Online Processing Library

Stéphane Drouin
Version 1.0
2001-11-28 12:28

Namespace Index ... x
Hierarchical Index ... xi
Compound Index ... xvi
File Index ... xxi
Cognois ... 2
Cognois::SpaceCarving ... 5
EspaceCalibrage ... 6
std ... 8
Class Documentation ... 9
Acquisition ... 9
Cognois::AcquisitionBalle ... 16
Cognois::AcquisitionCible ... 19
AcquisitionImage ... 21
AcquisitionImageListe ... 23
Cognois::AcquisitionSurDisque ... 25
Cognois::AcquisitionSurEcran ... 27
AGParamBin ... 29
alist ... 30
Cognois::Ball ... 31
BitmapImage ... 32
BitmapImageCoder ... 37
BitmapImageDecoder ... 39
BitmapImageEncoder ... 40
BlobColoring ... 41
BlobInfo ... 44
FormatBMP::bmpHeader ... 45
Cognois::Calibration ... 46
Calibration ... 49
Cognois::CalibrationClient ... 50
Cognois::CameraClientController ... 56
Cognois::CameraModel ... 59
cilist ... 65
cllist ... 66
BitmapImage::ColorArea ... 67
BitmapImage::ColorMapEntry ... 68
BitmapImage::ColorUsage ... 69
BitmapImage::ColorUsageTable ... 70
ColourSpaceConverter ... 71
complex ... 73
Contour ... 74
ContourSegment ... 77
CouleurHSI ... 79
CouleurRGB ... 81
Cognois::CounterOCR ... 84
DirectMilAccess ... 87
dmat ... 89
doublecomplex ... 90
EGifCircularTree ... 91
EGifCorruptFile ... 92
EGifError ... 93
EGifFileReadError ... 94
EGifNoMoreImages ... 95
EGifNotAGifFile ... 96
EGraphicsAbort ... 97
EGraphicsException ... 98

EInvalidBitCount ... 100
EJpegAbort ... 101
EJpegBadData ... 102
EJpegBadOutputParameter ... 103
EJpegError ... 104
EJpegFatal ... 105
EJpegIndexOutOfRange ... 106
EJpegValueOutOfRange ... 107
Ensemble ... 108
EPngError ... 109
Equivalence ... 110
ErreurFormat ... 111
ErreurFormatImage ... 112
ErreurFormatImageEcriture ... 113
ErreurFormatImageLecture ... 114
ErreurImage ... 115
ESubscriptOutOfRange ... 116
Cognois::EvaluationDeQualite ... 117
EXbmInvalid ... 118
Exception ... 119
std::exception ... 120
Cognois::Exception ... 121
Cognois::ExtractionDeCible ... 123
FormatBMP ... 128
FormatContour ... 130
FormatImage ... 132
FormatPBM ... 134
FormatPCX ... 136
FormatPGM ... 138
FormatPortable ... 140
FormatPPM ... 141
FormatSunRaster ... 143
FormatXPM ... 144
FormatXWD ... 147
Cognois::FrameCounter ... 150
Cognois::FrameSetClient ... 155
fstream ... 164
GenericCognoisClient ... 165
GifApplicationBlock ... 166
GifColorTableEntry ... 167
GifDecoder ... 168
GifEncoder ... 174
GifGraphicControlBlock ... 176
GifHeader ... 177
GifImageDescriptor ... 178
GifScreenDescriptor ... 179
PngEncoder::HashEntry ... 180
Contour::HoraudAngle ... 181
Contour::HoraudRelation ... 182
icilist ... 183
Image ... 184
ImageBinaire ... 190
ImageGradient ... 192
ImageGris ... 194
ImageRGB ... 196
ImageStream ... 198

IndiceAlias 202
inlist 204
PngDecoder::InterlaceInfo 207
Intervalle 208
JfifExtension 210
JfifHeader 211
JpegDecoder 212
JpegDecoderComponent 220
JpegDecoderDataUnit 225
JpegDecoderQuantizationTable 227
JpegEncoder 230
JpegEncoderCoefficientBlock 239
JpegEncoderComponent 240
JpegEncoderDataUnit 246
JpegEncoderHuffmanTable 248
JpegEncoderQuantizationTable 250
JpegHuffmanDecoder 252
Kernel 254
Liste 257
ListeTri 258
GifDecoder::LZ77String 259
Cognois::MakeAVI 260
Matrice 263
Cognois::MilGarbageCollector 272
MilGarbageCollector 275
MILSys 276
MINPACK 278
Multitype 280
Namelist 281
Cognois::NoopToServer 282
Cognois::OCRFontCreator 284
Cognois::OfflineProcessing 287
olist 291
Cognois::OnlineProcessing 292
Operateur 298
FormatPBM::pbmHeader 303
FormatPCX::pcxHeader 304
FormatPGM::pgmHeader 306
Picture 307
Cognois::PlanarToPacked 311
PngChromaticitiesData 313
PngDecoder 314
PngEncoder 323
PngHuffmanDecoder 333
PngHuffmanEncoder 335
PngImageHeader 337
PngPixelDimensions 338
PngTimeData 339
Point 340
EspaceCalibrage::Point 343
PointColore 344
EspaceCalibrage::PointXYZ 345
FormatPPM::ppmHeader 346
ImageRGB::PRep 347
ImageGris::PRep 348
Cognois::ProcessingObject 349

FormatSunRaster::rasHeader ... 352
RGBQUADsd ... 353
JpegEncoder::Scan ... 354
Cognois::SpaceCarving::SCVolume ... 355
Cognois::SequentialProcessing ... 358
StatisticsR ... 360
Table2D ... 362
TableHO ... 363
TesterContour ... 364
TesterEvolution ... 367
TstParam ... 369
TstParamNote ... 370
Vardesc ... 371
std::vector ... 372
XbmDecoder ... 373
XbmEncoder ... 376
FormatXPM::xpmColor ... 378
FormatXPM::xpmHeader ... 379
FormatXWD::xwdColor ... 380
FormatXWD::xwdHeader ... 381
File Documentation ... 384
Acquisition.cpp ... 384
Acquisition.h ... 386
AcquisitionBalle.cpp ... 387
AcquisitionBalle.h ... 388
AcquisitionCible.cpp ... 389
AcquisitionCible.h ... 390
AcquisitionImage.cpp ... 391
AcquisitionImage.h ... 392
AcquisitionImageListe.cpp ... 393
AcquisitionImageListe.h ... 394
AcquisitionSurDisque.cpp ... 395
AcquisitionSurDisque.h ... 396
AcquisitionSurEcran.cpp ... 397
AcquisitionSurEcran.h ... 398
bitimage.cpp ... 399
bitimage.h ... 400
BlobColoring.cpp ... 401
BlobColoring.h ... 402
Calibration.cpp ... 403
Calibration.h ... 404
CalibrationClient.cpp ... 405
CalibrationClient.h ... 406
CameraClientController.cpp ... 407
CameraClientController.h ... 408
CameraModel.cpp ... 409
CameraModel.h ... 410
ChaineUtile.hpp ... 412
Cognois.h ... 413
ColourSpaceConverter.cpp ... 414
ColourSpaceConverter.h ... 415
Contour.cpp ... 416
Contour.hpp ... 417
CounterOCR.cpp ... 418
CounterOCR.h ... 419
datatype.h ... 420

DPMPAR.cpp 422
ENORM.cpp 423
Equivalence.hpp 424
EspaceCalibrage.cpp 425
EspaceCalibrage.h 426
EvaluationDeQualite.cpp 427
EvaluationDeQualite.h 428
Exception.h 429
ExtractionDeCible.cpp 430
ExtractionDeCible.h 433
F2C.H 434
FDJAC2.cpp 439
FormatBMP.cpp 440
FormatBMP.hpp 441
FormatContour.cpp 442
FormatContour.hpp 443
FormatImage.hpp 444
FormatPBM.cpp 445
FormatPBM.hpp 446
FormatPCX.cpp 447
FormatPCX.hpp 448
FormatPGM.cpp 449
FormatPGM.hpp 450
FormatPortable.hpp 451
FormatPPM.cpp 453
FormatPPM.hpp 454
FormatSunRaster.cpp 455
FormatSunRaster.hpp 456
FormatXPM.cpp 458
FormatXPM.hpp 459
FormatXWD.cpp 460
FormatXWD.hpp 461
FrameCounter.cpp 462
FrameCounter.h 463
FrameSetClient.cpp 464
FrameSetClient.h 465
gif.h 466
gifdecod.cpp 468
gifdecod.h 469
gifencod.cpp 470
gifencod.h 471
gifexcep.h 472
GrabInit.cpp 473
GrabInit.h 474
grexcept.h 475
Image.hpp 476
ImageBinaire.cpp 477
ImageBinaire.hpp 478
ImageGradient.cpp 479
ImageGradient.hpp 480
ImageGris.cpp 481
ImageGris.hpp 482
ImageRGB.cpp 483
ImageRGB.hpp 484
ImageStream.cpp 485
ImageStream.hpp 486

ImageStreamException.hpp 487
imagetyp.h 488
jfif.cpp 489
jfif.h 491
jpdecobk.h 492
jpdecomp.cpp 493
jpdecomp.h 494
jpdedu.cpp 495
jpdedu.h 496
jpdehuff.cpp 497
jpdehuff.h 498
jpdequan.cpp 499
jpdequan.h 500
jpeg.cpp 501
jpeg.h 502
jpegdeco.cpp 505
jpegdeco.h 506
jpegenco.cpp 507
jpegenco.h 508
jpencobk.h 509
jpencomp.cpp 510
jpencomp.h 511
jpendu.cpp 512
jpendu.h 513
jpenhuff.cpp 514
jpenhuff.h 515
jpenquan.cpp 516
jpenquan.h 517
jpgexcep.h 518
jpmarker.h 519
Kernel.cpp 521
Kernel.hpp 522
ListeTri.hpp 523
LMDIF.cpp 524
LMPAR.cpp 525
MakeAVI.cpp 526
MakeAVI.h 527
Matrice.cpp 528
Matrice.h 529
MATRIX.cpp 530
MATRIX.H 532
MilGarbageCollector.cpp 534
MilGarbageCollector.h 535
MILSys.h 536
MINPACK.cpp 537
MINPACK.h 538
NoopToServer.h 539
OCRFontCreator.cpp 540
OCRFontCreator.h 541
OfflineProcessing.cpp 542
OfflineProcessing.h 544
OnlineProcessing.cpp 545
OnlineProcessing.h 547
Operateur.cpp 548
Operateur.hpp 549
Picture.cpp 550

Picture.h ... 551
PlanarToPacked.cpp ... 552
PlanarToPacked.h ... 553
png.cpp ... 554
png.h ... 555
pngchksm.cpp ... 556
pngchksm.h ... 557
pngdecod.cpp ... 558
pngdecod.h ... 560
pngencod.cpp ... 561
pngencod.h ... 563
pnghuffd.cpp ... 564
pnghuffd.h ... 565
pnghuffe.cpp ... 566
pnghuffe.h ... 567
pngpvt.h ... 568
Point.hpp ... 571
ProcessingObject.h ... 572
QRFAC.cpp ... 573
QRSOLV.cpp ... 574
SCVolume.cpp ... 575
SCVolume.h ... 576
SequentialProcessing.h ... 577
Statistics.h ... 578
Systeme.hpp ... 579
TesterContour.hpp ... 580
TesterEvolution.hpp ... 581
xbmdecod.cpp ... 582
xbmdecod.h ... 583
xbmencod.cpp ... 584
xbmencod.h ... 585
xbmpvt.h ... 586
Index ... 587

Online Processing Library Namespace Index

Online Processing Library Namespace List

Here is a list of all namespaces with brief descriptions:

Cognois (Cognois-related resources) 2
Cognois::SpaceCarving 5
EspaceCalibrage 6
std 8

Online Processing Library Hierarchical Index

Online Processing Library Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- Acquisition 9
- AcquisitionImage 21
- AcquisitionImageListe 23
- AGParamBin 29
  - TesterEvolution 367
- alist 30
- Cognois::Ball 31
- BitmapImage 32
- BitmapImageCoder 37
  - BitmapImageDecoder 39
    - GifDecoder 168
    - JpegDecoder 212
    - PngDecoder 314
    - XbmDecoder 373
  - BitmapImageEncoder 40
    - GifEncoder 174
    - JpegEncoder 230
    - PngEncoder 323
    - XbmEncoder 376
- BlobColoring 41
- BlobInfo 44
- FormatBMP::bmpHeader 45
- Cognois::Calibration 46
- Calibration 49
- Cognois::CameraClientController 56
- Cognois::CameraModel 59
- cilist 65
- cllist 66
- BitmapImage::ColorArea 67
- BitmapImage::ColorMapEntry 68
- BitmapImage::ColorUsage 69
- BitmapImage::ColorUsageTable 70
- ColourSpaceConverter 71
- complex 73
- CouleurHSI 79
- CouleurRGB 81
- DirectMilAccess 87
- dmat 89
- doublecomplex 90
- Cognois::EvaluationDeQualite 117

Exception....119
EGraphicsException....98
EGifError....93
EGifCircularTree....91
EGifCorruptFile....92
EGifFileReadError....94
EGifNoMoreImages....95
EGifNotAGifFile....96
EGraphicsAbort....97
EInvalidBitCount....100
EJpegAbort....101
EJpegError....104
EJpegBadData....102
EJpegBadOutputParameter....103
EJpegIndexOutOfRange....106
EJpegValueOutOfRange....107
EJpegFatal....105
EPngError....109
ESubscriptOutOfRange....116
EXbmInvalid....118
ErreurFormat....111
ErreurFormatImage....112
ErreurFormatImageEcriture....113
ErreurFormatImageLecture....114
ErreurImage....115
std::exception....120
Cognois::Exception....121
Cognois::ExtractionDeCible....123
FormatImage....132
FormatBMP....128
FormatPCX....136
FormatPortable....140
FormatContour....130
FormatPBM....134
FormatPGM....138
FormatPPM....141
FormatSunRaster....143
FormatXPM....144
FormatXWD....147
Cognois::FrameCounter....150
fstream....164
ImageStream....198
GenericCognoisClient....165
Cognois::FrameSetClient....155
Cognois::CalibrationClient....50
GifApplicationBlock....166
GifColorTableEntry....167

GifGraphicControlBlock 176
GifHeader 177
GifImageDescriptor 178
GifScreenDescriptor 179
PngEncoder::HashEntry 180
Contour::HoraudAngle 181
icilist 183
IndiceAlias 202
inlist 204
PngDecoder::InterlaceInfo 207
Intervalle 208
JfifExtension 210
JfifHeader 211
JpegDecoderComponent 220
JpegDecoderDataUnit 225
JpegDecoderQuantizationTable 227
JpegEncoderCoefficientBlock 239
JpegEncoderComponent 240
JpegEncoderDataUnit 246
JpegEncoderHuffmanTable 248
JpegEncoderQuantizationTable 250
JpegHuffmanDecoder 252
Liste 257
    Contour 74
    Contour::HoraudRelation 182
    ContourSegment 77
    Ensemble 108
    ListeTri 258
GifDecoder::LZ77String 259
Matrice 263
Cognois::MilGarbageCollector 272
MilGarbageCollector 275
MILSys 276
MINPACK 278
Multitype 280
Namelist 281
Cognois::OfflineProcessing 287
olist 291
Cognois::OnlineProcessing 292
FormatPBM::pbmHeader 303
FormatPCX::pcxHeader 304
FormatPGM::pgmHeader 306
Picture 307
PngChromaticitiesData 313
PngHuffmanDecoder 333

PngHuffmanEncoder 335
PngImageHeader 337
PngPixelDimensions 338
PngTimeData 339
Point 340
EspaceCalibrage::Point 343
PointColore 344
EspaceCalibrage::PointXYZ 345
FormatPPM::ppmHeader 346
ImageRGB::PRep 347
ImageGris::PRep 348
Cognois::ProcessingObject 349
    Cognois::AcquisitionBalle 16
    Cognois::AcquisitionCible 19
    Cognois::AcquisitionSurDisque 25
    Cognois::AcquisitionSurEcran 27
    Cognois::CounterOCR 84
    Cognois::MakeAVI 260
    Cognois::NoopToServer 282
    Cognois::OCRFontCreator 284
    Cognois::PlanarToPacked 311
    Cognois::SequentialProcessing 358
FormatSunRaster::rasHeader 352
RGBQUADsd 353
JpegEncoder::Scan 354
Cognois::SpaceCarving::SCVolume 357
StatisticsR 360
Table2D 362
    Image 184
        ImageGradient 192
        ImageGris 194
            ImageBinaire 190
        ImageRGB 196
    Operateur 298
        Kernel 254
TableHO 363
    Equivalence 110
TesterContour 364
TstParam 369
TstParamNote 370
Vardesc 371
std::vector 372
    Cognois::SequentialProcessing 358
FormatXPM::xpmColor 378
FormatXPM::xpmHeader 379
FormatXWD::xwdColor 380

FormatXWD::xwdHeader 381

Online Processing Library Compound Index

Online Processing Library Compound List

Here are the classes, structs, unions and interfaces with brief descriptions:

Acquisition 9
Cognois::AcquisitionBalle 16
Cognois::AcquisitionCible (Find a calibration pattern in an image) 19
AcquisitionImage 21
AcquisitionImageListe 23
Cognois::AcquisitionSurDisque (Export grabbed images to files) 25
Cognois::AcquisitionSurEcran (Display of grabbed images) 27
AGParamBin 29
alist 30
Cognois::Ball 31
BitmapImage 32
BitmapImageCoder 37
BitmapImageDecoder 39
BitmapImageEncoder 40
BlobColoring 41
BlobInfo 44
FormatBMP::bmpHeader 45
Cognois::Calibration 46
Calibration (Calilbrate one, two or more cameras using known planar points and its images) 49
Cognois::CalibrationClient (Read and write calibration data on the server) 50
Cognois::CameraClientController (Get the current mode from the server and use the appropriate processing object) 56
Cognois::CameraModel (**Calibration** (*p.46*) parameters of n cameras in a stereo environment) 59
cilist 65
cllist 66
BitmapImage::ColorArea 67
BitmapImage::ColorMapEntry 68
BitmapImage::ColorUsage 69
BitmapImage::ColorUsageTable 70
ColourSpaceConverter 71
complex 73
Contour 74
ContourSegment 77
CouleurHSI 79
CouleurRGB 81
Cognois::CounterOCR (Optical character recognition of the 5 digits of the HP 5214L Preset Counter) 84
DirectMilAccess 87
dmat 89
doublecomplex 90

EGifCircularTree 91
EGifCorruptFile 92
EGifError 93
EGifFileReadError 94
EGifNoMoreImages 95
EGifNotAGifFile 96
EGraphicsAbort 97
EGraphicsException 98
EInvalidBitCount 100
EJpegAbort 101
EJpegBadData 102
EJpegBadOutputParameter 103
EJpegError 104
EJpegFatal 105
EJpegIndexOutOfRange 106
EJpegValueOutOfRange 107
Ensemble 108
EPngError 109
Equivalence 110
ErreurFormat 111
ErreurFormatImage 112
ErreurFormatImageEcriture 113
ErreurFormatImageLecture 114
ErreurImage 115
ESubscriptOutOfRange 116
Cognois::EvaluationDeQualite 117
EXbmInvalid 118
Exception 119
std::exception 120
Cognois::Exception (**Exception** (*p.121*) class for COGNOIS) 121
Cognois::ExtractionDeCible (Extraction des des points d'une cible de calibrage dans une image. La cible est formee // de cercles et son modele est connu) 123
FormatBMP 128
FormatContour 130
FormatImage 132
FormatPBM 134
FormatPCX 136
FormatPGM 138
FormatPortable 140
FormatPPM 141
FormatSunRaster 143
FormatXPM 144
FormatXWD 147
Cognois::FrameCounter (Count grabbed frames according to their time stamp) 150
Cognois::FrameSetClient (Read and write blob information on the COGNOIS server) 155
fstream 164

GenericCognoisClient 165
GifApplicationBlock 166
GifColorTableEntry 167
GifDecoder 168
GifEncoder 174
GifGraphicControlBlock 176
GifHeader 177
GifImageDescriptor 178
GifScreenDescriptor 179
PngEncoder::HashEntry 180
Contour::HoraudAngle 181
Contour::HoraudRelation 182
icilist 183
Image 184
ImageBinaire 190
ImageGradient 192
ImageGris 194
ImageRGB 196
ImageStream 198
IndiceAlias 202
inlist 204
PngDecoder::InterlaceInfo 207
Intervalle 208
JfifExtension 210
JfifHeader 211
JpegDecoder 212
JpegDecoderComponent 220
JpegDecoderDataUnit 225
JpegDecoderQuantizationTable 227
JpegEncoder 230
JpegEncoderCoefficientBlock 239
JpegEncoderComponent 240
JpegEncoderDataUnit 246
JpegEncoderHuffmanTable 248
JpegEncoderQuantizationTable 250
JpegHuffmanDecoder 252
Kernel 254
Liste 257
ListeTri 258
GifDecoder::LZ77String 259
Cognois::MakeAVI (Export grabbed images to an AVI movie) 260
Matrice (Fonctions de base pour le calcul matriciel)....263
Cognois::MilGarbageCollector 272
MilGarbageCollector (A garbage collector for MIL image buffers)....275
MILSys (Mil system data) 276

MINPACK (Levenberg-Marquardt algorithm to minimize the sum of the squares) 278
Multitype 280
Namelist 281
Cognois::NoopToServer (Perform a sequence of nothing, but keep the server informed of the current frame ID) 282
Cognois::OCRFontCreator (Optical character recognition of the 5 digits of the HP 5214L Preset Counter) 284
Cognois::OfflineProcessing (Processing of images from AVI files) 287
olist 291
Cognois::OnlineProcessing (Online processing of images) 292
Operateur 298
FormatPBM::pbmHeader 303
FormatPCX::pcxHeader 304
FormatPGM::pgmHeader 306
Picture 307
Cognois::PlanarToPacked (Transform a MIL planar buffer to a MIL packed buffer) 311
PngChromaticitiesData 313
PngDecoder 314
PngEncoder 323
PngHuffmanDecoder 333
PngHuffmanEncoder 335
PngImageHeader 337
PngPixelDimensions 338
PngTimeData 339
Point 340
EspaceCalibrage::Point 343
PointColore 344
EspaceCalibrage::PointXYZ 345
FormatPPM::ppmHeader 346
ImageRGB::PRep 347
ImageGris::PRep 348
Cognois::ProcessingObject (Virtual class for user-supplied processing objects) 349
FormatSunRaster::rasHeader 352
RGBQUADsd 353
JpegEncoder::Scan 354
Cognois::SpaceCarving::SCVolume (Volume for space carving) 357
Cognois::SequentialProcessing (Perform a sequence of image processing) 358
StatisticsR 360
Table2D 362
TableHO 363
TesterContour 364
TesterEvolution 367
TstParam 369
TstParamNote 370
Vardesc 371
std::vector 372

XbmDecoder 373
XbmEncoder 376
FormatXPM::xpmColor 378
FormatXPM::xpmHeader 379
FormatXWD::xwdColor 380
FormatXWD::xwdHeader 381

Online Processing Library File Index

Online Processing Library File List

Here is a list of all files with brief descriptions:

Acquisition.cpp 384
Acquisition.h 386
AcquisitionBalle.cpp 387
AcquisitionBalle.h 388
AcquisitionCible.cpp 389
AcquisitionCible.h 390
AcquisitionImage.cpp 391
AcquisitionImage.h 392
AcquisitionImageListe.cpp 393
AcquisitionImageListe.h 394
AcquisitionSurDisque.cpp 395
AcquisitionSurDisque.h 396
AcquisitionSurEcran.cpp 397
AcquisitionSurEcran.h 398
bitimage.cpp 399
bitimage.h 400
BlobColoring.cpp 401
BlobColoring.h 402
Calibration.cpp 403
Calibration.h 404
CalibrationClient.cpp 405
CalibrationClient.h 406
CameraClientController.cpp 407
CameraClientController.h 408
CameraModel.cpp 409
CameraModel.h 410
ChaineUtile.hpp 412
Cognois.h 413
ColourSpaceConverter.cpp 414
ColourSpaceConverter.h 415
Contour.cpp 416
Contour.hpp 417
CounterOCR.cpp 418
CounterOCR.h 419
datatype.h 420
DPMPAR.cpp 422
ENORM.cpp 423
Equivalence.hpp 424
EspaceCalibrage.cpp 425

EspaceCalibrage.h 426
EvaluationDeQualite.cpp 427
EvaluationDeQualite.h 428
Exception.h 429
ExtractionDeCible.cpp 430
ExtractionDeCible.h 433
F2C.H 434
FDJAC2.cpp 439
FormatBMP.cpp 440
FormatBMP.hpp 441
FormatContour.cpp 442
FormatContour.hpp 443
FormatImage.hpp 444
FormatPBM.cpp 445
FormatPBM.hpp 446
FormatPCX.cpp 447
FormatPCX.hpp 448
FormatPGM.cpp 449
FormatPGM.hpp 450
FormatPortable.hpp 451
FormatPPM.cpp 453
FormatPPM.hpp 454
FormatSunRaster.cpp 455
FormatSunRaster.hpp 456
FormatXPM.cpp 458
FormatXPM.hpp 459
FormatXWD.cpp 460
FormatXWD.hpp 461
FrameCounter.cpp 462
FrameCounter.h 463
FrameSetClient.cpp 464
FrameSetClient.h 465
gif.h 466
gifdecod.cpp 468
gifdecod.h 469
gifencod.cpp 470
gifencod.h 471
gifexcep.h 472
GrabInit.cpp 473
GrabInit.h 474
grexcept.h 475
Image.hpp 476
ImageBinaire.cpp 477
ImageBinaire.hpp 478
ImageGradient.cpp 479

ImageGradient.hpp 480
ImageGris.cpp 481
ImageGris.hpp 482
ImageRGB.cpp 483
ImageRGB.hpp 484
ImageStream.cpp 485
ImageStream.hpp 486
ImageStreamException.hpp 487
imagetyp.h 488
jfif.cpp 489
jfif.h 491
jpdecobk.h 492
jpdecomp.cpp 493
jpdecomp.h 494
jpdedu.cpp 495
jpdedu.h 496
jpdehuff.cpp 497
jpdehuff.h 498
jpdequan.cpp 499
jpdequan.h 500
jpeg.cpp 501
jpeg.h 502
jpegdeco.cpp 505
jpegdeco.h 506
jpegenco.cpp 507
jpegenco.h 508
jpencobk.h 509
jpencomp.cpp 510
jpencomp.h 511
jpendu.cpp 512
jpendu.h 513
jpenhuff.cpp 514
jpenhuff.h 515
jpenquan.cpp 516
jpenquan.h 517
jpgexcep.h 518
jpmarker.h 519
Kernel.cpp 521
Kernel.hpp 522
ListeTri.hpp 523
LMDIF.cpp 524
LMPAR.cpp 525
MakeAVI.cpp 526
MakeAVI.h 527
Matrice.cpp 528

Matrice.h 529
MATRIX.cpp 530
MATRIX.H 532
MilGarbageCollector.cpp 534
MilGarbageCollector.h 535
MILSys.h 536
MINPACK.cpp 537
MINPACK.h 538
NoopToServer.h 539
OCRFontCreator.cpp 540
OCRFontCreator.h 541
OfflineProcessing.cpp 542
OfflineProcessing.h 544
OnlineProcessing.cpp 545
OnlineProcessing.h 547
Operateur.cpp 548
Operateur.hpp 549
Picture.cpp 550
Picture.h 551
PlanarToPacked.cpp 552
PlanarToPacked.h 553
png.cpp 554
png.h 555
pngchksm.cpp 556
pngchksm.h 557
pngdecod.cpp 558
pngdecod.h 560
pngencod.cpp 561
pngencod.h 563
pnghuffd.cpp 564
pnghuffd.h 565
pnghuffe.cpp 566
pnghuffe.h 567
pngpvt.h 568
Point.hpp 571
ProcessingObject.h 572
QRFAC.cpp 573
QRSOLV.cpp 574
SCVolume.cpp 575
SCVolume.h 576
SequentialProcessing.h 577
Statistics.h 578
Systeme.hpp 579
TesterContour.hpp 580
TesterEvolution.hpp 581

xbmdecod.cpp 582
xbmdecod.h 583
xbmencod.cpp 584
xbmencod.h 585
xbmpvt.h 586

Online Processing Library Namespace Documentation

Cognois Namespace Reference

Cognois-related resources.

Compounds

- class **Cognois::AcquisitionBalle**
- class **Cognois::AcquisitionCible**
  Find a calibration pattern in an image.

- class **Cognois::AcquisitionSurDisque**
  Export grabbed images to files.

- class **Cognois::AcquisitionSurEcran**
  Display of grabbed images.

- struct **Cognois::Ball**
- class **Cognois::Calibration**
- class **Cognois::CalibrationClient**
  Read and write calibration data on the server.

- class **Cognois::CameraClientController**
  Get the current mode from the server and use the appropriate processing object.

- class **Cognois::CameraModel**
  ***Calibration** (p.46) parameters of n cameras in a stereo environment.*

- class **Cognois::CounterOCR**
  Optical character recognition of the 5 digits of the HP 5214L Preset Counter.

- class **Cognois::EvaluationDeQualite**
- class **Cognois::Exception**
  ***Exception** (p.121) class for COGNOIS.*

- class **Cognois::ExtractionDeCible**
  Extraction des des points d'une cible de calibrage dans une image. La cible est formee // de cercles et son modele est connu.

- class **Cognois::FrameCounter**
  Count grabbed frames according to their time stamp.

- class **Cognois::FrameSetClient**
  Read and write blob information on the COGNOIS server.

- class **Cognois::MakeAVI**
  Export grabbed images to an AVI movie.

- class **Cognois::MilGarbageCollector**
- class **Cognois::NoopToServer**
  Perform a sequence of nothing, but keep the server informed of the current frame ID.

- class **Cognois::OCRFontCreator**
  Optical character recognition of the 5 digits of the HP 5214L Preset Counter.

- class **Cognois::OfflineProcessing**
  Processing of images from AVI files.

- class **Cognois::OnlineProcessing**
  Online processing of images.

- class **Cognois::PlanarToPacked**
  Transform a MIL planar buffer to a MIL packed buffer.

- class **Cognois::ProcessingObject**
  Virtual class for user-supplied processing objects.

- class **Cognois::SequentialProcessing**
  Perform a sequence of image processing.

Typedefs

- typedef struct Cognois::Ball **Ball**

Detailed Description

Cognois-related resources.

Master header file for COGNOIS. Include this file to use the COGNOIS Server, COGNOIS Exceptions and COGNOIS data types.

Typedef Documentation

typedef struct Cognois::Ball Cognois::Ball

Cognois::SpaceCarving Namespace Reference

Compounds

- class **Cognois::SpaceCarving::SCVolume**
  Volume for space carving.

EspaceCalibrage Namespace Reference

Compounds

- struct **EspaceCalibrage::Point**
- struct **EspaceCalibrage::PointXYZ**

Typedefs

- typedef float **DataType**
- typedef **std::vector**<**Point**> **VPoints**
- typedef **std::vector**<**VPoints**> **MPoints**
- typedef **std::vector**<**PointXYZ**> **VPointsXYZ**
- typedef **std::vector**<**VPointsXYZ**> **MPointsXYZ**
- typedef **std::vector**<double> **VDoubles**
- typedef **std::vector**<**VDoubles**> **MDoubles**

Functions

- int **operator==** (const **Point** &, const **Point** &)
- int **operator==** (const **PointXYZ** &, const **PointXYZ** &)
- std::ostream& **operator<<** (std::ostream &out, const **Point** &)
- std::ostream& **operator<<** (std::ostream &out, const **PointXYZ** &)
- void **CalculerAngles** (const **VDoubles** &parExt, double &Rx, double &Ry, double &Rz, double &altRx, double &altRy, double &altRz)
- void **CalculerAngles** (const **Matrice** &inParExt, double &Rx, double &Ry, double &Rz, double &altRx, double &altRy, double &altRz)
- void **CalculerMatrice** (**VDoubles** &parExt, const double &Rx, const double &Ry, const double &Rz)
- void **CalculerMatrice** (**Matrice** &parExt, const double &Rx, const double &Ry, const double &Rz)
- **Matrice MatriceProduitVectoriel** (const **Matrice** &x)

Typedef Documentation

typedef float EspaceCalibrage::DataType

typedef std::vector< VDoubles > EspaceCalibrage::MDoubles<VDoubles>

typedef std::vector< VPoints > EspaceCalibrage::MPoints<VPoints>

typedef std::vector< VPointsXYZ > EspaceCalibrage::MPointsXYZ<VPointsXYZ>

typedef std::vector< double > EspaceCalibrage::VDoubles<double>

typedef std::vector< Point > EspaceCalibrage::VPoints<Point>

typedef std::vector< PointXYZ > EspaceCalibrage::VPointsXYZ<PointXYZ>

Function Documentation

void EspaceCalibrage::CalculerAngles (const Matrice & *inParExt*, double & *Rx*, double & *Ry*, double & *Rz*, double & *altRx*, double & *altRy*, double & *altRz*)

void EspaceCalibrage::CalculerAngles (const VDoubles & *parExt*, double & *Rx*, double & *Ry*, double & *Rz*, double & *altRx*, double & *altRy*, double & *altRz*)

void EspaceCalibrage::CalculerMatrice (Matrice & *parExt*, const double & *Rx*, const double & *Ry*, const double & *Rz*)

void EspaceCalibrage::CalculerMatrice (VDoubles & *parExt*, const double & *Rx*, const double & *Ry*, const double & *Rz*)

Matrice EspaceCalibrage::MatriceProduitVectoriel (const Matrice & *x*)

std::ostream & EspaceCalibrage::operator<< (std::ostream & *out*, const PointXYZ & *pt*)

std::ostream & EspaceCalibrage::operator<< (std::ostream & *out*, const Point & *pt*)

int EspaceCalibrage::operator== (const PointXYZ & *a*, const PointXYZ & *b*)

int EspaceCalibrage::operator== (const Point & *a*, const Point & *b*)

std Namespace Reference

Online Processing Library Class Documentation

Acquisition Class Reference

```
#include <Acquisition.h>
```

Public Methods

- **Acquisition** ()
  DÉFINITION DES MÉTHODES ////////////

- int **mMilBufferDernier**
- int **mMilBufferProchainDernier**
- int **mMilBufferBarre**
- HANDLE **mMilBufferAcces**
- MIL_ID **mMilDigitizer**
- MIL_ID **mMilSysteme**
- MIL_ID **mMilApplication**
- HANDLE **mMilAcces**
- int **mGrabEnCourt**
- HANDLE **mEndGrabEvent**
- int **mGrabStartHookSucceed**
- int **mGrabEndHookSucceed**
- int **mFrameStartHookSucceed**
- int **mFrameEndHookSucceed**
- int **mFieldEndHookSucceed**
- int **mFieldOddEndHookSucceed**
- int **mFieldEvenEndHookSucceed**
- LARGE_INTEGER **mCompteurDebut**
- LARGE_INTEGER **mCompteurFrequence**
- double **mTemps** [6]
- int **mCount**

Friends

- long MFTYPE **GrabStartHookHandler** (long HookType, MIL_ID EventId, void MPTYPE *UserDataPtr)
  DÉFINITION DES PROCÉDURES ////////////

Acquisition::~Acquisition () `[virtual]`

Member Function Documentation

void Acquisition::Arreter ()

int Acquisition::Demarrer ()

void Acquisition::EcrireBits (int *pBit*, int *pDonnee*)

int Acquisition::Fermer ()

void Acquisition::FieldEndHookHandler () `[private]`

void Acquisition::FieldEvenEndHookHandler () `[private]`

void Acquisition::FieldOddEndHookHandler () `[private]`

void Acquisition::FrameEndHookHandler () `[private]`

void Acquisition::FrameStartHookHandler () `[private]`

void Acquisition::GrabEndHookHandler () `[private]`

void Acquisition::GrabStartHookHandler () `[private]`

int Acquisition::LibererDerniereImage ()

void Acquisition::LireBits (std::vector< int > *pBit*, std::vector< int > & *pDonnee*)

void Acquisition::LireBits (int *pBit*, int & *pDonnee*)

int Acquisition::LireDimensions (int & *pLargeur*, int & *pHauteur*)

int Acquisition::LireEvenement (HANDLE & *pEvent*)

void Acquisition::LireSysteme (MIL_ID & *pApplication*, MIL_ID & *pSysteme*, MIL_ID & *pDigitizer*)

int Acquisition::Ouvrir (double *pImageScale* = 0.5, bool *pOutputVD* = false)

int Acquisition::ReserverDerniereImage (MIL_ID & *pId*, LARGE_INTEGER & *pTempsDebut*, LARGE_INTEGER & *pTempsFin*)

int Acquisition::ReserverDerniereImage (char *& *pAdresse*, double & *pTempsDebut*, double & *pTempsFin*)

Friends And Related Function Documentation

long MFTYPE FieldEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[friend]`

long MFTYPE FieldEvenEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[friend]`

long MFTYPE FieldOddEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[friend]`

long MFTYPE FrameEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[friend]`

long MFTYPE FrameStartHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[friend]`

long MFTYPE GrabEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[friend]`

long MFTYPE GrabStartHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[friend]`

DÉFINITION DES PROCÉDURES ////////////

Member Data Documentation

LARGE_INTEGER Acquisition::mCompteurDebut `[private]`

LARGE_INTEGER Acquisition::mCompteurFrequence `[private]`

int Acquisition::mCount `[private]`

HANDLE Acquisition::mEndGrabEvent `[private]`

int Acquisition::mFieldEndHookSucceed `[private]`

int Acquisition::mFieldEvenEndHookSucceed `[private]`

int Acquisition::mFieldOddEndHookSucceed `[private]`

int Acquisition::mFrameEndHookSucceed `[private]`

int Acquisition::mFrameStartHookSucceed `[private]`

int Acquisition::mGrabEnCourt `[private]`

int Acquisition::mGrabEndHookSucceed `[private]`

int Acquisition::mGrabStartHookSucceed `[private]`

int Acquisition::mImageHauteur `[private]`

int Acquisition::mImageLargeur `[private]`

int Acquisition::mImageLongueur `[private]`

HANDLE Acquisition::mMilAcces `[private]`

MIL_ID Acquisition::mMilApplication `[private]`

AcquisitionImageListe Acquisition::mMilBuffer `[private]`

HANDLE Acquisition::mMilBufferAcces `[private]`

int Acquisition::mMilBufferAttente `[private]`

int Acquisition::mMilBufferBarre `[private]`

int Acquisition::mMilBufferCourant `[private]`

int Acquisition::mMilBufferDernier `[private]`

int Acquisition::mMilBufferNombre `[private]`

int Acquisition::mMilBufferProchainDernier `[private]`

MIL_ID Acquisition::mMilDigitizer `[private]`

MIL_ID Acquisition::mMilSysteme `[private]`

int Acquisition::mOuvert `[private]`

double Acquisition::mTemps `[private]`

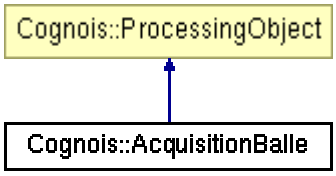
The documentation for this class was generated from the following files:

- **Acquisition.h**
- **Acquisition.cpp**

Cognois::AcquisitionBalle Class Reference

`#include <AcquisitionBalle.h>`

Inheritance diagram for Cognois::AcquisitionBalle:



Public Methods

- **AcquisitionBalle** (int inArgc, char **inArgv, float inFacteurX, float inFacteurY, int inResX, int inResY)
- **~AcquisitionBalle** ()
- void **Process** (MIL_ID &_milImage, unsigned long &_frameID, unsigned long &_timeStamp)
  Process an image with know frameID and timeStamp.

Private Attributes

- ColorAnalysis* **ac** [100]
- **Ball*** **ball** [100]
- BlobExtractor* **bem**
- int **compteur**
- float **facteurX**
- float **facteurY**
- int **blobCount**
- BackgroundSub **bg**
- OpticalFlow* **of**
- IplImage* **imageTraitee**
- IplImage* **background**
- IplImage* **mask**
- IplImage* **imageDest**
- int **mIterator**
- int **mBlocCount**
- **Cognois::FrameSetClient*** **transmission**
- LARGE_INTEGER **start**
- LARGE_INTEGER **stop**
- LARGE_INTEGER **frequency**

Constructor & Destructor Documentation

AcquisitionBalle::AcquisitionBalle (int *inArgc*, char ** *inArgv*, float *inFacteurX*, float *inFacteurY*, int *inResX*, int *inResY*)

AcquisitionBalle::~AcquisitionBalle ()

Member Function Documentation

void AcquisitionBalle::Process (MIL_ID & *milImage*, unsigned long & *frameID*, unsigned long & *timeStamp*) `[virtual]`

Process an image with know frameID and timeStamp.

Parameters:

inMilImage is a MIL image identifier.
inFrameID is the frame identifier associated with the image.
inTimeStamp is the time, in milliseconds, associated with the image.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

Member Data Documentation

ColorAnalysis * Cognois::AcquisitionBalle::ac `[private]`

IplImage * Cognois::AcquisitionBalle::background `[private]`

Ball * Cognois::AcquisitionBalle::ball `[private]`

BlobExtractor * Cognois::AcquisitionBalle::bem `[private]`

BackgroundSub Cognois::AcquisitionBalle::bg `[private]`

int Cognois::AcquisitionBalle::blobCount `[private]`

int Cognois::AcquisitionBalle::compteur `[private]`

float Cognois::AcquisitionBalle::facteurX `[private]`

float Cognois::AcquisitionBalle::facteurY `[private]`

LARGE_INTEGER Cognois::AcquisitionBalle::frequency `[private]`

IplImage * Cognois::AcquisitionBalle::imageDest `[private]`

IplImage * Cognois::AcquisitionBalle::imageTraitee `[private]`

int Cognois::AcquisitionBalle::mBlocCount `[private]`

int Cognois::AcquisitionBalle::mIterator `[private]`

IplImage * Cognois::AcquisitionBalle::mask `[private]`

OpticalFlow * Cognois::AcquisitionBalle::of `[private]`

LARGE_INTEGER Cognois::AcquisitionBalle::start `[private]`

LARGE_INTEGER Cognois::AcquisitionBalle::stop `[private]`

Cognois::FrameSetClient * Cognois::AcquisitionBalle::transmission `[private]`

The documentation for this class was generated from the following files:

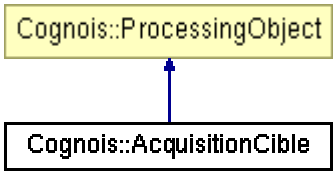
- **AcquisitionBalle.h**
- **AcquisitionBalle.cpp**

Cognois::AcquisitionCible Class Reference

Find a calibration pattern in an image.

`#include <AcquisitionCible.h>`

Inheritance diagram for Cognois::AcquisitionCible:



Public Methods

- **AcquisitionCible** (const VPoints &inPattern, **CalibrationClient** *inCognoisServer=0)
  Constructor with a calibration pattern and a cognois server interface.

- **AcquisitionCible** (const MPoints &inPatterns, **CalibrationClient** *inCognoisServer=0)
  Constructor with multiple calibration patterns and a cognois server interface.

- void **Process** (MIL_ID &inMilImage, unsigned long &inFrameID, unsigned long &inTimeStamp)
  Find a calibration pattern in an image.

Private Attributes

- **ExtractionDeCible mPatternFinder**
  Calibration *(p.46) pattern finder.*

- **CalibrationClient* mCognoisServer**
  Cognois *(p.2) server interface.*

Detailed Description

Find a calibration pattern in an image.

Constructor & Destructor Documentation

Cognois::AcquisitionCible::AcquisitionCible (const VPoints & *inPattern*, CalibrationClient * *inCognoisServer* = 0) `[inline]`

Constructor with a calibration pattern and a cognois server interface.

Cognois::AcquisitionCible::AcquisitionCible (const MPoints & *inPatterns*, CalibrationClient * *inCognoisServer* = 0) `[inline]`

Constructor with multiple calibration patterns and a cognois server interface.

Member Function Documentation

void Cognois::AcquisitionCible::Process (MIL_ID & *inMilImage*, unsigned long & *inFrameID*, unsigned long & *inTimeStamp*) `[virtual]`

Find a calibration pattern in an image.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

Member Data Documentation

CalibrationClient * Cognois::AcquisitionCible::mCognoisServer `[private]`

Cognois (*p.2*) server interface.

ExtractionDeCible Cognois::AcquisitionCible::mPatternFinder `[private]`

Calibration (*p.46*) pattern finder.

The documentation for this class was generated from the following files:

- **AcquisitionCible.h**
- **AcquisitionCible.cpp**

AcquisitionImage Class Reference

`#include <AcquisitionImage.h>`

Public Methods

- **AcquisitionImage** ()
  DÉFINITION DES MÉTHODES
  *////////////

Member Function Documentation

int AcquisitionImage::Creer (MIL_ID *pMilSysteme*, int *pImageLargeur*, int *pImageHauteur*)

char * AcquisitionImage::GetAdd ()

MIL_ID AcquisitionImage::GetId ()

LARGE_INTEGER AcquisitionImage::GetTempsDebut ()

LARGE_INTEGER AcquisitionImage::GetTempsFin ()

void AcquisitionImage::SetTempsDebut (LARGE_INTEGER *pTemps*)

void AcquisitionImage::SetTempsFin (LARGE_INTEGER *pTemps*)

Member Data Documentation

MIL_ID AcquisitionImage::mMilBuffer `[private]`

char * AcquisitionImage::mMilBufferAdd `[private]`

MIL_ID AcquisitionImage::mMilBufferBand `[private]`

LARGE_INTEGER AcquisitionImage::mTempsDebut `[private]`

LARGE_INTEGER AcquisitionImage::mTempsFin `[private]`

The documentation for this class was generated from the following files:

- **AcquisitionImage.h**
- **AcquisitionImage.cpp**

AcquisitionImageListe Class Reference

`#include <AcquisitionImageListe.h>`

Public Methods

- **AcquisitionImageListe** ()
  DÉFINITION DES MÉTHODES
  *////////////

Member Function Documentation

int AcquisitionImageListe::CreerListe (int *pLongueur*, MIL_ID *pMilSysteme*, int *pImageLargeur*, int *pImageHauteur*)

void AcquisitionImageListe::EffacerListe ()

char * AcquisitionImageListe::GetAdd (int *pIndex*)

MIL_ID AcquisitionImageListe::GetId (int *pIndex*)

LARGE_INTEGER AcquisitionImageListe::GetTempsDebut (int *pIndex*)

LARGE_INTEGER AcquisitionImageListe::GetTempsFin (int *pIndex*)

void AcquisitionImageListe::SetTempsDebut (int *pIndex*, LARGE_INTEGER *pTemps*)

void AcquisitionImageListe::SetTempsFin (int *pIndex*, LARGE_INTEGER *pTemps*)

Member Data Documentation

int AcquisitionImageListe::mImageHauteur `[private]`

int AcquisitionImageListe::mImageLargeur `[private]`

AcquisitionImage ** AcquisitionImageListe::mListe `[private]`

int AcquisitionImageListe::mLongueur `[private]`

MIL_ID AcquisitionImageListe::mMilSysteme `[private]`

The documentation for this class was generated from the following files:

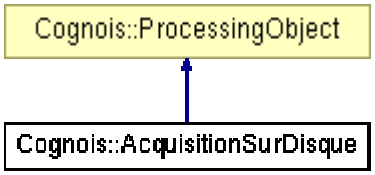
- **AcquisitionImageListe.h**
- **AcquisitionImageListe.cpp**

Cognois::AcquisitionSurDisque Class Reference

Export grabbed images to files.

`#include <AcquisitionSurDisque.h>`

Inheritance diagram for Cognois::AcquisitionSurDisque:



Public Methods

- **AcquisitionSurDisque** (unsigned long _queueLength, const char *_fileNameBase)
  Constructor with a maximum number of files to write and a base filename.

- **~AcquisitionSurDisque** ()
  Destructor.

- void **Process** (MIL_ID &_milImage, unsigned long &_frameID, unsigned long &_timeStamp)
  Export an image to a file.

Private Attributes

- unsigned long **compteur**
  Current file number.

- unsigned long **queueLimit**
  Maximum number of files.

- char* **fileName**
  Base filename.

Detailed Description

Export grabbed images to files.

Constructor & Destructor Documentation

Cognois::AcquisitionSurDisque::AcquisitionSurDisque (unsigned long *_queueLength*, const char * *_fileNameBase*)

Constructor with a maximum number of files to write and a base filename.

Parameters:

__queueLength is the maximum number of files to save
_fileNameBase is the base file name. Files will be named [_fileNameBase]000000.bmp

Cognois::AcquisitionSurDisque::~AcquisitionSurDisque ()

Destructor.

Member Function Documentation

void Cognois::AcquisitionSurDisque::Process (MIL_ID & *_milImage*, unsigned long & *_frameID*, unsigned long & *_timeStamp*) `[virtual]`

Export an image to a file.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

Member Data Documentation

unsigned long Cognois::AcquisitionSurDisque::compteur `[private]`

Current file number.

char * Cognois::AcquisitionSurDisque::fileName `[private]`

Base filename.

unsigned long Cognois::AcquisitionSurDisque::queueLimit `[private]`

Maximum number of files.

The documentation for this class was generated from the following files:

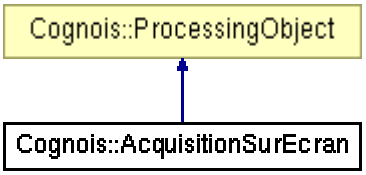
- **AcquisitionSurDisque.h**
- **AcquisitionSurDisque.cpp**

Cognois::AcquisitionSurEcran Class Reference

Display of grabbed images.

`#include <AcquisitionSurEcran.h>`

Inheritance diagram for Cognois::AcquisitionSurEcran:



Public Methods

- **AcquisitionSurEcran** (const **MILSys** &inNewMilSys)
  *Constructor with a reference to the **MILSys** (p.276) of the **OnlineProcessing** (p.292) object.*

- **~AcquisitionSurEcran** ()
  Destructor.

- void **Process** (MIL_ID &inMilImage, unsigned long &inFrameID, unsigned long &inTimeStamp)
  Display a grabbed image.

Private Attributes

- **MILSys MilSys**
  MIL identifiers; contains a display.

- MIL_ID **MilBufferDisplay**
  A display buffer.

- MIL_ID **mMilDisplay**

Detailed Description

Display of grabbed images.

Constructor & Destructor Documentation

Cognois::AcquisitionSurEcran::AcquisitionSurEcran (const MILSys & *inNewMilSys*)

Constructor with a reference to the **MILSys** (*p.276*) of the **OnlineProcessing** (*p.292*) object.

Parameters:

_newMilSys the MIL identifiers of a **OnlineProcessing** (*p.292*) object

Cognois::AcquisitionSurEcran::~AcquisitionSurEcran ()

Destructor.

Member Function Documentation

void Cognois::AcquisitionSurEcran::Process (MIL_ID & *inMilImage*, unsigned long & *inFrameID*, unsigned long & *inTimeStamp*) `[virtual]`

Display a grabbed image.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

Member Data Documentation

MIL_ID Cognois::AcquisitionSurEcran::MilBufferDisplay `[private]`

A display buffer.

MILSys Cognois::AcquisitionSurEcran::MilSys `[private]`

MIL identifiers; contains a display.

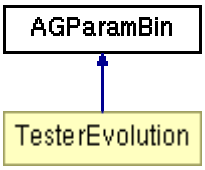
MIL_ID Cognois::AcquisitionSurEcran::mMilDisplay `[private]`

The documentation for this class was generated from the following files:

- **AcquisitionSurEcran.h**
- **AcquisitionSurEcran.cpp**

AGParamBin Class Reference

Inheritance diagram for AGParamBin:



The documentation for this class was generated from the following file:

- **TesterEvolution.hpp**

alist Struct Reference

```
#include <F2C.H>
```

Public Attributes

- **flag aerr**
- **ftnint aunit**

Member Data Documentation

flag alist::aerr

ftnint alist::aunit

The documentation for this struct was generated from the following file:

- **F2C.H**

Cognois::Ball Struct Reference

```
#include <AcquisitionBalle.h>
```

Public Attributes

- double **centerX**
- double **centerY**
- int **offsetX**
- int **offsetY**
- int **width**
- int **height**
- unsigned char* **array**

Member Data Documentation

unsigned char * Cognois::Ball::array

double Cognois::Ball::centerX

double Cognois::Ball::centerY

int Cognois::Ball::height

int Cognois::Ball::offsetX

int Cognois::Ball::offsetY

int Cognois::Ball::width

The documentation for this struct was generated from the following file:

- **AcquisitionBalle.h**

BitmapImage Class Reference

```
#include <bitimage.h>
```

Public Types

- enum { **RedOffset** = 2, **GreenOffset** = 1, **BlueOffset** = 0 }
- enum { **RowRounding** = 4 }

Public Methods

- **BitmapImage** ()
- **BitmapImage** (const BitmapImage &)
- virtual **~BitmapImage** ()
- BitmapImage& **operator=** (const BitmapImage &)
- void **SetSize** (unsigned int cc, unsigned int bits, unsigned int ww, unsigned int hh)
- **ColorMapEntry**& **ColorMap** (unsigned int index)
- **ColorMapEntry ColorMap** (unsigned int index) const
- **UBYTE1*** **operator[]** (unsigned int) const
- void **Clear** ()
- **UBYTE1*** **ImageData** ()
- unsigned int **Width** () const
- unsigned int **Height** () const
- unsigned int **BitCount** () const
- unsigned int **ColorCount** () const
- void **GetRGB** (unsigned int row, unsigned int col, **UBYTE1** &red, **UBYTE1** &green, **UBYTE1** &blue) const
- void **EightBitQuantization** (const BitmapImage &)
- void **SetProgressFunction** (**IMAGEPROGRESSFUNCTION**, void *)
- unsigned int **BytesPerRow** () const

Protected Methods

- void **Initialize** ()
- void **DoCopy** (const BitmapImage &)

Private Methods

- void **ClearData** ()
- int **LargestColorRange** (**ColorArea** &area)
- void **AddColor** (**UBYTE1** red, **UBYTE1** green, **UBYTE1** blue)
- void **SplitAreaInHalf** (unsigned int depth, unsigned int retrydepth, unsigned int area, unsigned int splitcolor)
- void **CreateColor** (unsigned int color)
- **ColorUsage*** **FindColor** (**UBYTE1** red, **UBYTE1** green, **UBYTE1** blue)
- void **FindColorUsage** (const BitmapImage &image)
- void **FreeColorQuantizationData** ()
- unsigned int **QuantizedColor** (**UBYTE1** red, **UBYTE1** green, **UBYTE1** blue)
- void **QuantizeSourceImage** (const BitmapImage &image)
- void **CallProgressFunction** (unsigned int, unsigned int, unsigned int)

Private Attributes

- unsigned int **row_width**
- unsigned int **bit_count**
- unsigned int **image_width**
- unsigned int **image_height**
- unsigned char* **image_data**
- unsigned int **color_count**
- **ColorMapEntry* color_map**
- **ColorUsageTable* color_usage**
- **ColorArea* color_areas**
- unsigned int **color_area_count**
- **IMAGEPROGRESSFUNCTION progress_function**
- void* **progress_data**

Member Enumeration Documentation

anonymous enum

Enumeration values:

RedOffset
GreenOffset
BlueOffset

anonymous enum

Enumeration values:

RowRounding

Constructor & Destructor Documentation

BitmapImage::BitmapImage ()

BitmapImage::BitmapImage (const BitmapImage & *source*)

BitmapImage::~BitmapImage () `[virtual]`

Member Function Documentation

void BitmapImage::AddColor (UBYTE1 *red*, UBYTE1 *green*, UBYTE1 *blue*) `[private]`

unsigned int BitmapImage::BitCount () const `[inline]`

unsigned int BitmapImage::BytesPerRow () const `[inline]`

void BitmapImage::CallProgressFunction (unsigned int *percent*, unsigned int *pass*, unsigned int *passcount*) `[private]`

void BitmapImage::Clear ()

void BitmapImage::ClearData () `[private]`

unsigned int BitmapImage::ColorCount () const `[inline]`

BitmapImage::ColorMapEntry BitmapImage::ColorMap (unsigned int *index*) const

BitmapImage::ColorMapEntry & BitmapImage::ColorMap (unsigned int *index*)

void BitmapImage::CreateColor (unsigned int *color*) `[private]`

void BitmapImage::DoCopy (const BitmapImage & *source*) `[protected]`

void BitmapImage::EightBitQuantization (const BitmapImage & *image*)

BitmapImage::ColorUsage * BitmapImage::FindColor (UBYTE1 *red*, UBYTE1 *green*, UBYTE1 *blue*) `[private]`

void BitmapImage::FindColorUsage (const BitmapImage & *image*) `[private]`

void BitmapImage::FreeColorQuantizationData () `[private]`

void BitmapImage::GetRGB (unsigned int *row*, unsigned int *col*, UBYTE1 & *red*, UBYTE1 & *green*, UBYTE1 & *blue*) const

unsigned int BitmapImage::Height () const `[inline]`

UBYTE1 * BitmapImage::ImageData () `[inline]`

void BitmapImage::Initialize () `[protected]`

int BitmapImage::LargestColorRange (ColorArea & *area*) `[private]`

void BitmapImage::QuantizeSourceImage (const BitmapImage & *src*) `[private]`

unsigned int BitmapImage::QuantizedColor (UBYTE1 *red*, UBYTE1 *green*, UBYTE1 *blue*) `[private]`

void BitmapImage::SetProgressFunction (IMAGEPROGRESSFUNCTION *function*, void * *data*)

void BitmapImage::SetSize (unsigned int *cc*, unsigned int *bits*, unsigned int *ww*, unsigned int *hh*)

void BitmapImage::SplitAreaInHalf (unsigned int *depth*, unsigned int *retrydepth*, unsigned int *areaid*, unsigned int *splitcolor*) `[private]`

unsigned int BitmapImage::Width () const `[inline]`

BitmapImage & BitmapImage::operator= (const BitmapImage & *source*)

UBYTE1 * BitmapImage::operator[] (unsigned int *xx*) const `[inline]`

Member Data Documentation

unsigned int BitmapImage::bit_count `[private]`

unsigned int BitmapImage::color_area_count `[private]`

ColorArea * BitmapImage::color_areas `[private]`

unsigned int BitmapImage::color_count `[private]`

ColorMapEntry * BitmapImage::color_map `[private]`

ColorUsageTable * BitmapImage::color_usage `[private]`

unsigned char * BitmapImage::image_data `[private]`

unsigned int BitmapImage::image_height `[private]`

unsigned int BitmapImage::image_width `[private]`

void * BitmapImage::progress_data `[private]`

IMAGEPROGRESSFUNCTION BitmapImage::progress_function `[private]`

unsigned int BitmapImage::row_width `[private]`

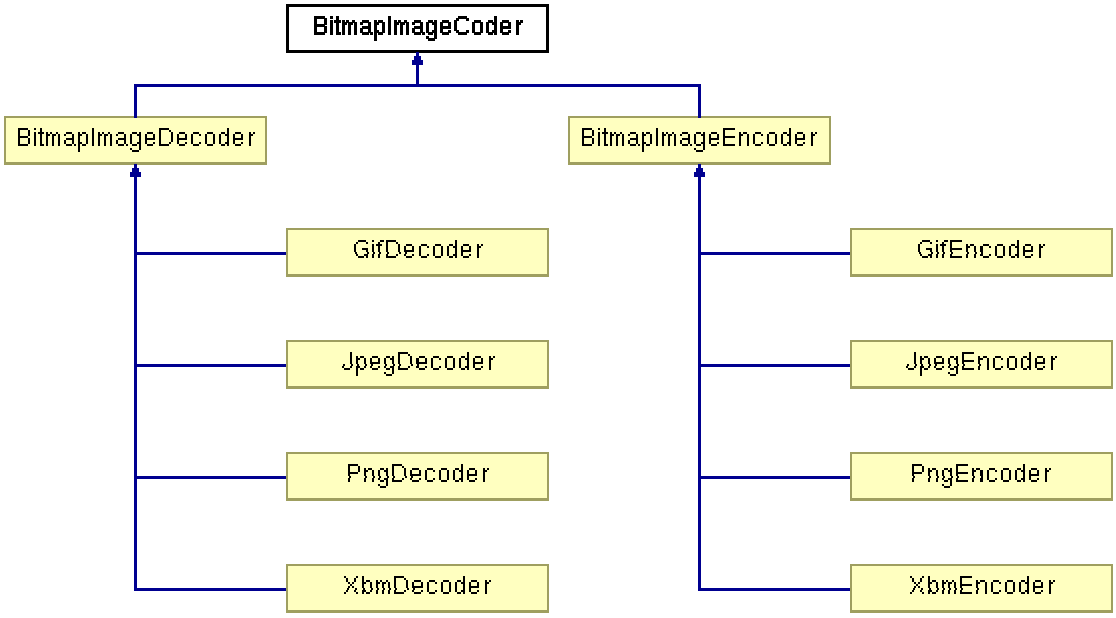
The documentation for this class was generated from the following files:

- **bitimage.h**
- **bitimage.cpp**

BitmapImageCoder Class Reference

`#include <bitimage.h>`

Inheritance diagram for BitmapImageCoder:



Public Methods

- **BitmapImageCoder** ()
- **BitmapImageCoder** (const BitmapImageCoder &source)
- virtual **~BitmapImageCoder** ()
- BitmapImageCoder& **operator=** (const BitmapImageCoder &source)
- void **SetProgressFunction** (**PROGRESSFUNCTION**, void *)
- virtual void **UpdateImage** ()

Protected Methods

- void **Initialize** ()
- void **DoCopy** (const BitmapImageCoder &source)

Protected Attributes

- **PROGRESSFUNCTION progress_function**
- void* **progress_data**

Constructor & Destructor Documentation

BitmapImageCoder::BitmapImageCoder () `[inline]`

BitmapImageCoder::BitmapImageCoder (const BitmapImageCoder & *source*) `[inline]`

BitmapImageCoder::~BitmapImageCoder () `[inline, virtual]`

Member Function Documentation

void BitmapImageCoder::DoCopy (const BitmapImageCoder & *source*) `[inline, protected]`

void BitmapImageCoder::Initialize () `[protected]`

Reimplemented in **GifDecoder** (*p.170*), **GifEncoder** (*p.175*), **JpegDecoder** (*p.214*), **JpegEncoder** (*p.234*), **PngDecoder** (*p.317*), **PngEncoder** (*p.328*), **XbmDecoder** (*p.374*), and **XbmEncoder** (*p.377*).

void BitmapImageCoder::SetProgressFunction (PROGRESSFUNCTION *func*, void * *data*) `[inline]`

void BitmapImageCoder::UpdateImage () `[inline, virtual]`

Reimplemented in **GifDecoder** (*p.171*), and **JpegDecoder** (*p.216*).

BitmapImageCoder & BitmapImageCoder::operator= (const BitmapImageCoder & *source*) `[inline]`

Member Data Documentation

void * BitmapImageCoder::progress_data `[protected]`

PROGRESSFUNCTION BitmapImageCoder::progress_function `[protected]`

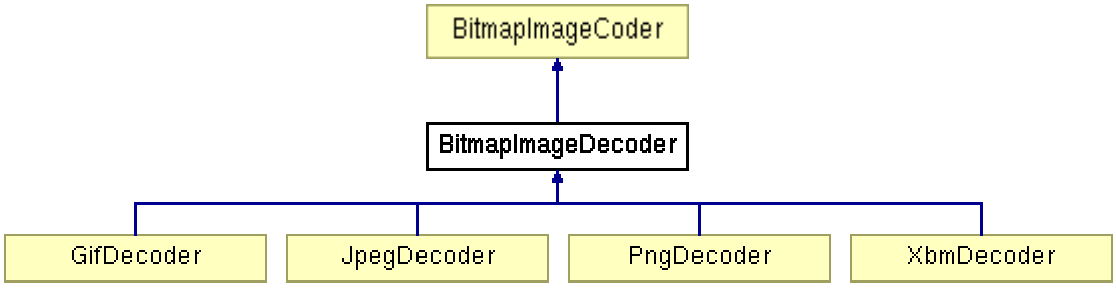
The documentation for this class was generated from the following file:

- **bitimage.h**

BitmapImageDecoder Class Reference

`#include <bitimage.h>`

Inheritance diagram for BitmapImageDecoder:



Public Methods

- virtual **~BitmapImageDecoder** ()
- virtual void **ReadImage** (istream &, **BitmapImage** &)=0

Constructor & Destructor Documentation

BitmapImageDecoder::~BitmapImageDecoder () `[inline, virtual]`

Member Function Documentation

void BitmapImageDecoder::ReadImage (istream & *strm*, BitmapImage & *image*) `[pure virtual]`

Reimplemented in **GifDecoder** (*p.170*), **JpegDecoder** (*p.215*), **PngDecoder** (*p.318*), and **XbmDecoder** (*p.374*).

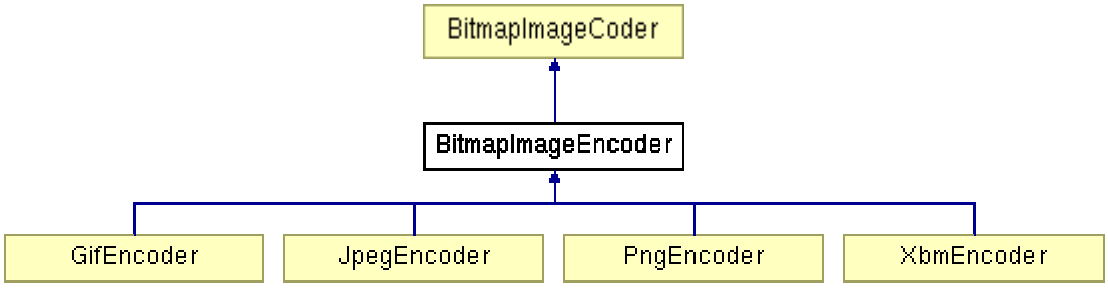
The documentation for this class was generated from the following file:

- **bitimage.h**

BitmapImageEncoder Class Reference

`#include <bitimage.h>`

Inheritance diagram for BitmapImageEncoder:



Public Methods

- virtual **~BitmapImageEncoder** ()
- virtual void **WriteImage** (ostream &, **BitmapImage** &)=0

Constructor & Destructor Documentation

BitmapImageEncoder::~BitmapImageEncoder () `[inline, virtual]`

Member Function Documentation

void BitmapImageEncoder::WriteImage (ostream & *strm*, BitmapImage & *timage*) `[pure virtual]`

Reimplemented in **GifEncoder** (*p.175*), **JpegEncoder** (*p.236*), **PngEncoder** (*p.330*), and **XbmEncoder** (*p.377*).

The documentation for this class was generated from the following file:

- **bitimage.h**

BlobColoring Class Reference

`#include <BlobColoring.h>`

Public Methods

- **BlobColoring** ()
- virtual **~BlobColoring** ()
- int **Process** (unsigned char *InputPict, unsigned int PictWidth, unsigned int PictHeight, unsigned int *Output, unsigned char BackgroundColor=0)
- int **CollectBlobInfo** (unsigned int *BlobImage, unsigned int ImWidth, unsigned int ImHeight, unsigned int BlobInfoArraySize, **BlobInfo** *BlobInfoArray)
- unsigned int **GetNbBlob** ()

Private Methods

- int **Blob** ()
- void **Propagate** (int CurPos, int CurPosX, int CurPosY, int CurBlob)
- void **FillOffsetTable** ()

Private Attributes

- int **PWidth**
- int **PHeight**
- int **PSize**
- unsigned char* **InPict**
- unsigned int* **OutPict**
- unsigned char **BackgrndColor**
- int **OffsetTable** [8]
- int **OffsetTableX** [8]
- int **OffsetTableY** [8]
- unsigned int **NbBlob**
- int* **PositionStack**
- short* **PositionXStack**
- short* **PositionYStack**
- unsigned char* **NgbrStack**

Constructor & Destructor Documentation

BlobColoring::BlobColoring ()

BlobColoring::~BlobColoring () `[virtual]`

Member Function Documentation

int BlobColoring::Blob () `[private]`

int BlobColoring::CollectBlobInfo (unsigned int * *BlobImage*, unsigned int *ImWidth*, unsigned int *ImHeight*, unsigned int *BlobInfoArraySize*, BlobInfo * *BlobInfoArray*)

void BlobColoring::FillOffsetTable () `[private]`

unsigned int BlobColoring::GetNbBlob () `[inline]`

int BlobColoring::Process (unsigned char * *InputPict*, unsigned int *PictWidth*, unsigned int *PictHeight*, unsigned int * *Output*, unsigned char *BackgroundColor* = 0)

void BlobColoring::Propagate (int *CurPos*, int *CurPosX*, int *CurPosY*, int *CurBlob*) `[private]`

Member Data Documentation

unsigned char BlobColoring::BackgrndColor `[private]`

unsigned char * BlobColoring::InPict `[private]`

unsigned int BlobColoring::NbBlob `[private]`

unsigned char * BlobColoring::NgbrStack `[private]`

int BlobColoring::OffsetTable `[private]`

int BlobColoring::OffsetTableX `[private]`

int BlobColoring::OffsetTableY `[private]`

unsigned int * BlobColoring::OutPict `[private]`

int BlobColoring::PHeight `[private]`

int BlobColoring::PSize `[private]`

int BlobColoring::PWidth `[private]`

int * BlobColoring::PositionStack `[private]`

short * BlobColoring::PositionXStack `[private]`

short * BlobColoring::PositionYStack `[private]`

The documentation for this class was generated from the following files:

- **BlobColoring.h**
- **BlobColoring.cpp**

BlobInfo Struct Reference

```
#include <BlobColoring.h>
```

Public Attributes

- float **CentreX**
- float **CentreY**
- int **NbPixels**
- short **MinX**
- short **MaxX**
- short **MinY**
- short **MaxY**

Member Data Documentation

float BlobInfo::CentreX

float BlobInfo::CentreY

short BlobInfo::MaxX

short BlobInfo::MaxY

short BlobInfo::MinX

short BlobInfo::MinY

int BlobInfo::NbPixels

The documentation for this struct was generated from the following file:

- **BlobColoring.h**

FormatBMP::bmpHeader Struct Reference

```
#include <FormatBMP.hpp>
```

Public Attributes

- **UBYTE4 width**
- **UBYTE4 height**
- **UBYTE4 nbBitsPerPixel**
- **UBYTE4 compressionType**
- **UBYTE4 imageSize**
- **UBYTE4 horizontalResolution**
- **UBYTE4 verticalResolution**
- **UBYTE4 nbColorIndexes**
- **UBYTE4 nbImportantColorIndexes**

Member Data Documentation

UBYTE4 FormatBMP::bmpHeader::compressionType

UBYTE4 FormatBMP::bmpHeader::height

UBYTE4 FormatBMP::bmpHeader::horizontalResolution

UBYTE4 FormatBMP::bmpHeader::imageSize

UBYTE4 FormatBMP::bmpHeader::nbBitsPerPixel

UBYTE4 FormatBMP::bmpHeader::nbColorIndexes

UBYTE4 FormatBMP::bmpHeader::nbImportantColorIndexes

UBYTE4 FormatBMP::bmpHeader::verticalResolution

UBYTE4 FormatBMP::bmpHeader::width

The documentation for this struct was generated from the following file:

- **FormatBMP.hpp**

Cognois::Calibration Class Reference

```
#include <Calibration.h>
```

Public Methods

- void **Heikkila** (const VPoints &surCible, const MPoints &surImages, **CameraModel** &cam) const
  Calibrate one camera Heikkila.

- void **Heikkila** (const VPoints &surCible, const **std::vector**< MPoints > &surImages, **CameraModel** &cam) const
  Calibrate a group of cameras using Heikkila.

- void **Heikkila** (const MPoints &surCible, const MPoints &surImages, **CameraModel** &cam) const
  Calibrate using a specefic planar pattern for each image.

- void **Heikkila** (const MPoints &surCibles, const **std::vector**< MPoints > &surImages, **CameraModel** &cam) const
  Calibrate using a specefic planar pattern for each image.

- void **Zhang** (const VPoints &surCible, const MPoints &surImages, **CameraModel** &cam) const
  Calibrate one camera using Zhang.

- void **Zhang** (const VPoints &surCible, const **std::vector**< MPoints > &surImages, **CameraModel** &cam) const
  Calibrate a group of cameras using Zhang.

- **Matrice HomographyToTransformation** (const VPoints &surCible, const VPoints &surImage, const **Matrice** &MparInt) const
  Get a transformation matrix from the homography and the intrinsic parameters (camera -> target).

- void **CombinerRT** (const MDoubles &parExt1, const MDoubles &parExt2, VDoubles &parExt) const
  Combine many samples of the extrinsic parametres into a closer-to-reality version.

Private Methods

- void **CalculerRt** (const VDoubles &parExt1, const VDoubles &parExt2, VDoubles &parExt) const
  Calculate extrinsic parameters, i.e. the position of camera 2 w/r to camera 1.

- **Matrice Homographie** (const VPoints &U, const VPoints &X) const
  Calculate an homography matrix.

Member Function Documentation

void Cognois::Calibration::CalculerRt (const VDoubles & *parExt1*, const VDoubles & *parExt2*, VDoubles & *parExt*) const `[private]`

Calculate extrinsic parameters, i.e. the position of camera 2 w/r to camera 1.

void Cognois::Calibration::CombinerRT (const MDoubles & *parExt1*, const MDoubles & *parExt2*, VDoubles & *parExt*) const

Combine many samples of the extrinsic parametres into a closer-to-reality version.

void Cognois::Calibration::Heikkila (const MPoints & *surCibles*, const std::vector< MPoints > & *surImages*, CameraModel & *cam*) const

Calibrate using a specefic planar pattern for each image.

void Cognois::Calibration::Heikkila (const MPoints & *surCibles*, const MPoints & *surImages*, CameraModel & *cam*) const

Calibrate using a specefic planar pattern for each image.

void Cognois::Calibration::Heikkila (const VPoints & *surCible*, const std::vector< MPoints > & *surImages*, CameraModel & *cam*) const

Calibrate a group of cameras using Heikkila.

void Cognois::Calibration::Heikkila (const VPoints & *surCible*, const MPoints & *surImages*, CameraModel & *cam*) const

Calibrate one camera Heikkila.

Matrice Cognois::Calibration::Homographie (const VPoints & *U*, const VPoints & *X*) const `[private]`

Calculate an homography matrix.

Donne l'homographie H tel que X = HU.

Matrice Cognois::Calibration::HomographyToTransformation (const VPoints & *surCible*, const VPoints & *surImage*, const Matrice & *MparInt*) const

Get a transformation matrix from the homography and the intrinsic parameters (camera -> target).

Retourne la transformation pour passer du referentiel cible vers le referentiel camera. i.e.: Exprimer la cible dans le repere de la camera **Matrice** (*p.263*) 4x4

void Cognois::Calibration::Zhang (const VPoints & *surCible*, const std::vector< MPoints > & *surImages*, CameraModel & *cam*) const

Calibrate a group of cameras using Zhang.

void Cognois::Calibration::Zhang (const VPoints & *surCible*, const MPoints & *surImages*, CameraModel & *cam*) const

Calibrate one camera using Zhang.

The documentation for this class was generated from the following files:

- **Calibration.h**

- **Calibration.cpp**

Calibration Class Reference

Calilbrate one, two or more cameras using known planar points and its images.

```
#include <Calibration.h>
```

Detailed Description

Calilbrate one, two or more cameras using known planar points and its images.

The documentation for this class was generated from the following file:

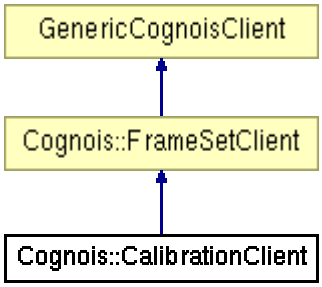
- **Calibration.h**

Cognois::CalibrationClient Class Reference

Read and write calibration data on the server.

`#include <CalibrationClient.h>`

Inheritance diagram for Cognois::CalibrationClient:



Public Methods

- **CalibrationClient** (int inArgc=0, char **inArgv=0)
  Constructor.

- CORBA::Boolean **SendPoints** (const VPoints &inPointsImage, const VPoints &inPointsPattern, unsigned long inFrameId, unsigned long inTimeStamp)
  Send calibration points image to the server for processing by a calibration client.

- CORBA::Boolean **GetLatestPoints** (MPoints &outPointsImage, VPoints &outPointsPattern, unsigned long &outFrameId, unsigned long &outTimeStamp)
  Get calibration points image from the server for processing.

- CORBA::Boolean **GetNewPoints** (MPoints &outPointsImage, VPoints &outPointsPattern, unsigned long &outFrameId, unsigned long &outTimeStamp, unsigned long inMaxWaitTime=2000)
  Get NEW calibration points image from the server for processing.

- CORBA::Boolean **SendPoints** (const VPointsXYZ &inPoints, unsigned long inFrameId, unsigned long inTimeStamp)
  Send 3D points to the server.

- CORBA::Boolean **GetLatestPoints** (MPoints &outPointsBlob, unsigned long &outFrameId, unsigned long &outTimeStamp)
  Get blob points from the server.

- CORBA::Boolean **SendCalibrationParameters** (const **CameraModel** &inParameters)
  Put the calibration parameters on the server.

- CORBA::Boolean **GetCalibrationParameters** (**CameraModel** &outParameters)
  Get the calibration parameters from the server.

- CORBA::Boolean **Send3dPoint** (const char *inId, const float &inXPosition, const float &inYPosition, const float &inZPosition, const float &inXVelocity, const float &inYVelocity, const float &inZVelocity, const unsigned long &inFrameId, const unsigned long &inTimeStamp)
  Upload a point with an identification to the server.

- CORBA::Boolean **Get3dPoint** (const char *inId, float &outXPosition, float &outYPosition, float &outZPosition, float &outXVelocity, float &outYVelocity, float &outZVelocity, unsigned long &outFrameId, unsigned long &outTimeStamp, const CORBA::Boolean &inLock=0)
  Download a point with an identification from the server.

- CORBA::Boolean **GetReal2DPoints** (unsigned long &outFrameId, unsigned long &outTimeStamp, VPoints &inoutBallCenters)
  Download 2D information on the ball.

- int **GetNumberOfCameraOnline** ()
  Return the number of cameras connected to the server.

Private Methods

- void **GetCalibrationPoints** (MPoints &outPointsImage, VPoints &outPointsPattern)
  Get calibration points from the current frame set.

Detailed Description

Read and write calibration data on the server.

Constructor & Destructor Documentation

Cognois::CalibrationClient::CalibrationClient (int *inArgc* = 0, char ** *inArgv* = 0)

Constructor.

Parameters:

inArgc is the number of arguments on the command line
inArgv is the command line agruments

Member Function Documentation

CORBA::Boolean Cognois::CalibrationClient::Get3dPoint (const char * *inId*, float & *outXPosition*, float & *outYPosition*, float & *outZPosition*, float & *outXVelocity*, float & *outYVelocity*, float & *outZVelocity*, unsigned long & *outFrameId*, unsigned long & *outTimeStamp*, const CORBA::Boolean & *inLock* = 0)

Download a point with an identification from the server.

Parameters:

inId is the identification of the point
outXPosition is the X position of the point
outYPosition is the Y position of the point
outZPosition is the Z position of the point
outXVelocity is the X velocity associated to the point
outYVelocity is the Y velocity associated to the point
outZVelocity is the Z velocity associated to the point
outFrameId is the frame id of the point
outTimeStamp is the time stamp of the point
inLock is set to true if the client wants to be locked until a new data is sent

Returns:

If the upload was successfull.

CORBA::Boolean Cognois::CalibrationClient::GetCalibrationParameters (CameraModel & *outParameters*)

Get the calibration parameters from the server.

Parameters:

outParameters are the most recent calibration parameters of the cameras

Returns:

If the download was successfull.

void Cognois::CalibrationClient::GetCalibrationPoints (MPoints & *outPointsImage*, VPoints & *outPointsPattern*) `[private]`

Get calibration points from the current frame set.

Parameters:

outPointsImage is a vector of vector of points to put the latest information
outPointsPattern is a vectof of the points on the calibration pattern

CORBA::Boolean Cognois::CalibrationClient::GetLatestPoints (MPoints & *outPointsBlob*, unsigned long & *outFrameId*, unsigned long & *outTimeStamp*)

Get blob points from the server.

CORBA::Boolean Cognois::CalibrationClient::GetLatestPoints (MPoints & *outPointsImage*, VPoints & *outPointsPattern*, unsigned long & *outFrameId*, unsigned long & *outTimeStamp*)

Get calibration points image from the server for processing.

Parameters:

outPointsImage is a vector of vector of points to put the latest information
outPointsPattern is a vectof of the points on the calibration pattern
outFrameId is the frame id of the points
outTimeStamp is the time stamp of the points

Returns:

If the frame set has never been downloaded from the server.

CORBA::Boolean Cognois::CalibrationClient::GetNewPoints (MPoints & *outPointsImage*, VPoints & *outPointsPattern*, unsigned long & *outFrameId*, unsigned long & *outTimeStamp*, unsigned long *inMaxWaitTime* = 2000)

Get NEW calibration points image from the server for processing.

Parameters:

outPointsImage is a vector of vector of points to put the latest information
outPointsPattern is a vectof of the points on the calibration pattern
outFrameId is the frame id of the points
outTimeStamp is the time stamp of the points
inMaxWaitingTime is the maximum time (in ms) to wait for a new frame set

Returns:

If the frame set has never been downloaded from the server.

int Cognois::CalibrationClient::GetNumberOfCameraOnline ()

Return the number of cameras connected to the server.

Returns:

The number of connected cameras.

CORBA::Boolean Cognois::CalibrationClient::GetReal2DPoints (unsigned long & *outFrameId*, unsigned long & *outTimeStamp*, VPoints & *inoutBallCenters*)

Download 2D information on the ball.

Parameters:

outFrameId is the frame id of the points
outTimeStamp is the time stamp of the points
inoutBallCenters is the position of the ball in all cameras.

Returns:

If the upload was successfull.

CORBA::Boolean Cognois::CalibrationClient::Send3dPoint (const char * *inId*, const float & *inXPosition*, const float & *inYPosition*, const float & *inZPosition*, const float & *inXVelocity*,

const float & *inYVelocity*, const float & *inZVelocity*, const unsigned long & *inFrameId*, const unsigned long & *inTimeStamp*)

Upload a point with an identification to the server.

Parameters:

inId is the identification of the point
inXPosition is the X position of the point
inYPosition is the Y position of the point
inZPosition is the Z position of the point
inXVelocity is the X velocity associated to the point
inYVelocity is the Y velocity associated to the point
inZVelocity is the Z velocity associated to the point
inFrameId is the frame ID of the point
inTimeStamp is the time stamp of the point

Returns:

If the upload was successfull.

CORBA::Boolean Cognois::CalibrationClient::SendCalibrationParameters (const CameraModel & *inParameters*)

Put the calibration parameters on the server.

Parameters:

inParameters are the calibration parameters of the cameras

Returns:

If the upload was successfull.

CORBA::Boolean Cognois::CalibrationClient::SendPoints (const VPointsXYZ & *inPoints*, unsigned long *inFrameId*, unsigned long *inTimeStamp*)

Send 3D points to the server.

CORBA::Boolean Cognois::CalibrationClient::SendPoints (const VPoints & *inPointsImage*, const VPoints & *inPointsPattern*, unsigned long *inFrameId*, unsigned long *inTimeStamp*)

Send calibration points image to the server for processing by a calibration client.

Parameters:

inPointsImage is a vector of the points on the image to send to the server
inPointsPattern is a vector of the points on the calibration pattern
inFrameId is the frame id of the points
inTimeStamp is the time stamp of the points

Returns:

If the upload was successfull.

The documentation for this class was generated from the following files:

- **CalibrationClient.h**
- **CalibrationClient.cpp**

Cognois::CameraClientController Class Reference

Get the current mode from the server and use the appropriate processing object.

`#include <CameraClientController.h>`

Public Methods

- **CameraClientController** (**OnlineProcessing** *inGrabber, **FrameSetClient** *inServer)
  Constructor.

- void **Begin** ()
  Begin the online processing and pool the server to get the modes.

- void **AddMode** (const std::string &inMode, **ProcessingObject** *inProcessor)
  Set the processing object for a mode.

- void **ListModes** () const
  Print all the modes.

Private Methods

- **ProcessingObject*** **Processor** (const std::string &inMode) const
  Get the processing object corresponding to a mode identifier.

Private Attributes

- **std::vector**<**ProcessingObject***> **mProcessors**
  List of valid processors.

- **std::vector**<std::string> **mModes**
  List of valid modes.

- **OnlineProcessing*** **mGrabber**
  Processing object to control.

- **FrameSetClient*** **mServer**
  Client to get modes from.

- std::string **mCurrentMode**
  Last valid mode received.

Detailed Description

Get the current mode from the server and use the appropriate processing object.

Constructor & Destructor Documentation

Cognois::CameraClientController::CameraClientController (OnlineProcessing * *inGrabber*, FrameSetClient * *inServer*)

Constructor.

Parameters:

inArgc is the number of arguments on the command line
inArgv is the command line agruments

Member Function Documentation

void Cognois::CameraClientController::AddMode (const std::string & *inMode*, ProcessingObject * *inProcessor*)

Set the processing object for a mode.

Parameters:

inMode is the mode identifier
inProcessor is the relevant processing object

void Cognois::CameraClientController::Begin ()

Begin the online processing and pool the server to get the modes.

void Cognois::CameraClientController::ListModes () const

Print all the modes.

Cognois::ProcessingObject * Cognois::CameraClientController::Processor (const std::string & *inMode*) const `[inline, private]`

Get the processing object corresponding to a mode identifier.

Parameters:

inMode is the mode identifier

Returns:

A pointer to the corresponding processing object (or null if mode is not found)

Member Data Documentation

std::string Cognois::CameraClientController::mCurrentMode `[private]`

Last valid mode received.

OnlineProcessing * Cognois::CameraClientController::mGrabber `[private]`

Processing object to control.

std::vector< std::string > Cognois::CameraClientController::mModes<std::string> `[private]`

List of valid modes.

std::vector< ProcessingObject *> Cognois::CameraClientController::mProcessors<ProcessingObject*> `[private]`

List of valid processors.

FrameSetClient * Cognois::CameraClientController::mServer `[private]`

Client to get modes from.

The documentation for this class was generated from the following files:

- **CameraClientController.h**
- **CameraClientController.cpp**

Cognois::CameraModel Class Reference

Calibration (*p.46*) parameters of n cameras in a stereo environment.

`#include <CameraModel.h>`

Public Methods

- **CameraModel** (int nbCams=1)
  Constructeur: determine la taille des vecteurs (nombre de cameras).

- int **Size** () const
  Retourne/fixe le nombre de cameras.

- void **Size** (int s)
- long **Version** () const
- **Matrice MatriceIntrinseque** (unsigned int i) const
  ***Matrice** (p.263) des parametres intrinseques de la camera i.*

- **Matrice MatriceRotation** (unsigned int i) const
  ***Matrice** (p.263) des parametres de rotation de la camera i.*

- **Matrice MatriceTranslation** (unsigned int i) const
  ***Matrice** (p.263) des parametres de translation de la camera i.*

- void **Rt** (**Matrice** mParExt, unsigned int cameraID)
  *Affecter un matrice de Transformation a la camera cameraID du **CameraModel** (p.59) courant.*

- VDoubles& **GetParametresIntrinseques** (unsigned int cameraID)
- const VDoubles& **GetParametresIntrinseques** (unsigned int cameraID) const
- VDoubles& **GetParametresExtrinseques** (unsigned int cameraID)
- const VDoubles& **GetParametresExtrinseques** (unsigned int cameraID) const
- void **SingleCam2MultiCam** (CameraModel inSingleCam, unsigned int inCameraID)
  Affecter une matrice de parametre intrinseques a la cameraID (pas de distorsion).

- void **SetParametresIntrinseques** (const CameraModel &inCameraModel, unsigned int inCameraID)
- **Matrice TransformInverse** (const **Matrice** &inTransform) const
  Calcul l'inverse d'une matrice de transformation.

- void **EcrireParametresIntrinseques** (unsigned int i, ostream &out) const
  Afficher les resultats du calibrage (avec formatage): parametres intrinseques.

- void **EcrireParametresExtrinseques** (unsigned int i, ostream &out) const
  Afficher les resultats du calibrage (avec formatage): parametres extrinseques.

- void **EcrireParametres** (ostream &out) const
  Afficher les resultats du calibrage (avec formatage): tous les parametres.

- CameraModel **MultiCam2SingleCam** (unsigned int inCameraID)
  Get the camera model of a single camera of the system.

- **Point Dedistorsion** (const **Point** &PointDist, unsigned int i) const
  Correction de la distorsion radiale d'un point dans la camera i.

- **Point Distorsion** (const **Point** &PointDist, unsigned int i) const
  Ajout de la distorsion radiale d'un point dans la camera i.

- int **ValiderCorrespondance** (const VPoints &mi) const
  Valider la mise en correspondance (peut contenir des mi hors des images).

- PointXYZ **CalculerXYZ** (const VPoints &mi) const
  Utiliser les parametres de calibrage pour calculer un point 3D.

- VPoints **Projeter** (const PointXYZ &p) const
  Projeter un point de l'espace dans les images.

- **Matrice MatriceProjection** (unsigned int i) const
- **Matrice MatriceEssentielle** (unsigned int i, unsigned int j) const
  ***Matrice** (p.263) essentielle/fondamentale reliant la camera i a la camera j (Eij).*

- **Matrice MatriceFondamentale** (unsigned int i, unsigned int j) const
- void **CalculerFondamentales** ()
  Pre-calculer toutes les matrices fondamentales.

- **Matrice DroiteEpipolaire** (const **Point** &m, unsigned int i, unsigned int j) const
  Droite epipolaire dans la camera j, correspondant au point m dans la camera i.

Private Types

- typedef **std::vector**<**Matrice**> **VMatrices**

Private Methods

- int **Dedistorsion** (const **Point** &Xd, const VDoubles &Cd, **Point** &Xu) const
- int **Distorsion** (const **Point** &Xu, const VDoubles &Cd, **Point** &Xd) const
- **Point ImageToCamera** (const **Point** &U, unsigned int i) const
- **Point CameraToImage** (const **Point** &X, unsigned int i) const

Private Attributes

- MDoubles **parInt**
- MDoubles **parExt**
- long **version**
- **std::vector**<**VMatrices**> **Fij**
- bool **Fij_valid**

Friends

- class **Calibration**
- class **CalibrationClient**
- ostream& **operator**<< (ostream &out, const CameraModel &cam)
- istream& **operator**>> (istream &in, CameraModel &cam)

Detailed Description

Calibration (*p.46*) parameters of n cameras in a stereo environment.

Intrinsic and extrinsic parameters of cameras forming a stereo environment. When representing one camera, contains its intrinsic parameters 'n' calibration pattern positions. When representing multiple cameras, contains 'n' intrinsic parameters, the position of the first camera in a global system and 'n-1' camera positions with respect with the first one.

Thus, the projection equation is: mi = Ai(Inv(Ri) * X - Inv(Ri)*Ti) and the line of sight is given by: li = Ri * Inv(Ai)* mi + Ti

---

Member Typedef Documentation

typedef std::vector< Matrice > Cognois::CameraModel::VMatrices<Matrice> `[private]`

Constructor & Destructor Documentation

Cognois::CameraModel::CameraModel (int *nbCams* = 1) `[inline]`

Constructeur: determine la taille des vecteurs (nombre de cameras).

Member Function Documentation

void Cognois::CameraModel::CalculerFondamentales ()

Pre-calculer toutes les matrices fondamentales.

PointXYZ Cognois::CameraModel::CalculerXYZ (const VPoints & *inMi*) const

Utiliser les parametres de calibrage pour calculer un point 3D.

Point Cognois::CameraModel::CameraToImage (const Point & *X*, unsigned int *i*) const `[inline, private]`

int Cognois::CameraModel::Dedistorsion (const Point & *Xd*, const VDoubles & *Cd*, Point & *Xu*) const `[private]`

Point Cognois::CameraModel::Dedistorsion (const Point & *Ud*, unsigned int *i*) const

Correction de la distorsion radiale d'un point dans la camera i.

int Cognois::CameraModel::Distorsion (const Point & *Xu*, const VDoubles & *Cd*, Point & *Xd*) const `[private]`

Point Cognois::CameraModel::Distorsion (const Point & *Uu*, unsigned int *i*) const

Ajout de la distorsion radiale d'un point dans la camera i.

Matrice Cognois::CameraModel::DroiteEpipolaire (const Point & *m*, unsigned int *i*, unsigned int *j*) const

Droite epipolaire dans la camera j, correspondant au point m dans la camera i.

void Cognois::CameraModel::EcrireParametres (ostream & *out*) const

Afficher les resultats du calibrage (avec formatage): tous les parametres.

void Cognois::CameraModel::EcrireParametresExtrinseques (unsigned int *i*, ostream & *out*) const

Afficher les resultats du calibrage (avec formatage): parametres extrinseques.

void Cognois::CameraModel::EcrireParametresIntrinseques (unsigned int *i*, ostream & *out*) const

Afficher les resultats du calibrage (avec formatage): parametres intrinseques.

const VDoubles & Cognois::CameraModel::GetParametresExtrinseques (unsigned int *cameraID*) const `[inline]`

VDoubles & Cognois::CameraModel::GetParametresExtrinseques (unsigned int *cameraID*) `[inline]`

const VDoubles & Cognois::CameraModel::GetParametresIntrinseques (unsigned int *cameraID*) const `[inline]`

VDoubles & Cognois::CameraModel::GetParametresIntrinseques (unsigned int *cameraID*) `[inline]`

Point Cognois::CameraModel::ImageToCamera (const Point & *U*, unsigned int *i*) const `[inline, private]`

Matrice Cognois::CameraModel::MatriceEssentielle (unsigned int *i*, unsigned int *j*) const

Matrice (*p.263*) essentielle/fondamentale reliant la camera i a la camera j (Eij).

Matrice Cognois::CameraModel::MatriceFondamentale (unsigned int *i*, unsigned int *j*) const

Matrice Cognois::CameraModel::MatriceIntrinseque (unsigned int *i*) const

Matrice (*p.263*) des parametres intrinseques de la camera i.

Matrice Cognois::CameraModel::MatriceProjection (unsigned int *i*) const

Matrice Cognois::CameraModel::MatriceRotation (unsigned int *i*) const

Matrice (*p.263*) des parametres de rotation de la camera i.

Matrice Cognois::CameraModel::MatriceTranslation (unsigned int *i*) const

Matrice (*p.263*) des parametres de translation de la camera i.

Cognois::CameraModel Cognois::CameraModel::MultiCam2SingleCam (unsigned int *inCameraID*)

Get the camera model of a single camera of the system.

VPoints Cognois::CameraModel::Projeter (const PointXYZ & *p*) const

Projeter un point de l'espace dans les images.

void Cognois::CameraModel::Rt (Matrice *mParExt*, unsigned int *inCameraID*)

Affecter un matrice de Transformation a la camera cameraID du **CameraModel** (*p.59*) courant.

void Cognois::CameraModel::SetParametresIntrinseques (const CameraModel & *inCameraModel*, unsigned int *inCameraID*)

void Cognois::CameraModel::SingleCam2MultiCam (CameraModel *inSingleCam*, unsigned int *inCameraID*)

Affecter une matrice de parametre intrinseques a la cameraID (pas de distorsion).

void Cognois::CameraModel::Size (int *s*) `[inline]`

int Cognois::CameraModel::Size () const `[inline]`

Retourne/fixe le nombre de cameras.

Matrice Cognois::CameraModel::TransformInverse (const Matrice & *inTransform*) const

Calcul l'inverse d'une matrice de transformation.

int Cognois::CameraModel::ValiderCorrespondance (const VPoints & *mi*) const

Valider la mise en correspondance (peut contenir des mi hors des images).

long Cognois::CameraModel::Version () const `[inline]`

Friends And Related Function Documentation

class Calibration `[friend]`

class CalibrationClient `[friend]`

ostream & operator<< (ostream & *out*, const CameraModel & *cam*) `[friend]`

istream & operator>> (istream & *in*, CameraModel & *cam*) `[friend]`

Member Data Documentation

std::vector< VMatrices > Cognois::CameraModel::Fij<VMatrices> `[private]`

bool Cognois::CameraModel::Fij_valid `[private]`

MDoubles Cognois::CameraModel::parExt `[private]`

MDoubles Cognois::CameraModel::parInt `[private]`

long Cognois::CameraModel::version `[private]`

The documentation for this class was generated from the following files:

- **CameraModel.h**
- **CameraModel.cpp**

cilist Struct Reference

```
#include <F2C.H>
```

Public Attributes

- **flag cierr**
- **ftnint ciunit**
- **flag ciend**
- char* **cifmt**
- **ftnint cirec**

Member Data Documentation

flag cilist::ciend

flag cilist::cierr

char * cilist::cifmt

ftnint cilist::cirec

ftnint cilist::ciunit

The documentation for this struct was generated from the following file:

- **F2C.H**

cllist Struct Reference

```
#include <F2C.H>
```

Public Attributes

- **flag cerr**
- **ftnint cunit**
- char* **csta**

Member Data Documentation

flag cllist::cerr

char * cllist::csta

ftnint cllist::cunit

The documentation for this struct was generated from the following file:

- **F2C.H**

BitmapImage::ColorArea Struct Reference

Public Attributes

- struct {
- **UBYTE1 low**
- **UBYTE1 high**
- } **color_values**
- unsigned int **color_count**
- unsigned int **pixel_count**

Member Data Documentation

unsigned int BitmapImage::ColorArea::color_count

struct { ... } BitmapImage::ColorArea::color_values

UBYTE1 BitmapImage::ColorArea::high

UBYTE1 BitmapImage::ColorArea::low

unsigned int BitmapImage::ColorArea::pixel_count

The documentation for this struct was generated from the following file:

- **bitimage.h**

BitmapImage::ColorMapEntry Struct Reference

```
#include <bitimage.h>
```

Public Attributes

- **UBYTE1 blue**
- **UBYTE1 green**
- **UBYTE1 red**

Member Data Documentation

UBYTE1 BitmapImage::ColorMapEntry::blue

UBYTE1 BitmapImage::ColorMapEntry::green

UBYTE1 BitmapImage::ColorMapEntry::red

The documentation for this struct was generated from the following file:

- **bitimage.h**

BitmapImage::ColorUsage Struct Reference

```
#include <bitimage.h>
```

Public Attributes

- **UBYTE1 colors** [3]
- **UBYTE4 usage**
- ColorUsage* **next** [3]

Member Data Documentation

UBYTE1 BitmapImage::ColorUsage::colors

ColorUsage * BitmapImage::ColorUsage::next

UBYTE4 BitmapImage::ColorUsage::usage

The documentation for this struct was generated from the following file:

- **bitimage.h**

BitmapImage::ColorUsageTable Struct Reference

Public Attributes

- **ColorUsage* lists** [256][3]
- unsigned int **color_count**

Member Data Documentation

unsigned int BitmapImage::ColorUsageTable::color_count

ColorUsage * BitmapImage::ColorUsageTable::lists

The documentation for this struct was generated from the following file:

- **bitimage.h**

ColourSpaceConverter Class Reference

`#include <ColourSpaceConverter.h>`

Public Methods

- **ColourSpaceConverter** ()
- virtual **~ColourSpaceConverter** ()
- int **RGB2HSV** (unsigned char R, unsigned char G, unsigned char B, double &H, double &S, double &V)
- int **HSV2RGB** (double H, double S, double V, unsigned char &R, unsigned char &G, unsigned char &B)

Private Attributes

- double **pi_x_2**
- double **pi_div_3**

Constructor & Destructor Documentation

ColourSpaceConverter::ColourSpaceConverter ()

ColourSpaceConverter::~ColourSpaceConverter () `[virtual]`

Member Function Documentation

int ColourSpaceConverter::HSV2RGB (double *H*, double *S*, double *V*, unsigned char & *R*, unsigned char & *G*, unsigned char & *B*)

int ColourSpaceConverter::RGB2HSV (unsigned char *R*, unsigned char *G*, unsigned char *B*, double & *H*, double & *S*, double & *V*)

Member Data Documentation

double ColourSpaceConverter::pi_div_3 `[private]`

double ColourSpaceConverter::pi_x_2 `[private]`

The documentation for this class was generated from the following files:

- **ColourSpaceConverter.h**

- **ColourSpaceConverter.cpp**

complex Struct Reference

```
#include <F2C.H>
```

Public Attributes

- **real r**
- **real i**

Member Data Documentation

real complex::i

real complex::r

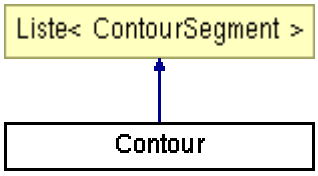
The documentation for this struct was generated from the following file:

- **F2C.H**

Contour Class Reference

`#include <Contour.hpp>`

Inheritance diagram for Contour:



Public Methods

- **Contour** ()
- Contour& **EdgeLinking** (**ImageBinaire** &Tmin, **ImageBinaire** &Tmax)
- Contour& **EdgeLinking** (**ImageBinaire** &Tim)
- void **SegmentsDroite** (unsigned int maxangle=10, double maxdist=10, double tolinter=5, double tolcourbe=0.1)
- void **AjusterSegments** (double tol=5)
- void **Abouter** (const unsigned int maxangle=10, const double maxdist=10)
- void **Filtrer** (const double len=10)
- void **RegrouperUnitaires** (double maxdist=10)
- void **SegmenterCourbes** (const double maxerreur=0.1)
- void **TrouverIntersections** (double tol=5, int ang=10, double distmin=5)
- **Liste**<**Point**> **HoraudJunctionDetection** (**Liste**< **HoraudRelation** > &, int e1=15, int w=5, double max_o=5) const
- void **HoraudCurveDetection** (const **Liste**< **HoraudRelation** > &, **Liste**< **HoraudRelation** > &,int e2=0, int e3=90) const
- void **Vider** ()
- void **EcrireSegments** (ostream &out) const

Public Attributes

- **Liste**<**Point**> **intersections**

Protected Methods

- void **HoraudCurveSuivant** (**HoraudRelation** &,int th, int e2, int e3, **ContourSegment** *, **Liste**< **HoraudRelation** > &) const

Friends

- ostream& **operator<<** (ostream &out, const Contour &c)
- istream& **operator>>** (istream &in, Contour &c)

Constructor & Destructor Documentation

Contour::Contour () `[inline]`

Member Function Documentation

void Contour::Abouter (const unsigned int *maxangle* = 10, const double *_maxdist* = 10)

void Contour::AjusterSegments (double *tol* = 5)

void Contour::EcrireSegments (ostream & *out*) const

Contour & Contour::EdgeLinking (ImageBinaire & *Tim*)

Contour & Contour::EdgeLinking (ImageBinaire & *Tmin*, ImageBinaire & *Tmax*)

void Contour::Filtrer (const double *len* = 10)

void Contour::HoraudCurveDetection (const Liste< HoraudRelation > & *_jonctions*, Liste< HoraudRelation > & *courbes*, int *e2* = 0, int *e3* = 90) const

void Contour::HoraudCurveSuivant (HoraudRelation & *courbe_courante*, int *theta1*, int *e2*, int *e3*, ContourSegment * *lo*, Liste< HoraudRelation > & *jonctions*) const `[protected]`

Liste< Point > Contour::HoraudJunctionDetection (Liste< HoraudRelation > & *jonctions*, int *e* = 15, int *w* = 5, double *max_overlap* = 5) const

void Contour::RegrouperUnitaires (double *maxdist* = 10)

void Contour::SegmenterCourbes (const double *maxerreur* = 0.1)

void Contour::SegmentsDroite (unsigned int *maxangle* = 10, double *maxdist* = 10, double *tolinter* = 5, double *tolcourbe* = 0.1)

void Contour::TrouverIntersections (double *tol* = 5, int *ang* = 10, double *distmin* = 5)

void Contour::Vider ()

Friends And Related Function Documentation

ostream & operator<< (ostream & *out*, const Contour & *c*) `[friend]`

istream & operator>> (istream & *in*, Contour & *c*) `[friend]`

Member Data Documentation

Liste< Point > Contour::intersections<Point>

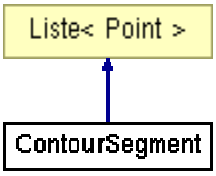
The documentation for this class was generated from the following files:

- **Contour.hpp**
- **Contour.cpp**

ContourSegment Class Reference

`#include <Contour.hpp>`

Inheritance diagram for ContourSegment:



Public Methods

- **Point**& **Origine** () const
- **Point**& **Fin** () const
- void **EquationDroite** (int &a, int &b, int &c) const
- double **LongueurDroite** () const
- int **Orientation** () const
- double **Distance** (const **Point** &) const
- ContourSegment **Separer** (const **Point** &ps)
- double **DroiteMaxErreur** (**Point** &pme) const
- int **DiffOrientation** (const ContourSegment &) const
- **Point Intersection** (const ContourSegment &) const
- int **Intersection** (const ContourSegment &, **Point** &, double tol=0) const
- void **EcrireExtremites** (ostream &out) const

Friends

- ostream& **operator**<< (ostream &out, const ContourSegment &sc)
- istream& **operator**>> (istream &in, ContourSegment &sc)

Member Function Documentation

int ContourSegment::DiffOrientation (const ContourSegment & *seg*) const

double ContourSegment::Distance (const Point & *pnt*) const

double ContourSegment::DroiteMaxErreur (Point & *pme*) const

void ContourSegment::EcrireExtremites (ostream & *out*) const

void ContourSegment::EquationDroite (int & *a*, int & *b*, int & *c*) const

Point & ContourSegment::Fin () const

int ContourSegment::Intersection (const ContourSegment & *seg*, Point & *ptint*, double *tol* = 0) const

Point ContourSegment::Intersection (const ContourSegment & *seg*) const

double ContourSegment::LongueurDroite () const

int ContourSegment::Orientation () const

Point & ContourSegment::Origine () const

ContourSegment ContourSegment::Separer (const Point & *ps*)

Friends And Related Function Documentation

ostream & operator<< (ostream & *out*, const ContourSegment & *sc*) `[friend]`

istream & operator>> (istream & *in*, ContourSegment & *sc*) `[friend]`

The documentation for this class was generated from the following files:

- **Contour.hpp**
- **Contour.cpp**

CouleurHSI Class Reference

```
#include <ImageRGB.hpp>
```

Public Methods

- **CouleurHSI** (double _h=0, double _s=0, double _i=0)
- const CouleurHSI& **operator=** (const unsigned long val)
- const CouleurHSI& **operator=** (const **CouleurRGB** &)
- **operator unsigned long** () const
- int **Valider** () const
- unsigned char **Red** () const
- unsigned char **Green** () const
- unsigned char **Blue** () const
- unsigned long **getInt** () const
- unsigned char **getChar** () const

Public Attributes

- double **hue**
- double **saturation**
- double **intensite**

Friends

- ostream& **operator<<** (ostream &out, const CouleurHSI &p)
- istream& **operator>>** (istream &in, CouleurHSI &p)

Constructor & Destructor Documentation

CouleurHSI::CouleurHSI (double _*h* = 0, double _*s* = 0, double _*i* = 0)

Member Function Documentation

unsigned char CouleurHSI::Blue () const

unsigned char CouleurHSI::Green () const

unsigned char CouleurHSI::Red () const

int CouleurHSI::Valider () const

unsigned char CouleurHSI::getChar () const

unsigned long CouleurHSI::getInt () const

CouleurHSI::operator unsigned long () const

const CouleurHSI & CouleurHSI::operator= (const CouleurRGB & *rgb*)

const CouleurHSI & CouleurHSI::operator= (const unsigned long *val*)

Friends And Related Function Documentation

ostream & operator<< (ostream & *out*, const CouleurHSI & *p*) `[friend]`

istream & operator>> (istream & *in*, CouleurHSI & *p*) `[friend]`

Member Data Documentation

double CouleurHSI::hue

double CouleurHSI::intensite

double CouleurHSI::saturation

The documentation for this class was generated from the following files:

- **ImageRGB.hpp**
- **ImageRGB.cpp**

CouleurRGB Class Reference

```
#include <ImageRGB.hpp>
```

Public Methods

- **CouleurRGB** (unsigned char _r=0, unsigned char _g=0, unsigned char _b=0)
- const CouleurRGB& **operator=** (const unsigned long val)
- const CouleurRGB& **operator=** (const **CouleurHSI** &)
- **operator unsigned long** () const
- const CouleurRGB& **operator+=** (const CouleurRGB &)
- CouleurRGB **operator+** (const CouleurRGB &) const
- const CouleurRGB& **operator-=** (const CouleurRGB &)
- CouleurRGB **operator-** (const CouleurRGB &) const
- double **Intensite** () const
- double **Saturation** () const
- double **Hue** () const
- unsigned long **getInt** () const
- unsigned char **getChar** () const

Public Attributes

- unsigned char **redValue**
- unsigned char **greenValue**
- unsigned char **blueValue**

Friends

- ostream& **operator<<** (ostream &out, const CouleurRGB &p)
- istream& **operator>>** (istream &in, CouleurRGB &p)

Constructor & Destructor Documentation

CouleurRGB::CouleurRGB (unsigned char *_r* = 0, unsigned char *_g* = 0, unsigned char *_b* = 0)

Member Function Documentation

double CouleurRGB::Hue () const

double CouleurRGB::Intensite () const

double CouleurRGB::Saturation () const

unsigned char CouleurRGB::getChar () const

unsigned long CouleurRGB::getInt () const

CouleurRGB::operator unsigned long () const

CouleurRGB CouleurRGB::operator+ (const CouleurRGB & *rh*) const

const CouleurRGB & CouleurRGB::operator+= (const CouleurRGB & *rh*)

CouleurRGB CouleurRGB::operator- (const CouleurRGB & *rh*) const

const CouleurRGB & CouleurRGB::operator-= (const CouleurRGB & *rh*)

const CouleurRGB & CouleurRGB::operator= (const CouleurHSI & *hsi*)

const CouleurRGB & CouleurRGB::operator= (const unsigned long *val*)

Friends And Related Function Documentation

ostream & operator<< (ostream & *out*, const CouleurRGB & *p*) `[friend]`

istream & operator>> (istream & *in*, CouleurRGB & *p*) `[friend]`

Member Data Documentation

unsigned char CouleurRGB::blueValue

unsigned char CouleurRGB::greenValue

unsigned char CouleurRGB::redValue

The documentation for this class was generated from the following files:

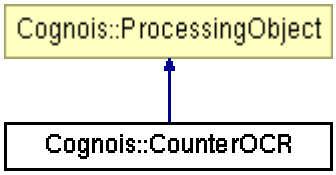
- **ImageRGB.hpp**
- **ImageRGB.cpp**

Cognois::CounterOCR Class Reference

Optical character recognition of the 5 digits of the HP 5214L Preset Counter.

`#include <CounterOCR.h>`

Inheritance diagram for Cognois::CounterOCR:



Public Methods

- **CounterOCR** (MIL_ID &inCalibrationImage, char *inFontFilename, **MILSys** &inMilSys, **FrameSetClient** *inCognoisServer=0)
  Constructor with interactive calibration of the font.

- **~CounterOCR** ()
- void **Process** (MIL_ID &inMilImage, unsigned long &inFrameID, unsigned long &inTimeStamp)
  Find a calibration pattern in an image.

Private Attributes

- MIL_ID **mFont**
- MIL_ID **mCounterImage**
- MIL_ID **mGrayImage**
- MIL_ID **mOCRResult**
- **FrameSetClient*** **mCognoisServer**
  Cognois *(p.2) server interface.*

- long **mTopX**
- long **mTopY**
- long **mBotX**
- long **mBotY**
  Bounding box of the number.

Detailed Description

Optical character recognition of the 5 digits of the HP 5214L Preset Counter.

Constructor & Destructor Documentation

Cognois::CounterOCR::CounterOCR (MIL_ID & *inCalibrationImage*, char * *inFontFilename*, MILSys & *inMilSys*, FrameSetClient * *inCognoisServer* = 0)

Constructor with interactive calibration of the font.

Cognois::CounterOCR::~CounterOCR ()

Member Function Documentation

void Cognois::CounterOCR::Process (MIL_ID & *inMilImage*, unsigned long & *inFrameID*, unsigned long & *inTimeStamp*) `[virtual]`

Find a calibration pattern in an image.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

Member Data Documentation

long Cognois::CounterOCR::mBotX `[private]`

long Cognois::CounterOCR::mBotY `[private]`

Bounding box of the number.

FrameSetClient * Cognois::CounterOCR::mCognoisServer `[private]`

Cognois (*p.2*) server interface.

MIL_ID Cognois::CounterOCR::mCounterImage `[private]`

MIL_ID Cognois::CounterOCR::mFont `[private]`

MIL_ID Cognois::CounterOCR::mGrayImage `[private]`

MIL_ID Cognois::CounterOCR::mOCRResult `[private]`

long Cognois::CounterOCR::mTopX `[private]`

long Cognois::CounterOCR::mTopY `[private]`

The documentation for this class was generated from the following files:

- **CounterOCR.h**
- **CounterOCR.cpp**

DirectMilAccess Struct Reference

Public Methods

- **DirectMilAccess** (const MIL_ID &_milImage)
- **~DirectMilAccess** ()
- unsigned char& **RedValue** (int x, int y)
- unsigned char& **GreenValue** (int x, int y)
- unsigned char& **BlueValue** (int x, int y)
- void **AllValues** (int x, int y, unsigned char &red, unsigned char &green, unsigned char &blue)

Public Attributes

- MIL_ID **MilImageRed**
- MIL_ID **MilImageGreen**
- MIL_ID **MilImageBlue**
- unsigned char* **MilRedPtr**
- unsigned char * **MilGreenPtr**
- unsigned char * **MilBluePtr**
- long **MilImageWorkPitch**

Constructor & Destructor Documentation

DirectMilAccess::DirectMilAccess (const MIL_ID & *_milImage*) `[inline]`

DirectMilAccess::~DirectMilAccess () `[inline]`

Member Function Documentation

void DirectMilAccess::AllValues (int *x*, int *y*, unsigned char & *red*, unsigned char & *green*, unsigned char & *blue*) `[inline]`

unsigned char & DirectMilAccess::BlueValue (int *x*, int *y*) `[inline]`

unsigned char & DirectMilAccess::GreenValue (int *x*, int *y*) `[inline]`

unsigned char & DirectMilAccess::RedValue (int *x*, int *y*) `[inline]`

Member Data Documentation

unsigned char * DirectMilAccess::MilBluePtr

unsigned char * DirectMilAccess::MilGreenPtr

MIL_ID DirectMilAccess::MilImageBlue

MIL_ID DirectMilAccess::MilImageGreen

MIL_ID DirectMilAccess::MilImageRed

long DirectMilAccess::MilImageWorkPitch

unsigned char * DirectMilAccess::MilRedPtr

The documentation for this struct was generated from the following file:

- **ExtractionDeCible.cpp**

dmat Struct Reference

```
#include <MATRIX.H>
```

Public Attributes

- int **lb1**
- int **ub1**
- int **lb2**
- int **ub2**
- char* **mat_sto**
- double** **el**

Member Data Documentation

double ** dmat::el

int dmat::lb1

int dmat::lb2

char * dmat::mat_sto

int dmat::ub1

int dmat::ub2

The documentation for this struct was generated from the following file:

- **MATRIX.H**

doublecomplex Struct Reference

```
#include <F2C.H>
```

Public Attributes

- **doublereal r**
- **doublereal i**

Member Data Documentation

doublereal doublecomplex::i

doublereal doublecomplex::r

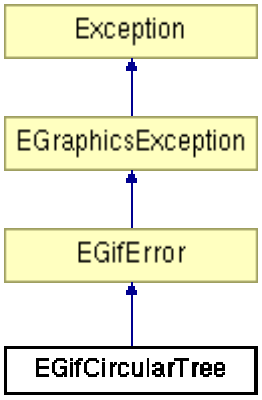
The documentation for this struct was generated from the following file:

- **F2C.H**

EGifCircularTree Class Reference

`#include <gifexcep.h>`

Inheritance diagram for EGifCircularTree:



Public Methods

- **EGifCircularTree** ()

Constructor & Destructor Documentation

EGifCircularTree::EGifCircularTree () `[inline]`

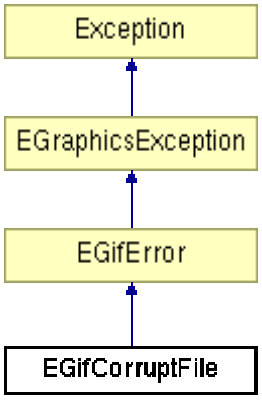
The documentation for this class was generated from the following file:

- **gifexcep.h**

EGifCorruptFile Class Reference

`#include <gifexcep.h>`

Inheritance diagram for EGifCorruptFile:



Public Methods

- **EGifCorruptFile** ()

Constructor & Destructor Documentation

EGifCorruptFile::EGifCorruptFile () `[inline]`

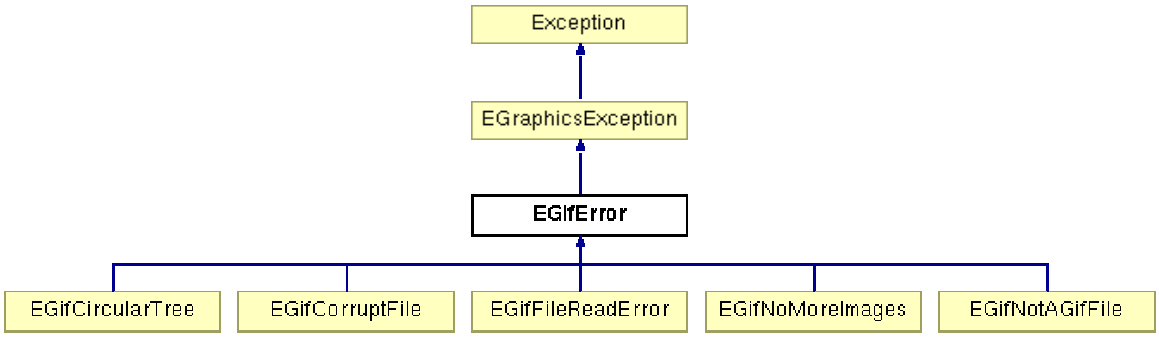
The documentation for this class was generated from the following file:

- **gifexcep.h**

EGifError Class Reference

`#include <gifexcep.h>`

Inheritance diagram for EGifError:



Public Methods

- **EGifError** (const Chaine &msg)

Constructor & Destructor Documentation

EGifError::EGifError (const Chaine & *msg*) `[inline]`

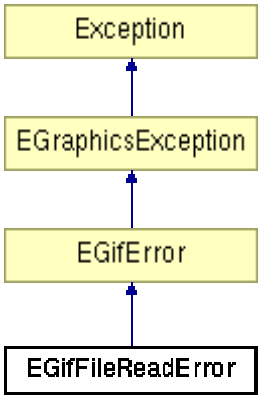
The documentation for this class was generated from the following file:

- **gifexcep.h**

EGifFileReadError Class Reference

`#include <gifexcep.h>`

Inheritance diagram for EGifFileReadError:



Public Methods

- **EGifFileReadError** ()

Constructor & Destructor Documentation

EGifFileReadError::EGifFileReadError () `[inline]`

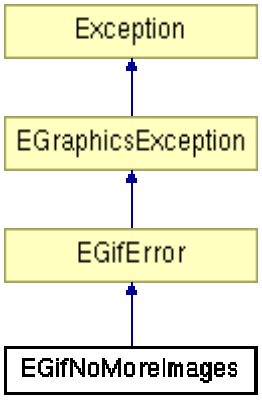
The documentation for this class was generated from the following file:

- **gifexcep.h**

EGifNoMoreImages Class Reference

`#include <gifexcep.h>`

Inheritance diagram for EGifNoMoreImages:



Public Methods

- **EGifNoMoreImages** ()

Constructor & Destructor Documentation

EGifNoMoreImages::EGifNoMoreImages () `[inline]`

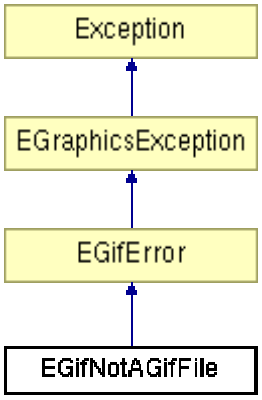
The documentation for this class was generated from the following file:

- **gifexcep.h**

EGifNotAGifFile Class Reference

`#include <gifexcep.h>`

Inheritance diagram for EGifNotAGifFile:



Public Methods

- **EGifNotAGifFile** ()

Constructor & Destructor Documentation

EGifNotAGifFile::EGifNotAGifFile () `[inline]`

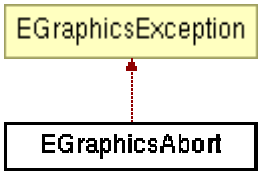
The documentation for this class was generated from the following file:

- **gifexcep.h**

EGraphicsAbort Class Reference

`#include <grexcept.h>`

Inheritance diagram for EGraphicsAbort:



Public Methods

- **EGraphicsAbort** ()

Constructor & Destructor Documentation

EGraphicsAbort::EGraphicsAbort () `[inline]`

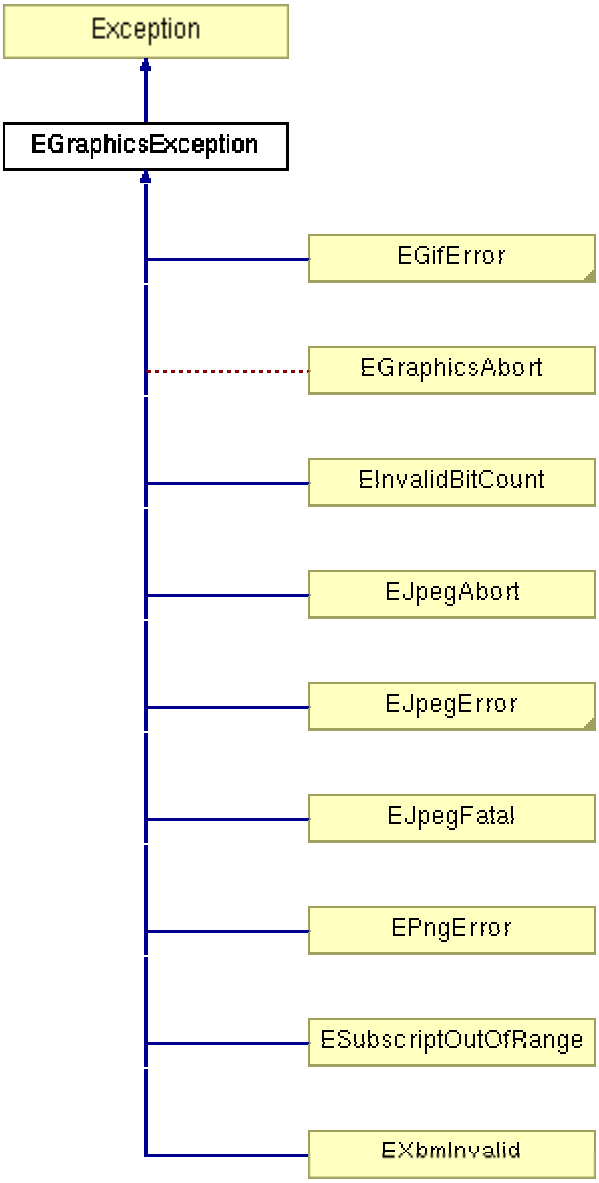
The documentation for this class was generated from the following file:

- **grexcept.h**

EGraphicsException Class Reference

`#include <grexcept.h>`

Inheritance diagram for EGraphicsException:



Public Methods

- **EGraphicsException** (Chaine msg)
- const EGraphicsException& **operator=** (const EGraphicsException &ge)

Constructor & Destructor Documentation

EGraphicsException::EGraphicsException (Chaine *msg*) `[inline]`

Member Function Documentation

const EGraphicsException & EGraphicsException::operator= (const EGraphicsException & *ge*) `[inline]`

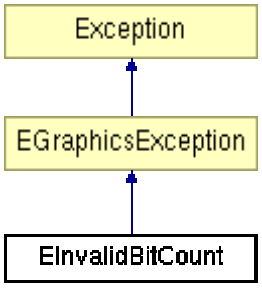
The documentation for this class was generated from the following file:

- **grexcept.h**

EInvalidBitCount Class Reference

`#include <grexcept.h>`

Inheritance diagram for EInvalidBitCount:



Public Methods

- **EInvalidBitCount** ()

Constructor & Destructor Documentation

EInvalidBitCount::EInvalidBitCount () `[inline]`

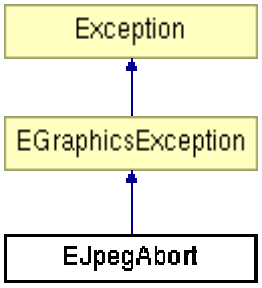
The documentation for this class was generated from the following file:

- **grexcept.h**

EJpegAbort Class Reference

`#include <jpgexcep.h>`

Inheritance diagram for EJpegAbort:



Public Methods

- **EJpegAbort** ()

Constructor & Destructor Documentation

EJpegAbort::EJpegAbort () `[inline]`

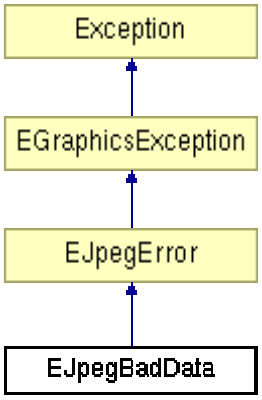
The documentation for this class was generated from the following file:

- **jpgexcep.h**

EJpegBadData Class Reference

`#include <jpgexcep.h>`

Inheritance diagram for EJpegBadData:



Public Methods

- **EJpegBadData** (Chaine msg="Bad input")

Constructor & Destructor Documentation

EJpegBadData::EJpegBadData (Chaine *msg* = "Bad input") `[inline]`

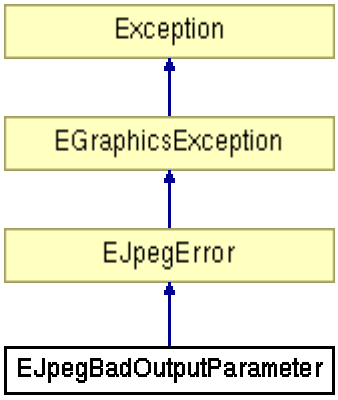
The documentation for this class was generated from the following file:

- **jpgexcep.h**

EJpegBadOutputParameter Class Reference

`#include <jpgexcep.h>`

Inheritance diagram for EJpegBadOutputParameter:



Public Methods

- **EJpegBadOutputParameter** (Chaine msg)

Constructor & Destructor Documentation

EJpegBadOutputParameter::EJpegBadOutputParameter (Chaine *msg*) `[inline]`

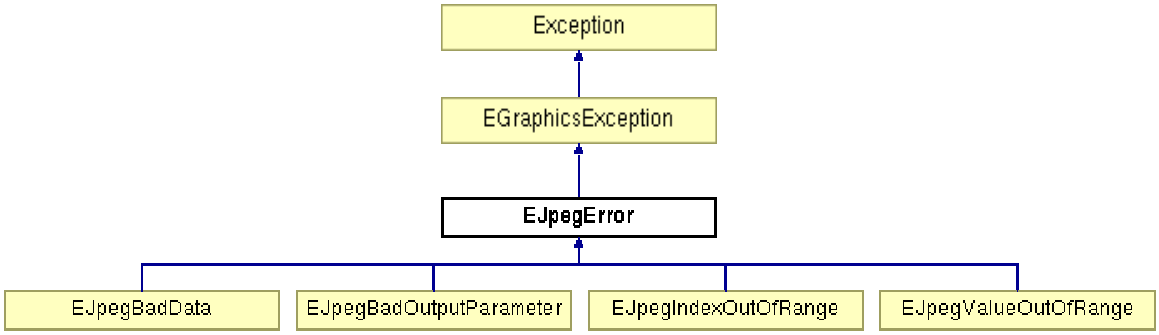
The documentation for this class was generated from the following file:

- **jpgexcep.h**

EJpegError Class Reference

`#include <jpgexcep.h>`

Inheritance diagram for EJpegError:



Public Methods

- **EJpegError** (const Chaine &msg)

Constructor & Destructor Documentation

EJpegError::EJpegError (const Chaine & *msg*) `[inline]`

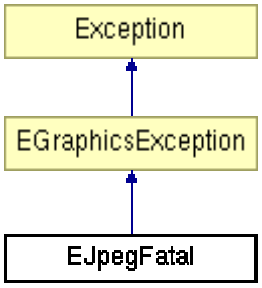
The documentation for this class was generated from the following file:

- **jpgexcep.h**

EJpegFatal Class Reference

`#include <jpgexcep.h>`

Inheritance diagram for EJpegFatal:



Public Methods

- **EJpegFatal** (const Chaine &msg)

Constructor & Destructor Documentation

EJpegFatal::EJpegFatal (const Chaine & *msg*) `[inline]`

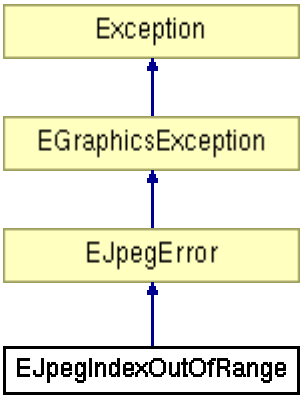
The documentation for this class was generated from the following file:

- **jpgexcep.h**

EJpegIndexOutOfRange Class Reference

`#include <jpgexcep.h>`

Inheritance diagram for EJpegIndexOutOfRange:



Public Methods

- **EJpegIndexOutOfRange** ()

Constructor & Destructor Documentation

EJpegIndexOutOfRange::EJpegIndexOutOfRange () `[inline]`

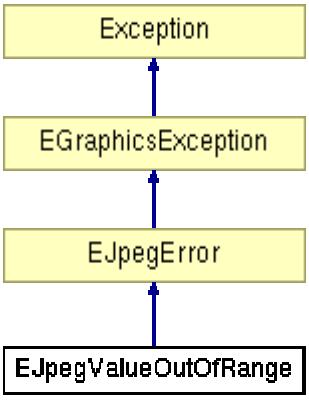
The documentation for this class was generated from the following file:

- **jpgexcep.h**

EJpegValueOutOfRange Class Reference

`#include <jpgexcep.h>`

Inheritance diagram for EJpegValueOutOfRange:



Public Methods

- **EJpegValueOutOfRange** ()

Constructor & Destructor Documentation

EJpegValueOutOfRange::EJpegValueOutOfRange () `[inline]`

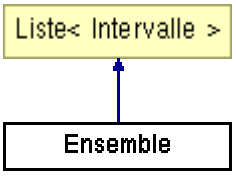
The documentation for this class was generated from the following file:

- **jpgexcep.h**

Ensemble Class Reference

`#include <ImageBinaire.hpp>`

Inheritance diagram for Ensemble:



Public Methods

- **Ensemble** ()
- **Ensemble** (long _min)
- int **Appartient** (const long &nb) const

Constructor & Destructor Documentation

Ensemble::Ensemble () `[inline]`

Ensemble::Ensemble (long *_min*) `[inline]`

Member Function Documentation

int Ensemble::Appartient (const long & *nb*) const

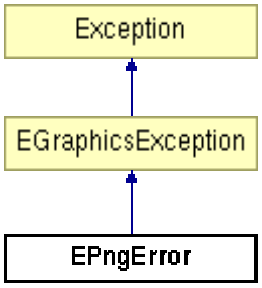
The documentation for this class was generated from the following files:

- **ImageBinaire.hpp**
- **ImageBinaire.cpp**

EPngError Class Reference

`#include <png.h>`

Inheritance diagram for EPngError:



Public Methods

- **EPngError** (char *msg)

Constructor & Destructor Documentation

EPngError::EPngError (char * *msg*) `[inline]`

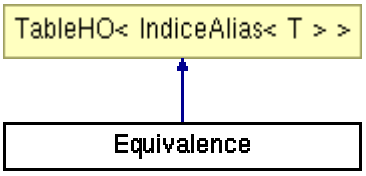
The documentation for this class was generated from the following file:

- **png.h**

Equivalence Class Template Reference

`#include <Equivalence.hpp>`

Inheritance diagram for Equivalence:



Public Methods

- void **Inserer** (const T &effectif, const T &desire)
- T **Chercher** (const T &effectif)

template<class T> class Equivalence

Member Function Documentation

template<class T> T Equivalence< T >::Chercher (const T & *effectif*)

template<class T> void Equivalence< T >::Inserer (const T & *effectif*, const T & *desire*)

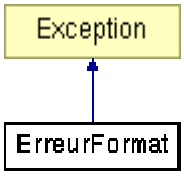
The documentation for this class was generated from the following file:

- **Equivalence.hpp**

ErreurFormat Class Reference

`#include <ImageStreamException.hpp>`

Inheritance diagram for ErreurFormat:



Public Methods

- **ErreurFormat** ()

Constructor & Destructor Documentation

ErreurFormat::ErreurFormat () `[inline]`

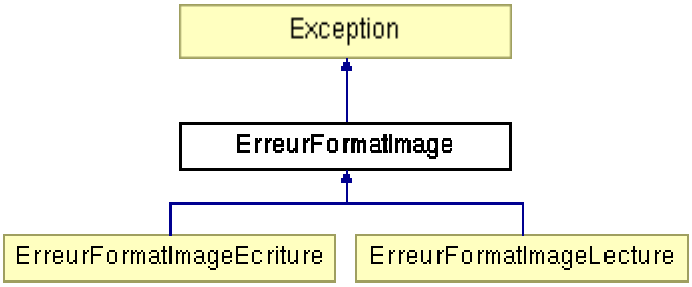
The documentation for this class was generated from the following file:

- **ImageStreamException.hpp**

ErreurFormatImage Class Reference

`#include <ImageStreamException.hpp>`

Inheritance diagram for ErreurFormatImage:



Public Methods

- **ErreurFormatImage** ()

Constructor & Destructor Documentation

ErreurFormatImage::ErreurFormatImage () `[inline]`

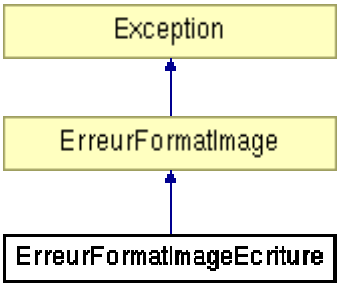
The documentation for this class was generated from the following file:

- **ImageStreamException.hpp**

ErreurFormatImageEcriture Class Reference

`#include <ImageStreamException.hpp>`

Inheritance diagram for ErreurFormatImageEcriture:



Public Methods

- **ErreurFormatImageEcriture** ()

Constructor & Destructor Documentation

ErreurFormatImageEcriture::ErreurFormatImageEcriture () `[inline]`

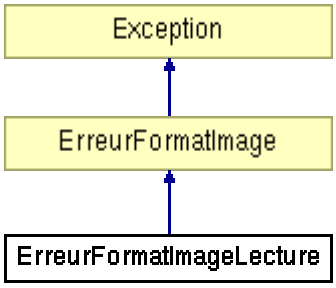
The documentation for this class was generated from the following file:

- **ImageStreamException.hpp**

ErreurFormatImageLecture Class Reference

`#include <ImageStreamException.hpp>`

Inheritance diagram for ErreurFormatImageLecture:



Public Methods

- **ErreurFormatImageLecture** ()

Constructor & Destructor Documentation

ErreurFormatImageLecture::ErreurFormatImageLecture () `[inline]`

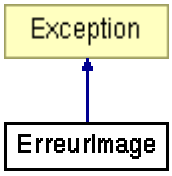
The documentation for this class was generated from the following file:

- **ImageStreamException.hpp**

ErreurImage Class Reference

`#include <ImageStreamException.hpp>`

Inheritance diagram for ErreurImage:



Public Methods

- **ErreurImage** (**fstream** *_us=0)

Private Attributes

- **fstream* us**

Constructor & Destructor Documentation

ErreurImage::ErreurImage (fstream * *_us* = 0) `[inline]`

Member Data Documentation

fstream * ErreurImage::us `[private]`

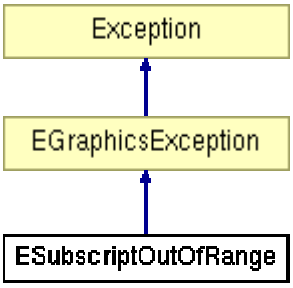
The documentation for this class was generated from the following file:

- **ImageStreamException.hpp**

ESubscriptOutOfRange Class Reference

`#include <grexcept.h>`

Inheritance diagram for ESubscriptOutOfRange:



Public Methods

- **ESubscriptOutOfRange** ()

Constructor & Destructor Documentation

ESubscriptOutOfRange::ESubscriptOutOfRange () `[inline]`

The documentation for this class was generated from the following file:

- **grexcept.h**

Cognois::EvaluationDeQualite Class Reference

```
#include <EvaluationDeQualite.h>
```

Public Methods

- double **QualiteExtraction** (const MPoints &surImages, int &nbImages) const
- double **QualiteReconstruction** (const **Cognois::CameraModel** &cam, const MPoints &surImg1, const MPoints &surImg2, const VPoints &surCible, int &nbImages) const

---

Member Function Documentation

double Cognois::EvaluationDeQualite::QualiteExtraction (const MPoints & *surImages*, int & *nbImages*) const

double Cognois::EvaluationDeQualite::QualiteReconstruction (const Cognois::CameraModel & *cam*, const MPoints & *surImg1*, const MPoints & *surImg2*, const VPoints & *surCible*, int & *nbImages*) const

---

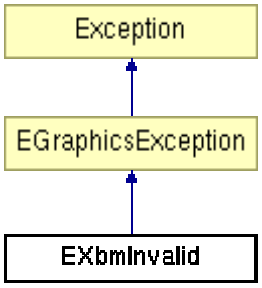
The documentation for this class was generated from the following files:

- **EvaluationDeQualite.h**
- **EvaluationDeQualite.cpp**

EXbmInvalid Class Reference

`#include <xbmdecod.h>`

Inheritance diagram for EXbmInvalid:



Public Methods

- **EXbmInvalid** ()
- **EXbmInvalid** (const Chaine &msg)

Constructor & Destructor Documentation

EXbmInvalid::EXbmInvalid () `[inline]`

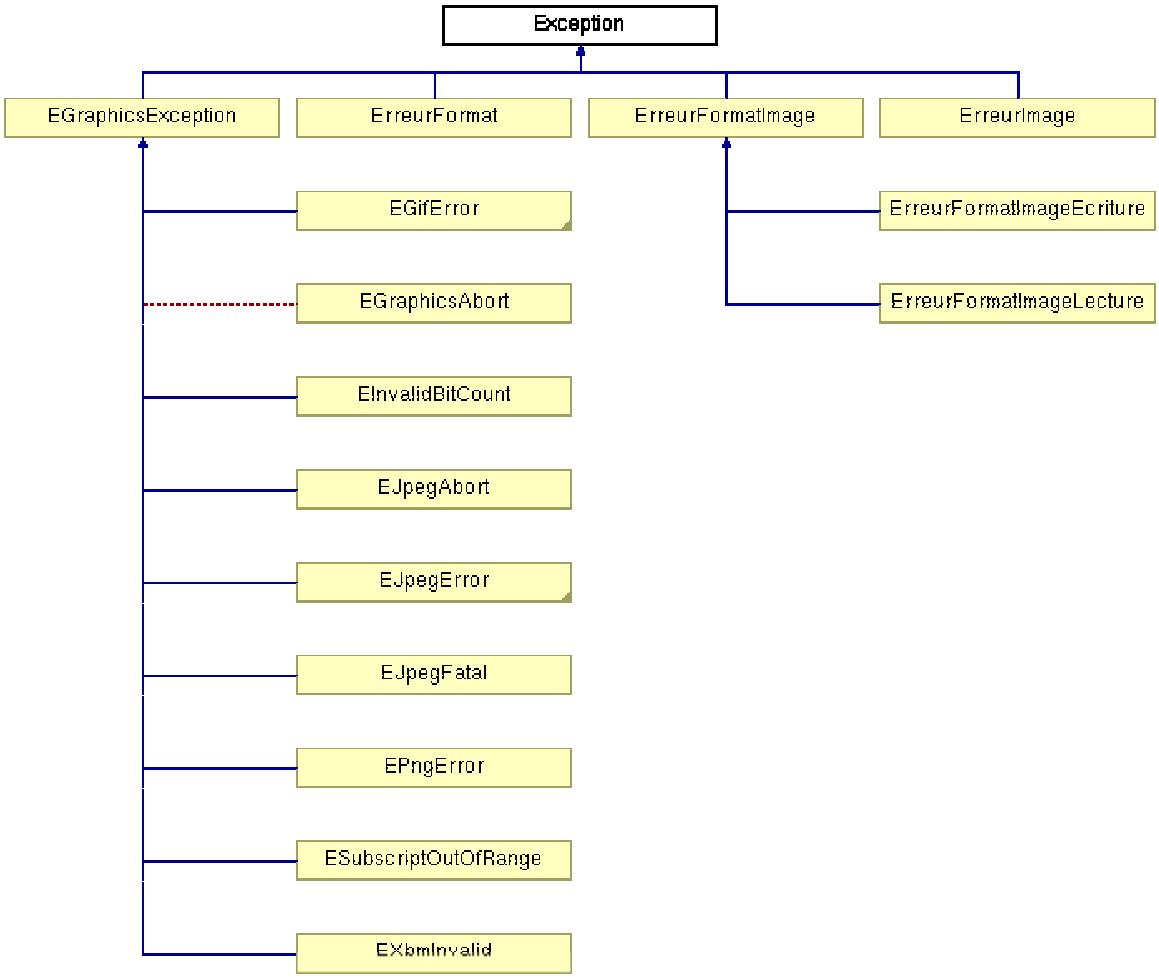
EXbmInvalid::EXbmInvalid (const Chaine & *msg*) `[inline]`

The documentation for this class was generated from the following file:

- **xbmdecod.h**

Exception Class Reference

Inheritance diagram for Exception:

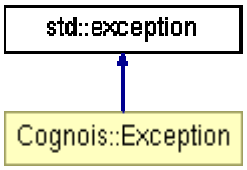


The documentation for this class was generated from the following file:

- **grexcept.h**

std::exception Class Reference

Inheritance diagram for std::exception:



The documentation for this class was generated from the following file:

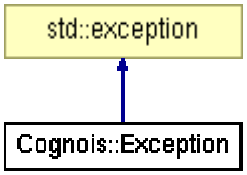
- **Exception.h**

Cognois::Exception Class Reference

Exception (*p.121*) class for COGNOIS.

`#include <Exception.h>`

Inheritance diagram for Cognois::Exception:



Public Methods

- **Exception** (const std::string &inMessage, const std::string &inSourceFile=0, int inSourceLine=0)
- void **Message** () const
  Print the error message.

- virtual const char* **what** () const

Private Attributes

- std::string **mMessage**
  The error message.

Detailed Description

Exception (*p.121*) class for COGNOIS.

Use this class to communicate unexpected events to a higher execution context that is better able to recover from such abnormal events.

Constructor & Destructor Documentation

Cognois::Exception::Exception (const std::string & *inMessage*, const std::string & *inSourceFile* = 0, int *inSourceLine* = 0) `[inline]`

Constructor for an exception; with the flag _COGNOIS_DEBUG, the message is printed

Parameters:

inMessage is an error message
inSourceFile is the source file in which the error occured
inSourceLigne is the line in which the error occured

Member Function Documentation

void Cognois::Exception::Message () const `[inline]`

Print the error message.

const char * Cognois::Exception::what () const `[inline, virtual]`

Member Data Documentation

std::string Cognois::Exception::mMessage `[private]`

The error message.

The documentation for this class was generated from the following file:

- **Exception.h**

Cognois::ExtractionDeCible Class Reference

Extraction des des points d'une cible de calibrage dans une image. La cible est formee // de cercles et son modele est connu.

```
#include <ExtractionDeCible.h>
```

Public Methods

- **ExtractionDeCible** ()
- **ExtractionDeCible** (const VPoints &surCible)
- **ExtractionDeCible** (const MPoints &surCibles)
- void **ObtenirPointsCoins** (const VPoints &surCible, const MIL_ID &image, VPoints &surImage)
- void **ObtenirPointsRGGG** (const VPoints &surCible, const MIL_ID &image, VPoints &surImage)
- const VPoints& **ObtenirPointsCoins** (const MIL_ID &image, VPoints &surImage)
- const VPoints& **ObtenirPointsRGGG** (const MIL_ID &image, VPoints &surImage)
- void **Afficher** (const MIL_ID &MilImage, const VPoints &surImage)

Public Attributes

- MIL_ID* **MilImageDisp**
- MIL_ID * **MilImageWork**

Private Methods

- void **ExtrairePoints** (const MIL_ID &image, const int nbamis, VPoints &surImage)
- unsigned char **AutoThresholdEstimation** (const MIL_ID &image) const
- void **GetBlobs** (void *_img, **BlobInfo** *&_blobs, int &_nbBlobs) const
- void **GetCircles** (void *img, const int nbamis, VPoints &surImage) const
- void **ConvertToGray** (const MIL_ID &_imgRGB, MIL_ID &_imgGray) const
- void **Binarize** (void *_imgGray, void *_binImage, unsigned char _treshold) const
- void **ConvertMilToPicture** (const MIL_ID &_milImage, void *_picImage) const
- void **AfficherDebug** (MIL_ID &_milBuffer, void *_binBuffer, const VPoints &_surImage) const
- template<class T> void **Seuils** (**std::vector**< T > &liste, int nbamis, float &moymin, float &ecartmin) const
- template<class T> void **Statistiques** (**std::vector**< T > &liste, int depart, int nbamis, float &moyenne, float &ecart_type) const
- void **TrouverPointRouge** (const MIL_ID &image, VPoints &surImage) const
- void **TrouverPointsRougeVert** (const MIL_ID &image, VPoints &surImage) const
- void **Ordonner** (const VPoints &surCible, const VPoints &surImage, VPoints &ordonnesImage)
- void **OrdonnerCouleur** (const VPoints &surCible, const VPoints &surImage, VPoints &ordonnesImage)
- VDoubles **Mapping** (const VPoints &U, const VPoints &X) const
- **Point SurCible** (const VDoubles &P, const **Point** &U) const
- int **Intersection** (const VPoints &points) const
- void **Iden4Coins** (const VPoints &points, VPoints &ordonnes) const
- void **SortMatrix** (const VPoints &A, VPoints &ordonnes) const
- void **Matching** (const VDoubles P, const VPoints &U, const VPoints &X, VPoints &Uout)
- void **RGBtoHSI** (unsigned char red, unsigned char green, unsigned char blue, double &hue, double &saturation, double &intensity) const

Private Attributes

- MPoints **ciblesParDefaut**
- VDoubles **distMinDefaut**
- unsigned int **cibleCourante**
- **Point mLastSize**
- **Point mLastPosition**
- **Point mImageSize**
- VPoints **mLastDisplay**
- bool **useCibleParDefaut**

Detailed Description

Extraction des des points d'une cible de calibrage dans une image. La cible est formee // de cercles et son modele est connu.

Constructor & Destructor Documentation

Cognois::ExtractionDeCible::ExtractionDeCible ()

Cognois::ExtractionDeCible::ExtractionDeCible (const VPoints & *surCible*)

Cognois::ExtractionDeCible::ExtractionDeCible (const MPoints & *surCibles*)

Member Function Documentation

void Cognois::ExtractionDeCible::Afficher (const MIL_ID & *MilImage*, const VPoints & *surImage*)

void Cognois::ExtractionDeCible::AfficherDebug (MIL_ID & *_milBuffer*, void * *_binBuffer*, const VPoints & *_surImage*) const `[private]`

unsigned char Cognois::ExtractionDeCible::AutoThresholdEstimation (const MIL_ID & *image*) const `[private]`

void Cognois::ExtractionDeCible::Binarize (void * *_imgGray*, void * *_binImage*, unsigned char *_treshold*) const `[inline, private]`

void Cognois::ExtractionDeCible::ConvertMilToPicture (const MIL_ID & *_milImage*, void * *_picImage*) const `[inline, private]`

void Cognois::ExtractionDeCible::ConvertToGray (const MIL_ID & *_imgRGB*, MIL_ID & *_imgGray*) const `[inline, private]`

void Cognois::ExtractionDeCible::ExtrairePoints (const MIL_ID & *image*, const int *nbamis*, VPoints & *surImage*) `[private]`

void Cognois::ExtractionDeCible::GetBlobs (void * *_img*, BlobInfo *& *_blobs*, int & *_nbBlobs*) const `[private]`

void Cognois::ExtractionDeCible::GetCircles (void * *img*, const int *nbamis*, VPoints & *surImage*) const `[private]`

void Cognois::ExtractionDeCible::Iden4Coins (const VPoints & *points*, VPoints & *ordonnes*) const `[private]`

int Cognois::ExtractionDeCible::Intersection (const VPoints & *points*) const `[private]`

VDoubles Cognois::ExtractionDeCible::Mapping (const VPoints & *U*, const VPoints & *X*) const `[private]`

void Cognois::ExtractionDeCible::Matching (const VDoubles *P*, const VPoints & *U*, const VPoints & *X*, VPoints & *Uout*) `[private]`

const VPoints & Cognois::ExtractionDeCible::ObtenirPointsCoins (const MIL_ID & *image*, VPoints & *surImage*)

void Cognois::ExtractionDeCible::ObtenirPointsCoins (const VPoints & *surCible*, const MIL_ID & *image*, VPoints & *surImage*)

const VPoints & Cognois::ExtractionDeCible::ObtenirPointsRGGG (const MIL_ID & *image*, VPoints & *surImage*)

void Cognois::ExtractionDeCible::ObtenirPointsRGGG (const VPoints & *surCible*, const MIL_ID & *image*, VPoints & *surImage*)

void Cognois::ExtractionDeCible::Ordonner (const VPoints & *surCible*, const VPoints & *surImage*, VPoints & *ordonnesImage*) `[private]`

void Cognois::ExtractionDeCible::OrdonnerCouleur (const VPoints & *surCible*, const VPoints & *surImage*, VPoints & *ordonnesImage*) `[private]`

void Cognois::ExtractionDeCible::RGBtoHSI (unsigned char *red*, unsigned char *green*, unsigned char *blue*, double & *hue*, double & *saturation*, double & *intensity*) const `[private]`

template<class T> void Cognois::ExtractionDeCible::Seuils (std::vector< T > & *liste*, int *nbamis*, float & *moymin*, float & *ecartmin*) const `[private]`

void Cognois::ExtractionDeCible::SortMatrix (const VPoints & *A*, VPoints & *ordonnes*) const `[private]`

template<class T> void Cognois::ExtractionDeCible::Statistiques (std::vector< T > & *liste*, int *depart*, int *nbamis*, float & *moyenne*, float & *ecart_type*) const `[private]`

Point Cognois::ExtractionDeCible::SurCible (const VDoubles & *P*, const Point & *U*) const `[private]`

void Cognois::ExtractionDeCible::TrouverPointRouge (const MIL_ID & *image*, VPoints & *surImage*) const `[private]`

void Cognois::ExtractionDeCible::TrouverPointsRougeVert (const MIL_ID & *image*, VPoints & *surImage*) const `[private]`

Member Data Documentation

MIL_ID * Cognois::ExtractionDeCible::MilImageDisp

MIL_ID * Cognois::ExtractionDeCible::MilImageWork

unsigned int Cognois::ExtractionDeCible::cibleCourante `[private]`

MPoints Cognois::ExtractionDeCible::ciblesParDefaut `[private]`

VDoubles Cognois::ExtractionDeCible::distMinDefaut `[private]`

Point Cognois::ExtractionDeCible::mImageSize `[private]`

VPoints Cognois::ExtractionDeCible::mLastDisplay `[private]`

Point Cognois::ExtractionDeCible::mLastPosition `[private]`

Point Cognois::ExtractionDeCible::mLastSize `[private]`

bool Cognois::ExtractionDeCible::useCibleParDefaut `[private]`

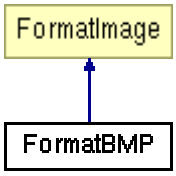
The documentation for this class was generated from the following files:

- **ExtractionDeCible.h**
- **ExtractionDeCible.cpp**

FormatBMP Class Reference

`#include <FormatBMP.hpp>`

Inheritance diagram for FormatBMP:



Public Methods

- void **ReadImage** (istream &, **ImageRGB** &)
- void **WriteImage** (ostream &, const **ImageRGB** &)

Protected Types

- enum { **BRGB**, **BRLE8**, **BRLE4** }
- enum { **EOL**, **EOB**, **DELTA** }

Member Enumeration Documentation

anonymous enum `[protected]`

Enumeration values:

BRGB
BRLE8
BRLE4

anonymous enum `[protected]`

Enumeration values:

EOL
EOB
DELTA

Member Function Documentation

void FormatBMP::ReadImage (istream & *in*, ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.132*).

void FormatBMP::WriteImage (ostream & *out*, const ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.133*).

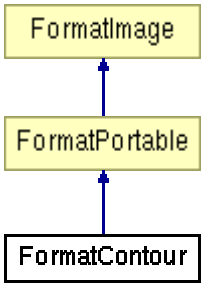
The documentation for this class was generated from the following files:

- **FormatBMP.hpp**
- **FormatBMP.cpp**

FormatContour Class Reference

`#include <FormatContour.hpp>`

Inheritance diagram for FormatContour:



Public Methods

- void **ReadImage** (istream &, **Contour** &)
- void **WriteImage** (ostream &, const **Contour** &)
- void **ReadImage** (istream &i, **ImageRGB** &r)
- void **WriteImage** (ostream &o, const **ImageRGB** &r)

Protected Methods

- void **ReadV1** (istream &, **Contour** &)

Member Function Documentation

void FormatContour::ReadImage (istream & *i*, ImageRGB & *r*) `[inline, virtual]`

Reimplemented from **FormatImage** (*p.132*).

void FormatContour::ReadImage (istream & *in*, Contour & *ctr*)

void FormatContour::ReadV1 (istream & *in*, Contour & *ctr*) `[protected]`

void FormatContour::WriteImage (ostream & *o*, const ImageRGB & *r*) `[inline, virtual]`

Reimplemented from **FormatImage** (*p.133*).

void FormatContour::WriteImage (ostream & *out*, const Contour & *ctr*)

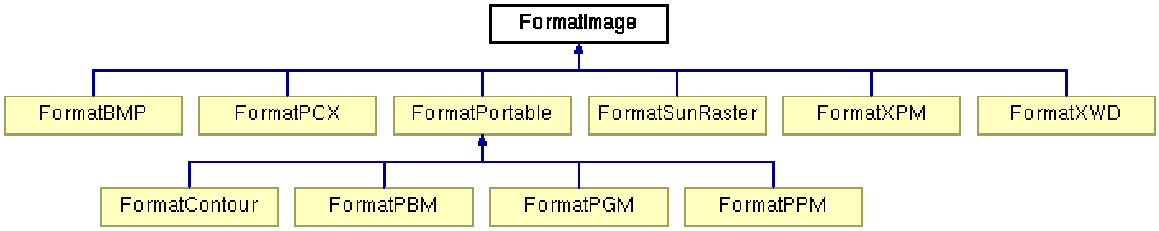
The documentation for this class was generated from the following files:

- **FormatContour.hpp**
- **FormatContour.cpp**

FormatImage Class Reference

`#include <FormatImage.hpp>`

Inheritance diagram for FormatImage:



Public Methods

- **FormatImage** ()
- virtual **~FormatImage** ()
- virtual void **ReadImage** (istream &, **ImageRGB** &)=0
- virtual void **WriteImage** (ostream &, const **ImageRGB** &)=0
- void **SetVerbose** (bool _v)
- int **GetVerbose** () const

Protected Attributes

- bool **verbose_flag**

Constructor & Destructor Documentation

FormatImage::FormatImage () `[inline]`

FormatImage::~FormatImage () `[inline, virtual]`

Member Function Documentation

int FormatImage::GetVerbose () const `[inline]`

void FormatImage::ReadImage (istream & *in*, ImageRGB & *img*) `[pure virtual]`

Reimplemented in **FormatBMP** (*p.128*), **FormatContour** (*p.130*), **FormatPBM** (*p.134*), **FormatPCX** (*p.137*), **FormatPGM** (*p.138*), **FormatPPM** (*p.141*), **FormatSunRaster** (*p.143*), **FormatXPM** (*p.145*), and **FormatXWD** (*p.148*).

void FormatImage::SetVerbose (bool _*v*) `[inline]`

void FormatImage::WriteImage (ostream & *out*, const ImageRGB & *img*) `[pure virtual]`

Reimplemented in **FormatBMP** (*p.129*), **FormatContour** (*p.130*), **FormatPBM** (*p.134*), **FormatPCX** (*p.137*), **FormatPGM** (*p.138*), **FormatPPM** (*p.141*), **FormatSunRaster** (*p.143*), **FormatXPM** (*p.145*), and **FormatXWD** (*p.148*).

Member Data Documentation

bool FormatImage::verbose_flag `[protected]`

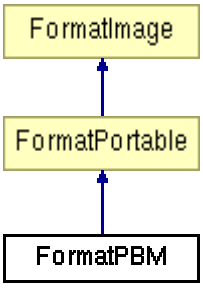
The documentation for this class was generated from the following file:

- **FormatImage.hpp**

FormatPBM Class Reference

`#include <FormatPBM.hpp>`

Inheritance diagram for FormatPBM:



Public Methods

- void **ReadImage** (istream &, **ImageGris** &)
- void **WriteImage** (ostream &, const **ImageGris** &)
- void **ReadImage** (istream &i, **ImageRGB** &r)
- void **WriteImage** (ostream &o, const **ImageRGB** &r)

Protected Methods

- void **ReadRAW** (istream &, **ImageGris** &)
- void **ReadASCII** (istream &, **ImageGris** &)

Protected Attributes

- **pbmHeader hd**

Member Function Documentation

void FormatPBM::ReadASCII (istream & *in*, ImageGris & *img*) `[protected]`

void FormatPBM::ReadImage (istream & *i*, ImageRGB & *r*) `[inline, virtual]`

Reimplemented from **FormatImage** (*p.132*).

void FormatPBM::ReadImage (istream & *in*, ImageGris & *img*)

void FormatPBM::ReadRAW (istream & *in*, ImageGris & *img*) `[protected]`

void FormatPBM::WriteImage (ostream & *o*, const ImageRGB & *r*) `[inline, virtual]`

Reimplemented from **FormatImage** (*p.133*).

void FormatPBM::WriteImage (ostream & *out*, const ImageGris & *img*)

Member Data Documentation

pbmHeader FormatPBM::hd `[protected]`

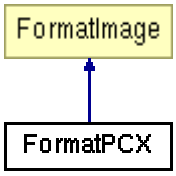
The documentation for this class was generated from the following files:

- **FormatPBM.hpp**
- **FormatPBM.cpp**

FormatPCX Class Reference

`#include <FormatPCX.hpp>`

Inheritance diagram for FormatPCX:



Public Methods

- void **ReadImage** (istream &, **ImageRGB** &)

Protected Types

- enum { **PMAGIC** = 0xA }
- enum { **PRLE** = 1 }
- enum { **PGSCALE** = 1, **PCOLOR** }
- enum { **PCXMONO** = 0, **PCX16c**, **PCX16c_pal**, **PCX256c_pal**, **PCX16Mc** }

Private Methods

- void **WriteImage** (ostream &, const **ImageRGB** &)

Member Enumeration Documentation

anonymous enum `[protected]`

Enumeration values:

PMAGIC

anonymous enum `[protected]`

Enumeration values:

PRLE

anonymous enum `[protected]`

Enumeration values:

PGSCALE
PCOLOR

anonymous enum `[protected]`

Enumeration values:

PCXMONO
PCX16c
PCX16c_pal
PCX256c_pal
PCX16Mc

Member Function Documentation

void FormatPCX::ReadImage (istream & *in*, ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.132*).

void FormatPCX::WriteImage (ostream & *out*, const ImageRGB & *img*) `[private, virtual]`

Reimplemented from **FormatImage** (*p.133*).

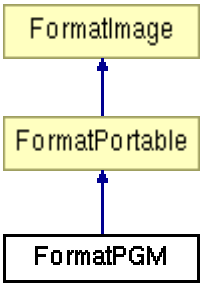
The documentation for this class was generated from the following files:

- **FormatPCX.hpp**
- **FormatPCX.cpp**

FormatPGM Class Reference

`#include <FormatPGM.hpp>`

Inheritance diagram for FormatPGM:



Public Methods

- void **ReadImage** (istream &, **ImageGris** &)
- void **WriteImage** (ostream &, const **ImageGris** &)
- void **ReadImage** (istream &i, **ImageRGB** &r)
- void **WriteImage** (ostream &o, const **ImageRGB** &r)

Protected Methods

- void **ReadRAW** (istream &, **ImageGris** &)
- void **ReadASCII** (istream &, **ImageGris** &)

Protected Attributes

- **pgmHeader hd**

Member Function Documentation

void FormatPGM::ReadASCII (istream & *in*, ImageGris & *img*) `[protected]`

void FormatPGM::ReadImage (istream & *i*, ImageRGB & *r*) `[inline, virtual]`

Reimplemented from **FormatImage** (*p.132*).

void FormatPGM::ReadImage (istream & *in*, ImageGris & *img*)

void FormatPGM::ReadRAW (istream & *in*, ImageGris & *img*) `[protected]`

void FormatPGM::WriteImage (ostream & *o*, const ImageRGB & *r*) `[inline, virtual]`

Reimplemented from **FormatImage** (*p.133*).

void FormatPGM::WriteImage (ostream & *out*, const ImageGris & *img*)

Member Data Documentation

pgmHeader FormatPGM::hd `[protected]`

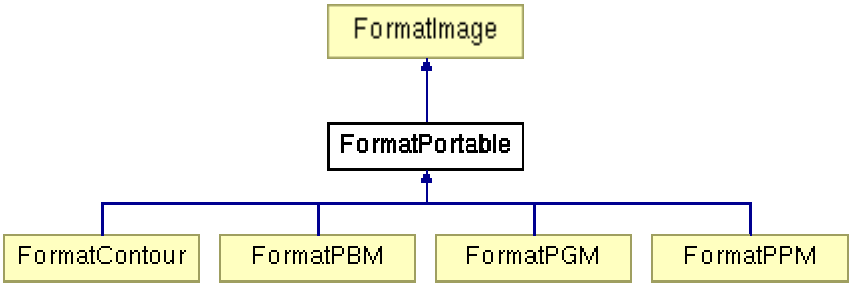
The documentation for this class was generated from the following files:

- **FormatPGM.hpp**
- **FormatPGM.cpp**

FormatPortable Class Reference

`#include <FormatPortable.hpp>`

Inheritance diagram for FormatPortable:



Protected Methods

- void **NextLine** (istream &)
- Chaine **NextInformation** (istream &)

Member Function Documentation

Chaine FormatPortable::NextInformation (istream & *in*) `[inline, protected]`

void FormatPortable::NextLine (istream & *in*) `[inline, protected]`

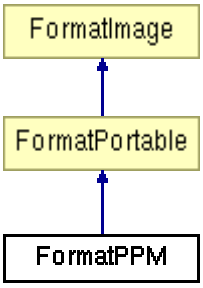
The documentation for this class was generated from the following file:

- **FormatPortable.hpp**

FormatPPM Class Reference

`#include <FormatPPM.hpp>`

Inheritance diagram for FormatPPM:



Public Methods

- void **ReadImage** (istream &, **ImageRGB** &)
- void **WriteImage** (ostream &, const **ImageRGB** &)

Protected Methods

- void **ReadRAW** (istream &, **ImageRGB** &)
- void **ReadASCII** (istream &, **ImageRGB** &)

Protected Attributes

- **ppmHeader hd**

Member Function Documentation

void FormatPPM::ReadASCII (istream & *in*, ImageRGB & *img*) `[protected]`

void FormatPPM::ReadImage (istream & *in*, ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.132*).

void FormatPPM::ReadRAW (istream & *in*, ImageRGB & *img*) `[protected]`

void FormatPPM::WriteImage (ostream & *out*, const ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.133*).

Member Data Documentation

ppmHeader FormatPPM::hd `[protected]`

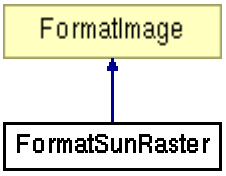
The documentation for this class was generated from the following files:

- **FormatPPM.hpp**
- **FormatPPM.cpp**

FormatSunRaster Class Reference

`#include <FormatSunRaster.hpp>`

Inheritance diagram for FormatSunRaster:



Public Methods

- void **ReadImage** (istream &, **ImageRGB** &)
- void **WriteImage** (ostream &, const **ImageRGB** &)
- void **WriteImage** (ostream &, const **ImageGris** &)

Member Function Documentation

void FormatSunRaster::ReadImage (istream & *in*, ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.132*).

void FormatSunRaster::WriteImage (ostream & *out*, const ImageGris & *img*)

void FormatSunRaster::WriteImage (ostream & *out*, const ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.133*).

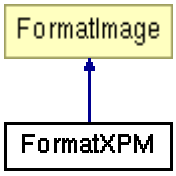
The documentation for this class was generated from the following files:

- **FormatSunRaster.hpp**
- **FormatSunRaster.cpp**

FormatXPM Class Reference

`#include <FormatXPM.hpp>`

Inheritance diagram for FormatXPM:



Public Methods

- void **ReadImage** (istream &, **ImageRGB** &)
- void **WriteImage** (ostream &, const **ImageRGB** &)

Protected Types

- enum **Token** { **Invalid**, **Define**, **Comma**, **Integer**, **Static**, **Char**, **Bits**, **LBrace**, **RBrace**, **Equal**, **LBracket**, **RBracket**, **Guillemet**, **Slash**, **Format**, **Width**, **Height**, **NColors**, **Pixel**, **Mono**, **Pointeur**, **Semicolon**, **Unsigned** }

Protected Methods

- int **NextChar** (istream &)
- int **NextToken** (istream &)
- void **NextColor** (istream &, **xpmColor** &tmp, int bpp=1)
- int **NextPixel** (istream &, int)

Protected Attributes

- int **last_char**
- unsigned int **lexical_value**
- bool **openstring**

Member Enumeration Documentation

enum FormatXPM::Token `[protected]`

Enumeration values:

Invalid
Define
Comma
Integer
Static
Char
Bits
LBrace

RBrace
Equal
LBracket
RBracket
Guillemet
Slash
Format
Width
Height
NColors
Pixel
Mono
Pointeur
Semicolon
Unsigned

Member Function Documentation

int FormatXPM::NextChar (istream & *strm*) `[protected]`

void FormatXPM::NextColor (istream & *in*, xpmColor & *retour*, int *bpp* = 1) `[protected]`

int FormatXPM::NextPixel (istream & *in*, int *bpp*) `[protected]`

int FormatXPM::NextToken (istream & *strm*) `[protected]`

void FormatXPM::ReadImage (istream & *in*, ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.132*).

void FormatXPM::WriteImage (ostream & *out*, const ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.133*).

Member Data Documentation

int FormatXPM::last_char `[protected]`

unsigned int FormatXPM::lexical_value `[protected]`

bool FormatXPM::openstring `[protected]`

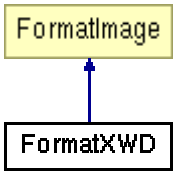
The documentation for this class was generated from the following files:

- **FormatXPM.hpp**
- **FormatXPM.cpp**

FormatXWD Class Reference

`#include <FormatXWD.hpp>`

Inheritance diagram for FormatXWD:



Public Methods

- void **ReadImage** (istream &, **ImageRGB** &)
- void **WriteImage** (ostream &, const **ImageRGB** &)

Protected Types

- enum { **XYPixmap** = 1, **ZPixmap** }
- enum { **StaticG** = 0, **ScaleG**, **StaticC**, **PseudoC**, **TrueC**, **DirectC** }
- enum { **LSBFirst** = 0, **MSBFirst** }

Protected Methods

- **UBYTE4 SystemToEndian** (**UBYTE4** val)
- **UBYTE4 EndianToSystem** (**UBYTE4** val)
- **UBYTE2 SystemToEndian** (**UBYTE2** val)
- **UBYTE2 EndianToSystem** (**UBYTE2** val)
- Table1D<**UBYTE4**> **DecomposeUnit** (**UBYTE4** unit, **UBYTE1** bbp, **UBYTE1** bu)

Protected Attributes

- bool **bytebigendian**
- bool **bitbigendian**
- **UBYTE4 pixmask**

Member Enumeration Documentation

anonymous enum `[protected]`

Enumeration values:

StaticG
ScaleG
StaticC
PseudoC
TrueC
DirectC

anonymous enum `[protected]`

Enumeration values:

LSBFirst
MSBFirst

anonymous enum `[protected]`

Enumeration values:

XYPixmap
ZPixmap

Member Function Documentation

Table1D< UBYTE4 > FormatXWD::DecomposeUnit (UBYTE4 *unit*, UBYTE1 *bpp*, UBYTE1 *bu*) `[inline, protected]`

UBYTE2 FormatXWD::EndianToSystem (UBYTE2 *val*) `[inline, protected]`

UBYTE4 FormatXWD::EndianToSystem (UBYTE4 *val*) `[inline, protected]`

void FormatXWD::ReadImage (istream & *in*, ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.132*).

UBYTE2 FormatXWD::SystemToEndian (UBYTE2 *val*) `[inline, protected]`

UBYTE4 FormatXWD::SystemToEndian (UBYTE4 *val*) `[inline, protected]`

void FormatXWD::WriteImage (ostream & *out*, const ImageRGB & *img*) `[virtual]`

Reimplemented from **FormatImage** (*p.133*).

Member Data Documentation

bool FormatXWD::bitbigendian `[protected]`

bool FormatXWD::bytebigendian `[protected]`

UBYTE4 FormatXWD::pixmask `[protected]`

The documentation for this class was generated from the following files:

- **FormatXWD.hpp**
- **FormatXWD.cpp**

Cognois::FrameCounter Class Reference

Count grabbed frames according to their time stamp.

```
#include <FrameCounter.h>
```

Public Methods

- **FrameCounter** (double inFrameRate, unsigned int inQueueLength=10)
  Constructor with an estimation of the frame rate.

- **FrameCounter** (const FrameCounter &inFrameCounter)
  Copy constructor.

- **~FrameCounter** ()
  Destructor.

- void **SetVerbose** (strstream *inStream)
- const FrameCounter& **operator=** (const FrameCounter &inFrameCounter)
  *Assign a **FrameCounter** (p.150).*

- void **Reset** (unsigned long inNextFrameId=0)
  Reset the frame queue.

- void **GetFrameInformation** (LARGE_INTEGER &inPerformanceCounter, unsigned long &outFrameId, unsigned long &outTimeStamp)
  Assigns a frame number and a time stamp to a performance counter value.

- double **TimeSinceLastFrame** (LARGE_INTEGER &inPerformanceCounter)

Private Methods

- unsigned int **NextLeastRecentFrameIndex** () const
  Calculate the next index in the circular array.

- unsigned int **NextMostRecentFrameIndex** () const
  Calculate the next index in the circular array.

- unsigned int **NextIndex** (const unsigned int &inCurrentIndex) const
  Calculate the next index in the circular array.

- unsigned long **ElapsedMilliseconds** (LARGE_INTEGER &inPreviousTime, LARGE_INTEGER &inCurrentTime) const
  Calculate the number of milliseconds between two time stamps.

Private Attributes

- unsigned long* **mLastFrameIds**
  Array of the last frame ids.

- LARGE_INTEGER* **mLastTimeStamps**
  Array of the last time stamps.

- unsigned int **mLeastRecentFrameIndex**
  Index of the least recent frame.

- unsigned int **mMostRecentFrameIndex**
  Index of the most recent frame.

- LARGE_INTEGER **mTimeAtBeginning**
  Original performance counter value.

- unsigned int **mQueueLength**
  Number of frame to remember.

- double **mFrameRate**
  Estimated frame rate of the camera.

- LARGE_INTEGER **mPerformanceFrequency**
  Performance frequency value.

- bool **mTransientState**
  If no frame have been processed.

- bool **mVerbose**
- strstream* **mStream**

Detailed Description

Count grabbed frames according to their time stamp.

Constructor & Destructor Documentation

Cognois::FrameCounter::FrameCounter (double *inFrameRate*, unsigned int *inQueueLength* = 10)

Constructor with an estimation of the frame rate.

Parameters:

inFrameRate is the estimated frame rate of the camera
inQueueLength is the number of frame to remember

Cognois::FrameCounter::FrameCounter (const FrameCounter & *inFrameCounter*)

Copy constructor.

Cognois::FrameCounter::~FrameCounter ()

Destructor.

Member Function Documentation

unsigned long Cognois::FrameCounter::ElapsedMilliseconds (LARGE_INTEGER & *inPreviousTime*, LARGE_INTEGER & *inCurrentTime*) const `[inline, private]`

Calculate the number of milliseconds between two time stamps.

Parameters:

inPreviousTime is a time stamp
inCurrentTime is a time stamp

Returns:

The number of millisecondes elapsed from inPreviousTime to inCurrentTime

void Cognois::FrameCounter::GetFrameInformation (LARGE_INTEGER & *inPerformanceCounter*, unsigned long & *outFrameId*, unsigned long & *outTimeStamp*)

Assigns a frame number and a time stamp to a performance counter value.

Parameters:

inPerformanceCounter is a performance counter value associated with a frame
outFrameId is a the frame ID corresponding to the time stamp
outTimeStamp is the time stamp corresponding to the performance counter

unsigned int Cognois::FrameCounter::NextIndex (const unsigned int & *inCurrentIndex*) const `[private]`

Calculate the next index in the circular array.

Parameters:

inCurrentIndex is the current index in the circular array

Returns:

The next index in the circular array

unsigned int Cognois::FrameCounter::NextLeastRecentFrameIndex () const `[inline, private]`

Calculate the next index in the circular array.

Returns:

The next index to use to address the least recent frame

unsigned int Cognois::FrameCounter::NextMostRecentFrameIndex () const `[inline, private]`

Calculate the next index in the circular array.

Returns:

The next index to use to address the most recent frame

void Cognois::FrameCounter::Reset (unsigned long *inNextFrameId* = 0)

Reset the frame queue.

Parameters:

inNextFrameId is a the next frame ID to allocate

void Cognois::FrameCounter::SetVerbose (strstream * *inStream*) `[inline]`

double Cognois::FrameCounter::TimeSinceLastFrame (LARGE_INTEGER & *inPerformanceCounter*) `[inline]`

Evaluate the time (in ms) since the last valid frame.

Parameters:

inPerformanceCounter is the current frame time stamp.

Returns:

The time since the last valid frame.

const Cognois::FrameCounter & Cognois::FrameCounter::operator= (const FrameCounter & *inFrameCounter*)

Assign a **FrameCounter** (*p.150*).

Parameters:

inFrameCounter is the object to assign this

Returns:

A reference to this

Member Data Documentation

double Cognois::FrameCounter::mFrameRate `[private]`

Estimated frame rate of the camera.

unsigned long * Cognois::FrameCounter::mLastFrameIds `[private]`

Array of the last frame ids.

LARGE_INTEGER * Cognois::FrameCounter::mLastTimeStamps `[private]`

Array of the last time stamps.

unsigned int Cognois::FrameCounter::mLeastRecentFrameIndex `[private]`

Index of the least recent frame.

unsigned int Cognois::FrameCounter::mMostRecentFrameIndex `[private]`

Index of the most recent frame.

LARGE_INTEGER Cognois::FrameCounter::mPerformanceFrequency `[private]`

Performance frequency value.

unsigned int Cognois::FrameCounter::mQueueLength `[private]`

Number of frame to remember.

strstream * Cognois::FrameCounter::mStream `[private]`

LARGE_INTEGER Cognois::FrameCounter::mTimeAtBeginning `[private]`

Original performance counter value.

bool Cognois::FrameCounter::mTransientState `[private]`

If no frame have been processed.

bool Cognois::FrameCounter::mVerbose `[private]`

The documentation for this class was generated from the following files:

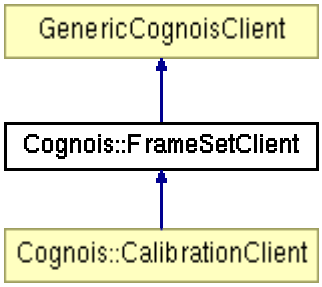
- **FrameCounter.h**
- **FrameCounter.cpp**

Cognois::FrameSetClient Class Reference

Read and write blob information on the COGNOIS server.

`#include <FrameSetClient.h>`

Inheritance diagram for Cognois::FrameSetClient:



Public Methods

- **FrameSetClient** (int inArgc=0, char **inArgv=0)
  Constructor with an optionnal command line.

- **~FrameSetClient** ()
  Destructor.

- void **CreateFrame** (const unsigned long &inBlobCount, const unsigned long &inFrameId, const unsigned long &inTimeStamp)
  Create a frame (blob set) of empty elements.

- CORBA::Boolean **SendFrame** ()
  Send a previously created frame, then destroys it.

- CORBA::Boolean **GetLatestFrameSet** (unsigned long &outFrameId, unsigned long &outTimeStamp, CORBA::Boolean inGetBlobData=0)
  Get a frame set from the server and put it in the private member variables.

- CORBA::Boolean **GetNewFrameSet** (unsigned long &outFrameId, unsigned long &outTimeStamp, CORBA::Boolean inGetBlobData=0, unsigned long inMaxWaitTime=2000)
  Get a NEW frame set from the server and put it in the private member variables.

- SimpleXmlParser& **XmlData** (const unsigned long &inBlobId, const unsigned long &inCameraId=0)
  Address the xml content of a particular blob (created or received).

- unsigned long **FrameCount** () const
  Get the number of frames.

- unsigned long **BlobCount** (const unsigned long &inCameraId=0) const

Get the number of blobs in a camera frame.

- CORBA::Boolean **Send2dPoint** (const char *inId, const float &inXPosition, const float &inYPosition, const short &inCameraId, const unsigned long &inFrameId, const unsigned long &inTimeStamp)
  Upload a point with an identification to the server.

- CORBA::Boolean **Get2dPoint** (const char *inId, float &outXPosition, float &outYPosition, const short &inCameraId, unsigned long &outFrameId, unsigned long &outTimeStamp, const CORBA::Boolean &inLock=0)
  Download a point with an identification from the server.

- CORBA::Boolean **SendMode** (const char *inMode)
  Set the working mode of the whole system.

- CORBA::Boolean **GetMode** (std::string &outMode)
  Get the working mode of the whole system.

- CORBA::Boolean **SendImageSize** (const unsigned int &inDimX, const unsigned int &inDimY)
  Upload the image size (in pixels) to the server.

- CORBA::Boolean **GetImageSize** (unsigned int &outDimX, unsigned int &outDimY)
  Download the image size (in pixels) from the server.

- CORBA::Boolean **ResetServer** ()
  Reset the current frame set number (to 0) and the camera client count.

- CORBA::Boolean **SendAngle** (double inAngle)
  Upload an angle value to the server.

- CORBA::Boolean **GetAngle** (double &outAngle)
  Download an angle value from the server.

- FrameSet_var& **Frame** ()
  Reference to the private FrameSet.

- void **MinPoolingPeriod** (unsigned int inMinPoolingPeriod)
  Change the minimum pooling period; must be at least 1.

Protected Methods

- void **WaitToPool** ()
  Wait for the minimum server pooling period.

Private Methods

- **FrameSetClient** (const FrameSetClient &)
  Copy constructor: we do not allow copy!

- FrameSetClient& **operator=** (const FrameSetClient &)
  Copy operator: we do not allow copy!

- void **DestroyFrame** ()
  Delete all frame-related information.

Private Attributes

- bool **mCameraClientInitialized**
  Status of the camera client connection.

- FrameSet_var **mFrame**
  The frame being created/received from the server.

- SimpleXmlParser** **mXmlData**
  Each blob has a xml parser (this is an matrix).

- LARGE_INTEGER **mLastPooling**
- LARGE_INTEGER **mPerformanceFrequency**
  Server pooling time stamp.

- unsigned int **mMinPoolingPeriod**
  Minimum server pooling period.

- unsigned long **mLatestFrameId**
  Frame id of the last frame; used in GetNewFrameSet.

Detailed Description

Read and write blob information on the COGNOIS server.

Constructor & Destructor Documentation

Cognois::FrameSetClient::FrameSetClient (int *inArgc* = 0, char ** *inArgv* = 0)

Constructor with an optionnal command line.

Parameters:

inArgc is the number of arguments on the command line
inArgv is the command line agruments

Cognois::FrameSetClient::~FrameSetClient ()

Destructor.

Cognois::FrameSetClient::FrameSetClient (const FrameSetClient &) `[private]`

Copy constructor: we do not allow copy!

Member Function Documentation

unsigned long Cognois::FrameSetClient::BlobCount (const unsigned long & *inCameraId* = 0) const

Get the number of blobs in a camera frame.

Parameters:

inCameraId is the camera in wich to count the blobs.

Returns:

The number of blobs in a frame.

void Cognois::FrameSetClient::CreateFrame (const unsigned long & *inBlobCount*, const unsigned long & *inFrameId*, const unsigned long & *inTimeStamp*)

Create a frame (blob set) of empty elements.

Parameters:

inBlobCount is the number of blobs to send to the server
inFrameId is the frame id of the points
inTimeStamp is the time stamp of the points

void Cognois::FrameSetClient::DestroyFrame () `[private]`

Delete all frame-related information.

FrameSet_var & Cognois::FrameSetClient::Frame () `[inline]`

Reference to the private FrameSet.

unsigned long Cognois::FrameSetClient::FrameCount () const

Get the number of frames.

Returns:

The number of frames.

CORBA::Boolean Cognois::FrameSetClient::Get2dPoint (const char * *inId*, float & *outXPosition*, float & *outYPosition*, const short & *inCameraId*, unsigned long & *outFrameId*, unsigned long & *outTimeStamp*, const CORBA::Boolean & *inLock* = 0)

Download a point with an identification from the server.

Parameters:

inId is the identification of the point
outXPosition is the X position of the point
outYPosition is the Y position of the point
inCameraId is the camera ID to which the desired point was intended
outFrameId is the frame id of the point
outTimeStamp is the time stamp of the point
inLock is set to true if the client wants to be locked until a new data is sent

Returns:

If the download was successfull.

CORBA::Boolean Cognois::FrameSetClient::GetAngle (double & *outAngle*)

Download an angle value from the server.

Parameters:

outAngle is the angle value downloaded from the server.

Returns:

If the download was successfull.

CORBA::Boolean Cognois::FrameSetClient::GetImageSize (unsigned int & *outDimX*, unsigned int & *outDimY*)

Download the image size (in pixels) from the server.

Parameters:

outDimX is the image size in the X (horizontal) dimension
outDimY is the image size in the Y (vertical) dimension

Returns:

If the download was successfull.

CORBA::Boolean Cognois::FrameSetClient::GetLatestFrameSet (unsigned long & *outFrameId*, unsigned long & *outTimeStamp*, CORBA::Boolean *inGetBlobData* = 0)

Get a frame set from the server and put it in the private member variables.

Parameters:

outFrameId is the frame id of the points
outTimeStamp is the time stamp of the points
inGetBlobData is set to 1 if the blob contents is wanted

Returns:

If the frame set has never been downloaded from the server.

CORBA::Boolean Cognois::FrameSetClient::GetMode (std::string & *outMode*)

Get the working mode of the whole system.

Parameters:

outMode is the current working mode of the system

Returns:

If the download was successfull.

CORBA::Boolean Cognois::FrameSetClient::GetNewFrameSet (unsigned long & *outFrameId*, unsigned long & *outTimeStamp*, CORBA::Boolean *inGetBlobData* = 0, unsigned long *inMaxWaitTime* = 2000)

Get a NEW frame set from the server and put it in the private member variables.

Parameters:

outFrameId is the frame id of the points
outTimeStamp is the time stamp of the points
inGetBlobData is set to 1 if the blob contents is wanted
inMaxWaitingTime is the maximum time (in ms) to wait for a new frame set; throws an exception if no new frame set is received within this limit.

Returns:

If the frame set has never been downloaded from the server.

void Cognois::FrameSetClient::MinPoolingPeriod (unsigned int *inMinPoolingPeriod*)

Change the minimum pooling period; must be at least 1.

Parameters:

inMinPoolingPeriod is the new minimum pooling period; must be at least 1.

CORBA::Boolean Cognois::FrameSetClient::ResetServer ()

Reset the current frame set number (to 0) and the camera client count.

Returns:

If the operation was successfull.

CORBA::Boolean Cognois::FrameSetClient::Send2dPoint (const char * *inId*, const float & *inXPosition*, const float & *inYPosition*, const short & *inCameraId*, const unsigned long & *inFrameId*, const unsigned long & *inTimeStamp*)

Upload a point with an identification to the server.

Parameters:

inId is the identification of the point
inXPosition is the X position of the point
inYPosition is the Y position of the point
inCameraId is the camera ID to which the point is intended
inFrameId is the frame ID of the point
inTimeStamp is the time stamp of the point

Returns:

If the upload was successfull.

CORBA::Boolean Cognois::FrameSetClient::SendAngle (double *inAngle*)

Upload an angle value to the server.

Parameters:

inAngle is the angle to put on the server.

Returns:

If the upload was successfull.

CORBA::Boolean Cognois::FrameSetClient::SendFrame ()

Send a previously created frame, then destroys it.

Parameters:

inBlobCount is the number of blobs to send to the server
inFrameId is the frame id of the points
inTimeStamp is the time stamp of the points

Returns:

If the upload was successfull.

CORBA::Boolean Cognois::FrameSetClient::SendImageSize (const unsigned int & *inDimX*, const unsigned int & *inDimY*)

Upload the image size (in pixels) to the server.

Parameters:

inDimX is the image size in the X (horizontal) dimension
inDimY is the image size in the Y (vertical) dimension

Returns:

If the upload was successfull.

CORBA::Boolean Cognois::FrameSetClient::SendMode (const char * *inMode*)

Set the working mode of the whole system.

Parameters:

inMode is the new working mode of the system

Returns:

If the upload was successfull.

void Cognois::FrameSetClient::WaitToPool () [protected]

Wait for the minimum server pooling period.

Parameters:

inMinPoolingPeriod is the new minimum pooling period; must be at least 1.

Cognois::SimpleXmlParser & Cognois::FrameSetClient::XmlData (const unsigned long & *inBlobId*, const unsigned long & *inCameraId* = 0)

Address the xml content of a particular blob (created or received).

Parameters:

inBlobId is the blob ID of the desired XML data.
inCameraId is the camera in wich to look for the blob.

Returns:

A reference to the blob XML data.

FrameSetClient & Cognois::FrameSetClient::operator= (const FrameSetClient &) [private]

Copy operator: we do not allow copy!

Member Data Documentation

bool Cognois::FrameSetClient::mCameraClientInitialized [private]

Status of the camera client connection.

FrameSet_var Cognois::FrameSetClient::mFrame [private]

The frame being created/received from the server.

LARGE_INTEGER Cognois::FrameSetClient::mLastPooling [private]

unsigned long Cognois::FrameSetClient::mLatestFrameId [private]

Frame id of the last frame; used in GetNewFrameSet.

unsigned int Cognois::FrameSetClient::mMinPoolingPeriod [private]

Minimum server pooling period.

LARGE_INTEGER Cognois::FrameSetClient::mPerformanceFrequency [private]

Server pooling time stamp.

SimpleXmlParser ** Cognois::FrameSetClient::mXmlData `[private]`

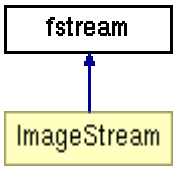
Each blob has a xml parser (this is an matrix).

The documentation for this class was generated from the following files:

- **FrameSetClient.h**
- **FrameSetClient.cpp**

fstream Class Reference

Inheritance diagram for fstream:

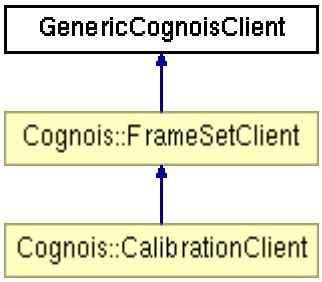


The documentation for this class was generated from the following file:

- **ImageStream.hpp**

GenericCognoisClient Class Reference

Inheritance diagram for GenericCognoisClient:



The documentation for this class was generated from the following file:

- **FrameSetClient.h**

GifApplicationBlock Struct Reference

```
#include <gif.h>
```

Public Attributes

- **UBYTE1 block_size**
- **UBYTE1 application_name** [8]
- **UBYTE1 authentication_code** [3]

Member Data Documentation

UBYTE1 GifApplicationBlock::application_name

UBYTE1 GifApplicationBlock::authentication_code

UBYTE1 GifApplicationBlock::block_size

The documentation for this struct was generated from the following file:

- **gif.h**

GifColorTableEntry Struct Reference

```
#include <gif.h>
```

Public Attributes

- **UBYTE1 red**
- **UBYTE1 green**
- **UBYTE1 blue**

Member Data Documentation

UBYTE1 GifColorTableEntry::blue

UBYTE1 GifColorTableEntry::green

UBYTE1 GifColorTableEntry::red

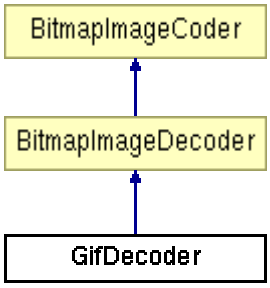
The documentation for this struct was generated from the following file:

- **gif.h**

GifDecoder Class Reference

`#include <gifdecod.h>`

Inheritance diagram for GifDecoder:



Public Methods

- **GifDecoder** ()
- virtual **~GifDecoder** ()
- **GifDecoder** (const GifDecoder &)
- GifDecoder& **operator=** (const GifDecoder &)
- virtual void **ReadImage** (istream &, **BitmapImage** &)
- void **ReadNextImage** (**BitmapImage** &)
- void **Reset** ()
- void **SetReadMultipleImages** (bool)
- bool **GetReadMultipleImages** () const
- bool **GetVerboseFlag** () const
- void **SetVerboseFlag** (bool)
- bool **MoreImages** () const
- unsigned int **ImageTop** () const
- unsigned int **ImageLeft** () const
- unsigned int **ScreenWidth** () const
- unsigned int **ScreenHeight** () const
- void **SetVerbose** (bool)
- bool **GetVerbose** () const

Private Methods

- void **Initialize** ()
- void **DoCopy** (const GifDecoder &source)
- void **ReadStreamHeader** ()
- void **ReadToNextImage** ()
- void **InitializeLocalData** ()
- int **NextBit** ()
- unsigned int **NextCode** ()
- void **LoadBuffer** ()
- void **AllocateReadBuffers** ()
- void **FreeReadBuffers** ()
- void **CallProgressFunction** (unsigned int percent)
- virtual void **UpdateImage** ()
- void **ReadCompressedData** ()

- void **StoreByte** (unsigned long offset, **UBYTE1** data)
- **UBYTE1 GetImageByte** (unsigned long offset)

Private Attributes

- unsigned int **global_color_count**
- **GifColorTableEntry*** **global_color_table**
- bool **more_images**
- unsigned int **image_width**
- unsigned int **image_height**
- bool **interlaced_image**
- unsigned int **current_row**
- unsigned int **current_column**
- unsigned int **interlace_group**
- unsigned int **rows_done**
- struct **GifDecoder::LZ77String** * **length_distance_table**
- unsigned char **code_size**
- unsigned int **end_code**
- unsigned int **bit_offset**
- unsigned int **byte_offset**
- unsigned char **current_block_size**
- unsigned char* **input_buffer**
- istream* **input_stream**
- bool **verbose_flag**
- bool **read_multiple_images**
- unsigned int **image_top**
- unsigned int **image_left**
- unsigned int **screen_width**
- unsigned int **screen_height**
- signed int **stack_pointer**
- **BitmapImage*** **current_image**
- **UBYTE1*** **lz_window**

Constructor & Destructor Documentation

GifDecoder::GifDecoder ()

GifDecoder::~GifDecoder () `[virtual]`

GifDecoder::GifDecoder (const GifDecoder & *source*)

Member Function Documentation

void GifDecoder::AllocateReadBuffers () [private]

void GifDecoder::CallProgressFunction (unsigned int *percent*) [private]

void GifDecoder::DoCopy (const GifDecoder & *source*) [private]

void GifDecoder::FreeReadBuffers () [private]

UBYTE1 GifDecoder::GetImageByte (unsigned long *offset*) [private]

bool GifDecoder::GetReadMultipleImages () const [inline]

bool GifDecoder::GetVerbose () const [inline]

bool GifDecoder::GetVerboseFlag () const [inline]

unsigned int GifDecoder::ImageLeft () const [inline]

unsigned int GifDecoder::ImageTop () const [inline]

void GifDecoder::Initialize () [private]

Reimplemented from **BitmapImageCoder** (*p.38*).

void GifDecoder::InitializeLocalData () [private]

void GifDecoder::LoadBuffer () [private]

bool GifDecoder::MoreImages () const [inline]

int GifDecoder::NextBit () [private]

unsigned int GifDecoder::NextCode () [private]

void GifDecoder::ReadCompressedData () [private]

void GifDecoder::ReadImage (istream & *strm*, BitmapImage & *image*) [virtual]

Reimplemented from **BitmapImageDecoder** (*p.39*).

void GifDecoder::ReadNextImage (BitmapImage & *image*)

void GifDecoder::ReadStreamHeader () `[private]`

void GifDecoder::ReadToNextImage () `[private]`

void GifDecoder::Reset ()

unsigned int GifDecoder::ScreenHeight () const `[inline]`

unsigned int GifDecoder::ScreenWidth () const `[inline]`

void GifDecoder::SetReadMultipleImages (bool *value*) `[inline]`

void GifDecoder::SetVerbose (bool *value*) `[inline]`

void GifDecoder::SetVerboseFlag (bool *value*) `[inline]`

void GifDecoder::StoreByte (unsigned long *offset*, UBYTE1 *data*) `[private]`

void GifDecoder::UpdateImage () `[private, virtual]`

Reimplemented from **BitmapImageCoder** (*p.38*).

GifDecoder & GifDecoder::operator= (const GifDecoder & *source*)

Member Data Documentation

unsigned int GifDecoder::bit_offset `[private]`

unsigned int GifDecoder::byte_offset `[private]`

unsigned char GifDecoder::code_size `[private]`

unsigned char GifDecoder::current_block_size `[private]`

unsigned int GifDecoder::current_column `[private]`

BitmapImage * GifDecoder::current_image `[private]`

unsigned int GifDecoder::current_row `[private]`

unsigned int GifDecoder::end_code `[private]`

unsigned int GifDecoder::global_color_count `[private]`

GifColorTableEntry * GifDecoder::global_color_table `[private]`

unsigned int GifDecoder::image_height `[private]`

unsigned int GifDecoder::image_left `[private]`

unsigned int GifDecoder::image_top `[private]`

unsigned int GifDecoder::image_width `[private]`

unsigned char * GifDecoder::input_buffer `[private]`

istream * GifDecoder::input_stream `[private]`

unsigned int GifDecoder::interlace_group `[private]`

bool GifDecoder::interlaced_image `[private]`

struct GifDecoder::LZ77String * GifDecoder::length_distance_table `[private]`

UBYTE1 * GifDecoder::lz_window `[private]`

bool GifDecoder::more_images `[private]`

bool GifDecoder::read_multiple_images `[private]`

unsigned int GifDecoder::rows_done `[private]`

unsigned int GifDecoder::screen_height `[private]`

unsigned int GifDecoder::screen_width `[private]`

signed int GifDecoder::stack_pointer `[private]`

bool GifDecoder::verbose_flag `[private]`

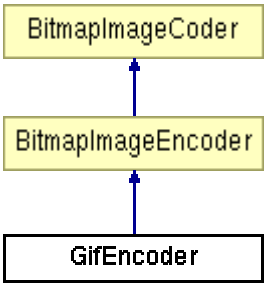
The documentation for this class was generated from the following files:

- **gifdecod.h**
- **gifdecod.cpp**

GifEncoder Class Reference

`#include <gifencod.h>`

Inheritance diagram for GifEncoder:



Public Methods

- **GifEncoder** ()
- **GifEncoder** (const GifEncoder &)
- virtual **~GifEncoder** ()
- GifEncoder& **operator=** (const GifEncoder &)
- virtual void **WriteImage** (ostream &, **BitmapImage** &)

Private Methods

- void **Initialize** ()
- void **DoCopy** (const GifEncoder &)
- void **CallProgressFunction** (unsigned int)
- void **OutputBits** (unsigned int value, unsigned int count)
- void **OutputCode** (unsigned int code)

Private Attributes

- ostream* **output_stream**
- unsigned int **bit_position**
- unsigned int **code_count**
- **BitmapImage quantized_image**
- **BitmapImage*** **current_image**
- **UBYTE1 data_buffer** [256]

Constructor & Destructor Documentation

GifEncoder::GifEncoder ()

GifEncoder::GifEncoder (const GifEncoder & *source*)

GifEncoder::~GifEncoder () `[virtual]`

Member Function Documentation

void GifEncoder::CallProgressFunction (unsigned int *percent*) `[private]`

void GifEncoder::DoCopy (const GifEncoder & *source*) `[private]`

void GifEncoder::Initialize () `[private]`

Reimplemented from **BitmapImageCoder** (*p.38*).

void GifEncoder::OutputBits (unsigned int *value*, unsigned int *count*) `[private]`

void GifEncoder::OutputCode (unsigned int *code*) `[private]`

void GifEncoder::WriteImage (ostream & *strm*, BitmapImage & *image*) `[virtual]`

Reimplemented from **BitmapImageEncoder** (*p.40*).

GifEncoder & GifEncoder::operator= (const GifEncoder & *source*)

Member Data Documentation

unsigned int GifEncoder::bit_position `[private]`

unsigned int GifEncoder::code_count `[private]`

BitmapImage * GifEncoder::current_image `[private]`

UBYTE1 GifEncoder::data_buffer `[private]`

ostream * GifEncoder::output_stream `[private]`

BitmapImage GifEncoder::quantized_image `[private]`

The documentation for this class was generated from the following files:

- **gifencod.h**
- **gifencod.cpp**

GifGraphicControlBlock Struct Reference

```
#include <gif.h>
```

Public Attributes

- **UBYTE1 block_size**
- **UBYTE1 packedfield**
- **UBYTE2 delay_time**
- **UBYTE1 transparent_color**

Member Data Documentation

UBYTE1 GifGraphicControlBlock::block_size

UBYTE2 GifGraphicControlBlock::delay_time

UBYTE1 GifGraphicControlBlock::packedfield

UBYTE1 GifGraphicControlBlock::transparent_color

The documentation for this struct was generated from the following file:

- **gif.h**

GifHeader Struct Reference

```
#include <gif.h>
```

Public Attributes

- **BYTE1 file_signature** [3]
- **BYTE1 file_version** [3]

Member Data Documentation

BYTE1 GifHeader::file_signature

BYTE1 GifHeader::file_version

The documentation for this struct was generated from the following file:

- **gif.h**

GifImageDescriptor Struct Reference

```
#include <gif.h>
```

Public Attributes

- **UBYTE2 left_position**
- **UBYTE2 top_position**
- **UBYTE2 image_width**
- **UBYTE2 image_height**
- **UBYTE1 packedfield**

Member Data Documentation

UBYTE2 GifImageDescriptor::image_height

UBYTE2 GifImageDescriptor::image_width

UBYTE2 GifImageDescriptor::left_position

UBYTE1 GifImageDescriptor::packedfield

UBYTE2 GifImageDescriptor::top_position

The documentation for this struct was generated from the following file:

- **gif.h**

GifScreenDescriptor Struct Reference

```
#include <gif.h>
```

Public Attributes

- **UBYTE2 logical_screen_width**
- **UBYTE2 logical_screen_height**
- **UBYTE1 packedfield**
- **UBYTE1 background_color_index**
- **UBYTE1 pixel_aspect_ratio**

Member Data Documentation

UBYTE1 GifScreenDescriptor::background_color_index

UBYTE2 GifScreenDescriptor::logical_screen_height

UBYTE2 GifScreenDescriptor::logical_screen_width

UBYTE1 GifScreenDescriptor::packedfield

UBYTE1 GifScreenDescriptor::pixel_aspect_ratio

The documentation for this struct was generated from the following file:

- **gif.h**

PngEncoder::HashEntry Struct Reference

Public Attributes

- **UBYTE2 index**
- HashEntry* **next**
- HashEntry* **previous**

Member Data Documentation

UBYTE2 PngEncoder::HashEntry::index

HashEntry * PngEncoder::HashEntry::next

HashEntry * PngEncoder::HashEntry::previous

The documentation for this struct was generated from the following file:

- **pngencod.h**

Contour::HoraudAngle Struct Reference

```
#include <Contour.hpp>
```

Public Attributes

- **Point q**
- **ContourSegment* l1**
- **ContourSegment * l2**

Member Data Documentation

ContourSegment * Contour::HoraudAngle::l1

ContourSegment * Contour::HoraudAngle::l2

Point Contour::HoraudAngle::q

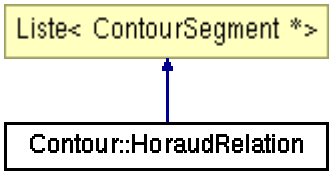
The documentation for this struct was generated from the following file:

- **Contour.hpp**

Contour::HoraudRelation Class Reference

`#include <Contour.hpp>`

Inheritance diagram for Contour::HoraudRelation:



The documentation for this class was generated from the following file:

- **Contour.hpp**

icilist Struct Reference

```
#include <F2C.H>
```

Public Attributes

- **flag icierr**
- char* **iciunit**
- **flag iciend**
- char* **icifmt**
- **ftnint icirlen**
- **ftnint icirnum**

Member Data Documentation

flag icilist::iciend

flag icilist::icierr

char * icilist::icifmt

ftnint icilist::icirlen

ftnint icilist::icirnum

char * icilist::iciunit

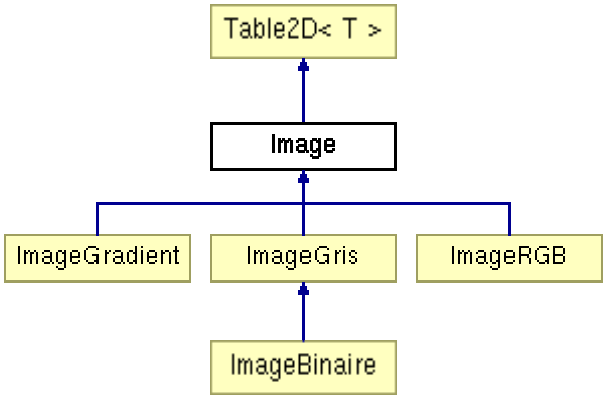
The documentation for this struct was generated from the following file:

- **F2C.H**

Image Class Template Reference

`#include <Image.hpp>`

Inheritance diagram for Image:



Public Methods

- **Image** ()
- **Image** (T min, T max)
- const Image& **operator=** (const Image &rh)
- **operator const T *** () const
- T& **operator()** (const **Point** &pn)
- T& **operator()** (int i, int j)
- const T& **operator()** (int i, int j) const
- const Image& **operator *=** (const **Operateur** &mask)
- Image **operator *** (const **Operateur** &mask) const
- const Image& **operator-=** (const Image &rh)
- Image **operator-** (const Image &rh) const
- const Image& **operator-=** (const T &rh)
- Image **operator-** (const T &rh) const
- const Image& **operator+=** (const Image &rh)
- Image **operator+** (const Image &rh) const
- const Image& **operator+=** (const T &rh)
- Image **operator+** (const T &rh) const
- const Image& **operator *=** (const Image &rh)
- Image **operator *** (const Image &rh) const
- const Image& **operator *=** (const double &rh)
- Image **operator *** (const double &rh) const
- const Image& **operator/=** (const Image &rh)
- Image **operator/** (const Image &rh) const
- const Image& **operator/=** (const double &rh)
- Image **operator/** (const double &rh) const
- const Image& **operator &=** (const Image &rh)
- Image **operator &** (const Image &rh) const
- const Image& **operator &=** (const T &rh)
- Image **operator &** (const T &rh) const
- const Image& **operator|=** (const Image &rh)
- Image **operator|** (const Image &rh) const

- const Image& **operator|=** (const T &rh)
- Image **operator|** (const T &rh) const
- const Image& **operator^=** (const Image &rh)
- Image **operator^** (const Image &rh) const
- const Image& **operator^=** (const T &rh)
- Image **operator^** (const T &rh) const
- Image **operator~** () const
- void **BruitFlipBits** (double prob=0.1)
- void **BruitGaussien** (double sigma=5)
- void **BruitImpulsion** (double prob=0.1)
- void **BruitSaltPepper** (double prob=0.1, double salt=0.5)
- void **FiltreConservateur** (unsigned int _taille=3)
- void **FiltreGaussien** (unsigned int _taille=3)
- void **FiltreMaxMin** (unsigned int _taille=3)
- void **FiltreMediane** (unsigned int _taille=3)
- void **FiltreMoyenne** (unsigned int _taille=3)
- void **FiltreUnsharp** (double k=0.5)
- Image& **HistoEqual** (double(*distrib)(double))
- Table1D<unsigned long> **Histogramme** (unsigned long classes=0) const
- Image& **HistoScale** (T min, T max)
- Image **RegionGet** (int ligne, int col, int dligne, int dcol) const
- void **RegionPut** (const Image &,int ligne, int col)
- T **Min** () const
- T **Max** () const
- void **MaxMin** (T &max, T &min) const
- T **MinValue** () const
- T **MaxValue** () const
- int **Width** () const
- int **Height** () const
- int **Length** () const

Protected Methods

- void **ObtenirMiroir** (Image &miroir, int h, int b, int g, int d)

Private Attributes

- T **minVal**
- T **maxVal**

template<class T> class Image

Constructor & Destructor Documentation

template<class T> Image< T >::Image<T> ()

template<class T> Image< T >::Image<T> (T *min*, T *max*)

Member Function Documentation

template<class T> void Image< T >::BruitFlipBits (double *prob* = 0.1)

template<class T> void Image< T >::BruitGaussien (double *sigma* = 5)

template<class T> void Image< T >::BruitImpulsion (double *prob* = 0.1)

template<class T> void Image< T >::BruitSaltPepper (double *prob* = 0.1, double *salt* = 0.5)

template<class T> void Image< T >::FiltreConservateur (unsigned int *_taille* = 3)

template<class T> void Image< T >::FiltreGaussien (unsigned int *_taille* = 3)

template<class T> void Image< T >::FiltreMaxMin (unsigned int *_taille* = 3)

template<class T> void Image< T >::FiltreMediane (unsigned int *_taille* = 3)

template<class T> void Image< T >::FiltreMoyenne (unsigned int *_taille* = 3)

template<class T> void Image< T >::FiltreUnsharp (double *k* = 0.5)

template<class T> int Image<T>::Height () const `[inline]`

template<class T> Image< T > & Image< T >::HistoEqual (double(* *distrib*)(double))

Reimplemented in **ImageRGB** (*p.196*).

template<class T> Image< T > & Image< T >::HistoScale (T *min*, T *max*)

template<class T> Table1D< unsigned long > Image< T >::Histogramme (unsigned long *classes* = 0) const

Reimplemented in **ImageRGB** (*p.197*).

template<class T> int Image<T>::Length () const `[inline]`

template<class T> T Image< T >::Max () const

template<class T> void Image< T >::MaxMin (T & *max*, T & *min*) const

template<class T> T Image<T>::MaxValue () const `[inline]`

template<class T> T Image< T >::Min () const

template<class T> T Image<T>::MinValue () const `[inline]`

template<class T> void Image< T >::ObtenirMiroir (Image<T> & *miroir*, int *h*, int *b*, int *g*, int *d*) `[protected]`

template<class T> Image< T > Image< T >::RegionGet (int *ligne*, int *col*, int *dligne*, int *dcol*) const

template<class T> void Image< T >::RegionPut (const Image<T> & *img*, int *ligne*, int *col*)

template<class T> int Image<T>::Width () const `[inline]`

template<class T> Image< T > Image< T >::operator & (const T & *rh*) const

template<class T> Image< T > Image< T >::operator & (const Image<T> & *rh*) const

template<class T> const Image< T > & Image< T >::operator &= (const T & *rh*)

template<class T> const Image< T > & Image< T >::operator &= (const Image<T> & *rh*)

template<class T> Image< T > Image< T >::operator * (const double & *rh*) const

template<class T> Image< T > Image< T >::operator * (const Image<T> & *rh*) const

template<class T> Image< T > Image< T >::operator * (const Operateur & *mask*) const

template<class T> const Image< T > & Image< T >::operator *= (const double & *rh*)

template<class T> const Image< T > & Image< T >::operator *= (const Image<T> & *rh*)

template<class T> const Image< T > & Image< T >::operator *= (const Operateur & *mask*)

template<class T> Image<T>::operator const T * () const `[inline]`

template<class T> const T & Image<T>::operator() (int *i*, int *j*) const `[inline]`

template<class T> T & Image<T>::operator() (int *i*, int *j*) `[inline]`

template<class T> T & Image< T >::operator() (const Point & *pn*) `[inline]`

template<class T> Image< T > Image< T >::operator+ (const T & *rh*) const

template<class T> Image< T > Image< T >::operator+ (const Image<T> & *rh*) const

template<class T> const Image< T > & Image< T >::operator+= (const T & *rh*)

template<class T> const Image< T > & Image< T >::operator+= (const Image<T> & *rh*)

template<class T> Image< T > Image< T >::operator- (const T & *rh*) const

template<class T> Image< T > Image< T >::operator- (const Image<T> & *rh*) const

template<class T> const Image< T > & Image< T >::operator-= (const T & *rh*)

template<class T> const Image< T > & Image< T >::operator-= (const Image<T> & *rh*)

template<class T> Image< T > Image< T >::operator/ (const double & *rh*) const

template<class T> Image< T > Image< T >::operator/ (const Image<T> & *rh*) const

template<class T> const Image< T > & Image< T >::operator/= (const double & *rh*)

template<class T> const Image< T > & Image< T >::operator/= (const Image<T> & *rh*)

template<class T> const Image< T > & Image< T >::operator= (const Image<T> & *rh*)
`[inline]`

template<class T> Image< T > Image< T >::operator^ (const T & *rh*) const

template<class T> Image< T > Image< T >::operator^ (const Image<T> & *rh*) const

template<class T> const Image< T > & Image< T >::operator^= (const T & *rh*)

template<class T> const Image< T > & Image< T >::operator^= (const Image<T> & *rh*)

template<class T> Image< T > Image< T >::operator| (const T & *rh*) const

template<class T> Image< T > Image< T >::operator| (const Image<T> & *rh*) const

template<class T> const Image< T > & Image< T >::operator|= (const T & *rh*)

template<class T> const Image< T > & Image< T >::operator|= (const Image<T> & *rh*)

template<class T> Image< T > Image< T >::operator~ () const

Member Data Documentation

template<class T> T Image<T>::maxVal `[private]`

template<class T> T Image<T>::minVal `[private]`

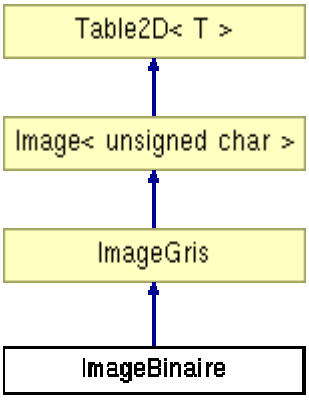
The documentation for this class was generated from the following file:

- **Image.hpp**

ImageBinaire Class Reference

`#include <ImageBinaire.hpp>`

Inheritance diagram for ImageBinaire:



Public Types

- enum **Statistique** { **MOYENNE**, **MEDIANE**, **MAXMIN** }

Public Methods

- ImageBinaire& **Treshold** (const **ImageGradient** &, int)
- ImageBinaire& **Treshold** (const **ImageGris** &, unsigned char)
- ImageBinaire& **Treshold** (const **ImageGris** &, const **Ensemble** &)
- ImageBinaire& **AdaptiveTresholding** (const **ImageGris** &,**Statistique** s=MOYENNE, int v=7, int cte=5)
- ImageBinaire& **IterativeTresholding** (const **ImageGris** &,unsigned char _t=128)
- ImageBinaire& **DoubleTresholding** (const **ImageGris** &, unsigned char _t1, unsigned char _t2)
- ImageBinaire& **ZeroCrossing** (const **ImageGradient** &)
- void **Dilate** (const **Kernel** &kernel=**Kernel**(Kernel::CROIX))
- void **Erode** (const **Kernel** &kernel=**Kernel**(Kernel::CROIX))
- void **Close** (const **Kernel** &kernel=**Kernel**(Kernel::CROIX))
- void **Open** (const **Kernel** &kernel=**Kernel**(Kernel::CROIX))
- void **Expand** (unsigned int k=1)
- void **Shrink** (unsigned int k=1)
- void **Boundary** ()
- **Table2D**<unsigned int> **ComponentLabeling** () const
- void **Skeleton** ()

Member Enumeration Documentation

enum ImageBinaire::Statistique

Enumeration values:

MOYENNE

MEDIANE
MAXMIN

Member Function Documentation

ImageBinaire & ImageBinaire::AdaptiveTresholding (const ImageGris & *img*, Statistique *s* = MOYENNE, int *v* = 7, int *cte* = 5)

void ImageBinaire::Boundary ()

void ImageBinaire::Close (const Kernel & *kernel* = Kernel(Kernel::CROIX))

Table2D< unsigned int > ImageBinaire::ComponentLabeling () const

void ImageBinaire::Dilate (const Kernel & *kernel* = Kernel(Kernel::CROIX))

ImageBinaire & ImageBinaire::DoubleTresholding (const ImageGris & *img*, unsigned char *_t1*, unsigned char *_t2*)

void ImageBinaire::Erode (const Kernel & *kernel* = Kernel(Kernel::CROIX))

void ImageBinaire::Expand (unsigned int *k* = 1)

ImageBinaire & ImageBinaire::IterativeTresholding (const ImageGris & *img*, unsigned char *tres* = 128)

void ImageBinaire::Open (const Kernel & *kernel* = Kernel(Kernel::CROIX))

void ImageBinaire::Shrink (unsigned int *k* = 1)

void ImageBinaire::Skeleton ()

ImageBinaire & ImageBinaire::Treshold (const ImageGris & *img*, const Ensemble & *ens*)

ImageBinaire & ImageBinaire::Treshold (const ImageGris & *img*, unsigned char *c*)

ImageBinaire & ImageBinaire::Treshold (const ImageGradient & *img*, int *seuil*)

ImageBinaire & ImageBinaire::ZeroCrossing (const ImageGradient & *img*)

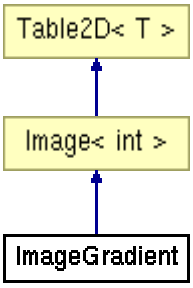
The documentation for this class was generated from the following files:

- **ImageBinaire.hpp**
- **ImageBinaire.cpp**

ImageGradient Class Reference

`#include <ImageGradient.hpp>`

Inheritance diagram for ImageGradient:



Public Methods

- **ImageGradient** ()
- ImageGradient& **Gradient** (const **ImageGris** &, int _taille=2)
- ImageGradient& **GradientMaxima** (const **ImageGris** &, int _taille=2)
- ImageGradient& **GradientMaxima** (const **ImageGris** &, const **Operateur** &dx, const **Operateur** &dy)
- void **GradXY** (const **ImageGris** &, ImageGradient &gradx, ImageGradient &grady, int _taille) const
- void **GradXY** (const **ImageGris** &, ImageGradient &gx, ImageGradient &gy, const **Operateur** &dx, const **Operateur** &dy) const
- void **DirectionGradient** (**Image**< int > &, const ImageGradient &gx, const ImageGradient &gy) const
- void **ModuleGradient** (ImageGradient &, const ImageGradient &gx, const ImageGradient &gy) const
- const ImageGradient& **operator=** (const **ImageGris** &rh)
- const ImageGradient& **operator=** (const **Image**< int > &rh)
- const ImageGradient& **operator=** (const ImageGradient &rh)

Public Attributes

- **Image**<int> **direction**

Constructor & Destructor Documentation

ImageGradient::ImageGradient () `[inline]`

Member Function Documentation

void ImageGradient::DirectionGradient (Image< int > & *dgr*, const ImageGradient & *gx*, const ImageGradient & *gy*) const

void ImageGradient::GradXY (const ImageGris & *entree*, ImageGradient & *gx*, ImageGradient & *gy*, const Operateur & *dx*, const Operateur & *dy*) const

void ImageGradient::GradXY (const ImageGris & *entree*, ImageGradient & *gradx*, ImageGradient & *grady*, int *_taille*) const

ImageGradient & ImageGradient::Gradient (const ImageGris & *entree*, int *_taille* = 2)

ImageGradient & ImageGradient::GradientMaxima (const ImageGris & *entree*, const Operateur & *dx*, const Operateur & *dy*)

ImageGradient & ImageGradient::GradientMaxima (const ImageGris & *entree*, int *_taille* = 2)

void ImageGradient::ModuleGradient (ImageGradient & *mg*, const ImageGradient & *gx*, const ImageGradient & *gy*) const

const ImageGradient & ImageGradient::operator= (const ImageGradient & *rh*)

const ImageGradient & ImageGradient::operator= (const Image< int > & *rh*)

const ImageGradient & ImageGradient::operator= (const ImageGris & *rh*)

Member Data Documentation

Image< int > ImageGradient::direction<int>

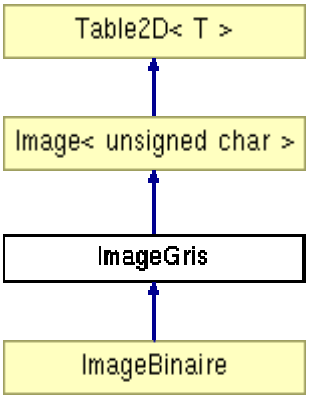
The documentation for this class was generated from the following files:

- **ImageGradient.hpp**
- **ImageGradient.cpp**

ImageGris Class Reference

`#include <ImageGris.hpp>`

Inheritance diagram for ImageGris:



Public Methods

- **ImageGris** ()
- **Contour Canny** (int t2=20, int g=3, int f=5) const
- **Contour Canny** (const **Operateur** &dx, const **Operateur** &dy, int t2=20, int f=5) const
- void **Canny** (**Contour** &, int t2=20, int g=3, int f=5) const
- void **Canny** (**Contour** &, const **Operateur** &dx, const **Operateur** &dy, int t2=20, int f=5) const
- **Liste**<**Point**> **Harris** (int np=0, double s=1, double k=0.06) const
- **Liste**<**Point**> **Horaud** (int w=5, int e1=5, int t=20) const
- void **LoG** (**Contour** &, int f=5, int t=20) const
- const ImageGris& **operator=** (const **ImageRGB** &)
- const ImageGris& **operator=** (const **ImageGradient** &)
- const ImageGris& **operator=** (const **Table2D**< unsigned > &)
- const ImageGris& **operator=** (const **BitmapImage** &)
- const ImageGris& **operator=** (const ImageGris &rh)
- const ImageGris& **operator=** (const **Image**< unsigned char > &rh)
- **operator BitmapImage** () const

Constructor & Destructor Documentation

ImageGris::ImageGris () `[inline]`

Member Function Documentation

void ImageGris::Canny (Contour & *ct*, const Operateur & *dx*, const Operateur & *dy*, int *_t2* = 20, int *_fsiz* = 5) const

void ImageGris::Canny (Contour & *ct*, int *t2* = 20, int *g* = 3, int *f* = 5) const

Contour ImageGris::Canny (const Operateur & *dx*, const Operateur & *dy*, int *t2* = 20, int *f* = 5) const

Contour ImageGris::Canny (int *t2* = 20, int *g* = 3, int *f* = 5) const

Liste< Point > ImageGris::Harris (int *np* = 0, double *sigma* = 1, double *k* = 0.06) const

Liste< Point > ImageGris::Horaud (int *w* = 5, int *e1* = 5, int *t* = 20) const

void ImageGris::LoG (Contour & *ct*, int *_fsiz* = 5, int *_t* = 20) const

ImageGris::operator BitmapImage () const

const ImageGris & ImageGris::operator= (const Image< unsigned char > & *rh*)

const ImageGris & ImageGris::operator= (const ImageGris & *rh*)

const ImageGris & ImageGris::operator= (const BitmapImage & *bi*)

const ImageGris & ImageGris::operator= (const Table2D< unsigned > & *rh*)

const ImageGris & ImageGris::operator= (const ImageGradient & *rh*)

const ImageGris & ImageGris::operator= (const ImageRGB & *rh*)

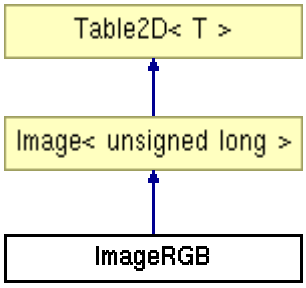
The documentation for this class was generated from the following files:

- **ImageGris.hpp**
- **ImageGris.cpp**

ImageRGB Class Reference

`#include <ImageRGB.hpp>`

Inheritance diagram for ImageRGB:



Public Methods

- **ImageRGB** ()
- const ImageRGB& **operator=** (const **ImageGris** &rh)
- const ImageRGB& **operator=** (const **BitmapImage** &bi)
- **Liste**<**Point**> **Harris** (int np=0, double s=1, double k=0.06) const
- **operator BitmapImage** () const

Protected Methods

- int **getCouleur** () const

Private Methods

- ImageRGB& **HistoEqual** (double(*distrib)(double))
- Table1D<unsigned long> **Histogramme** (unsigned long classes=0) const

---

Constructor & Destructor Documentation

ImageRGB::ImageRGB () `[inline]`

Member Function Documentation

Liste< Point > ImageRGB::Harris (int *np* = 0, double *sigma* = 1, double *k* = 0.06) const

ImageRGB & ImageRGB::HistoEqual (double(* *distrib*)(double)) `[private]`

Reimplemented from **Image** (*p.186*).

Table1D< unsigned long > ImageRGB::Histogramme<unsigned long> (unsigned long *classes* = 0) const `[private]`

Reimplemented from **Image** (*p.186*).

int ImageRGB::getCouleur () const `[protected]`

ImageRGB::operator BitmapImage () const

const ImageRGB & ImageRGB::operator= (const BitmapImage & *bi*)

const ImageRGB & ImageRGB::operator= (const ImageGris & *rh*)

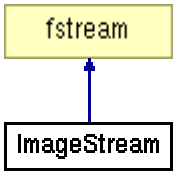
The documentation for this class was generated from the following files:

- **ImageRGB.hpp**
- **ImageRGB.cpp**

ImageStream Class Reference

`#include <ImageStream.hpp>`

Inheritance diagram for ImageStream:



Public Types

- enum **imgFormat** { **RAS** = 0, **BMP**, **JPG**, **GIF**, **PNG**, **XBM**, **XPM**, **PCX**, **XWD**, **PGM**, **PBM**, **PPM**, **CTR**, **NbFormats** }

Public Methods

- **ImageStream** (Chaine _ndf, int _mode, **imgFormat** _format=NbFormats)
- ImageStream& **operator**<< (const **ImageGris** &img)
- ImageStream& **operator**>> (**ImageGris** &img)
- ImageStream& **operator**<< (const **ImageRGB** &img)
- ImageStream& **operator**>> (**ImageRGB** &img)
- ImageStream& **operator**<< (const **Contour** &ctr)
- ImageStream& **operator**>> (**Contour** &ctr)
- void **SetVerbose** (bool _v=1)
- int **GetVerbose** () const

Protected Methods

- void **LireRaster** (**ImageRGB** &img)
- void **EcrireRaster** (const **ImageGris** &img)
- void **EcrireRaster** (const **ImageRGB** &img)
- void **LireBMP** (**ImageRGB** &img)
- void **EcrireBMP** (const **ImageRGB** &img)
- void **LireJPG** (**ImageRGB** &img)
- void **EcrireJPG** (const **ImageRGB** &img)
- void **EcrireJPG** (const **ImageGris** &img)
- void **LireGIF** (**ImageRGB** &img)
- void **EcrireGIF** (const **ImageRGB** &img)
- void **LirePNG** (**ImageRGB** &img)
- void **EcrirePNG** (const **ImageRGB** &img)
- void **LireXBM** (**ImageGris** &img)
- void **EcrireXBM** (const **ImageGris** &img)
- void **LireXPM** (**ImageRGB** &img)
- void **LirePCX** (**ImageRGB** &img)
- void **LireXWD** (**ImageRGB** &img)
- void **EcrireXWD** (const **ImageRGB** &img)
- void **LirePGM** (**ImageGris** &img)
- void **EcrirePGM** (const **ImageGris** &img)
- void **LirePBM** (**ImageGris** &img)

- void **EcrirePBM** (const **ImageGris** &img)
- void **LirePPM** (**ImageRGB** &img)
- void **EcrirePPM** (const **ImageRGB** &img)
- void **LireContour** (**Contour** &ctr)
- void **EcrireContour** (const **Contour** &ctr)
- Chaine **GetExtension** (Chaine filename)

Protected Attributes

- int **mode**
- bool **verbose_flag**
- **imgFormat format**
- streampos **beg_of_file**

Member Enumeration Documentation

enum ImageStream::imgFormat

Enumeration values:

RAS
BMP
JPG
GIF
PNG
XBM
XPM
PCX
XWD
PGM
PBM
PPM
CTR
NbFormats

Constructor & Destructor Documentation

ImageStream::ImageStream (Chaine *_ndf*, int *_mode*, imgFormat *_format* = NbFormats)

Member Function Documentation

void ImageStream::EcrireBMP (const ImageRGB & *img*) `[protected]`

void ImageStream::EcrireContour (const Contour & *ctr*) `[protected]`

void ImageStream::EcrireGIF (const ImageRGB & *img*) `[protected]`

void ImageStream::EcrireJPG (const ImageGris & *img*) `[protected]`

void ImageStream::EcrireJPG (const ImageRGB & *img*) `[protected]`

void ImageStream::EcrirePBM (const ImageGris & *img*) `[protected]`

void ImageStream::EcrirePGM (const ImageGris & *img*) `[protected]`

void ImageStream::EcrirePNG (const ImageRGB & *img*) `[protected]`

void ImageStream::EcrirePPM (const ImageRGB & *img*) `[protected]`

void ImageStream::EcrireRaster (const ImageRGB & *img*) `[protected]`

void ImageStream::EcrireRaster (const ImageGris & *img*) `[protected]`

void ImageStream::EcrireXBM (const ImageGris & *img*) `[protected]`

void ImageStream::EcrireXWD (const ImageRGB & *img*) `[protected]`

Chaine ImageStream::GetExtension (Chaine *filename*) `[protected]`

int ImageStream::GetVerbose () const

void ImageStream::LireBMP (ImageRGB & *img*) `[protected]`

void ImageStream::LireContour (Contour & *ctr*) `[protected]`

void ImageStream::LireGIF (ImageRGB & *img*) `[protected]`

void ImageStream::LireJPG (ImageRGB & *img*) `[protected]`

void ImageStream::LirePBM (ImageGris & *img*) `[protected]`

void ImageStream::LirePCX (ImageRGB & *img*) `[protected]`

void ImageStream::LirePGM (ImageGris & *img*) `[protected]`

void ImageStream::LirePNG (ImageRGB & *img*) `[protected]`

void ImageStream::LirePPM (ImageRGB & *img*) `[protected]`

void ImageStream::LireRaster (ImageRGB & *img*) `[protected]`

void ImageStream::LireXBM (ImageGris & *img*) `[protected]`

void ImageStream::LireXPM (ImageRGB & *img*) `[protected]`

void ImageStream::LireXWD (ImageRGB & *img*) `[protected]`

void ImageStream::SetVerbose (bool _*v* = 1)

ImageStream & ImageStream::operator<< (const Contour & *ctr*)

ImageStream & ImageStream::operator<< (const ImageRGB & *img*)

ImageStream & ImageStream::operator<< (const ImageGris & *img*)

ImageStream & ImageStream::operator>> (Contour & *ctr*)

ImageStream & ImageStream::operator>> (ImageRGB & *img*)

ImageStream & ImageStream::operator>> (ImageGris & *img*)

Member Data Documentation

streampos ImageStream::beg_of_file `[protected]`

imgFormat ImageStream::format `[protected]`

int ImageStream::mode `[protected]`

bool ImageStream::verbose_flag `[protected]`

The documentation for this class was generated from the following files:

- **ImageStream.hpp**
- **ImageStream.cpp**

IndiceAlias Class Template Reference

`#include <Equivalence.hpp>`

Public Methods

- **IndiceAlias** ()
- **IndiceAlias** (const T &_e)
- **IndiceAlias** (const T &_e, const T &_d)
- int **operator==** (const IndiceAlias &c) const

Public Attributes

- T **eff**
- T **des**

Friends

- ostream& **operator<<** (ostream &out, const IndiceAlias< T > &ia)
- istream& **operator>>** (istream &in, IndiceAlias< T > &ia)

template<class T> class IndiceAlias

Constructor & Destructor Documentation

template<class T> IndiceAlias<T>::IndiceAlias<T> () `[inline]`

template<class T> IndiceAlias<T>::IndiceAlias<T> (const T & _*e*) `[inline]`

template<class T> IndiceAlias<T>::IndiceAlias<T> (const T & _*e*, const T & _*d*) `[inline]`

Member Function Documentation

template<class T> int IndiceAlias<T>::operator== (const IndiceAlias<T> & *c*) const `[inline]`

Friends And Related Function Documentation

template<class T> ostream & operator<< (ostream & *out*, const IndiceAlias< T > & *ia*) `[friend]`

template<class T> istream & operator>> (istream & *in*, IndiceAlias< T > & *ia*) `[friend]`

Member Data Documentation

template<class T> T IndiceAlias<T>::des

template<class T> T IndiceAlias<T>::eff

The documentation for this class was generated from the following file:

- **Equivalence.hpp**

inlist Struct Reference

```
#include <F2C.H>
```

Public Attributes

- **flag inerr**
- **ftnint inunit**
- char* **infile**
- **ftnlen infilen**
- **ftnint* inex**
- **ftnint* inopen**
- **ftnint* innum**
- **ftnint* innamed**
- char* **inname**
- **ftnlen innamlen**
- char* **inacc**
- **ftnlen inacclen**
- char* **inseq**
- **ftnlen inseqlen**
- char* **indir**
- **ftnlen indirlen**
- char* **infmt**
- **ftnlen infmtlen**
- char* **inform**
- **ftnint informlen**
- char* **inunf**
- **ftnlen inunflen**
- **ftnint* inrecl**
- **ftnint* innrec**
- char* **inblank**
- **ftnlen inblanklen**

Member Data Documentation

char * inlist::inacc

ftnlen inlist::inacclen

char * inlist::inblank

ftnlen inlist::inblanklen

char * inlist::indir

ftnlen inlist::indirlen

flag inlist::inerr

ftnint * inlist::inex

char * inlist::infile

ftnlen inlist::infilen

char * inlist::infmt

ftnlen inlist::infmtlen

char * inlist::inform

ftnint inlist::informlen

char * inlist::inname

ftnint * inlist::innamed

ftnlen inlist::innamlen

ftnint * inlist::innrec

ftnint * inlist::innum

ftnint * inlist::inopen

ftnint * inlist::inrecl

char * inlist::inseq

ftnlen inlist::inseqlen

char * inlist::inunf

ftnlen inlist::inunflen

ftnint inlist::inunit

The documentation for this struct was generated from the following file:

- **F2C.H**

PngDecoder::InterlaceInfo Struct Reference

```
#include <pngdecod.h>
```

Public Attributes

- unsigned int **row_interval**
- unsigned int **col_interval**
- unsigned int **start_row**
- unsigned int **start_col**

Member Data Documentation

unsigned int PngDecoder::InterlaceInfo::col_interval

unsigned int PngDecoder::InterlaceInfo::row_interval

unsigned int PngDecoder::InterlaceInfo::start_col

unsigned int PngDecoder::InterlaceInfo::start_row

The documentation for this struct was generated from the following file:

- **pngdecod.h**

Intervalle Class Reference

`#include <ImageBinaire.hpp>`

Public Methods

- **Intervalle** (long _min=0)
- **Intervalle** (long _min, long _max)
- int **Appartient** (const long &nb)
- void **Entre** (long _min, long _max)
- void **Est** (long _min)
- void **PlusGrand** (long _min)
- void **PlusPetit** (long _max)

Public Attributes

- long **min**
- long **max**

Constructor & Destructor Documentation

Intervalle::Intervalle (long *_min* = 0) `[inline]`

Intervalle::Intervalle (long *_min*, long *_max*) `[inline]`

Member Function Documentation

int Intervalle::Appartient (const long & *nb*) `[inline]`

void Intervalle::Entre (long *_min*, long *_max*) `[inline]`

void Intervalle::Est (long *_min*) `[inline]`

void Intervalle::PlusGrand (long *_min*) `[inline]`

void Intervalle::PlusPetit (long *_max*) `[inline]`

Member Data Documentation

long Intervalle::max

long Intervalle::min

The documentation for this class was generated from the following file:

- **ImageBinaire.hpp**

JfifExtension Struct Reference

```
#include <jfif.h>
```

Public Types

- enum { **JpegThumbnail** = 0x10, **OneByteThumbnail** = 0x11, **ThreeByteThumbnail** = 0x13 }

Public Attributes

- **UBYTE2 length**
- **UBYTE1 identifier** [5]
- **UBYTE1 extension_code**

Member Enumeration Documentation

anonymous enum

Enumeration values:

JpegThumbnail
OneByteThumbnail
ThreeByteThumbnail

Member Data Documentation

UBYTE1 JfifExtension::extension_code

UBYTE1 JfifExtension::identifier

UBYTE2 JfifExtension::length

The documentation for this struct was generated from the following file:

- **jfif.h**

JfifHeader Struct Reference

```
#include <jfif.h>
```

Public Attributes

- **UBYTE2 length**
- **UBYTE1 identifier** [5]
- **UBYTE1 version** [2]
- **UBYTE1 units**
- **UBYTE2 xdensity**
- **UBYTE2 ydensity**
- **UBYTE1 xthumbnail**
- **UBYTE1 ythumbnail**

Member Data Documentation

UBYTE1 JfifHeader::identifier

UBYTE2 JfifHeader::length

UBYTE1 JfifHeader::units

UBYTE1 JfifHeader::version

UBYTE2 JfifHeader::xdensity

UBYTE1 JfifHeader::xthumbnail

UBYTE2 JfifHeader::ydensity

UBYTE1 JfifHeader::ythumbnail

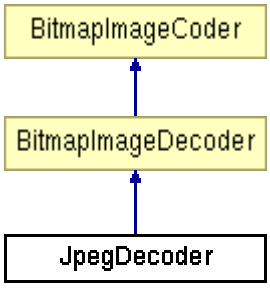
The documentation for this struct was generated from the following file:

- **jfif.h**

JpegDecoder Class Reference

`#include <jpegdeco.h>`

Inheritance diagram for JpegDecoder:



Public Methods

- **JpegDecoder** ()
- **JpegDecoder** (const JpegDecoder &)
- virtual **~JpegDecoder** ()
- JpegDecoder& **operator=** (const JpegDecoder &)
- virtual void **ReadImage** (istream &istrm, **BitmapImage** &image)
- bool **GetVerbose** () const
- void **SetVerbose** (bool)
- virtual void **UpdateImage** ()

Protected Methods

- **UBYTE1 ReadByte** ()
- **UBYTE2 ReadWord** ()
- int **NextBit** ()
- int **Receive** (unsigned int count)
- bool **IsProgressive** () const
- unsigned int **McuRows** () const
- unsigned int **McuCols** () const
- unsigned int **FrameHeight** () const
- unsigned int **FrameWidth** () const
- unsigned int **MaxVFrequency** () const
- unsigned int **MaxHFrequency** () const
- unsigned int **RestartInterval** () const

Private Methods

- void **Initialize** ()
- void **DoCopy** (const JpegDecoder &)
- void **CallProgressFunction** (unsigned int progress)
- void **GetScanCount** ()
- void **ReadStreamHeader** ()
- void **ReadMarker** ()
- void **ReadApplication** (**UBYTE1** type)
- void **ReadHuffmanTable** ()

- void **ReadQuantization** ()
- void **ReadRestartInterval** ()
- void **ReadStartOfFrame** (**UBYTE1** type)
- void **ReadStartOfScan** ()
- void **CalculateMcuDimensions** ()
- bool **ScanIsInterleaved** () const
- void **ReadProgressiveScanData** (unsigned int sss, unsigned int sse, unsigned int sah, unsigned int sal)
- void **ReadDcFirst** (unsigned int ssa)
- void **ReadDcRefine** (unsigned int ssa)
- void **ReadAcFirst** (unsigned int sss, unsigned int sse, unsigned int ssa)
- void **ReadAcRefine** (unsigned int sss, unsigned int sse, unsigned int ssa)
- void **FreeAllocatedResources** ()
- void **ReadSequentialScanData** ()
- void **ReadSequentialInterleavedScan** ()
- void **ReadSequentialNonInterleavedScan** ()
- void **ResetDcDifferences** ()
- void **ProcessRestartMarker** ()
- void **RefineAcCoefficient** (**BYTE2** &value, unsigned int ssa)

Private Attributes

- **JpegHuffmanDecoder*** **ac_tables**
- **JpegHuffmanDecoder*** **dc_tables**
- **JpegDecoderQuantizationTable*** **quantization_tables**
- int **bit_position**
- unsigned char **bit_data**
- bool **eoi_found**
- bool **sof_found**
- bool **verbose_flag**
- istream* **input_stream**
- unsigned int **restart_interval**
- unsigned int **frame_width**
- unsigned int **frame_height**
- unsigned int **frame_type**
- unsigned **max_horizontal_frequency**
- unsigned **max_vertical_frequency**
- bool **progressive_frame**
- unsigned int **component_count**
- **JpegDecoderComponent*** **components**
- unsigned int* **component_indices**
- **BitmapImage*** **current_image**
- unsigned int **current_scan**
- unsigned int **scan_count**
- unsigned int **mcu_rows**
- unsigned int **mcu_cols**
- unsigned int **mcu_height**
- unsigned int **mcu_width**
- unsigned int **scan_component_count**
- **JpegDecoderComponent**** **scan_components**
- unsigned int **expected_restart**
- bool **strict_jfif**

Friends

- class **JpegDecoderQuantizationTable**
- class **JpegHuffmanDecoder**
- class **JpegDecoderComponent**

Constructor & Destructor Documentation

JpegDecoder::JpegDecoder ()

JpegDecoder::JpegDecoder (const JpegDecoder & *source*)

JpegDecoder::~JpegDecoder () `[virtual]`

Member Function Documentation

void JpegDecoder::CalculateMcuDimensions () `[private]`

void JpegDecoder::CallProgressFunction (unsigned int *progress*) `[private]`

void JpegDecoder::DoCopy (const JpegDecoder & *source*) `[private]`

unsigned int JpegDecoder::FrameHeight () const `[inline, protected]`

unsigned int JpegDecoder::FrameWidth () const `[inline, protected]`

void JpegDecoder::FreeAllocatedResources () `[private]`

void JpegDecoder::GetScanCount () `[private]`

bool JpegDecoder::GetVerbose () const `[inline]`

void JpegDecoder::Initialize () `[private]`

Reimplemented from **BitmapImageCoder** (*p.38*).

bool JpegDecoder::IsProgressive () const `[inline, protected]`

unsigned int JpegDecoder::MaxHFrequency () const `[inline, protected]`

unsigned int JpegDecoder::MaxVFrequency () const `[inline, protected]`

unsigned int JpegDecoder::McuCols () const `[inline, protected]`

unsigned int JpegDecoder::McuRows () const `[inline, protected]`

int JpegDecoder::NextBit () `[protected]`

void JpegDecoder::ProcessRestartMarker () `[private]`

void JpegDecoder::ReadAcFirst (unsigned int *sss*, unsigned int *sse*, unsigned int *ssa*) `[private]`

void JpegDecoder::ReadAcRefine (unsigned int *sss*, unsigned int *sse*, unsigned int *ssa*) `[private]`

void JpegDecoder::ReadApplication (UBYTE1 *type*) `[private]`

UBYTE1 JpegDecoder::ReadByte () `[protected]`

void JpegDecoder::ReadDcFirst (unsigned int *ssa*) `[private]`

void JpegDecoder::ReadDcRefine (unsigned int *ssa*) `[private]`

void JpegDecoder::ReadHuffmanTable () `[private]`

void JpegDecoder::ReadImage (istream & *istrm*, BitmapImage & *image*) `[virtual]`

Reimplemented from **BitmapImageDecoder** (*p.39*).

void JpegDecoder::ReadMarker () `[private]`

void JpegDecoder::ReadProgressiveScanData (unsigned int *sss*, unsigned int *sse*, unsigned int *sah*, unsigned int *sal*) `[private]`

void JpegDecoder::ReadQuantization () `[private]`

void JpegDecoder::ReadRestartInterval () `[private]`

void JpegDecoder::ReadSequentialInterleavedScan () `[private]`

void JpegDecoder::ReadSequentialNonInterleavedScan () `[private]`

void JpegDecoder::ReadSequentialScanData () `[private]`

void JpegDecoder::ReadStartOfFrame (UBYTE1 *type*) `[private]`

void JpegDecoder::ReadStartOfScan () `[private]`

void JpegDecoder::ReadStreamHeader () `[private]`

UBYTE2 JpegDecoder::ReadWord () `[protected]`

int JpegDecoder::Receive (unsigned int *count*) `[protected]`

void JpegDecoder::RefineAcCoefficient (BYTE2 & *value*, unsigned int *ssa*) `[private]`

void JpegDecoder::ResetDcDifferences () `[private]`

unsigned int JpegDecoder::RestartInterval () const `[inline, protected]`

bool JpegDecoder::ScanIsInterleaved () const `[inline, private]`

void JpegDecoder::SetVerbose (bool *value*) `[inline]`

void JpegDecoder::UpdateImage () `[virtual]`

Reimplemented from **BitmapImageCoder** (*p.38*).

JpegDecoder & JpegDecoder::operator= (const JpegDecoder & *source*)

Friends And Related Function Documentation

class JpegDecoderComponent `[friend]`

class JpegDecoderQuantizationTable `[friend]`

class JpegHuffmanDecoder `[friend]`

Member Data Documentation

JpegHuffmanDecoder * JpegDecoder::ac_tables `[private]`

unsigned char JpegDecoder::bit_data `[private]`

int JpegDecoder::bit_position `[private]`

unsigned int JpegDecoder::component_count `[private]`

unsigned int * JpegDecoder::component_indices `[private]`

JpegDecoderComponent * JpegDecoder::components `[private]`

BitmapImage * JpegDecoder::current_image `[private]`

unsigned int JpegDecoder::current_scan `[private]`

JpegHuffmanDecoder * JpegDecoder::dc_tables `[private]`

bool JpegDecoder::eoi_found `[private]`

unsigned int JpegDecoder::expected_restart `[private]`

unsigned int JpegDecoder::frame_height `[private]`

unsigned int JpegDecoder::frame_type `[private]`

unsigned int JpegDecoder::frame_width `[private]`

istream * JpegDecoder::input_stream `[private]`

unsigned JpegDecoder::max_horizontal_frequency `[private]`

unsigned JpegDecoder::max_vertical_frequency `[private]`

unsigned int JpegDecoder::mcu_cols `[private]`

unsigned int JpegDecoder::mcu_height `[private]`

unsigned int JpegDecoder::mcu_rows `[private]`

unsigned int JpegDecoder::mcu_width `[private]`

bool JpegDecoder::progressive_frame `[private]`

JpegDecoderQuantizationTable * JpegDecoder::quantization_tables `[private]`

unsigned int JpegDecoder::restart_interval `[private]`

unsigned int JpegDecoder::scan_component_count `[private]`

JpegDecoderComponent ** JpegDecoder::scan_components `[private]`

unsigned int JpegDecoder::scan_count `[private]`

bool JpegDecoder::sof_found `[private]`

bool JpegDecoder::strict_jfif `[private]`

bool JpegDecoder::verbose_flag `[private]`

The documentation for this class was generated from the following files:

- **jpegdeco.h**
- **jpegdeco.cpp**

JpegDecoderComponent Class Reference

```
#include <jpdecomp.h>
```

Public Methods

- **JpegDecoderComponent** ()
- **~JpegDecoderComponent** ()
- unsigned int **HorizontalFrequency** () const
- void **HorizontalFrequency** (unsigned int)
- unsigned int **VerticalFrequency** () const
- void **VerticalFrequency** (unsigned int)
- void **SetQuantizationTable** (**JpegDecoderQuantizationTable** &table)
- void **AllocateComponentBuffers** (const **JpegDecoder** &decoder)
- void **FreeComponentBuffers** ()
- void **SetHuffmanTables** (**JpegHuffmanDecoder** &dc, **JpegHuffmanDecoder** &ac)
- void **Upsample** ()
- void **CheckAcTable** ()
- void **CheckDcTable** ()
- void **CheckQuantizationTable** ()
- void **DecodeSequential** (**JpegDecoder** &decoder, unsigned int mcurow, unsigned int mcucol)
- unsigned long **NoninterleavedRows** () const
- unsigned long **NoninterleavedCols** () const
- void **ResetDcDifference** ()
- void **DecodeDcFirst** (**JpegDecoder** &decoder, unsigned int row, unsigned int col, unsigned int ssa)
- void **DecodeDcRefine** (**JpegDecoder** &decoder, unsigned int row, unsigned int col, unsigned int ssa)
- void **DecodeAcFirst** (**JpegDecoder** &decoder, unsigned int row, unsigned int col, unsigned int sss, unsigned int sse, unsigned int ssa)
- void **DecodeAcRefine** (**JpegDecoder** &decoder, unsigned int row, unsigned int col, unsigned int sss, unsigned int sse, unsigned int ssa)
- void **ProgressiveInverseDct** ()

Static Public Methods

- void **RGBConvert** (JpegDecoderComponent &c1, JpegDecoderComponent &c2, JpegDecoderComponent &c3, **BitmapImage** &image)
- void **GrayscaleConvert** (JpegDecoderComponent &cc, **BitmapImage** &image)

Private Methods

- **JpegDecoderComponent** (const JpegDecoderComponent &)
- JpegDecoderComponent **operator=** (const JpegDecoderComponent &)

Private Attributes

- unsigned int **component_id**
- unsigned int **horizontal_frequency**
- unsigned int **vertical_frequency**
- unsigned int **v_sampling**
- unsigned int **h_sampling**
- int **last_dc_value**

- **JpegHuffmanDecoder* ac_table**
- **JpegHuffmanDecoder* dc_table**
- **JpegDecoderQuantizationTable* quantization_table**
- unsigned int **eob_run**
- unsigned int **noninterleaved_rows**
- unsigned int **noninterleaved_cols**
- unsigned int **du_rows**
- unsigned int **du_cols**
- **JpegDecoderDataUnit* data_units**
- **JPEGSAMPLE* upsample_data**
- **JpegDecoderCoefficientBlock* coefficient_blocks**

Constructor & Destructor Documentation

JpegDecoderComponent::JpegDecoderComponent ()

JpegDecoderComponent::~JpegDecoderComponent ()

JpegDecoderComponent::JpegDecoderComponent (const JpegDecoderComponent &) [private]

Member Function Documentation

void JpegDecoderComponent::AllocateComponentBuffers (const JpegDecoder & *decoder*)

void JpegDecoderComponent::CheckAcTable ()

void JpegDecoderComponent::CheckDcTable ()

void JpegDecoderComponent::CheckQuantizationTable ()

void JpegDecoderComponent::DecodeAcFirst (JpegDecoder & *decoder*, unsigned int *row*, unsigned int *col*, unsigned int *sss*, unsigned int *sse*, unsigned int *ssa*)

void JpegDecoderComponent::DecodeAcRefine (JpegDecoder & *decoder*, unsigned int *row*, unsigned int *col*, unsigned int *sss*, unsigned int *sse*, unsigned int *ssa*)

void JpegDecoderComponent::DecodeDcFirst (JpegDecoder & *decoder*, unsigned int *row*, unsigned int *col*, unsigned int *ssa*)

void JpegDecoderComponent::DecodeDcRefine (JpegDecoder & *decoder*, unsigned int *row*, unsigned int *col*, unsigned int *ssa*)

void JpegDecoderComponent::DecodeSequential (JpegDecoder & *decoder*, unsigned int *mcurow*, unsigned int *mcucol*)

void JpegDecoderComponent::FreeComponentBuffers ()

void JpegDecoderComponent::GrayscaleConvert (JpegDecoderComponent & *cc*, BitmapImage & *image*) `[static]`

void JpegDecoderComponent::HorizontalFrequency (unsigned int *value*)

unsigned int JpegDecoderComponent::HorizontalFrequency () const `[inline]`

unsigned long JpegDecoderComponent::NoninterleavedCols () const `[inline]`

unsigned long JpegDecoderComponent::NoninterleavedRows () const `[inline]`

void JpegDecoderComponent::ProgressiveInverseDct ()

void JpegDecoderComponent::RGBConvert (JpegDecoderComponent & *c1*, JpegDecoderComponent & *c2*, JpegDecoderComponent & *c3*, BitmapImage & *image*) `[static]`

void JpegDecoderComponent::ResetDcDifference () `[inline]`

void JpegDecoderComponent::SetHuffmanTables (JpegHuffmanDecoder & *dc*, JpegHuffmanDecoder & *ac*)

void JpegDecoderComponent::SetQuantizationTable (JpegDecoderQuantizationTable & *table*)

void JpegDecoderComponent::Upsample ()

void JpegDecoderComponent::VerticalFrequency (unsigned int *value*)

unsigned int JpegDecoderComponent::VerticalFrequency () const `[inline]`

JpegDecoderComponent JpegDecoderComponent::operator= (const JpegDecoderComponent &) `[private]`

Member Data Documentation

JpegHuffmanDecoder * JpegDecoderComponent::ac_table `[private]`

JpegDecoderCoefficientBlock * JpegDecoderComponent::coefficient_blocks `[private]`

unsigned int JpegDecoderComponent::component_id `[private]`

JpegDecoderDataUnit * JpegDecoderComponent::data_units `[private]`

JpegHuffmanDecoder * JpegDecoderComponent::dc_table `[private]`

unsigned int JpegDecoderComponent::du_cols `[private]`

unsigned int JpegDecoderComponent::du_rows `[private]`

unsigned int JpegDecoderComponent::eob_run `[private]`

unsigned int JpegDecoderComponent::h_sampling `[private]`

unsigned int JpegDecoderComponent::horizontal_frequency `[private]`

int JpegDecoderComponent::last_dc_value `[private]`

unsigned int JpegDecoderComponent::noninterleaved_cols `[private]`

unsigned int JpegDecoderComponent::noninterleaved_rows `[private]`

JpegDecoderQuantizationTable * JpegDecoderComponent::quantization_table
`[private]`

JPEGSAMPLE * JpegDecoderComponent::upsample_data `[private]`

unsigned int JpegDecoderComponent::v_sampling `[private]`

unsigned int JpegDecoderComponent::vertical_frequency `[private]`

The documentation for this class was generated from the following files:

- **jpdecomp.h**
- **jpdecomp.cpp**

JpegDecoderDataUnit Class Reference

`#include <jpdedu.h>`

Public Types

- typedef JpegDecoderDataUnit& (JpegDecoderDataUnit::* **IDctFunction**)(const **JpegDecoderCoefficientBlock**, const **JpegDecoderQuantizationTable** &)

Public Methods

- **JpegDecoderDataUnit** ()
- virtual **~JpegDecoderDataUnit** ()
- void **Print** (ostream &) const
- JpegDecoderDataUnit& **InverseDCT** (const **JpegDecoderCoefficientBlock** cb, const **JpegDecoderQuantizationTable** &qt)
- JpegDecoderDataUnit& **FloatInverseDCT** (const **JpegDecoderCoefficientBlock**, const **JpegDecoderQuantizationTable** &)
- JpegDecoderDataUnit& **IntegerInverseDCT** (const **JpegDecoderCoefficientBlock**, const **JpegDecoderQuantizationTable** &)
- **JPEGSAMPLE*** **operator[]** (unsigned int ii)
- **JPEGSAMPLE** const* **operator[]** (unsigned int ii) const

Private Methods

- **JpegDecoderDataUnit** (const JpegDecoderDataUnit &)
- JpegDecoderDataUnit& **operator=** (const JpegDecoderDataUnit &)

Private Attributes

- **UBYTE1 values** [**JpegSampleWidth**][**JpegSampleWidth**]

Static Private Attributes

- **IDctFunction idct_function** = &JpegDecoderDataUnit::IntegerInverseDCT

Member Typedef Documentation

typedef JpegDecoderDataUnit &(JpegDecoderDataUnit::* JpegDecoderDataUnit::IDctFunction)(const JpegDecoderCoefficientBlock, const JpegDecoderQuantizationTable &)

Constructor & Destructor Documentation

JpegDecoderDataUnit::JpegDecoderDataUnit () `[inline]`

JpegDecoderDataUnit::~JpegDecoderDataUnit () `[inline, virtual]`

JpegDecoderDataUnit::JpegDecoderDataUnit (const JpegDecoderDataUnit & *du*) `[private]`

Member Function Documentation

JpegDecoderDataUnit & JpegDecoderDataUnit::FloatInverseDCT (const *data*, const JpegDecoderQuantizationTable & *qt*)

JpegDecoderDataUnit & JpegDecoderDataUnit::IntegerInverseDCT (const *data*, const JpegDecoderQuantizationTable & *qt*)

JpegDecoderDataUnit & JpegDecoderDataUnit::InverseDCT (const JpegDecoderCoefficientBlock *cb*, const JpegDecoderQuantizationTable & *qt*) `[inline]`

void JpegDecoderDataUnit::Print (ostream & *strm*) const

JpegDecoderDataUnit & JpegDecoderDataUnit::operator= (const JpegDecoderDataUnit & *du*) `[private]`

JPEGSAMPLE const * JpegDecoderDataUnit::operator[] (unsigned int *ii*) const `[inline]`

JPEGSAMPLE * JpegDecoderDataUnit::operator[] (unsigned int *ii*) `[inline]`

Member Data Documentation

JpegDecoderDataUnit::IDctFunction JpegDecoderDataUnit::idct_function = &JpegDecoderDataUnit::IntegerInverseDCT `[static, private]`

UBYTE1 JpegDecoderDataUnit::values `[private]`

The documentation for this class was generated from the following files:

- **jpdedu.h**
- **jpdedu.cpp**

JpegDecoderQuantizationTable Class Reference

`#include <jpdequan.h>`

Public Methods

- **JpegDecoderQuantizationTable** ()
- **~JpegDecoderQuantizationTable** ()
- bool **Defined** () const
- void **ReadTable** (**JpegDecoder** &decoder, unsigned int precision)
- void **BuildScaledTables** ()
- void **Print** (ostream &) const

Private Types

- enum { **QuantizationIntegerScale** = 12 }

Private Methods

- **JpegDecoderQuantizationTable** (const JpegDecoderQuantizationTable &)
- JpegDecoderQuantizationTable& **operator=** (const JpegDecoderQuantizationTable &)

Private Attributes

- **UBYTE2 data_values** [**JpegSampleSize**]
- double **float_scaling** [**JpegSampleWidth**][**JpegSampleWidth**]
- long **integer_scaling** [**JpegSampleWidth**][**JpegSampleWidth**]
- bool **table_defined**

Friends

- class **JpegDecoderDataUnit**

Member Enumeration Documentation

anonymous enum `[private]`

Enumeration values:

QuantizationIntegerScale

Constructor & Destructor Documentation

JpegDecoderQuantizationTable::JpegDecoderQuantizationTable ()

JpegDecoderQuantizationTable::~JpegDecoderQuantizationTable () `[inline]`

JpegDecoderQuantizationTable::JpegDecoderQuantizationTable (const JpegDecoderQuantizationTable &) `[private]`

Member Function Documentation

void JpegDecoderQuantizationTable::BuildScaledTables ()

bool JpegDecoderQuantizationTable::Defined () const `[inline]`

void JpegDecoderQuantizationTable::Print (ostream & *strm*) const

void JpegDecoderQuantizationTable::ReadTable (JpegDecoder & *decoder*, unsigned int *precision*)

JpegDecoderQuantizationTable & JpegDecoderQuantizationTable::operator= (const JpegDecoderQuantizationTable &) `[private]`

Friends And Related Function Documentation

class JpegDecoderDataUnit `[friend]`

Member Data Documentation

UBYTE2 JpegDecoderQuantizationTable::data_values `[private]`

double JpegDecoderQuantizationTable::float_scaling `[private]`

long JpegDecoderQuantizationTable::integer_scaling `[private]`

bool JpegDecoderQuantizationTable::table_defined `[private]`

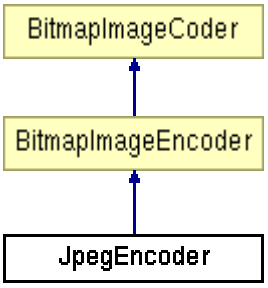
The documentation for this class was generated from the following files:

- **jpdequan.h**
- **jpdequan.cpp**

JpegEncoder Class Reference

`#include <jpegenco.h>`

Inheritance diagram for JpegEncoder:



Public Types

- enum { **MaxScans** = 256 }
- enum { **YComponent** = 1, **CbComponent** = 2, **CrComponent** = 3 }

Public Methods

- **JpegEncoder** ()
- virtual **~JpegEncoder** ()
- **JpegEncoder** (const JpegEncoder &)
- JpegEncoder& **operator=** (const JpegEncoder &)
- virtual void **WriteImage** (ostream &strm, **BitmapImage** &image)
- unsigned int **GetQuality** () const
- void **SetQuality** (unsigned int)
- bool **GetGrayscale** () const
- void **SetGrayscale** (bool)
- bool **GetProgressive** () const
- void **SetProgressive** (bool)
- unsigned int **GetRowsPerRestart** () const
- void **SetRowsPerRestart** (unsigned int)
- Chaine **GetComment** () const
- void **SetComment** (const Chaine &)
- void **SetSamplingFrequency** (unsigned int component, unsigned int hf, unsigned int vf)
- void **GetSamplingFrequency** (unsigned int component, unsigned int &hf, unsigned int &vf)
- void **SetScanAttributes** (unsigned int scan, unsigned long components, unsigned int sse, unsigned int ssa)
- void **GetScanAttributes** (unsigned int scan, unsigned long &components, unsigned int &sse, unsigned int &ssa)

Protected Methods

- void **OutputBits** (int bits, unsigned int count)
- void **OutputByte** (**UBYTE1**)
- void **OutputWord** (**UBYTE2**)
- void **CallProgressFunction** (unsigned int)

Private Types

- enum { **MaxComponents** = 4 }

Private Methods

- void **DoCopy** (const JpegEncoder &)
- void **Initialize** ()
- void **ValidateParameters** ()
- void **FlushBitBuffer** ()
- void **OutputMarker** (**UBYTE1**)
- void **PrintQuantizationTables** ()
- void **PrintSequentialFrame** (**BitmapImage** &image)
- void **PrintProgressiveFrame** (**BitmapImage** &image)
- void **PrintComment** (const Chaine &)
- void **OutputJfifHeader** ()
- void **OutputRestartInterval** (unsigned int restartinterval)
- void **PrintHuffmanTables** (const **Scan** &scan, bool usedc, bool useac)
- void **PrintSequentialScan** (const **Scan** &scan)
- void **PrintProgressiveScan** (const **Scan** &scan)
- void **PrintDcFirst** (const **Scan** &scan)
- void **PrintDcRefine** (const **Scan** &scan)
- void **PrintAcFirst** (const **Scan** &scan)
- void **PrintAcRefine** (const **Scan** &scan)
- void **FirstAcData** (const **Scan** &scan, bool outputrestarts, **JpegEncoderComponent::ACOUTPUTFUNCTION** acfunction)
- void **RefineAcData** (const **Scan** &scan, bool outputrestarts, **JpegEncoderComponent::ACOUTPUTFUNCTION** acfunction)
- void **InterleavedPass** (bool writedata, **JpegEncoderComponent::COMPONENTPASSFUNCTION** passfunction, **JpegEncoderComponent::DCOUTPUTFUNCTION** dcfunction, **JpegEncoderComponent::ACOUTPUTFUNCTION** acfunction, unsigned int sss, unsigned int sse, unsigned int ssa)
- void **NoninterleavedPass** (bool writedata, **JpegEncoderComponent::COMPONENTPASSFUNCTION** passfunction, **JpegEncoderComponent::DCOUTPUTFUNCTION** dcfunction, **JpegEncoderComponent::ACOUTPUTFUNCTION** acfunction, unsigned int sss, unsigned int sse, unsigned int ssa)
- void **ResetDcValues** ()
- void **CalculateMcuDimensions** ()
- void **CountPassesForProgressReporting** ()
- void **FindComponentsInScan** (**Scan** &scan)
- void **CreateQuantizationTables** (unsigned int quality)

Private Attributes

- unsigned int **bit_count**
- **UBYTE1 bit_buffer**
- **JpegEncoderQuantizationTable*** **chrominance_quanttbl**
- **JpegEncoderQuantizationTable*** **luminance_quanttbl**
- **JpegEncoderHuffmanTable*** **ac_tables**
- **JpegEncoderHuffmanTable*** **dc_tables**
- bool **gray_scale**
- unsigned int **rows_per_restart**
- unsigned int **restart_interval**

- unsigned int **image_quality**
- Chaine **comment_string**
- bool **progressive_mode**
- unsigned int **total_passes**
- unsigned int **current_pass**
- unsigned int **frame_width**
- unsigned int **frame_height**
- unsigned int **max_horizontal_frequency**
- unsigned int **max_vertical_frequency**
- unsigned int **mcu_rows**
- unsigned int **mcu_cols**
- **BitmapImage*** **current_image**
- ostream* **output_stream**
- unsigned int **scan_count**
- **Scan image_scans** [MaxScans]
- **JpegEncoderComponent image_components** [MaxComponents]
- unsigned int **scan_component_count**
- **JpegEncoderComponent*** **scan_components** [**JpegMaxComponentsPerScan**]

Friends

- class **JpegEncoderHuffmanTable**
- class **JpegEncoderComponent**

Member Enumeration Documentation

anonymous enum

Enumeration values:

MaxScans

anonymous enum

Enumeration values:

YComponent
CbComponent
CrComponent

anonymous enum `[private]`

Enumeration values:

MaxComponents

Constructor & Destructor Documentation

JpegEncoder::JpegEncoder ()

JpegEncoder::~JpegEncoder () `[virtual]`

JpegEncoder::JpegEncoder (const JpegEncoder & *source*)

Member Function Documentation

void JpegEncoder::CalculateMcuDimensions () `[private]`

void JpegEncoder::CallProgressFunction (unsigned int *progress*) `[inline, protected]`

void JpegEncoder::CountPassesForProgressReporting () `[private]`

void JpegEncoder::CreateQuantizationTables (unsigned int *quality*) `[private]`

void JpegEncoder::DoCopy (const JpegEncoder & *source*) `[private]`

void JpegEncoder::FindComponentsInScan (Scan & *scan*) `[private]`

void JpegEncoder::FirstAcData (const Scan & *scan*, bool *outputrestarts*, JpegEncoderComponent::ACOUTPUTFUNCTION *acfunction*) `[private]`

void JpegEncoder::FlushBitBuffer () `[private]`

Chaine JpegEncoder::GetComment () const

bool JpegEncoder::GetGrayscale () const `[inline]`

bool JpegEncoder::GetProgressive () const

unsigned int JpegEncoder::GetQuality () const

unsigned int JpegEncoder::GetRowsPerRestart () const `[inline]`

void JpegEncoder::GetSamplingFrequency (unsigned int *component*, unsigned int & *hf*, unsigned int & *vf*)

void JpegEncoder::GetScanAttributes (unsigned int *scan*, unsigned long & *components*, unsigned int & *sse*, unsigned int & *ssa*)

void JpegEncoder::Initialize () `[private]`

Reimplemented from **BitmapImageCoder** (*p.38*).

void JpegEncoder::InterleavedPass (bool *writedata*, JpegEncoderComponent::COMPONENTPASSFUNCTION *passfunction*, JpegEncoderComponent::DCOUTPUTFUNCTION *dcfunction*, JpegEncoderComponent::ACOUTPUTFUNCTION *acfunction*, unsigned int *sss*, unsigned int *sse*, unsigned int *ssa*) `[private]`

void JpegEncoder::NoninterleavedPass (bool *writedata*, JpegEncoderComponent::COMPONENTPASSFUNCTION *passfunction*, JpegEncoderComponent::DCOUTPUTFUNCTION *dcfunction*, JpegEncoderComponent::ACOUTPUTFUNCTION *acfunction*, unsigned int *sss*, unsigned int *sse*, unsigned int *ssa*) `[private]`

void JpegEncoder::OutputBits (int *bits*, unsigned int *count*) `[protected]`

void JpegEncoder::OutputByte (UBYTE1 *value*) `[protected]`

void JpegEncoder::OutputJfifHeader () `[private]`

void JpegEncoder::OutputMarker (UBYTE1 *marker*) `[private]`

void JpegEncoder::OutputRestartInterval (unsigned int *restartinterval*) `[private]`

void JpegEncoder::OutputWord (UBYTE2 *value*) `[protected]`

void JpegEncoder::PrintAcFirst (const Scan & *scan*) `[private]`

void JpegEncoder::PrintAcRefine (const Scan & *scan*) `[private]`

void JpegEncoder::PrintComment (const Chaine & *str*) `[private]`

void JpegEncoder::PrintDcFirst (const Scan & *scan*) `[private]`

void JpegEncoder::PrintDcRefine (const Scan & *scan*) `[private]`

void JpegEncoder::PrintHuffmanTables (const Scan & *scan*, bool *usedc*, bool *useac*) `[private]`

void JpegEncoder::PrintProgressiveFrame (BitmapImage & *image*) `[private]`

void JpegEncoder::PrintProgressiveScan (const Scan & *scan*) `[private]`

void JpegEncoder::PrintQuantizationTables () `[private]`

void JpegEncoder::PrintSequentialFrame (BitmapImage & *image*) `[private]`

void JpegEncoder::PrintSequentialScan (const Scan & *scan*) `[private]`

void JpegEncoder::RefineAcData (const Scan & *scan*, bool *outputrestarts*, JpegEncoderComponent::ACOUTPUTFUNCTION *acfunction*) `[private]`

void JpegEncoder::ResetDcValues () `[private]`

void JpegEncoder::SetComment (const Chaine &)

void JpegEncoder::SetGrayscale (bool *value*) `[inline]`

void JpegEncoder::SetProgressive (bool *value*)

void JpegEncoder::SetQuality (unsigned int *value*)

void JpegEncoder::SetRowsPerRestart (unsigned int *rows*) `[inline]`

void JpegEncoder::SetSamplingFrequency (unsigned int *component*, unsigned int *hf*, unsigned int *vf*)

void JpegEncoder::SetScanAttributes (unsigned int *scan*, unsigned long *components*, unsigned int *sse*, unsigned int *ssa*)

void JpegEncoder::ValidateParameters () `[private]`

void JpegEncoder::WriteImage (ostream & *strm*, BitmapImage & *image*) `[virtual]`

Reimplemented from **BitmapImageEncoder** (*p.40*).

JpegEncoder & JpegEncoder::operator= (const JpegEncoder & *source*)

Friends And Related Function Documentation

class JpegEncoderComponent `[friend]`

class JpegEncoderHuffmanTable `[friend]`

Member Data Documentation

JpegEncoderHuffmanTable * JpegEncoder::ac_tables `[private]`

UBYTE1 JpegEncoder::bit_buffer `[private]`

unsigned int JpegEncoder::bit_count `[private]`

JpegEncoderQuantizationTable * JpegEncoder::chrominance_quanttbl `[private]`

Chaine JpegEncoder::comment_string `[private]`

BitmapImage * JpegEncoder::current_image `[private]`

unsigned int JpegEncoder::current_pass `[private]`

JpegEncoderHuffmanTable * JpegEncoder::dc_tables `[private]`

unsigned int JpegEncoder::frame_height `[private]`

unsigned int JpegEncoder::frame_width `[private]`

bool JpegEncoder::gray_scale `[private]`

JpegEncoderComponent JpegEncoder::image_components `[private]`

unsigned int JpegEncoder::image_quality `[private]`

Scan JpegEncoder::image_scans `[private]`

JpegEncoderQuantizationTable * JpegEncoder::luminance_quanttbl `[private]`

unsigned int JpegEncoder::max_horizontal_frequency `[private]`

unsigned int JpegEncoder::max_vertical_frequency `[private]`

unsigned int JpegEncoder::mcu_cols `[private]`

unsigned int JpegEncoder::mcu_rows `[private]`

ostream * JpegEncoder::output_stream `[private]`

bool JpegEncoder::progressive_mode `[private]`

unsigned int JpegEncoder::restart_interval `[private]`

unsigned int JpegEncoder::rows_per_restart `[private]`

unsigned int JpegEncoder::scan_component_count `[private]`

JpegEncoderComponent * JpegEncoder::scan_components `[private]`

unsigned int JpegEncoder::scan_count `[private]`

unsigned int JpegEncoder::total_passes `[private]`

The documentation for this class was generated from the following files:

- **jpegenco.h**
- **jpegenco.cpp**

JpegEncoderCoefficientBlock Class Reference

`#include <jpencobk.h>`

Public Methods

- **BYTE2*** **operator[]** (unsigned int index)
- **BYTE2**& **ZigZagInput** (unsigned int index)

Private Attributes

- **BYTE2 data** [**JpegSampleWidth**][**JpegSampleWidth**]

Member Function Documentation

BYTE2 & JpegEncoderCoefficientBlock::ZigZagInput (unsigned int *index*) `[inline]`

BYTE2 * JpegEncoderCoefficientBlock::operator[] (unsigned int *index*) `[inline]`

Member Data Documentation

BYTE2 JpegEncoderCoefficientBlock::data `[private]`

The documentation for this class was generated from the following file:

- **jpencobk.h**

JpegEncoderComponent Class Reference

```
#include <jpencomp.h>
```

Public Types

- typedef void (JpegEncoderComponent::* **DCOUTPUTFUNCTION**)(int, int)
- typedef void (JpegEncoderComponent::* **ACOUTPUTFUNCTION**)(int, int, int)
- typedef void (JpegEncoderComponent::* **COMPONENTPASSFUNCTION**)(unsigned int row, unsigned int col, **DCOUTPUTFUNCTION**, **ACOUTPUTFUNCTION**, unsigned int sss, unsigned int sse, unsigned int ssa)

Public Methods

- **JpegEncoderComponent** ()
- virtual **~JpegEncoderComponent** ()
- void **PrintAcData** (int, int, int)
- void **PrintDcData** (int, int)
- void **GatherAcData** (int, int, int)
- void **GatherDcData** (int, int)
- void **EncodeSequential** (unsigned int row, unsigned int col, **DCOUTPUTFUNCTION** dcfunction, **ACOUTPUTFUNCTION** acfunction, unsigned int, unsigned int, unsigned int)
- void **ProgressiveDcFirst** (unsigned int row, unsigned int col, **DCOUTPUTFUNCTION** dcfunction, **ACOUTPUTFUNCTION**, unsigned int, unsigned int, unsigned int ssa)
- void **ProgressiveDcRefine** (unsigned int row, unsigned int col, **DCOUTPUTFUNCTION** dcfunction, **ACOUTPUTFUNCTION**, unsigned int, unsigned int, unsigned int)
- void **ProgressiveAcFirst** (unsigned int row, unsigned int col, **ACOUTPUTFUNCTION** acfunction, unsigned int sss, unsigned int sse, unsigned int ssa)
- void **ProgressiveAcRefine** (unsigned int row, unsigned int col, **ACOUTPUTFUNCTION** acfunction, unsigned int sss, unsigned int sse, unsigned int ssa)
- void **ResetEobRun** ()
- void **PrintEobRun** (**ACOUTPUTFUNCTION** acfunction)
- void **PrintRefineEobRun** (**ACOUTPUTFUNCTION** acfunction, unsigned int sss, unsigned int sse, unsigned int ssa)
- void **SampleYComponent** (**JpegEncoder** &encoder, const **BitmapImage** &, unsigned int maxhf, unsigned int maxvf)
- void **SampleCbComponent** (**JpegEncoder** &encoder, const **BitmapImage** &, unsigned int maxhf, unsigned int maxvf)
- void **SampleCrComponent** (**JpegEncoder** &encoder, const **BitmapImage** &, unsigned int maxhf, unsigned int maxvf)
- unsigned int **GetHorizontalFrequency** () const
- void **SetHorizontalFrequency** (unsigned int)
- unsigned int **GetVerticalFrequency** () const
- void **SetVerticalFrequency** (unsigned int)
- unsigned int **DataUnitRows** () const
- unsigned int **DataUnitCols** () const
- void **ResetDcDifference** ()
- void **SetHuffmanTables** (**JpegEncoderHuffmanTable** &dc, **JpegEncoderHuffmanTable** &ac)
- void **SetQuantizationTable** (**JpegEncoderQuantizationTable** &table)
- void **FreeDynamicStorage** ()

Private Types

- typedef **JPEGSAMPLE** (* **RGBTOYCBCRFUNCTION**)(**JPEGSAMPLE** red, **JPEGSAMPLE** green, **JPEGSAMPLE** blue)

Private Methods

- **JpegEncoderComponent** (const JpegEncoderComponent &)
- JpegEncoderComponent& **operator=** (const JpegEncoderComponent &)
- void **CalculateDuDimensions** (const **BitmapImage** &image, unsigned int maxhf, unsigned int maxvf)
- void **Sample1to1Component** (const **BitmapImage** &, **RGBTOYCBCRFUNCTION**)
- void **SampleNtoNComponent** (const **BitmapImage** &image, **RGBTOYCBCRFUNCTION** function)

Private Attributes

- **JpegEncoder*** **jpeg_encoder**
- unsigned int **du_rows**
- unsigned int **du_cols**
- **JpegEncoderCoefficientBlock*** **dct_coefficients**
- unsigned int **eob_run**
- unsigned int **eob_start_du_row**
- unsigned int **eob_start_du_col**
- unsigned int **eob_start_position**
- unsigned int **v_frequency**
- unsigned int **h_frequency**
- unsigned int **v_period**
- unsigned int **h_period**
- **JpegEncoderHuffmanTable*** **ac_table**
- **JpegEncoderHuffmanTable*** **dc_table**
- **JpegEncoderQuantizationTable*** **quantization_table**
- int **last_dc_value**

Member Typedef Documentation

typedef void(JpegEncoderComponent::* JpegEncoderComponent::ACOUTPUTFUNCTION)(int, int, int)

typedef void(JpegEncoderComponent::* JpegEncoderComponent::COMPONENTPASSFUNCTION)(unsigned int row, unsigned int col, DCOUTPUTFUNCTION, ACOUTPUTFUNCTION, unsigned int sss, unsigned int sse, unsigned int ssa)

typedef void(JpegEncoderComponent::* JpegEncoderComponent::DCOUTPUTFUNCTION)(int, int)

typedef JPEGSAMPLE(* JpegEncoderComponent::RGBTOYCBCRFUNCTION)(JPEGSAMPLE red, JPEGSAMPLE green, JPEGSAMPLE blue) `[private]`

Constructor & Destructor Documentation

JpegEncoderComponent::JpegEncoderComponent ()

JpegEncoderComponent::~JpegEncoderComponent () `[virtual]`

JpegEncoderComponent::JpegEncoderComponent (const JpegEncoderComponent &) `[private]`

Member Function Documentation

void JpegEncoderComponent::CalculateDuDimensions (const BitmapImage & *image*, unsigned int *maxhf*, unsigned int *maxvf*) `[private]`

unsigned int JpegEncoderComponent::DataUnitCols () const `[inline]`

unsigned int JpegEncoderComponent::DataUnitRows () const `[inline]`

void JpegEncoderComponent::EncodeSequential (unsigned int *row*, unsigned int *col*, DCOUTPUTFUNCTION *dcfunction*, ACOUTPUTFUNCTION *acfunction*, unsigned *int*, unsigned *int*, unsigned *int*)

void JpegEncoderComponent::FreeDynamicStorage ()

void JpegEncoderComponent::GatherAcData (int *huffvalue*, int, int)

void JpegEncoderComponent::GatherDcData (int *huffvalue*, int)

unsigned int JpegEncoderComponent::GetHorizontalFrequency () const `[inline]`

unsigned int JpegEncoderComponent::GetVerticalFrequency () const `[inline]`

void JpegEncoderComponent::PrintAcData (int *huffvalue*, int *value*, int *size*)

void JpegEncoderComponent::PrintDcData (int *huffvalue*, int *bits*)

void JpegEncoderComponent::PrintEobRun (ACOUTPUTFUNCTION *acfunction*)

void JpegEncoderComponent::PrintRefineEobRun (ACOUTPUTFUNCTION *acfunction*, unsigned int *sss*, unsigned int *sse*, unsigned int *ssa*)

void JpegEncoderComponent::ProgressiveAcFirst (unsigned int *row*, unsigned int *col*, ACOUTPUTFUNCTION *acfunction*, unsigned int *sss*, unsigned int *sse*, unsigned int *ssa*)

void JpegEncoderComponent::ProgressiveAcRefine (unsigned int *row*, unsigned int *col*, ACOUTPUTFUNCTION *acfunction*, unsigned int *sss*, unsigned int *sse*, unsigned int *ssa*)

void JpegEncoderComponent::ProgressiveDcFirst (unsigned int *row*, unsigned int *col*, DCOUTPUTFUNCTION *dcfunction*, ACOUTPUTFUNCTION, unsigned *int*, unsigned *int*, unsigned int *successiveapproximation*)

void JpegEncoderComponent::ProgressiveDcRefine (unsigned int *row*, unsigned int *col*, DCOUTPUTFUNCTION *dcfunction*, ACOUTPUTFUNCTION, unsigned *int*, unsigned *int*, unsigned int *ssa*)

void JpegEncoderComponent::ResetDcDifference () `[inline]`

void JpegEncoderComponent::ResetEobRun ()

void JpegEncoderComponent::Sample1to1Component (const BitmapImage & *image*, RGBTOYCBCRFUNCTION *function*) `[private]`

void JpegEncoderComponent::SampleCbComponent (JpegEncoder & *encoder*, const BitmapImage & *image*, unsigned int *maxhf*, unsigned int *maxvf*)

void JpegEncoderComponent::SampleCrComponent (JpegEncoder & *encoder*, const BitmapImage & *image*, unsigned int *maxhf*, unsigned int *maxvf*)

void JpegEncoderComponent::SampleNtoNComponent (const BitmapImage & *image*, RGBTOYCBCRFUNCTION *function*) `[private]`

void JpegEncoderComponent::SampleYComponent (JpegEncoder & *encoder*, const BitmapImage & *image*, unsigned int *maxhf*, unsigned int *maxvf*)

void JpegEncoderComponent::SetHorizontalFrequency (unsigned int *value*)

void JpegEncoderComponent::SetHuffmanTables (JpegEncoderHuffmanTable & *dc*, JpegEncoderHuffmanTable & *ac*) `[inline]`

void JpegEncoderComponent::SetQuantizationTable (JpegEncoderQuantizationTable & *table*) `[inline]`

void JpegEncoderComponent::SetVerticalFrequency (unsigned int *value*)

JpegEncoderComponent & JpegEncoderComponent::operator= (const JpegEncoderComponent &) `[private]`

Member Data Documentation

JpegEncoderHuffmanTable * JpegEncoderComponent::ac_table `[private]`

JpegEncoderHuffmanTable * JpegEncoderComponent::dc_table `[private]`

JpegEncoderCoefficientBlock * JpegEncoderComponent::dct_coefficients `[private]`

unsigned int JpegEncoderComponent::du_cols `[private]`

unsigned int JpegEncoderComponent::du_rows `[private]`

unsigned int JpegEncoderComponent::eob_run `[private]`

unsigned int JpegEncoderComponent::eob_start_du_col `[private]`

unsigned int JpegEncoderComponent::eob_start_du_row `[private]`

unsigned int JpegEncoderComponent::eob_start_position `[private]`

unsigned int JpegEncoderComponent::h_frequency `[private]`

unsigned int JpegEncoderComponent::h_period `[private]`

JpegEncoder * JpegEncoderComponent::jpeg_encoder `[private]`

int JpegEncoderComponent::last_dc_value `[private]`

JpegEncoderQuantizationTable * JpegEncoderComponent::quantization_table `[private]`

unsigned int JpegEncoderComponent::v_frequency `[private]`

unsigned int JpegEncoderComponent::v_period `[private]`

The documentation for this class was generated from the following files:

- **jpencomp.h**
- **jpencomp.cpp**

JpegEncoderDataUnit Class Reference

`#include <jpendu.h>`

Public Types

- typedef **JPEGSAMPLE DATAUNITVALUE**

Public Methods

- **JpegEncoderDataUnit** ()
- **~JpegEncoderDataUnit** ()
- **DATAUNITVALUE* operator[]** (unsigned int)
- **DATAUNITVALUE**& **ZigZagInput** (unsigned int)
- void **ForwardDct** (**JpegEncoderQuantizationTable** &, **JpegEncoderCoefficientBlock** &)

Private Methods

- **JpegEncoderDataUnit** (const JpegEncoderDataUnit &)
- JpegEncoderDataUnit& **operator=** (const JpegEncoderDataUnit &)

Private Attributes

- **DATAUNITVALUE data** [**JpegSampleWidth**][**JpegSampleWidth**]

Member Typedef Documentation

typedef JPEGSAMPLE JpegEncoderDataUnit::DATAUNITVALUE

Constructor & Destructor Documentation

JpegEncoderDataUnit::JpegEncoderDataUnit () `[inline]`

JpegEncoderDataUnit::~JpegEncoderDataUnit () `[inline]`

JpegEncoderDataUnit::JpegEncoderDataUnit (const JpegEncoderDataUnit &) `[private]`

Member Function Documentation

void JpegEncoderDataUnit::ForwardDct (JpegEncoderQuantizationTable & *qt*, JpegEncoderCoefficientBlock & *output*)

JpegEncoderDataUnit::DATAUNITVALUE & JpegEncoderDataUnit::ZigZagInput (unsigned int *ii*) `[inline]`

JpegEncoderDataUnit & JpegEncoderDataUnit::operator= (const JpegEncoderDataUnit &) `[private]`

JpegEncoderDataUnit::DATAUNITVALUE * JpegEncoderDataUnit::operator[] (unsigned int *ii*) `[inline]`

Member Data Documentation

DATAUNITVALUE JpegEncoderDataUnit::data `[private]`

The documentation for this class was generated from the following files:

- **jpendu.h**
- **jpendu.cpp**

JpegEncoderHuffmanTable Class Reference

`#include <jpenhuff.h>`

Public Methods

- **JpegEncoderHuffmanTable** ()
- virtual **~JpegEncoderHuffmanTable** ()
- void **Reset** ()
- void **IncrementFrequency** (unsigned int value)
- void **BuildTable** ()
- void **Encode** (unsigned int value, **UBYTE2** &code, **UBYTE1** &size) const
- unsigned int **OutputSize** () const
- void **PrintTable** (**JpegEncoder** &encoder) const

Private Methods

- **JpegEncoderHuffmanTable** (const JpegEncoderHuffmanTable &)
- JpegEncoderHuffmanTable& **operator=** (const JpegEncoderHuffmanTable &)
- void **MakeTable** ()

Private Attributes

- unsigned int **frequencies** [**JpegMaxNumberOfHuffmanCodes**+1]
- **UBYTE1 huff_bits** [2 ***JpegMaxHuffmanCodeLength**]
- **UBYTE1 huff_values** [**JpegMaxNumberOfHuffmanCodes**]
- **UBYTE1 ehufsi** [**JpegMaxNumberOfHuffmanCodes**+1]
- **UBYTE2 ehufco** [**JpegMaxNumberOfHuffmanCodes**+1]
- bool **sizes_found**

Constructor & Destructor Documentation

JpegEncoderHuffmanTable::JpegEncoderHuffmanTable ()

JpegEncoderHuffmanTable::~JpegEncoderHuffmanTable () `[inline, virtual]`

JpegEncoderHuffmanTable::JpegEncoderHuffmanTable (const JpegEncoderHuffmanTable &) `[private]`

Member Function Documentation

void JpegEncoderHuffmanTable::BuildTable ()

void JpegEncoderHuffmanTable::Encode (unsigned int *value*, UBYTE2 & *code*, UBYTE1 & *size*) const

void JpegEncoderHuffmanTable::IncrementFrequency (unsigned int *value*)

void JpegEncoderHuffmanTable::MakeTable () `[private]`

unsigned int JpegEncoderHuffmanTable::OutputSize () const

void JpegEncoderHuffmanTable::PrintTable (JpegEncoder & *encoder*) const

void JpegEncoderHuffmanTable::Reset ()

JpegEncoderHuffmanTable & JpegEncoderHuffmanTable::operator= (const JpegEncoderHuffmanTable &) `[private]`

Member Data Documentation

UBYTE2 JpegEncoderHuffmanTable::ehufco `[private]`

UBYTE1 JpegEncoderHuffmanTable::ehufsi `[private]`

unsigned int JpegEncoderHuffmanTable::frequencies `[private]`

UBYTE1 JpegEncoderHuffmanTable::huff_bits `[private]`

UBYTE1 JpegEncoderHuffmanTable::huff_values `[private]`

bool JpegEncoderHuffmanTable::sizes_found `[private]`

The documentation for this class was generated from the following files:

- **jpenhuff.h**
- **jpenhuff.cpp**

JpegEncoderQuantizationTable Class Reference

`#include <jpenquan.h>`

Public Methods

- **JpegEncoderQuantizationTable** ()
- **~JpegEncoderQuantizationTable** ()
- **UBYTE2 operator[]** (unsigned int index) const
- **UBYTE2**& **operator[]** (unsigned int index)
- void **BuildScaledTables** ()

Private Methods

- **JpegEncoderQuantizationTable** (const JpegEncoderQuantizationTable &)
- JpegEncoderQuantizationTable& **operator=** (const JpegEncoderQuantizationTable &)

Private Attributes

- **UBYTE2 data_values** [**JpegSampleSize**]
- double **float_scaling** [**JpegSampleWidth**][**JpegSampleWidth**]

Friends

- class **JpegEncoderDataUnit**

Constructor & Destructor Documentation

JpegEncoderQuantizationTable::JpegEncoderQuantizationTable ()

JpegEncoderQuantizationTable::~JpegEncoderQuantizationTable () `[inline]`

JpegEncoderQuantizationTable::JpegEncoderQuantizationTable (const JpegEncoderQuantizationTable &) `[private]`

Member Function Documentation

void JpegEncoderQuantizationTable::BuildScaledTables ()

JpegEncoderQuantizationTable & JpegEncoderQuantizationTable::operator= (const JpegEncoderQuantizationTable &) `[private]`

UBYTE2 & JpegEncoderQuantizationTable::operator[] (unsigned int *index*) `[inline]`

UBYTE2 JpegEncoderQuantizationTable::operator[] (unsigned int *index*) const `[inline]`

Friends And Related Function Documentation

class JpegEncoderDataUnit `[friend]`

Member Data Documentation

UBYTE2 JpegEncoderQuantizationTable::data_values `[private]`

double JpegEncoderQuantizationTable::float_scaling `[private]`

The documentation for this class was generated from the following files:

- **jpenquan.h**
- **jpenquan.cpp**

JpegHuffmanDecoder Class Reference

`#include <jpdehuff.h>`

Public Methods

- **JpegHuffmanDecoder** ()
- virtual **~JpegHuffmanDecoder** ()
- bool **Defined** () const
- unsigned int **ReadTable** (**JpegDecoder** &)
- int **Decode** (**JpegDecoder** &)
- void **Dump** (ostream &strm) const

Private Methods

- **JpegHuffmanDecoder** (const JpegHuffmanDecoder &)
- JpegHuffmanDecoder& **operator=** (const JpegHuffmanDecoder &)
- void **MakeTable** (**UBYTE1** huffbits[**JpegMaxHuffmanCodeLength**])

Private Attributes

- int **maxcode** [**JpegMaxHuffmanCodeLength**]
- int **mincode** [**JpegMaxHuffmanCodeLength**]
- **UBYTE1 valptr** [**JpegMaxHuffmanCodeLength**]
- **UBYTE1 huff_values** [**JpegMaxNumberOfHuffmanCodes**]
- bool **table_defined**

Constructor & Destructor Documentation

JpegHuffmanDecoder::JpegHuffmanDecoder ()

JpegHuffmanDecoder::~JpegHuffmanDecoder () `[inline, virtual]`

JpegHuffmanDecoder::JpegHuffmanDecoder (const JpegHuffmanDecoder &) `[private]`

Member Function Documentation

int JpegHuffmanDecoder::Decode (JpegDecoder & *decoder*)

bool JpegHuffmanDecoder::Defined () const

void JpegHuffmanDecoder::Dump (ostream & *strm*) const

void JpegHuffmanDecoder::MakeTable (UBYTE1 *huffbits*[JpegMaxHuffmanCodeLength]) `[private]`

unsigned int JpegHuffmanDecoder::ReadTable (JpegDecoder & *decoder*)

JpegHuffmanDecoder & JpegHuffmanDecoder::operator= (const JpegHuffmanDecoder &) `[private]`

Member Data Documentation

UBYTE1 JpegHuffmanDecoder::huff_values `[private]`

int JpegHuffmanDecoder::maxcode `[private]`

int JpegHuffmanDecoder::mincode `[private]`

bool JpegHuffmanDecoder::table_defined `[private]`

UBYTE1 JpegHuffmanDecoder::valptr `[private]`

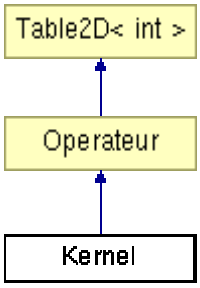
The documentation for this class was generated from the following files:

- **jpdehuff.h**
- **jpdehuff.cpp**

Kernel Class Reference

`#include <Kernel.hpp>`

Inheritance diagram for Kernel:



Public Types

- enum **EtatPixel** { **P_OFF** = 0, **P_ON** = 1, **P_X** = 2 }
- enum **KernelType** { **T**, **CROIX**, **CARRE**, **NbKernelType** }

Public Methods

- **Kernel** (**KernelType** _type=T)
- **Kernel** (**KernelType** _type, unsigned int _d)
- **Kernel** (**KernelType** _type, unsigned int _l, unsigned int _c)
- void **defType** (**KernelType** _type)
- void **defType** (**KernelType** _type, unsigned int _d)
- void **defType** (**KernelType** _type, int _l, int _c)
- **KernelType Type** () const
- void **defT** (unsigned int _l=2, unsigned int _c=3)
- void **defCroix** (unsigned int _a=3)
- void **defCarre** (unsigned int _a=3)

Protected Attributes

- **KernelType kerneltype**

Member Enumeration Documentation

enum Kernel::EtatPixel

Enumeration values:

P_OFF
P_ON
P_X

enum Kernel::KernelType

Enumeration values:

T
CROIX
CARRE
NbKernelType

Constructor & Destructor Documentation

Kernel::Kernel (KernelType _*type* = T)

Kernel::Kernel (KernelType _*type*, unsigned int _*d*)

Kernel::Kernel (Kernel::KernelType _*type*, unsigned int _*l*, unsigned int _*c*)

Member Function Documentation

KernelType Kernel::Type () const `[inline]`

Reimplemented from **Operateur** (*p.300*).

void Kernel::defCarre (unsigned int _*a* = 3)

void Kernel::defCroix (unsigned int _*a* = 3)

void Kernel::defT (unsigned int _*l* = 2, unsigned int _*c* = 3)

void Kernel::defType (Kernel::KernelType _*type*, int _*l*, int _*c*)

void Kernel::defType (KernelType _*type*, unsigned int _*d*)

void Kernel::defType (KernelType _*type*)

Member Data Documentation

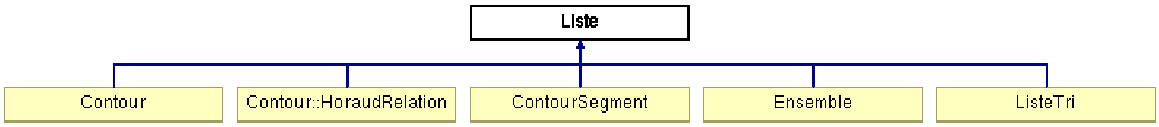
KernelType Kernel::kerneltype `[protected]`

The documentation for this class was generated from the following files:

- **Kernel.hpp**
- **Kernel.cpp**

Liste Class Reference

Inheritance diagram for Liste:



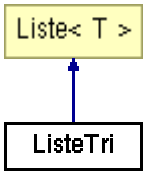
The documentation for this class was generated from the following file:

- **Contour.hpp**

ListeTri Class Template Reference

`#include <ListeTri.hpp>`

Inheritance diagram for ListeTri:



Public Methods

- void **Trier** ()
- void **Inserer** (const T &_e)

template<class T> class ListeTri

Member Function Documentation

template<class T> void ListeTri< T >::Inserer (const T & _*e*)

template<class T> void ListeTri< T >::Trier ()

The documentation for this class was generated from the following file:

- **ListeTri.hpp**

GifDecoder::LZ77String Struct Reference

Public Attributes

- unsigned long **offset**
- unsigned long **length**

Member Data Documentation

unsigned long GifDecoder::LZ77String::length

unsigned long GifDecoder::LZ77String::offset

The documentation for this struct was generated from the following file:

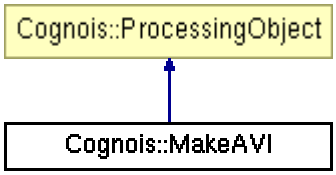
- **gifdecod.h**

Cognois::MakeAVI Class Reference

Export grabbed images to an AVI movie.

`#include <MakeAVI.h>`

Inheritance diagram for Cognois::MakeAVI:



Public Methods

- **MakeAVI** (unsigned long inImageLimit, const char *inBaseFileName)
  Constructor with a maximum number of images to write and a filename.

- **~MakeAVI** ()
  Destructor.

- void **Process** (MIL_ID &inMilImage, unsigned long &inFrameID, unsigned long &inTimeStamp)
  Export an image to a file.

Private Methods

- void **SaveData** ()
  Save images and sync data to a file.

Private Attributes

- unsigned long **mImageLimit**
  Maximum number of images.

- char* **mFileName**
  Base filename for the AVI and text info.

- char* **mAviFileName**
- **std::vector**<unsigned long> **mFrameIds**
  Frame IDs of all the images in the AVI.

- **std::vector**<unsigned long> **mTimeStamps**
  Time stamps of all the images.

- **std::vector**<char *> **mImageContents**
  Contents of the images.

- unsigned long **mImageCounter**
  Current number of images.

- long **mSizeX**
- long **mSizeY**
- long **mSizeBand**
  Image *(p.184) dimensions.*

Detailed Description

Export grabbed images to an AVI movie.

Constructor & Destructor Documentation

Cognois::MakeAVI::MakeAVI (unsigned long *inImageLimit*, const char * *inBaseFileName*)

Constructor with a maximum number of images to write and a filename.

Parameters:

inImageLimit is the maximum number of images to save. (If 0, write directly on disk).
inBaseFileName is the base file name. Files will be named [_fileNameBase].bmp / .txt

Cognois::MakeAVI::~MakeAVI ()

Destructor.

Member Function Documentation

void Cognois::MakeAVI::Process (MIL_ID & *inMilImage*, unsigned long & *inFrameID*, unsigned long & *inTimeStamp*) `[virtual]`

Export an image to a file.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

void Cognois::MakeAVI::SaveData () `[private]`

Save images and sync data to a file.

Member Data Documentation

char * Cognois::MakeAVI::mAviFileName [private]

char * Cognois::MakeAVI::mFileName [private]

Base filename for the AVI and text info.

std::vector< unsigned long > Cognois::MakeAVI::mFrameIds<unsigned long> [private]

Frame IDs of all the images in the AVI.

std::vector< char *> Cognois::MakeAVI::mImageContents<char *> [private]

Contents of the images.

unsigned long Cognois::MakeAVI::mImageCounter [private]

Current number of images.

unsigned long Cognois::MakeAVI::mImageLimit [private]

Maximum number of images.

long Cognois::MakeAVI::mSizeBand [private]

Image (*p.184*) dimensions.

long Cognois::MakeAVI::mSizeX [private]

long Cognois::MakeAVI::mSizeY [private]

std::vector< unsigned long > Cognois::MakeAVI::mTimeStamps<unsigned long> [private]

Time stamps of all the images.

The documentation for this class was generated from the following files:

- **MakeAVI.h**
- **MakeAVI.cpp**

Matrice Class Reference

Fonctions de base pour le calcul matriciel.

```
#include <Matrice.h>
```

Public Types

- enum **Type** { **ZEROS**, **UNS**, **IDENTITE** }

Public Methods

- **Matrice** ()
  *Matrice 0*0.*

- **Matrice** (const Matrice &)
  Constructeur copieur.

- **Matrice** (const char *)
  Matlab-style constructor.

- **Matrice** (unsigned int i, unsigned int j, double v=0, int rs=0, int cs=0)
- **Matrice** (unsigned int dim, **Type** t=ZEROS, int rs=0, int cs=0)
- **~Matrice** ()
  Destructeur.

- void **Resize** (unsigned int i, unsigned int j, double v=0, int rs=0, int cs=0)
- void **Resize** (unsigned int dim, **Type** t=ZEROS, int rs=0, int cs=0)
- double& **operator()** (int i, int j)
  Acceder a l'element i,j.

- double const& **operator()** (int i, int j) const
- Matrice& **operator=** (const Matrice &)
  Affecter.

- Matrice& **operator=** (const char *)
  Affecter une string (matlab style).

- Matrice **operator~** () const
  Transposer.

- Matrice **operator-** () const
  *Obtenir -1*Matrice.*

- Matrice **operator+** (const Matrice &) const
  Additionner Matrice.

- Matrice& **operator+=** (const Matrice &)
  Additionner/Affecter Matrice.

- Matrice **operator-** (const Matrice &) const
  Soustraire Matrice.

- Matrice& **operator-=** (const Matrice &)
  Soustraire/Affecter Matrice.

- Matrice **operator *** (const Matrice &) const
  Multiplier Matrice.

- Matrice& **operator *=** (const Matrice &)
  Multiplier/Affecter Matrice.

- Matrice **operator+** (double) const
  Additionner scalaire a chaque element.

- Matrice& **operator+=** (double)
  Additionner/Affecter scal. a chaque element.

- Matrice **operator-** (double) const
  Soustraire scalaire a chaque element.

- Matrice& **operator-=** (double)
  Soustraire/Affecter scal. a chaque element.

- Matrice **operator *** (double) const
  Multiplier scalaire.

- Matrice& **operator *=** (double)
  Multiplier/Affecter scalaire.

- Matrice **operator/** (double) const
  Diviser scalaire.

- Matrice& **operator/=** (double)
  Diviser/Affecter scalaire.

- Matrice **Cross** (const Matrice &m) const
  Calculation of cross product.

- double **Norm** () const

Calculats the norm of a vector (Matrix nx1).

- Matrice **AugmenterMatrice** ()
- Matrice **operator|** (const Matrice &) const
  Concatener deux matrices.

- Matrice& **operator|=** (const Matrice &)
  Concatener/Affecter deux matrices.

- Matrice **operator/** (const Matrice &) const
  *Resoudre x=A**(this** = A).*

- Matrice& **operator/=** (const Matrice &)
  *Resoudre x=A**et** ecrire resultat dans A.*

- int **operator==** (const Matrice &) const
  Comparer deux matrices.

- int **operator!=** (const Matrice &) const
- double **Det** () const
  Calculer determinant.

- Matrice **Inv** () const
  Retourner inverse.

- Matrice **GenInv** () const
  Retourner inverse generalisee.

- void **SVD** (Matrice &outU, Matrice &outW, Matrice &outV) const
  Retourner UWV' == A.

- int **MinLigne** () const
  Premiere ligne.

- int **MaxLigne** () const
  Derniere ligne.

- int **MinCol** () const
  Premiere colonne.

- int **MaxCol** () const
  Derniere colonne.

- int **NbLignes** () const

Nombre de lignes.

- int **NbCols** () const
  Nombre de colonnes.

- int **EstVide** () const
  Matrice existante?

- void **Afficher** () const
  Afficher a la console.

- void **AfficherFichier** (char *FileName) const
  Afficher a la console.

- char* **MatlabOutput** () const
  Retourner une string (matlab-style).

- Matrice **SubMatrix** (int i0, int i1, int j0, int j1) const
  this(i0:i1,j0:j1).

- double **Max** () const
  Maximum value in the matrix.

- double **Min** () const
  Minimum value in the matrix.

Public Attributes

- double** **el**
  *Acces direct aux elements de **dmat** (p.89).*

Private Attributes

- void* **mat**
  *Pointeur sur une structure **dmat** (p.89), le contenu de la matrice.*

Detailed Description

Fonctions de base pour le calcul matriciel.

Member Enumeration Documentation

enum Matrice::Type

Enumeration values:

ZEROS
UNS
IDENTITE

Constructor & Destructor Documentation

Matrice::Matrice ()

Matrice 0*0.

Matrice::Matrice (const Matrice & *m*)

Constructeur copieur.

Matrice::Matrice (const char * *inMatrix*)

Matlab-style constructor.

Matrice::Matrice (unsigned int *i*, unsigned int *j*, double *v* = 0, int *rs* = 0, int *cs* = 0)

Matrice::Matrice (unsigned int *dim*, Type *t* = ZEROS, int *rs* = 0, int *cs* = 0)

Matrice::~Matrice ()

Destructeur.

Member Function Documentation

void Matrice::Afficher () const

Afficher a la console.

void Matrice::AfficherFichier (char * *FileName*) const

Afficher a la console.

Matrice Matrice::AugmenterMatrice ()

Matrice Matrice::Cross (const Matrice & *m*) const

Calculation of cross product.

double Matrice::Det () const

Calculer determinant.

int Matrice::EstVide () const

Matrice existante?

Matrice Matrice::GenInv () const

Retourner inverse generalisee.

Matrice Matrice::Inv () const

Retourner inverse.

char * Matrice::MatlabOutput () const

Retourner une string (matlab-style).

double Matrice::Max () const

Maximum value in the matrix.

int Matrice::MaxCol () const

Derniere colonne.

int Matrice::MaxLigne () const

Derniere ligne.

double Matrice::Min () const

Minimum value in the matrix.

int Matrice::MinCol () const

Premiere colonne.

int Matrice::MinLigne () const

Premiere ligne.

int Matrice::NbCols () const

Nombre de colonnes.

int Matrice::NbLignes () const

Nombre de lignes.

double Matrice::Norm () const

Calculats the norm of a vector (Matrix nx1).

void Matrice::Resize (unsigned int *dim*, Type *t* = ZEROS, int *rs* = 0, int *cs* = 0)

void Matrice::Resize (unsigned int *i*, unsigned int *j*, double *v* = 0, int *rs* = 0, int *cs* = 0)

void Matrice::SVD (Matrice & *U*, Matrice & *W*, Matrice & *V*) const

Retourner UWV' == A.

Matrice Matrice::SubMatrix (int *i0*, int *i1*, int *j0*, int *j1*) const

this(i0:i1,j0:j1).

Matrice Matrice::operator * (double *sc*) const

Multiplier scalaire.

Matrice Matrice::operator * (const Matrice & *m*) const

Multiplier Matrice.

Matrice & Matrice::operator *= (double *sc*)

Multiplier/Affecter scalaire.

Matrice & Matrice::operator *= (const Matrice & *m*)

Multiplier/Affecter Matrice.

int Matrice::operator != (const Matrice & *m*) const

double const & Matrice::operator() (int *i*, int *j*) const

double & Matrice::operator() (int *i*, int *j*)

Acceder a l'element i,j.

Matrice Matrice::operator+ (double *sc*) const

Additionner scalaire a chaque element.

Matrice Matrice::operator+ (const Matrice & *m*) const

Additionner Matrice.

Matrice & Matrice::operator+= (double *sc*)

Additionner/Affecter scal. a chaque element.

Matrice & Matrice::operator+= (const Matrice & *m*)

Additionner/Affecter Matrice.

Matrice Matrice::operator- (double *sc*) const

Soustraire scalaire a chaque element.

Matrice Matrice::operator- (const Matrice & *m*) const

Soustraire Matrice.

Matrice Matrice::operator- () const

Obtenir -1*Matrice.

Matrice & Matrice::operator-= (double *sc*)

Soustraire/Affecter scal. a chaque element.

Matrice & Matrice::operator-= (const Matrice & *m*)

Soustraire/Affecter Matrice.

Matrice Matrice::operator/ (const Matrice & *b*) const

Resoudre x=A**(this** = A).

Matrice Matrice::operator/ (double *sc*) const

Diviser scalaire.

Matrice & Matrice::operator/= (const Matrice & *m*)

Resoudre x=A**et** ecrire resultat dans A.

Matrice & Matrice::operator/= (double *sc*)

Diviser/Affecter scalaire.

Matrice & Matrice::operator= (const char * *inMatrix*)

Affecter une string (matlab style).

Matrice & Matrice::operator= (const Matrice & *m*)

Affecter.

int Matrice::operator== (const Matrice & *m*) const

Comparer deux matrices.

Matrice Matrice::operator| (const Matrice & *m*) const

Concatener deux matrices.

Matrice & Matrice::operator|= (const Matrice & *m*)

Concatener/Affecter deux matrices.

Matrice Matrice::operator~ () const

Transposer.

Member Data Documentation

double ** Matrice::el

Acces direct aux elements de **dmat** (*p.89*).

void * Matrice::mat `[private]`

Pointeur sur une structure **dmat** (*p.89*), le contenu de la matrice.

The documentation for this class was generated from the following files:

- **Matrice.h**
- **Matrice.cpp**

Cognois::MilGarbageCollector Class Reference

```
#include <MilGarbageCollector.h>
```

Public Methods

- **~MilGarbageCollector** ()
- MIL_ID **MbufAllocColor** (MIL_ID _systemId, long _sizeBand, long _sizeX, long _sizeY, long _type, long _attribute, MIL_ID *_bufIdPtr)
  Allocate a color data buffer.

- MIL_ID **MbufAlloc2d** (MIL_ID _systemId, long _sizeX, long _sizeY, long _type, long _attribute, MIL_ID *_bufIdPtr)
  Allocate a 2D data buffer.

- MIL_ID **MbufChildColor** (MIL_ID _parentBufId, long _band, MIL_ID *_bufIdPtr)
  Allocate a color-band child data buffer within a color parent buffer.

- MIL_ID **MbufChild2d** (MIL_ID _parentBufId, long _offX, long _offY, long _sizeX, long _sizeY, MIL_ID *_bufIdPtr)
  Allocate a child buffer within a specific region of a parent buffer.

- void **MbufFree** (MIL_ID _bufId)
  Free a data buffer.

- void **FreeAllBuffers** ()
  Free all data buffers.

Private Methods

- void **Allocate** (MIL_ID _bufId)
  Add a buffer to the allocated MIL data buffers list.

Private Attributes

- **std::vector**<MIL_ID> **allocatedBuffers**
  List of all allocated MIL data buffers.

Constructor & Destructor Documentation

Cognois::MilGarbageCollector::~MilGarbageCollector () `[inline]`

Member Function Documentation

void Cognois::MilGarbageCollector::Allocate (MIL_ID *_bufId*) `[inline, private]`

Add a buffer to the allocated MIL data buffers list.

Parameters:

_bufId is the buffer identifier to allocate.

void Cognois::MilGarbageCollector::FreeAllBuffers ()

Free all data buffers.

MIL_ID Cognois::MilGarbageCollector::MbufAlloc2d (MIL_ID *_systemId*, long *_sizeX*, long *_sizeY*, long *_type*, long *_attribute*, MIL_ID * *_bufIdPtr*)

Allocate a 2D data buffer.

Parameters:

_systemId is a system identifier
_sizeX is the desired X dimension
_sizeY is the desired Y dimension
_type is the desired data depth and data type
_attribute is the buffer attributes
_bufIdPtr is a storage location for buffer identifier

MIL_ID Cognois::MilGarbageCollector::MbufAllocColor (MIL_ID *_systemId*, long *_sizeBand*, long *_sizeX*, long *_sizeY*, long *_type*, long *_attribute*, MIL_ID * *_bufIdPtr*)

Allocate a color data buffer.

Parameters:

_systemId is a system identifier
_sizeBand is the number of color bands
_sizeX is the desired X dimension
_sizeY is the desired Y dimension
_type is the desired data depth and data type
_attribute is the buffer attributes
_bufIdPtr is a storage location for buffer identifier

MIL_ID Cognois::MilGarbageCollector::MbufChild2d (MIL_ID *_parentBufId*, long *_offX*, long *_offY*, long *_sizeX*, long *_sizeY*, MIL_ID * *_bufIdPtr*)

Allocate a child buffer within a specific region of a parent buffer.

Parameters:

_parentBufId is the parent buffer identifier.
_offX is the X pixel offset relative to the parent buffer
_offY is the Y pixel offset relative to the parent buffer
_sizeX is the child buffer width
_sizeY is the child buffer height
_bufIdPtr is the storage location for child buffer identifier.

MIL_ID Cognois::MilGarbageCollector::MbufChildColor (MIL_ID *_parentBufId*, long *_band*, MIL_ID * *_bufIdPtr*)

Allocate a color-band child data buffer within a color parent buffer.

Parameters:

_parentBufId is the parent buffer identifier.
_band is the index of a color band.
_bufIdPtr is the storage location for child buffer identifier.

void Cognois::MilGarbageCollector::MbufFree (MIL_ID *_bufId*)

Free a data buffer.

Parameters:

_bufId is the buffer identifier to deallocate.

Member Data Documentation

std::vector< MIL_ID > Cognois::MilGarbageCollector::allocatedBuffers `[private]`

List of all allocated MIL data buffers.

The documentation for this class was generated from the following files:

- **MilGarbageCollector.h**
- **MilGarbageCollector.cpp**

MilGarbageCollector Class Reference

A garbage collector for MIL image buffers.

```
#include <MilGarbageCollector.h>
```

Detailed Description

A garbage collector for MIL image buffers.

When allocated with a MilGarbageCollector, MIL data buffers are automatically freed upon reallocation or when the MilGarbageCollector is destroyed.

The documentation for this class was generated from the following file:

- **MilGarbageCollector.h**

MILSys Struct Reference

Mil system data.

```
#include <MILSys.h>
```

Public Attributes

- MIL_ID **MilApplication**
- MIL_ID **MilSystem**
- MIL_ID **MilDisplay**
- MIL_ID **MilDigitizer**
- long **ImageDimX**
- long **ImageDimY**
- long **NbColors**
- double **ImageScale**

Detailed Description

Mil system data.

Member Data Documentation

long MILSys::ImageDimX

long MILSys::ImageDimY

double MILSys::ImageScale

MIL_ID MILSys::MilApplication

MIL_ID MILSys::MilDigitizer

MIL_ID MILSys::MilDisplay

MIL_ID MILSys::MilSystem

long MILSys::NbColors

The documentation for this struct was generated from the following file:

- **MILSys.h**

MINPACK Class Reference

Levenberg-Marquardt algorithm to minimize the sum of the squares.

`#include <MINPACK.h>`

Public Methods

- long **lmdif** (void(*fcn)(long, long, double *, double *, double *), long nbEquations, long nbInconnues, double *x, double *constantes=0, long maxfev=1000)
  Minimize the sum of the squares of an user-supplied function with user-supplied data.

Static Private Methods

- int **evaluation_erreur** (long *m_ptr, long *n_ptr, double *params, double *err, long *iflag)
  Error evaluation function.

Static Private Attributes

- void (* **user_fcn**)(long, long, double *, double *, double *)
  Pointer to the user function.

- double* **user_constants** = 0
  Pointer to the user constants.

Detailed Description

Levenberg-Marquardt algorithm to minimize the sum of the squares.

Minimization of the sum of the squares of m nonlinear functions in n variables by a modification of the levenberg-marquardt algorithm.

Member Function Documentation

int MINPACK::evaluation_erreur (long * *m_ptr*, long * *n_ptr*, double * *params*, double * *err*, long * *iflag*) `[static, private]`

Error evaluation function.

long MINPACK::lmdif (void(* *fcn*)(long, long, double *, double *, double *), long *nbEquations*, long *nbInconnues*, double * *x*, double * *constantes* = 0, long *maxfev* = 1000)

Minimize the sum of the squares of an user-supplied function with user-supplied data.

Member Data Documentation

double * MINPACK::user_constants = 0 `[static, private]`

Pointer to the user constants.

void(* MINPACK::user_fcn)(long, long, double *, double *, double *)=0 `[static, private]`

Pointer to the user function.

The documentation for this class was generated from the following files:

- **MINPACK.h**
- **MINPACK.cpp**

Multitype Union Reference

```
#include <F2C.H>
```

Public Attributes

- **shortint h**
- **integer i**
- **real r**
- **doublereal d**
- **complex c**
- **doublecomplex z**

Member Data Documentation

complex Multitype::c

doublereal Multitype::d

shortint Multitype::h

integer Multitype::i

real Multitype::r

doublecomplex Multitype::z

The documentation for this union was generated from the following file:

- **F2C.H**

Namelist Struct Reference

```
#include <F2C.H>
```

Public Attributes

- char* **name**
- **Vardesc**** **vars**
- int **nvars**

Member Data Documentation

char * Namelist::name

int Namelist::nvars

Vardesc ** Namelist::vars

The documentation for this struct was generated from the following file:

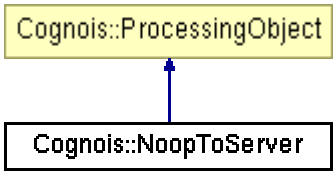
- **F2C.H**

Cognois::NoopToServer Class Reference

Perform a sequence of nothing, but keep the server informed of the current frame ID.

`#include <NoopToServer.h>`

Inheritance diagram for Cognois::NoopToServer:



Public Methods

- **NoopToServer** (**FrameSetClient** *inCognoisServer)
- void **Process** (MIL_ID &inMilImage, unsigned long &inFrameID, unsigned long &inTimeStamp)
  Process an image with know frameID and timeStamp.

Private Attributes

- **FrameSetClient* mCognoisServer**

Detailed Description

Perform a sequence of nothing, but keep the server informed of the current frame ID.

Constructor & Destructor Documentation

Cognois::NoopToServer::NoopToServer (FrameSetClient * *inCognoisServer*) `[inline]`

Member Function Documentation

void Cognois::NoopToServer::Process (MIL_ID & *inMilImage*, unsigned long & *inFrameID*, unsigned long & *inTimeStamp*) `[inline, virtual]`

Process an image with know frameID and timeStamp.

Parameters:

inMilImage is a MIL image identifier.
inFrameID is the frame identifier associated with the image.
inTimeStamp is the time, in milliseconds, associated with the image.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

Member Data Documentation

FrameSetClient * Cognois::NoopToServer::mCognoisServer `[private]`

The documentation for this class was generated from the following file:

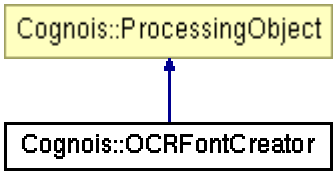
- **NoopToServer.h**

Cognois::OCRFontCreator Class Reference

Optical character recognition of the 5 digits of the HP 5214L Preset Counter.

`#include <OCRFontCreator.h>`

Inheritance diagram for Cognois::OCRFontCreator:



Public Methods

- **OCRFontCreator** (char *inFontFilename, **MILSys** &inMilSys, **OnlineProcessing** *inOnlineProcessing)
  Constructor with interactive calibration of the font.

- **~OCRFontCreator** ()
- void **Process** (MIL_ID &inMilImage, unsigned long &inFrameID, unsigned long &inTimeStamp)
  Find a calibration pattern in an image.

Private Methods

- void **allocateFont** (MIL_ID &inMilImage)
- void **getCharacterSamplesMean** ()

Private Attributes

- **MILSys*** **mMilSys**
- MIL_ID **mFont**
- MIL_ID **mCounterImage**
- MIL_ID **mGrayImage**
- **std::vector**<**std::vector**<MIL_ID> > **mCharacterSamples**
- **std::vector**<char> **mCharacters**
- bool **mFontIsAllocated**
- **std::vector**<long> **mTopX**
- **std::vector**<long> **mTopY**
- **std::vector**<long> **mBotX**
- **std::vector**<long> **mBotY**
  Bounding box of the digits.

- **OnlineProcessing*** **mOnlineProcessing**
- std::string **mFontFilename**

Detailed Description

Optical character recognition of the 5 digits of the HP 5214L Preset Counter.

Constructor & Destructor Documentation

Cognois::OCRFontCreator::OCRFontCreator (char * *inFontFilename*, MILSys & *inMilSys*, OnlineProcessing * *inOnlineProcessing*)

Constructor with interactive calibration of the font.

Cognois::OCRFontCreator::~OCRFontCreator ()

Member Function Documentation

void Cognois::OCRFontCreator::Process (MIL_ID & *inMilImage*, unsigned long & *inFrameID*, unsigned long & *inTimeStamp*) `[virtual]`

Find a calibration pattern in an image.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

void Cognois::OCRFontCreator::allocateFont (MIL_ID & *inMilImage*) `[private]`

void Cognois::OCRFontCreator::getCharacterSamplesMean () `[private]`

Member Data Documentation

std::vector< long > Cognois::OCRFontCreator::mBotX `[private]`

std::vector< long > Cognois::OCRFontCreator::mBotY `[private]`

Bounding box of the digits.

std::vector< std::vector< MIL_ID > > Cognois::OCRFontCreator::mCharacterSamples<std::vector<MIL_ID> > `[private]`

std::vector< char > Cognois::OCRFontCreator::mCharacters<char> `[private]`

MIL_ID Cognois::OCRFontCreator::mCounterImage `[private]`

MIL_ID Cognois::OCRFontCreator::mFont `[private]`

std::string Cognois::OCRFontCreator::mFontFilename `[private]`

bool Cognois::OCRFontCreator::mFontIsAllocated `[private]`

MIL_ID Cognois::OCRFontCreator::mGrayImage `[private]`

MILSys * Cognois::OCRFontCreator::mMilSys `[private]`

OnlineProcessing * Cognois::OCRFontCreator::mOnlineProcessing `[private]`

std::vector< long > Cognois::OCRFontCreator::mTopX<long> `[private]`

std::vector< long > Cognois::OCRFontCreator::mTopY `[private]`

The documentation for this class was generated from the following files:

- **OCRFontCreator.h**
- **OCRFontCreator.cpp**

Cognois::OfflineProcessing Class Reference

Processing of images from AVI files.

`#include <OfflineProcessing.h>`

Public Methods

- **OfflineProcessing** (const char *inBaseFilename)
  Destructor.

- **~OfflineProcessing** ()
- void **SetProcessor** (**ProcessingObject** *inNewProcessor)
  Set the processing to apply to grabbed images.

- void **LoopMe** ()
  Start, wait for the keyboard, then end the grabbing/processing.

- void **Begin** ()
  Start both the grabbing and the processing threads.

- void **End** ()
  Stop both the grabbing and the processing threads.

- void **ThreadAcquisition** ()
  Grabbing thread.

- ::MILSys& **MILSys** ()
  Returns the MIL identifiers.

Private Methods

- void **ResetEvents** ()
  Set all events initial value.

Private Attributes

- ::**MILSys mMilSys**
  Identificateurs MIL.

- MIL_ID **milBuffer**
  Buffer de traitement.

- HANDLE **threadGrabbingHasEnded**
  End of the grabbing thread.

- **ProcessingObject*** **processor**
  Traitement utilise.

- SynchClient* **mSyncServer**
  Interface to synch server.

- volatile bool **Acquisition**
  Drapeaux des boucles.

- std::string **mAVIFile**
- std::string **mTimeFile**

Detailed Description

Processing of images from AVI files.

Same as **OnlineProcessing** (*p.292*), but for **OfflineProcessing** (*p.287*). The processing is done by a user-supplied object.

Constructor & Destructor Documentation

Cognois::OfflineProcessing::OfflineProcessing (const char * *inBaseFilename*)

Destructor.

Parameters:

inBaseFilename is the avi file to get images from

Cognois::OfflineProcessing::~OfflineProcessing ()

Member Function Documentation

void Cognois::OfflineProcessing::Begin ()

Start both the grabbing and the processing threads.

void Cognois::OfflineProcessing::End ()

Stop both the grabbing and the processing threads.

void Cognois::OfflineProcessing::LoopMe ()

Start, wait for the keyboard, then end the grabbing/processing.

MILSys & Cognois::OfflineProcessing::MILSys () `[inline]`

Returns the MIL identifiers.

void Cognois::OfflineProcessing::ResetEvents () `[private]`

Set all events initial value.

void Cognois::OfflineProcessing::SetProcessor (ProcessingObject * *inNewProcessor*)

Set the processing to apply to grabbed images.

Parameters:

inNewProcessor is the desired process to apply.

void Cognois::OfflineProcessing::ThreadAcquisition ()

Grabbing thread.

Member Data Documentation

volatile bool Cognois::OfflineProcessing::Acquisition `[private]`

Drapeaux des boucles.

std::string Cognois::OfflineProcessing::mAVIFile `[private]`

MILSys Cognois::OfflineProcessing::mMilSys `[private]`

Identificateurs MIL.

SynchClient * Cognois::OfflineProcessing::mSyncServer `[private]`

Interface to synch server.

std::string Cognois::OfflineProcessing::mTimeFile `[private]`

MIL_ID Cognois::OfflineProcessing::milBuffer `[private]`

Buffer de traitement.

ProcessingObject * Cognois::OfflineProcessing::processor `[private]`

Traitement utilise.

HANDLE Cognois::OfflineProcessing::threadGrabbingHasEnded `[private]`

End of the grabbing thread.

The documentation for this class was generated from the following files:

- **OfflineProcessing.h**
- **OfflineProcessing.cpp**

olist Struct Reference

```
#include <F2C.H>
```

Public Attributes

- **flag oerr**
- **ftnint ounit**
- char* **ofnm**
- **ftnlen ofnmlen**
- char* **osta**
- char* **oacc**
- char* **ofm**
- **ftnint orl**
- char* **oblnk**

Member Data Documentation

char * olist::oacc

char * olist::oblnk

flag olist::oerr

char * olist::ofm

char * olist::ofnm

ftnlen olist::ofnmlen

ftnint olist::orl

char * olist::osta

ftnint olist::ounit

The documentation for this struct was generated from the following file:

- **F2C.H**

Cognois::OnlineProcessing Class Reference

Online processing of images.

`#include <OnlineProcessing.h>`

Public Methods

- **OnlineProcessing** (double inImageScale=1, double inFrameRate=60, bool inSynchro=true, unsigned long inResetPeriod=1000)
  Destructor.

- **~OnlineProcessing** ()
- void **SetProcessor** (**ProcessingObject** *inNewProcessor)
  Set the processing to apply to grabbed images.

- void **LoopMe** ()
  Start, wait for the keyboard, then end the grabbing/processing.

- void **Begin** ()
  Start both the grabbing and the processing threads.

- void **End** ()
  Stop both the grabbing and the processing threads.

- void **ThreadAcquisition** ()
  Grabbing thread.

- void **ThreadTraitement** ()
  Processing thread.

- void **ThreadSynchronization** ()
  Synchronization with server.

- ::MILSys **MILSys** ()
  Returns the MIL identifiers.

- bool **IsGrabbing** () const

Private Methods

- void **ResetEvents** ()
  Set all events initial value.

Private Attributes

- ::**MILSys mMilSys**
  Identificateurs MIL.

- **Acquisition mGrabber**
  The grabber and grabbing buffers.

- **FrameCounter mFrameCounter**
  Counts frames with their time.

- SynchClient* **mSyncServer**
  Interface to synch server.

- volatile **ProcessingObject*** **mProcessor**
  Traitement utilise.

- bool **mSynchro**
  Use synchronisation or not?

- HANDLE **mEventThreadProcessingHasEnded**
  End of the processing thread.

- HANDLE **mEventThreadGrabbingHasEnded**
  End of the grabbing thread.

- HANDLE **mEventThreadSynchronizationHasEnded**
  End of the sync thread.

- HANDLE **mEventImageGrabbed**
  An image has been grabbed.

- HANDLE **mEventRequestProcessing**
  Synchro acquisition-traitement.

- HANDLE **mEventGrabEnd**
  Fin de l'acquisition.

- HANDLE **mEventResetFrameId**
  Ask for a software barrier.

- HANDLE **mMutexImageBuffer**
- HANDLE **mEventEndProcessing**
- HANDLE **mEventBeginProcessing**
- HANDLE **mEventProcessingHasEnded**

- MIL_ID **mImageBuffer**
  Buffers de traitement.

- volatile bool **mAcquisition**
  Drapeaux des boucles.

- volatile bool **mArretTraitement**
- volatile bool **mNoValidFrameYet**
  Any valid since the beginning?

- volatile bool **mResetCompleted**
  A reset has been completed.

- volatile unsigned long **mTimeStampProcessing**
  Timestamp du buffer traite.

- volatile unsigned long **mFrameIdProcessing**
  Identificateur du buffer traite.

- volatile unsigned long **mInitialFrameId**
  First frame ID for the sequence.

- const unsigned long **mResetPeriod**

Detailed Description

Online processing of images.

The grabbing and the image processing are performed in separated threads. The processing is done by a user-supplied object.

Constructor & Destructor Documentation

Cognois::OnlineProcessing::OnlineProcessing (double *inImageScale* = 1, double *inFrameRate* = 60, bool *inSynchro* = true, unsigned long *inResetPeriod* = 1000)

Destructor.

Parameters:

inImageScale is the desired image scale for the digitizer
inFrameRate is the estimated frame rate of the camera
inSynchro is set to true if synchronisation is wanted

Cognois::OnlineProcessing::~OnlineProcessing ()

Member Function Documentation

void Cognois::OnlineProcessing::Begin ()

Start both the grabbing and the processing threads.

void Cognois::OnlineProcessing::End ()

Stop both the grabbing and the processing threads.

bool Cognois::OnlineProcessing::IsGrabbing () const `[inline]`

void Cognois::OnlineProcessing::LoopMe ()

Start, wait for the keyboard, then end the grabbing/processing.

MILSys Cognois::OnlineProcessing::MILSys () `[inline]`

Returns the MIL identifiers.

void Cognois::OnlineProcessing::ResetEvents () `[private]`

Set all events initial value.

void Cognois::OnlineProcessing::SetProcessor (ProcessingObject * *inNewProcessor*)

Set the processing to apply to grabbed images.

Parameters:

inNewProcessor is the desired process to apply.

void Cognois::OnlineProcessing::ThreadAcquisition ()

Grabbing thread.

void Cognois::OnlineProcessing::ThreadSynchronization ()

Synchronization with server.

void Cognois::OnlineProcessing::ThreadTraitement ()

Processing thread.

Member Data Documentation

volatile bool Cognois::OnlineProcessing::mAcquisition [private]

Drapeaux des boucles.

volatile bool Cognois::OnlineProcessing::mArretTraitement [private]

HANDLE Cognois::OnlineProcessing::mEventBeginProcessing [private]

HANDLE Cognois::OnlineProcessing::mEventEndProcessing [private]

HANDLE Cognois::OnlineProcessing::mEventGrabEnd [private]

Fin de l'acquisition.

HANDLE Cognois::OnlineProcessing::mEventImageGrabbed [private]

An image has been grabbed.

HANDLE Cognois::OnlineProcessing::mEventProcessingHasEnded [private]

HANDLE Cognois::OnlineProcessing::mEventRequestProcessing [private]

Synchro acquisition-traitement.

HANDLE Cognois::OnlineProcessing::mEventResetFrameId [private]

Ask for a software barrier.

HANDLE Cognois::OnlineProcessing::mEventThreadGrabbingHasEnded [private]

End of the grabbing thread.

HANDLE Cognois::OnlineProcessing::mEventThreadProcessingHasEnded [private]

End of the processing thread.

HANDLE Cognois::OnlineProcessing::mEventThreadSynchronizationHasEnded [private]

End of the sync thread.

FrameCounter Cognois::OnlineProcessing::mFrameCounter [private]

Counts frames with their time.

volatile unsigned long Cognois::OnlineProcessing::mFrameIdProcessing [private]

Identificateur du buffer traite.

Acquisition Cognois::OnlineProcessing::mGrabber [private]

The grabber and grabbing buffers.

MIL_ID Cognois::OnlineProcessing::mImageBuffer `[private]`

Buffers de traitement.

volatile unsigned long Cognois::OnlineProcessing::mInitialFrameId `[private]`

First frame ID for the sequence.

MILSys Cognois::OnlineProcessing::mMilSys `[private]`

Identificateurs MIL.

HANDLE Cognois::OnlineProcessing::mMutexImageBuffer `[private]`

volatile bool Cognois::OnlineProcessing::mNoValidFrameYet `[private]`

Any valid since the beginning?

volatile ProcessingObject * Cognois::OnlineProcessing::mProcessor `[private]`

Traitement utilise.

volatile bool Cognois::OnlineProcessing::mResetCompleted `[private]`

A reset has been completed.

const unsigned long Cognois::OnlineProcessing::mResetPeriod `[private]`

SynchClient * Cognois::OnlineProcessing::mSyncServer `[private]`

Interface to synch server.

bool Cognois::OnlineProcessing::mSynchro `[private]`

Use synchronisation or not?

volatile unsigned long Cognois::OnlineProcessing::mTimeStampProcessing `[private]`

Timestamp du buffer traite.

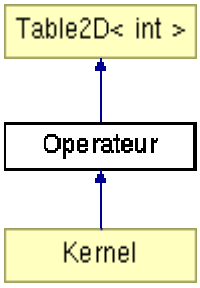
The documentation for this class was generated from the following files:

- **OnlineProcessing.h**
- **OnlineProcessing.cpp**

Operateur Class Reference

`#include <Operateur.hpp>`

Inheritance diagram for Operateur:



Public Types

- enum **OperateurType** { **IDENTITE**, **SOBEL**, **ROBERTS**, **PREWITT**, **SOBELH**, **SOBELV**, **ROBERTSH**, **ROBERTSV**, **PREWITTH**, **PREWITTV**, **GAUSSIENNE**, **GAUSSIENNEH**, **GAUSSIENNEV**, **MOYENNE**, **LAPLACIEN**, **UNKNOWN** }

Public Methods

- **Operateur** (**OperateurType** _type=IDENTITE)
- **Operateur** (**OperateurType** _type, unsigned int _a)
- void **defType** (**OperateurType** _type)
- void **defType** (**OperateurType** _type, unsigned int _a)
- void **defIdentite** ()
- void **defSobelH** ()
- void **defSobelV** ()
- void **defRobertsH** ()
- void **defRobertsV** ()
- void **defRobertsH** (unsigned int _a)
- void **defRobertsV** (unsigned int _a)
- void **defPrewittH** ()
- void **defPrewittV** ()
- void **defGaussienne** (unsigned int _a=5)
- void **defGaussienneH** (unsigned int _a=5)
- void **defGaussienneV** (unsigned int _a=5)
- void **defMoyenne** (unsigned int _a=5)
- void **defLaplacien** ()
- const int **Facteur** () const
- **OperateurType Type** () const
- const **Point Origine** () const
- void **Facteur** (int _facteur)
- void **Origine** (**Point** _org)
- int **ValiderPosition** () const
- void **CalculerPosition** ()
- **operator const int *** () const

Public Attributes

- int **xg**
- int **xd**
- int **yh**
- int **yb**

Protected Methods

- Table1D<int> **Gaussienne** (unsigned int largeur)

Protected Attributes

- **OperateurType type**
- **Point org**
- int **facteur**

Member Enumeration Documentation

enum Operateur::OperateurType

Enumeration values:

IDENTITE
SOBEL
ROBERTS
PREWITT
SOBELH
SOBELV
ROBERTSH
ROBERTSV
PREWITTH
PREWITTV
GAUSSIENNE
GAUSSIENNEH
GAUSSIENNEV
MOYENNE
LAPLACIEN
UNKNOWN

Constructor & Destructor Documentation

Operateur::Operateur (OperateurType *_type* = IDENTITE)

Operateur::Operateur (Operateur::OperateurType *_type*, unsigned int *_a*)

Member Function Documentation

void Operateur::CalculerPosition ()

void Operateur::Facteur (int _*facteur*)

const int Operateur::Facteur () const `[inline]`

Table1D< int > Operateur::Gaussienne (unsigned int _*a*) `[protected]`

void Operateur::Origine (Point _*org*)

const Point Operateur::Origine () const `[inline]`

OperateurType Operateur::Type () const `[inline]`

Reimplemented in **Kernel** (*p.255*).

int Operateur::ValiderPosition () const

void Operateur::defGaussienne (unsigned int _*a* = 5)

void Operateur::defGaussienneH (unsigned int _*a* = 5)

void Operateur::defGaussienneV (unsigned int _*a* = 5)

void Operateur::defIdentite ()

void Operateur::defLaplacien ()

void Operateur::defMoyenne (unsigned int _*a* = 5)

void Operateur::defPrewittH ()

void Operateur::defPrewittV ()

void Operateur::defRobertsH (unsigned int _*a*)

void Operateur::defRobertsH ()

void Operateur::defRobertsV (unsigned int _*a*)

void Operateur::defRobertsV ()

void Operateur::defSobelH ()

void Operateur::defSobelV ()

void Operateur::defType (Operateur::OperateurType _*type*, unsigned int _*a*)

void Operateur::defType (OperateurType _*type*)

Operateur::operator const int * () const `[inline]`

Member Data Documentation

int Operateur::facteur `[protected]`

Point Operateur::org `[protected]`

OperateurType Operateur::type `[protected]`

int Operateur::xd

int Operateur::xg

int Operateur::yb

int Operateur::yh

The documentation for this class was generated from the following files:

- **Operateur.hpp**
- **Operateur.cpp**

FormatPBM::pbmHeader Struct Reference

```
#include <FormatPBM.hpp>
```

Public Attributes

- **UBYTE4 width**
- **UBYTE4 height**

Member Data Documentation

UBYTE4 FormatPBM::pbmHeader::height

UBYTE4 FormatPBM::pbmHeader::width

The documentation for this struct was generated from the following file:

- **FormatPBM.hpp**

FormatPCX::pcxHeader Struct Reference

```
#include <FormatPCX.hpp>
```

Public Attributes

- **UBYTE1 signature**
- **UBYTE1 version**
- **UBYTE1 compression**
- **UBYTE1 bitsperpixel**
- **UBYTE2 xmin**
- **UBYTE2 ymin**
- **UBYTE2 xmax**
- **UBYTE2 ymax**
- **UBYTE2 horesolution**
- **UBYTE2 veresolution**
- **UBYTE1 palette** [48]
- **UBYTE1 reserved**
- **UBYTE1 colorplanes**
- **UBYTE2 bytesperline**
- **UBYTE2 palettetype**
- **UBYTE1 filler** [58]

Member Data Documentation

UBYTE1 FormatPCX::pcxHeader::bitsperpixel

UBYTE2 FormatPCX::pcxHeader::bytesperline

UBYTE1 FormatPCX::pcxHeader::colorplanes

UBYTE1 FormatPCX::pcxHeader::compression

UBYTE1 FormatPCX::pcxHeader::filler

UBYTE2 FormatPCX::pcxHeader::horesolution

UBYTE1 FormatPCX::pcxHeader::palette

UBYTE2 FormatPCX::pcxHeader::palettetype

UBYTE1 FormatPCX::pcxHeader::reserved

UBYTE1 FormatPCX::pcxHeader::signature

UBYTE2 FormatPCX::pcxHeader::veresolution

UBYTE1 FormatPCX::pcxHeader::version

UBYTE2 FormatPCX::pcxHeader::xmax

UBYTE2 FormatPCX::pcxHeader::xmin

UBYTE2 FormatPCX::pcxHeader::ymax

UBYTE2 FormatPCX::pcxHeader::ymin

The documentation for this struct was generated from the following file:

- **FormatPCX.hpp**

FormatPGM::pgmHeader Struct Reference

```
#include <FormatPGM.hpp>
```

Public Attributes

- **UBYTE4 width**
- **UBYTE4 height**
- **UBYTE4 maxvalue**

Member Data Documentation

UBYTE4 FormatPGM::pgmHeader::height

UBYTE4 FormatPGM::pgmHeader::maxvalue

UBYTE4 FormatPGM::pgmHeader::width

The documentation for this struct was generated from the following file:

- **FormatPGM.hpp**

Picture Class Reference

```
#include <Picture.h>
```

Public Methods

- **Picture** ()
- **Picture** (const Picture &)
- virtual **~Picture** ()
- int **SetWidth** (unsigned int)
- int **SetHeight** (unsigned int)
- int **SetType** (int)
- int **GetType** () const
- unsigned int **GetWidth** () const
- unsigned int **GetHeight** () const
- unsigned int **GetSize** () const
- int **GetLastError** ()
- void **GetHistogram** (int Histo[256])
- void **Clear** ()
- int **PutCross** (int x, int y, unsigned int w=5)
- void **SetCrossColor** (**RGBQUADsd** c)
- void **SetCrossColor** (unsigned char c)
- void **SetCrossColor** (unsigned short c)
- int **AutoThresholding** ()
- int **AutoThresholdEstimation** (unsigned char &EstimatedThreshold)
- int **ApplyThreshold** (unsigned char t)
- int **RepresentFloatData** (float *FDataPtr, unsigned int DataWidth, unsigned int DataHeight, int RepType=CS_COLOR)
- int **RepresentIndexedData** (unsigned int *IData, unsigned int NbIndex, unsigned int DataWidth, unsigned int DataHeight)
- int **ConvertToGrayScale** ()
- int **GraphLine** (unsigned int LineNb, float *FDataPtr, unsigned int DataWidth, unsigned int DataHeight)
- int **GraphLine** (unsigned int LineNb, Picture &Pict)
- void* **GetDataPtr** ()
- const Picture& **operator=** (const Picture &)

Private Methods

- int **AllocateDataBuffer** ()
- int **Equal** (const Picture &)

Private Attributes

- int **Type**
- int **LastError**
- **RGBQUADsd*** **DataRGB**
- unsigned char* **Data8**
- unsigned short* **Data16**
- unsigned int **Width**
- unsigned int **Height**

- unsigned int **Size**
- **RGBQUADsd CrossColorRGB**
- unsigned char **CrossColor8**
- unsigned short **CrossColor16**

Constructor & Destructor Documentation

Picture::Picture ()

Picture::Picture (const Picture & *PictToCopy*)

Picture::~Picture () `[virtual]`

Member Function Documentation

int Picture::AllocateDataBuffer () `[private]`

int Picture::ApplyThreshold (unsigned char *t*)

int Picture::AutoThresholdEstimation (unsigned char & *EstimatedThreshold*)

int Picture::AutoThresholding ()

void Picture::Clear ()

int Picture::ConvertToGrayScale ()

int Picture::Equal (const Picture & *PictToCopy*) `[private]`

void * Picture::GetDataPtr ()

unsigned int Picture::GetHeight () const

void Picture::GetHistogram (int *Histo*[256])

int Picture::GetLastError () `[inline]`

unsigned int Picture::GetSize () const

int Picture::GetType () const `[inline]`

unsigned int Picture::GetWidth () const

int Picture::GraphLine (unsigned int *LineNb*, Picture & *Pict*)

int Picture::GraphLine (unsigned int *LineNb*, float * *FDataPtr*, unsigned int *DataWidth*, unsigned int *DataHeight*)

int Picture::PutCross (int *x*, int *y*, unsigned int *w* = 5)

int Picture::RepresentFloatData (float * *FDataPtr*, unsigned int *DataWidth*, unsigned int *DataHeight*, int *RepType* = CS_COLOR)

int Picture::RepresentIndexedData (unsigned int * *IData*, unsigned int *NbIndex*, unsigned int *DataWidth*, unsigned int *DataHeight*)

void Picture::SetCrossColor (unsigned short *c*) `[inline]`

void Picture::SetCrossColor (unsigned char *c*) `[inline]`

void Picture::SetCrossColor (RGBQUADsd *c*) `[inline]`

int Picture::SetHeight (unsigned int *h*)

int Picture::SetType (int *t*)

int Picture::SetWidth (unsigned int *w*)

const Picture & Picture::operator= (const Picture & *PictToCopy*)

Member Data Documentation

unsigned short Picture::CrossColor16 `[private]`

unsigned char Picture::CrossColor8 `[private]`

RGBQUADsd Picture::CrossColorRGB `[private]`

unsigned short * Picture::Data16 `[private]`

unsigned char * Picture::Data8 `[private]`

RGBQUADsd * Picture::DataRGB `[private]`

unsigned int Picture::Height `[private]`

int Picture::LastError `[private]`

unsigned int Picture::Size `[private]`

int Picture::Type `[private]`

unsigned int Picture::Width `[private]`

The documentation for this class was generated from the following files:

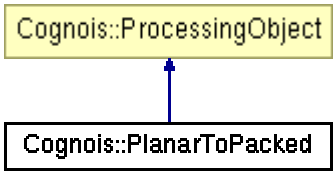
- **Picture.h**
- **Picture.cpp**

Cognois::PlanarToPacked Class Reference

Transform a MIL planar buffer to a MIL packed buffer.

`#include <PlanarToPacked.h>`

Inheritance diagram for Cognois::PlanarToPacked:



Public Methods

- **PlanarToPacked** (const::MILSys &_milSys)
  *Constructor with a reference to the **MILSys** (p.276) of the **OnlineProcessing** (p.292) object.*

- **~PlanarToPacked** ()
  Destructor.

- void **Process** (MIL_ID &_milImage, unsigned long &_frameID, unsigned long &_timeStamp)
  Process an image with know frameID and timeStamp.

Private Attributes

- MIL_ID **milBufferPacked**
  MIL packed buffer.

Detailed Description

Transform a MIL planar buffer to a MIL packed buffer.

Constructor & Destructor Documentation

Cognois::PlanarToPacked::PlanarToPacked (const::MILSys & *_milSys*)

Constructor with a reference to the **MILSys** (*p.276*) of the **OnlineProcessing** (*p.292*) object.

Cognois::PlanarToPacked::~PlanarToPacked ()

Destructor.

Member Function Documentation

void Cognois::PlanarToPacked::Process (MIL_ID & *_milImage*, unsigned long & *_frameID*, unsigned long & *_timeStamp*) `[virtual]`

Process an image with know frameID and timeStamp.

Parameters:

inMilImage is a MIL image identifier.
inFrameID is the frame identifier associated with the image.
inTimeStamp is the time, in milliseconds, associated with the image.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

Member Data Documentation

MIL_ID Cognois::PlanarToPacked::milBufferPacked `[private]`

MIL packed buffer.

The documentation for this class was generated from the following files:

- **PlanarToPacked.h**
- **PlanarToPacked.cpp**

PngChromaticitiesData Struct Reference

```
#include <pngpvt.h>
```

Public Attributes

- **UBYTE4 whitepointx**
- **UBYTE4 whitepointy**
- **UBYTE4 redx**
- **UBYTE4 redy**
- **UBYTE4 greenx**
- **UBYTE4 greeny**
- **UBYTE4 bluex**
- **UBYTE4 bluey**

Member Data Documentation

UBYTE4 PngChromaticitiesData::bluex

UBYTE4 PngChromaticitiesData::bluey

UBYTE4 PngChromaticitiesData::greenx

UBYTE4 PngChromaticitiesData::greeny

UBYTE4 PngChromaticitiesData::redx

UBYTE4 PngChromaticitiesData::redy

UBYTE4 PngChromaticitiesData::whitepointx

UBYTE4 PngChromaticitiesData::whitepointy

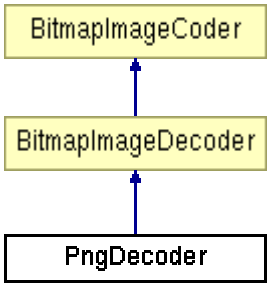
The documentation for this struct was generated from the following file:

- **pngpvt.h**

PngDecoder Class Reference

`#include <pngdecod.h>`

Inheritance diagram for PngDecoder:



Public Methods

- **PngDecoder** ()
- **PngDecoder** (const PngDecoder &)
- virtual **~PngDecoder** ()
- PngDecoder& **operator=** (PngDecoder &)
- virtual void **ReadImage** (istream &, **BitmapImage** &)
- void **SetVerbose** (bool)
- bool **GetVerbose** () const

Private Methods

- void **Initialize** ()
- void **DoCopy** (const PngDecoder &)
- void **ReadSignature** ()
- void **ReadChunk** ()
- void **ProcessHeader** ()
- void **ProcessPalette** ()
- void **ProcessBackground** ()
- void **ProcessGamma** ()
- void **ProcessData** ()
- void **ProcessChromaticities** ()
- void **ProcessHistogram** ()
- void **ProcessPhysicalPixelDimensions** ()
- void **ProcessSignificantBits** ()
- void **ProcessTextualData** ()
- void **ProcessImageTime** ()
- void **ProcessTransparency** ()
- void **ProcessCompressedText** ()
- void **CallProgressFunction** (unsigned int percent)
- int **GetBits** (unsigned int count=1)
- **UBYTE1 GetIDATByte** ()
- **UBYTE1 DecodeByte** ()
- void **StartNewDataSet** ()
- void **CheckAdler** ()
- void **ReadLengths** (**PngHuffmanDecoder** &, unsigned int[], unsigned int)

- **UBYTE1 DecodeLiteralByte** ()
- **UBYTE1 DecodeCompressedByte** ()
- void **FreeData** ()
- void **ReadPixelData** ()
- void **CopyNoninterlacedRowToImage** (unsigned int row)
- void **CopyInterlacedRowToImage** (unsigned int row, unsigned int width)
- void **ReadNoninterlaced** ()
- void **ReadInterlaced** ()
- void **FilterRow** (unsigned int filter)

Private Attributes

- **UBYTE1 background_red**
- **UBYTE1 background_green**
- **UBYTE1 background_blue**
- **UBYTE1 background_gray**
- **UBYTE4 chunk_length**
- **UBYTE4 chunk_type**
- **UBYTE1* chunk_data**
- **UBYTE4 chunk_data_size**
- **UBYTE4 chunk_crc**
- **UBYTE4 stream_adler**
- **UBYTE1* lz_window**
- unsigned int **window_position**
- unsigned int **copy_position**
- unsigned int **copy_count**
- **PngHuffmanDecoder* literal_table**
- **PngHuffmanDecoder* distance_table**
- bool **final_data_set**
- bool **literal_mode**
- unsigned int **literal_count**
- istream* **input_stream**
- **BitmapImage* current_image**
- bool **verbose_flag**
- unsigned int **palette_size**
- bool **reading_pixel_data**
- unsigned int **bit_offset**
- unsigned int **byte_offset**
- **UBYTE1 bit_buffer**
- **UBYTE4 image_height**
- **UBYTE4 image_width**
- **UBYTE1 image_depth**
- unsigned int **image_color_type**
- **UBYTE1 image_compression_method**
- **UBYTE1 image_filter_method**
- **UBYTE1 image_interlace_method**
- **UBYTE1* row_buffers** [2]
- int **current_row_buffer**
- unsigned int **row_buffer_width**
- unsigned int **interlace_pass**
- double **file_gamma**
- bool **data_read**

- bool **palette_read**
- bool **header_read**
- bool **end_read**

Static Private Attributes

- const **InterlaceInfo interlace_values** []

Friends

- class **PngHuffmanDecoder**

Constructor & Destructor Documentation

PngDecoder::PngDecoder ()

PngDecoder::PngDecoder (const PngDecoder & *source*)

PngDecoder::~PngDecoder () `[virtual]`

Member Function Documentation

void PngDecoder::CallProgressFunction (unsigned int *percent*) `[private]`

void PngDecoder::CheckAdler () `[private]`

void PngDecoder::CopyInterlacedRowToImage (unsigned int *row*, unsigned int *rowwidth*) `[private]`

void PngDecoder::CopyNoninterlacedRowToImage (unsigned int *row*) `[private]`

UBYTE1 PngDecoder::DecodeByte () `[private]`

UBYTE1 PngDecoder::DecodeCompressedByte () `[private]`

UBYTE1 PngDecoder::DecodeLiteralByte () `[private]`

void PngDecoder::DoCopy (const PngDecoder & *source*) `[private]`

void PngDecoder::FilterRow (unsigned int *filter*) `[private]`

void PngDecoder::FreeData () `[private]`

int PngDecoder::GetBits (unsigned int *count* = 1) `[private]`

UBYTE1 PngDecoder::GetIDATByte () `[private]`

bool PngDecoder::GetVerbose () const `[inline]`

void PngDecoder::Initialize () `[private]`

Reimplemented from **BitmapImageCoder** (*p.38*).

void PngDecoder::ProcessBackground () `[private]`

void PngDecoder::ProcessChromaticities () `[private]`

void PngDecoder::ProcessCompressedText () `[private]`

void PngDecoder::ProcessData () `[private]`

void PngDecoder::ProcessGamma () `[private]`

void PngDecoder::ProcessHeader () `[private]`

void PngDecoder::ProcessHistogram () `[private]`

void PngDecoder::ProcessImageTime () `[private]`

void PngDecoder::ProcessPalette () `[private]`

void PngDecoder::ProcessPhysicalPixelDimensions () `[private]`

void PngDecoder::ProcessSignificantBits () `[private]`

void PngDecoder::ProcessTextualData () `[private]`

void PngDecoder::ProcessTransparency () `[private]`

void PngDecoder::ReadChunk () `[private]`

void PngDecoder::ReadImage (istream & *strm*, BitmapImage & *image*) `[virtual]`

Reimplemented from **BitmapImageDecoder** (*p.39*).

void PngDecoder::ReadInterlaced () `[private]`

void PngDecoder::ReadLengths (PngHuffmanDecoder & *ht*, unsigned int *lengths*[], unsigned int *lengthcount*) `[private]`

void PngDecoder::ReadNoninterlaced () `[private]`

void PngDecoder::ReadPixelData () `[private]`

void PngDecoder::ReadSignature () `[private]`

void PngDecoder::SetVerbose (bool *value*) `[inline]`

void PngDecoder::StartNewDataSet () `[private]`

PngDecoder & PngDecoder::operator= (PngDecoder & *source*)

Friends And Related Function Documentation

class PngHuffmanDecoder `[friend]`

Member Data Documentation

UBYTE1 PngDecoder::background_blue `[private]`

UBYTE1 PngDecoder::background_gray `[private]`

UBYTE1 PngDecoder::background_green `[private]`

UBYTE1 PngDecoder::background_red `[private]`

UBYTE1 PngDecoder::bit_buffer `[private]`

unsigned int PngDecoder::bit_offset `[private]`

unsigned int PngDecoder::byte_offset `[private]`

UBYTE4 PngDecoder::chunk_crc `[private]`

UBYTE1 * PngDecoder::chunk_data `[private]`

UBYTE4 PngDecoder::chunk_data_size `[private]`

UBYTE4 PngDecoder::chunk_length `[private]`

UBYTE4 PngDecoder::chunk_type `[private]`

unsigned int PngDecoder::copy_count `[private]`

unsigned int PngDecoder::copy_position `[private]`

BitmapImage * PngDecoder::current_image `[private]`

int PngDecoder::current_row_buffer `[private]`

bool PngDecoder::data_read `[private]`

PngHuffmanDecoder * PngDecoder::distance_table `[private]`

bool PngDecoder::end_read `[private]`

double PngDecoder::file_gamma `[private]`

bool PngDecoder::final_data_set `[private]`

bool PngDecoder::header_read `[private]`

unsigned int PngDecoder::image_color_type `[private]`

UBYTE1 PngDecoder::image_compression_method `[private]`

UBYTE1 PngDecoder::image_depth `[private]`

UBYTE1 PngDecoder::image_filter_method `[private]`

UBYTE4 PngDecoder::image_height `[private]`

UBYTE1 PngDecoder::image_interlace_method `[private]`

UBYTE4 PngDecoder::image_width `[private]`

istream * PngDecoder::input_stream `[private]`

unsigned int PngDecoder::interlace_pass `[private]`

const PngDecoder::InterlaceInfo PngDecoder::interlace_values `[static, private]`

Initial value:

```
{
  { 8, 8, 0, 0, },
  { 8, 8, 0, 4, },
  { 8, 4, 4, 0, },
  { 4, 4, 0, 2, },
  { 4, 2, 2, 0, },
  { 2, 2, 0, 1, },
  { 2, 1, 1, 0, }
}
```

unsigned int PngDecoder::literal_count `[private]`

bool PngDecoder::literal_mode `[private]`

PngHuffmanDecoder * PngDecoder::literal_table `[private]`

UBYTE1 * PngDecoder::lz_window `[private]`

bool PngDecoder::palette_read `[private]`

unsigned int PngDecoder::palette_size `[private]`

bool PngDecoder::reading_pixel_data `[private]`

unsigned int PngDecoder::row_buffer_width `[private]`

UBYTE1 * PngDecoder::row_buffers `[private]`

UBYTE4 PngDecoder::stream_adler `[private]`

bool PngDecoder::verbose_flag `[private]`

unsigned int PngDecoder::window_position `[private]`

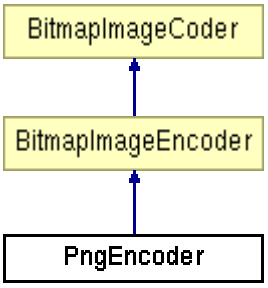
The documentation for this class was generated from the following files:

- **pngdecod.h**
- **pngdecod.cpp**

PngEncoder Class Reference

`#include <pngencod.h>`

Inheritance diagram for PngEncoder:



Public Types

- enum **CompressionLevel** { **FastestCompression** = 0, **FastCompression** = 1, **DefaultCompression** = 2, **MaximumCompression** = 3 }

Public Methods

- **PngEncoder** ()
- **PngEncoder** (const PngEncoder &)
- virtual **~PngEncoder** ()
- PngEncoder& **operator=** (const PngEncoder &)
- void **WriteImage** (ostream &, **BitmapImage** &)
- void **SetTitle** (const Chaine &)
- Chaine **GetTitle** () const
- void **SetAuthor** (const Chaine &)
- Chaine **GetAuthor** () const
- void **SetDescription** (const Chaine &)
- Chaine **GetDescription** () const
- void **SetCopyright** (const Chaine &)
- Chaine **GetCopyright** () const
- void **SetSoftware** (const Chaine &)
- Chaine **GetSoftware** () const
- void **SetDisclaimer** (const Chaine &)
- Chaine **GetDisclaimer** () const
- void **SetWarning** (const Chaine &)
- Chaine **GetWarning** () const
- void **SetSource** (const Chaine &)
- Chaine **GetSource** () const
- void **SetComment** (const Chaine &)
- Chaine **GetComment** () const
- void **SetCompressionLevel** (**CompressionLevel**)
- **CompressionLevel GetCompressionLevel** () const
- void **SetUseFilters** (bool)
- bool **GetUseFilters** () const
- unsigned long **GetBlockSize** () const
- void **SetBlockSize** (unsigned long)

Protected Methods

- void **Initialize** ()
- void **DoCopy** (const PngEncoder &)

Private Types

- typedef void (PngEncoder::* **LENGTHFUNCTION**)(unsigned int index, unsigned int code, unsigned int length, unsigned int extra)
- enum { **FilterBufferCount** = 5 }

Private Methods

- void **LongestMatch** (unsigned int &length, unsigned int &offset)
- unsigned int **HashValue** (unsigned int)
- void **MoveHashEntry** (unsigned int entry, unsigned int hashvalue)
- void **GatherLengthCounts** (unsigned int, unsigned int code, unsigned int, unsigned int)
- void **OutputLengthCounts** (unsigned int index, unsigned int code, unsigned int extra, unsigned int value)
- void **FindLengthCodes** (**LENGTHFUNCTION**, **UBYTE1** lengths[], unsigned int count)
- bool **ProcessImageData** ()
- void **StartChunk** (const char type[5])
- void **WriteCurrentChunk** ()
- void **OutputBlock** (const void *data, unsigned int length)
- void **OutputByte** (**UBYTE1**)
- void **OutputDataBits** (unsigned int data, unsigned int length)
- void **FlushBitBuffer** ()
- void **OutputAdler** ()
- void **FreeBuffers** ()
- void **FillBuffer** (unsigned int count)
- void **WriteText** (const Chaine &keyword, const Chaine &value)
- void **WriteTextBlocks** ()
- void **CallProgressFunction** (unsigned int)
- void **FilterRow** (unsigned int row)
- void **DoWrite** ()
- void **InitializeHashTable** ()
- void **InitializeLZWindow** ()
- void **OutputDeflateHeader** (bool lastblock)
- void **OutputZLibHeader** ()
- void **WriteImageData** ()
- void **OutputBlockData** ()

Private Attributes

- **HashEntry* hash_values**
- **HashEntry* hash_table**
- **BitmapImage* current_image**
- ostream* **output_stream**
- unsigned int **search_limit**
- **PngHuffmanEncoder* distance_table**
- **PngHuffmanEncoder* length_table**

- **PngHuffmanEncoder* length_length_table**
- **UBYTE1* output_buffer**
- **UBYTE4 chunk_type**
- **UBYTE4 byte_position**
- unsigned int **bit_position**
- unsigned int **row_width**
- Chaine **title_string**
- Chaine **author_string**
- Chaine **description_string**
- Chaine **copyright_string**
- Chaine **software_string**
- Chaine **disclaimer_string**
- Chaine **warning_string**
- Chaine **source_string**
- Chaine **comment_string**
- **UBYTE1* filter_buffers** [FilterBufferCount]
- unsigned int **current_filter**
- unsigned int **filter_mask**
- unsigned int **filter_width**
- **CompressionLevel compression_level**
- **UBYTE4 adler_value**
- **UBYTE1* lz_window**
- **UBYTE2* lookahead_buffer**
- unsigned int **lookahead_position**
- unsigned int **lz_position**
- int **image_row**
- unsigned int **image_col**
- bool **image_end**
- **UBYTE2* block_buffer**
- unsigned int **block_buffer_size**
- unsigned int **block_buffer_count**

Member Typedef Documentation

typedef void(PngEncoder::* PngEncoder::LENGTHFUNCTION)(unsigned int index, unsigned int code, unsigned int length, unsigned int extra) `[private]`

Member Enumeration Documentation

anonymous enum `[private]`

Enumeration values:

FilterBufferCount

enum PngEncoder::CompressionLevel

Enumeration values:

FastestCompression
FastCompression
DefaultCompression
MaximumCompression

Constructor & Destructor Documentation

PngEncoder::PngEncoder ()

PngEncoder::PngEncoder (const PngEncoder & *source*)

PngEncoder::~PngEncoder () `[virtual]`

Member Function Documentation

void PngEncoder::CallProgressFunction (unsigned int *percent*) `[private]`

void PngEncoder::DoCopy (const PngEncoder & *source*) `[protected]`

void PngEncoder::DoWrite () `[private]`

void PngEncoder::FillBuffer (unsigned int *count*) `[private]`

void PngEncoder::FilterRow (unsigned int *row*) `[private]`

void PngEncoder::FindLengthCodes (LENGTHFUNCTION *function*, UBYTE1 *lengths*[], unsigned int *count*) `[private]`

void PngEncoder::FlushBitBuffer () `[private]`

void PngEncoder::FreeBuffers () `[private]`

void PngEncoder::GatherLengthCounts (unsigned *int*, unsigned int *code*, unsigned *int*, unsigned *int*) `[private]`

Chaine PngEncoder::GetAuthor () const `[inline]`

unsigned long PngEncoder::GetBlockSize () const `[inline]`

Chaine PngEncoder::GetComment () const `[inline]`

PngEncoder::CompressionLevel PngEncoder::GetCompressionLevel () const

Chaine PngEncoder::GetCopyright () const `[inline]`

Chaine PngEncoder::GetDescription () const `[inline]`

Chaine PngEncoder::GetDisclaimer () const `[inline]`

Chaine PngEncoder::GetSoftware () const `[inline]`

Chaine PngEncoder::GetSource () const `[inline]`

Chaine PngEncoder::GetTitle () const `[inline]`

bool PngEncoder::GetUseFilters () const

Chaine PngEncoder::GetWarning () const `[inline]`

unsigned int PngEncoder::HashValue (unsigned int *index*) `[private]`

void PngEncoder::Initialize () `[protected]`

Reimplemented from **BitmapImageCoder** (*p.38*).

void PngEncoder::InitializeHashTable () `[private]`

void PngEncoder::InitializeLZWindow () `[private]`

void PngEncoder::LongestMatch (unsigned int & *bestlength*, unsigned int & *bestoffset*) `[private]`

void PngEncoder::MoveHashEntry (unsigned int *entry*, unsigned int *hashvalue*) `[private]`

void PngEncoder::OutputAdler () `[private]`

void PngEncoder::OutputBlock (const void * *data*, unsigned int *length*) `[private]`

void PngEncoder::OutputBlockData () `[private]`

void PngEncoder::OutputByte (UBYTE1 *value*) `[private]`

void PngEncoder::OutputDataBits (unsigned int *data*, unsigned int *length*) `[private]`

void PngEncoder::OutputDeflateHeader (bool *lastblock*) `[private]`

void PngEncoder::OutputLengthCounts (unsigned int *index*, unsigned int *code*, unsigned int *extra*, unsigned int *value*) `[private]`

void PngEncoder::OutputZLibHeader () `[private]`

bool PngEncoder::ProcessImageData () `[private]`

void PngEncoder::SetAuthor (const Chaine & *value*) `[inline]`

void PngEncoder::SetBlockSize (unsigned long *value*) `[inline]`

void PngEncoder::SetComment (const Chaine & *value*) `[inline]`

void PngEncoder::SetCompressionLevel (PngEncoder::CompressionLevel *value*)

void PngEncoder::SetCopyright (const Chaine & *value*) `[inline]`

void PngEncoder::SetDescription (const Chaine & *value*) `[inline]`

void PngEncoder::SetDisclaimer (const Chaine & *value*) `[inline]`

void PngEncoder::SetSoftware (const Chaine & *value*) `[inline]`

void PngEncoder::SetSource (const Chaine & *value*) `[inline]`

void PngEncoder::SetTitle (const Chaine & *value*) `[inline]`

void PngEncoder::SetUseFilters (bool *value*)

void PngEncoder::SetWarning (const Chaine & *value*) `[inline]`

void PngEncoder::StartChunk (const char *type*[5]) `[private]`

void PngEncoder::WriteCurrentChunk () `[private]`

void PngEncoder::WriteImage (ostream & *strm*, BitmapImage & *image*) `[virtual]`

Reimplemented from **BitmapImageEncoder** (*p.40*).

void PngEncoder::WriteImageData () `[private]`

void PngEncoder::WriteText (const Chaine & *keyword*, const Chaine & *value*) `[private]`

void PngEncoder::WriteTextBlocks () `[private]`

PngEncoder & PngEncoder::operator= (const PngEncoder & *source*)

Member Data Documentation

UBYTE4 PngEncoder::adler_value `[private]`

Chaine PngEncoder::author_string `[private]`

unsigned int PngEncoder::bit_position `[private]`

UBYTE2 * PngEncoder::block_buffer `[private]`

unsigned int PngEncoder::block_buffer_count `[private]`

unsigned int PngEncoder::block_buffer_size `[private]`

UBYTE4 PngEncoder::byte_position `[private]`

UBYTE4 PngEncoder::chunk_type `[private]`

Chaine PngEncoder::comment_string `[private]`

CompressionLevel PngEncoder::compression_level `[private]`

Chaine PngEncoder::copyright_string `[private]`

unsigned int PngEncoder::current_filter `[private]`

BitmapImage * PngEncoder::current_image `[private]`

Chaine PngEncoder::description_string `[private]`

Chaine PngEncoder::disclaimer_string `[private]`

PngHuffmanEncoder * PngEncoder::distance_table `[private]`

UBYTE1 * PngEncoder::filter_buffers `[private]`

unsigned int PngEncoder::filter_mask `[private]`

unsigned int PngEncoder::filter_width `[private]`

HashEntry * PngEncoder::hash_table `[private]`

HashEntry * PngEncoder::hash_values `[private]`

unsigned int PngEncoder::image_col `[private]`

bool PngEncoder::image_end `[private]`

int PngEncoder::image_row `[private]`

PngHuffmanEncoder * PngEncoder::length_length_table `[private]`

PngHuffmanEncoder * PngEncoder::length_table `[private]`

UBYTE2 * PngEncoder::lookahead_buffer `[private]`

unsigned int PngEncoder::lookahead_position `[private]`

unsigned int PngEncoder::lz_position `[private]`

UBYTE1 * PngEncoder::lz_window `[private]`

UBYTE1 * PngEncoder::output_buffer `[private]`

ostream * PngEncoder::output_stream `[private]`

unsigned int PngEncoder::row_width `[private]`

unsigned int PngEncoder::search_limit `[private]`

Chaine PngEncoder::software_string `[private]`

Chaine PngEncoder::source_string `[private]`

Chaine PngEncoder::title_string `[private]`

Chaine PngEncoder::warning_string `[private]`

The documentation for this class was generated from the following files:

- **pngencod.h**
- **pngencod.cpp**

PngHuffmanDecoder Class Reference

`#include <pnghuffd.h>`

Public Methods

- **PngHuffmanDecoder** ()
- virtual **~PngHuffmanDecoder** ()
- void **MakeTable** (unsigned int lengthcount, const unsigned int huffbits[])
- void **Dump** () const
- int **Decode** (**PngDecoder** &)

Private Types

- enum { **MaxHuffmanCodeLength** = 15 }

Private Methods

- **PngHuffmanDecoder** (const PngHuffmanDecoder &)
- PngHuffmanDecoder& **operator=** (const PngHuffmanDecoder &)

Private Attributes

- int **maxcode** [MaxHuffmanCodeLength]
- int **mincode** [MaxHuffmanCodeLength]
- **UBYTE2 valptr** [MaxHuffmanCodeLength]
- **UBYTE2 huff_values** [**PngMaxNumberOfHuffmanCodes**]
- unsigned int **value_count**

Member Enumeration Documentation

anonymous enum `[private]`

Enumeration values:

MaxHuffmanCodeLength

Constructor & Destructor Documentation

PngHuffmanDecoder::PngHuffmanDecoder () `[inline]`

PngHuffmanDecoder::~PngHuffmanDecoder () `[inline, virtual]`

PngHuffmanDecoder::PngHuffmanDecoder (const PngHuffmanDecoder &) `[private]`

Member Function Documentation

int PngHuffmanDecoder::Decode (PngDecoder & *decoder*)

void PngHuffmanDecoder::Dump () const

void PngHuffmanDecoder::MakeTable (unsigned int *valuecount*, const unsigned int *codelengths*[])

PngHuffmanDecoder & PngHuffmanDecoder::operator= (const PngHuffmanDecoder &) `[private]`

Member Data Documentation

UBYTE2 PngHuffmanDecoder::huff_values `[private]`

int PngHuffmanDecoder::maxcode `[private]`

int PngHuffmanDecoder::mincode `[private]`

UBYTE2 PngHuffmanDecoder::valptr `[private]`

unsigned int PngHuffmanDecoder::value_count `[private]`

The documentation for this class was generated from the following files:

- **pnghuffd.h**
- **pnghuffd.cpp**

PngHuffmanEncoder Class Reference

`#include <pnghuffe.h>`

Public Methods

- **PngHuffmanEncoder** ()
- virtual **~PngHuffmanEncoder** ()
- void **Reset** ()
- void **IncrementFrequency** (unsigned int value)
- void **BuildTable** (unsigned int maxlength)
- void **Encode** (unsigned int value, **UBYTE2** &code, **UBYTE1** &size) const

Private Methods

- **PngHuffmanEncoder** (const PngHuffmanEncoder &)
- PngHuffmanEncoder& **operator=** (const PngHuffmanEncoder &)
- void **MakeTable** ()

Private Attributes

- unsigned int* **frequencies**
- **UBYTE1*** **ehufsi**
- **UBYTE2*** **ehufco**

Friends

- class **PngEncoder**

Constructor & Destructor Documentation

PngHuffmanEncoder::PngHuffmanEncoder ()

PngHuffmanEncoder::~PngHuffmanEncoder () `[virtual]`

PngHuffmanEncoder::PngHuffmanEncoder (const PngHuffmanEncoder &) `[private]`

Member Function Documentation

void PngHuffmanEncoder::BuildTable (unsigned int *maxlength*)

void PngHuffmanEncoder::Encode (unsigned int *value*, UBYTE2 & *code*, UBYTE1 & *size*) const

void PngHuffmanEncoder::IncrementFrequency (unsigned int *value*)

void PngHuffmanEncoder::MakeTable () `[private]`

void PngHuffmanEncoder::Reset ()

PngHuffmanEncoder & PngHuffmanEncoder::operator= (const PngHuffmanEncoder &) `[private]`

Friends And Related Function Documentation

class PngEncoder `[friend]`

Member Data Documentation

UBYTE2 * PngHuffmanEncoder::ehufco `[private]`

UBYTE1 * PngHuffmanEncoder::ehufsi `[private]`

unsigned int * PngHuffmanEncoder::frequencies `[private]`

The documentation for this class was generated from the following files:

- **pnghuffe.h**
- **pnghuffe.cpp**

PngImageHeader Struct Reference

```
#include <pngpvt.h>
```

Public Attributes

- **UBYTE4 width**
- **UBYTE4 height**
- **UBYTE1 bitdepth**
- **UBYTE1 colortype**
- **UBYTE1 compressionmethod**
- **UBYTE1 filtermethod**
- **UBYTE1 interlacemethod**

Member Data Documentation

UBYTE1 PngImageHeader::bitdepth

UBYTE1 PngImageHeader::colortype

UBYTE1 PngImageHeader::compressionmethod

UBYTE1 PngImageHeader::filtermethod

UBYTE4 PngImageHeader::height

UBYTE1 PngImageHeader::interlacemethod

UBYTE4 PngImageHeader::width

The documentation for this struct was generated from the following file:

- **pngpvt.h**

PngPixelDimensions Struct Reference

```
#include <pngpvt.h>
```

Public Attributes

- **UBYTE4 pixelsx**
- **UBYTE4 pixelsy**
- **UBYTE1 unit**

Member Data Documentation

UBYTE4 PngPixelDimensions::pixelsx

UBYTE4 PngPixelDimensions::pixelsy

UBYTE1 PngPixelDimensions::unit

The documentation for this struct was generated from the following file:

- **pngpvt.h**

PngTimeData Struct Reference

```
#include <pngpvt.h>
```

Public Attributes

- **UBYTE2 year**
- **UBYTE1 month**
- **UBYTE1 day**
- **UBYTE1 hour**
- **UBYTE1 minute**
- **UBYTE1 second**

Member Data Documentation

UBYTE1 PngTimeData::day

UBYTE1 PngTimeData::hour

UBYTE1 PngTimeData::minute

UBYTE1 PngTimeData::month

UBYTE1 PngTimeData::second

UBYTE2 PngTimeData::year

The documentation for this struct was generated from the following file:

- **pngpvt.h**

Point Class Reference

`#include <Point.hpp>`

Public Methods

- **Point** (int _l=0, int _c=0)
- int **operator**< (const Point &p) const
- int **operator**> (const Point &p) const
- int **operator**<= (const Point &p) const
- int **operator**>= (const Point &p) const
- int **operator**== (const Point &p) const
- int **operator!=** (const Point &p) const
- int **Chessboard** (const Point &p) const
- int **CityBlock** (const Point &p) const
- double **Euclidean** (const Point &p) const
- int **X** () const
- int **Y** () const
- void **X** (int _x)
- void **Y** (int _y)

Public Attributes

- int **ligne**
- int **col**

Friends

- ostream& **operator**<< (ostream &out, const Point &p)
- istream& **operator**>> (istream &in, Point &p)

Constructor & Destructor Documentation

Point::Point (int _*l* = 0, int _*c* = 0) `[inline]`

Member Function Documentation

int Point::Chessboard (const Point & *p*) const `[inline]`

int Point::CityBlock (const Point & *p*) const `[inline]`

double Point::Euclidean (const Point & *p*) const `[inline]`

void Point::X (int *_x*) `[inline]`

int Point::X () const `[inline]`

void Point::Y (int *_y*) `[inline]`

int Point::Y () const `[inline]`

int Point::operator!= (const Point & *p*) const `[inline]`

int Point::operator< (const Point & *p*) const `[inline]`

int Point::operator<= (const Point & *p*) const `[inline]`

int Point::operator== (const Point & *p*) const `[inline]`

int Point::operator> (const Point & *p*) const `[inline]`

int Point::operator>= (const Point & *p*) const `[inline]`

Friends And Related Function Documentation

ostream & operator<< (std::ostream & *out*, const Point & *p*) `[friend]`

istream & operator>> (istream & *in*, Point & *p*) `[friend]`

Member Data Documentation

int Point::col

int Point::ligne

The documentation for this class was generated from the following file:

- **Point.hpp**

EspaceCalibrage::Point Struct Reference

```
#include <EspaceCalibrage.h>
```

Public Methods

- int **EstHorsImage** () const

Public Attributes

- **DataType x**
- **DataType y**

Member Function Documentation

int EspaceCalibrage::Point::EstHorsImage () const `[inline]`

Member Data Documentation

DataType EspaceCalibrage::Point::x

DataType EspaceCalibrage::Point::y

The documentation for this struct was generated from the following file:

- **EspaceCalibrage.h**

PointColore Struct Reference

Public Attributes

- **Point position**
- double **hue**
- double **saturation**
- double **intensity**
- unsigned long **indice**

Member Data Documentation

double PointColore::hue

unsigned long PointColore::indice

double PointColore::intensity

Point PointColore::position

double PointColore::saturation

The documentation for this struct was generated from the following file:

- **ExtractionDeCible.cpp**

EspaceCalibrage::PointXYZ Struct Reference

```
#include <EspaceCalibrage.h>
```

Public Attributes

- **DataType x**
- **DataType y**
- **DataType z**

Member Data Documentation

DataType EspaceCalibrage::PointXYZ::x

DataType EspaceCalibrage::PointXYZ::y

DataType EspaceCalibrage::PointXYZ::z

The documentation for this struct was generated from the following file:

- **EspaceCalibrage.h**

FormatPPM::ppmHeader Struct Reference

```
#include <FormatPPM.hpp>
```

Public Attributes

- **UBYTE4 width**
- **UBYTE4 height**
- **UBYTE4 maxvalue**

Member Data Documentation

UBYTE4 FormatPPM::ppmHeader::height

UBYTE4 FormatPPM::ppmHeader::maxvalue

UBYTE4 FormatPPM::ppmHeader::width

The documentation for this struct was generated from the following file:

- **FormatPPM.hpp**

ImageRGB::PRep Class Reference

Public Methods

- int **operator**< (const PRep &b) const
- int **operator**> (const PRep &b) const
- int **operator**== (const PRep &b) const

Public Attributes

- **Point pt**
- int **rep**

Member Function Documentation

int ImageRGB::PRep::operator< (const PRep & *b*) const `[inline]`

int ImageRGB::PRep::operator== (const PRep & *b*) const `[inline]`

int ImageRGB::PRep::operator> (const PRep & *b*) const `[inline]`

Member Data Documentation

Point ImageRGB::PRep::pt

int ImageRGB::PRep::rep

The documentation for this class was generated from the following file:

- **ImageRGB.hpp**

ImageGris::PRep Class Reference

Public Methods

- int **operator**< (const PRep &b) const
- int **operator**> (const PRep &b) const
- int **operator**== (const PRep &b) const

Public Attributes

- **Point pt**
- int **rep**

Member Function Documentation

int ImageGris::PRep::operator< (const PRep & *b*) const `[inline]`

int ImageGris::PRep::operator== (const PRep & *b*) const `[inline]`

int ImageGris::PRep::operator> (const PRep & *b*) const `[inline]`

Member Data Documentation

Point ImageGris::PRep::pt

int ImageGris::PRep::rep

The documentation for this class was generated from the following file:

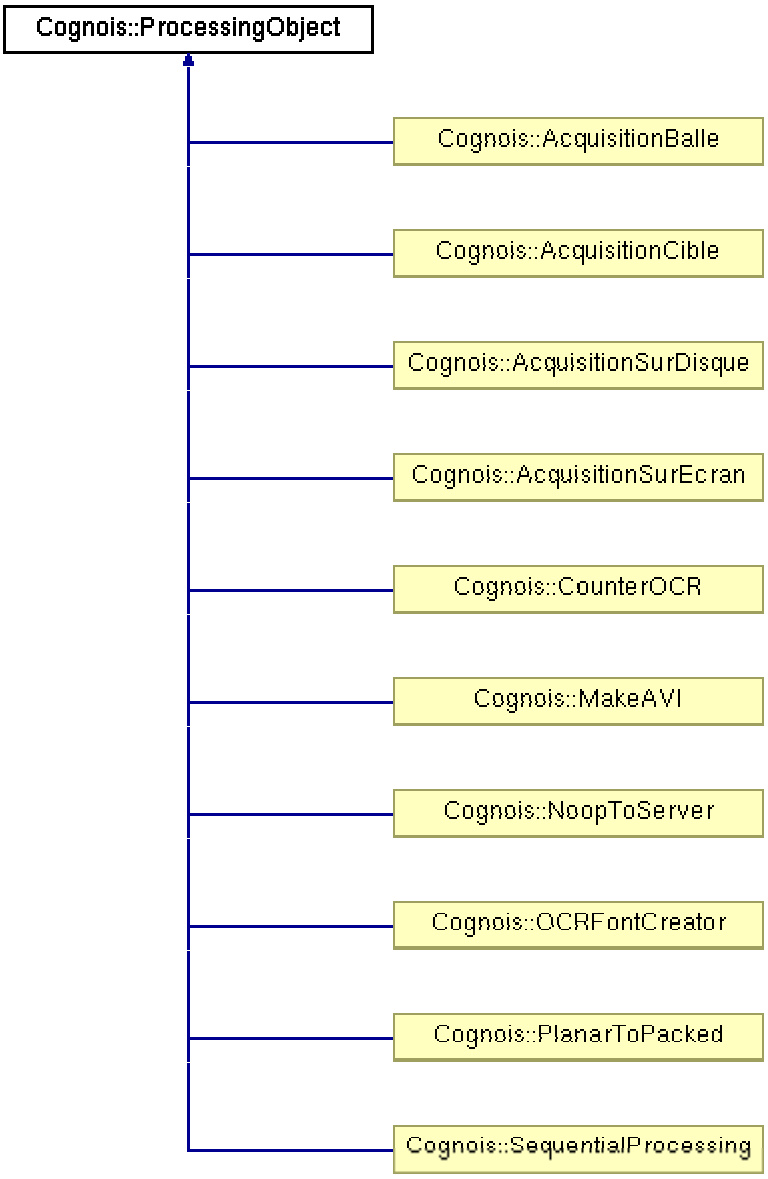
- **ImageGris.hpp**

Cognois::ProcessingObject Class Reference

Virtual class for user-supplied processing objects.

`#include <ProcessingObject.h>`

Inheritance diagram for Cognois::ProcessingObject:



Public Methods

- virtual **~ProcessingObject** ()
  Virtual destructor.

- virtual void **Process** (MIL_ID &inMilImage, unsigned long &inFrameID, unsigned long &inTimeStamp)=0

Process an image with know frameID and timeStamp.

- virtual void **Activate** ()
  *Activate the **ProcessingObject** (p.349) (initialization for a sequence).*

- virtual void **Deactivate** ()
  *Deactivate the **ProcessingObject** (p.349) (end of a sequence).*

Detailed Description

Virtual class for user-supplied processing objects.

Constructor & Destructor Documentation

Cognois::ProcessingObject::~ProcessingObject () `[inline, virtual]`

Virtual destructor.

Member Function Documentation

void Cognois::ProcessingObject::Activate () `[inline, virtual]`

Activate the **ProcessingObject** (*p.349*) (initialization for a sequence).

Reimplemented in **Cognois::SequentialProcessing** (*p.358*).

void Cognois::ProcessingObject::Deactivate () `[inline, virtual]`

Deactivate the **ProcessingObject** (*p.349*) (end of a sequence).

Reimplemented in **Cognois::SequentialProcessing** (*p.358*).

void Cognois::ProcessingObject::Process (MIL_ID & *_milImage*, unsigned long & *_frameID*, unsigned long & *_timeStamp*) `[pure virtual]`

Process an image with know frameID and timeStamp.

Parameters:

inMilImage is a MIL image identifier.
inFrameID is the frame identifier associated with the image.
inTimeStamp is the time, in milliseconds, associated with the image.

Reimplemented in **Cognois::AcquisitionBalle** (*p.17*), **Cognois::AcquisitionCible** (*p.20*), **Cognois::AcquisitionSurDisque** (*p.26*), **Cognois::AcquisitionSurEcran** (*p.28*), **Cognois::CounterOCR** (*p.85*), **Cognois::MakeAVI** (*p.261*), **Cognois::NoopToServer** (*p.282*), **Cognois::OCRFontCreator** (*p.285*), **Cognois::PlanarToPacked** (*p.312*), and **Cognois::SequentialProcessing** (*p.359*).

The documentation for this class was generated from the following file:

- **ProcessingObject.h**

FormatSunRaster::rasHeader Struct Reference

```
#include <FormatSunRaster.hpp>
```

Public Attributes

- **UBYTE4 magic**
- **UBYTE4 width**
- **UBYTE4 height**
- **UBYTE4 depth**
- **UBYTE4 length**
- **UBYTE4 type**
- **UBYTE4 maptype**
- **UBYTE4 maplength**

Member Data Documentation

UBYTE4 FormatSunRaster::rasHeader::depth

UBYTE4 FormatSunRaster::rasHeader::height

UBYTE4 FormatSunRaster::rasHeader::length

UBYTE4 FormatSunRaster::rasHeader::magic

UBYTE4 FormatSunRaster::rasHeader::maplength

UBYTE4 FormatSunRaster::rasHeader::maptype

UBYTE4 FormatSunRaster::rasHeader::type

UBYTE4 FormatSunRaster::rasHeader::width

The documentation for this struct was generated from the following file:

- **FormatSunRaster.hpp**

RGBQUADsd Struct Reference

```
#include <Picture.h>
```

Public Attributes

- **BYTE rgbBlue**
- **BYTE rgbGreen**
- **BYTE rgbRed**
- **BYTE rgbReserved**

Member Data Documentation

BYTE RGBQUADsd::rgbBlue

BYTE RGBQUADsd::rgbGreen

BYTE RGBQUADsd::rgbRed

BYTE RGBQUADsd::rgbReserved

The documentation for this struct was generated from the following file:

- **Picture.h**

JpegEncoder::Scan Struct Reference

Public Attributes

- unsigned long **component_mask**
- unsigned int **spectral_selection_start**
- unsigned int **spectral_selection_end**
- unsigned int **successive_approximation**
- int **successive_approximation_high**
- int **successive_approximation_low**

Member Data Documentation

unsigned long JpegEncoder::Scan::component_mask

unsigned int JpegEncoder::Scan::spectral_selection_end

unsigned int JpegEncoder::Scan::spectral_selection_start

unsigned int JpegEncoder::Scan::successive_approximation

int JpegEncoder::Scan::successive_approximation_high

int JpegEncoder::Scan::successive_approximation_low

The documentation for this struct was generated from the following file:

- **jpegenco.h**

Cognois::SpaceCarving::SCVolume Class Reference

Volume for space carving.

`#include <SCVolume.h>`

Public Types

- enum **Type** { **eType1**, **eType2** }

Public Methods

- **SCVolume** (const **CameraModel** &inCamera, double inZmax, double inVoxelSize, **Matrice** inImageSize="[640;480]", **Type** inType=eType2)
  Build an initial inType volume with known calibration parameters (all in mm).

Private Methods

- void **carveInitialVolume** (double inZmax, **Matrice** inImageSize, **Type** inType)
- void **getInitialVolumeVertices** (double inZmax, **Matrice** &outVo, **Matrice** &outVf) const

Private Attributes

- **CameraModel mCamera**
- double **mVoxelSize**
- **std::vector**<**std::vector**<**std::vector**<bool> > > **mImage**

Detailed Description

Volume for space carving.

Member Enumeration Documentation

enum Cognois::SpaceCarving::SCVolume::Type

Enumeration values:

eType1
eType2

Constructor & Destructor Documentation

Cognois::SpaceCarving::SCVolume::SCVolume (const CameraModel & *inCamera*, double *inZmax*, double *inVoxelSize*, Matrice *inImageSize* = "[640;480]", Type *inType* = eType2)

Build an initial inType volume with known calibration parameters (all in mm).

Parameters:

inCamera is the calibration parameters of the system (all in mm)
inZmax is the maximum Z distance for each camera (mm)
inVoxelSize is the size of the voxels (mm)
inImageSize is the image resolution (pixels)
inType is the type of volume (1 camera intersection or 2 cameras intersection)

Member Function Documentation

void Cognois::SpaceCarving::SCVolume::carveInitialVolume (double *inZmax*, Matrice *inImageSize*, Type *inType*) `[private]`

Parameters:

inZmax is the maximum Z distance for each camera (mm)
inImageSize is the image resolution (pixels)
inType is the type of volume (1 camera intersection or 2 cameras intersection)

void Cognois::SpaceCarving::SCVolume::getInitialVolumeVertices (double *inZmax*, Matrice & *outVo*, Matrice & *outVf*) const `[private]`

Parameters:

inZmax is the maximum Z distance for each camera (mm)
outVo is the origin of the volume
outVf is the opposing vertex in the volume cube

Member Data Documentation

CameraModel Cognois::SpaceCarving::SCVolume::mCamera `[private]`

std::vector< std::vector< std::vector< bool > > > Cognois::SpaceCarving::SCVolume::mImage<std::vector<std::vector<bool> > > `[private]`

double Cognois::SpaceCarving::SCVolume::mVoxelSize `[private]`

The documentation for this class was generated from the following files:

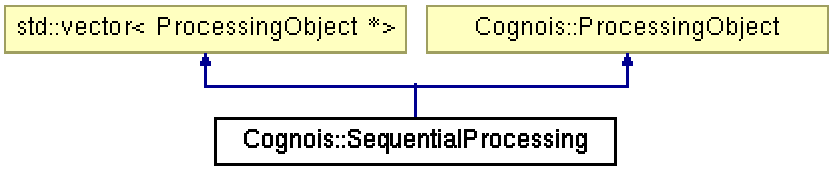
- **SCVolume.h**
- **SCVolume.cpp**

Cognois::SequentialProcessing Class Reference

Perform a sequence of image processing.

`#include <SequentialProcessing.h>`

Inheritance diagram for Cognois::SequentialProcessing:



Public Methods

- void **Process** (MIL_ID &_milImage, unsigned long &_frameID, unsigned long &_timeStamp)
  Process an image with know frameID and timeStamp.

- void **Activate** ()
  *Activate the **ProcessingObject** (p.349) (initialization for a sequence).*

- virtual void **Deactivate** ()
  *Deactivate the **ProcessingObject** (p.349) (end of a sequence).*

Detailed Description

Perform a sequence of image processing.

The user-supplied processing object are applied to an image according to their order in a vector of **ProcessingObject** (*p.349*) pointers.

Member Function Documentation

void Cognois::SequentialProcessing::Activate () `[inline, virtual]`

Activate the **ProcessingObject** (*p.349*) (initialization for a sequence).

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

void Cognois::SequentialProcessing::Deactivate () `[inline, virtual]`

Deactivate the **ProcessingObject** (*p.349*) (end of a sequence).

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

void Cognois::SequentialProcessing::Process (MIL_ID & *_milImage*, unsigned long & *_frameID*, unsigned long & *_timeStamp*) `[inline, virtual]`

Process an image with know frameID and timeStamp.

Parameters:

inMilImage is a MIL image identifier.
inFrameID is the frame identifier associated with the image.
inTimeStamp is the time, in milliseconds, associated with the image.

Reimplemented from **Cognois::ProcessingObject** (*p.350*).

The documentation for this class was generated from the following file:

- **SequentialProcessing.h**

StatisticsR Class Template Reference

`#include <Statistics.h>`

Public Methods

- T **Mean** (const **std::vector**< T > &) const
- T **Median** (const **std::vector**< T > &) const
- T **StandardDeviation** (const **std::vector**< T > &) const
- T **StandardDeviation** (const **std::vector**< T > &, const T &) const
- T **RobustStandardDeviation** (const **std::vector**< T > &) const
- T **RobustStandardDeviation** (const T &) const
- T **MedianDeviation** (const **std::vector**< T > &) const
- T **MedianDeviation** (const **std::vector**< T > &, const T &) const
- T **Max** (const **std::vector**< T > &) const

template<class T> class StatisticsR

Member Function Documentation

template<class T> T StatisticsR< T >::Max (const std::vector< T > & *x*) const `[inline]`

template<class T> T StatisticsR< T >::Mean (const std::vector< T > & *x*) const `[inline]`

template<class T> T StatisticsR< T >::Median (const std::vector< T > & *x*) const `[inline]`

template<class T> T StatisticsR< T >::MedianDeviation (const std::vector< T > & *x*, const T & *median*) const `[inline]`

template<class T> T StatisticsR< T >::MedianDeviation (const std::vector< T > & *x*) const `[inline]`

template<class T> T StatisticsR< T >::RobustStandardDeviation (const T & *median*) const `[inline]`

template<class T> T StatisticsR< T >::RobustStandardDeviation (const std::vector< T > & *x*) const `[inline]`

template<class T> T StatisticsR< T >::StandardDeviation (const std::vector< T > & *x*, const T & *mean*) const `[inline]`

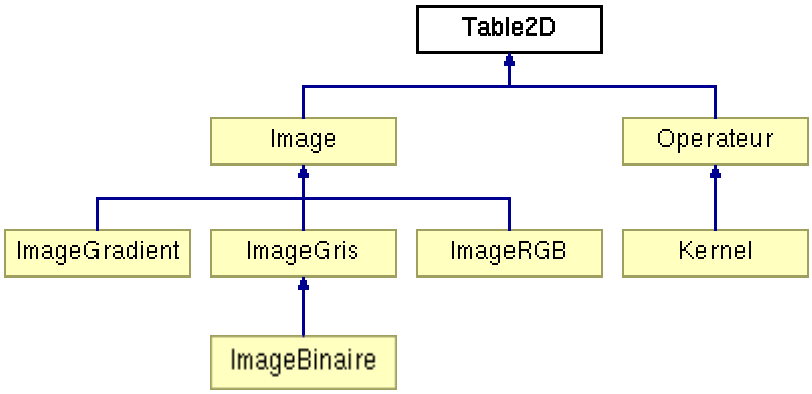
template<class T> T StatisticsR< T >::StandardDeviation (const std::vector< T > & *x*) const `[inline]`

The documentation for this class was generated from the following file:

- **Statistics.h**

Table2D Class Reference

Inheritance diagram for Table2D:

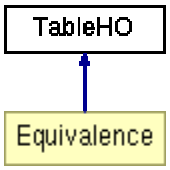


The documentation for this class was generated from the following file:

- **Image.hpp**

TableHO Class Reference

Inheritance diagram for TableHO:



The documentation for this class was generated from the following file:

- **Equivalence.hpp**

TesterContour Class Reference

`#include <TesterContour.hpp>`

Public Methods

- **TesterContour** (const **TstParam** &par, double _tp, double _ts, **ImageGris** *ig=0, **Contour** *ct=0)
- void **EffectuerTest** (double variance=0, double saltpepper=0)
- double **Note** () const
- void **Muet** (int _muet)
- double **NotePoints** () const
- double **NoteSegments** () const

Protected Methods

- void **TesterPoints** (**Liste**< **Point** >,**Liste**< **Point** >)
- void **TesterSegments** (**Contour**, **Contour**)
- void **Rapport** (const **Contour** &,const **Contour** &) const

Protected Attributes

- **TstParam param**
- double **tolpoints**
- double **tolsegments**
- double **pac**
- double **pae**
- double **pat**
- double **pdm**
- double **sac**
- double **sae**
- double **sat**
- double **sdm**
- double **slm**
- **ImageGris*** **imageorg**
- **Contour*** **compedges**
- int **muet**

Constructor & Destructor Documentation

TesterContour::TesterContour (const TstParam & *par*, double *_tp*, double *_ts*, ImageGris * *ig* = 0, Contour * *ct* = 0) `[inline]`

Member Function Documentation

void TesterContour::EffectuerTest (double *variance* = 0, double *saltpepper* = 0)

void TesterContour::Muet (int *_muet*) `[inline]`

double TesterContour::Note () const `[inline]`

double TesterContour::NotePoints () const `[inline]`

double TesterContour::NoteSegments () const `[inline]`

void TesterContour::Rapport (const Contour &, const Contour &) const `[protected]`

void TesterContour::TesterPoints (Liste< Point >, Liste< Point >) `[protected]`

void TesterContour::TesterSegments (Contour, Contour) `[protected]`

Member Data Documentation

Contour * TesterContour::compedges `[protected]`

ImageGris * TesterContour::imageorg `[protected]`

int TesterContour::muet `[protected]`

double TesterContour::pac `[protected]`

double TesterContour::pae `[protected]`

TstParam TesterContour::param `[protected]`

double TesterContour::pat `[protected]`

double TesterContour::pdm `[protected]`

double TesterContour::sac `[protected]`

double TesterContour::sae `[protected]`

double TesterContour::sat `[protected]`

double TesterContour::sdm `[protected]`

double TesterContour::slm `[protected]`

double TesterContour::tolpoints `[protected]`

double TesterContour::tolsegments `[protected]`

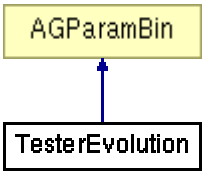
The documentation for this class was generated from the following file:

- **TesterContour.hpp**

TesterEvolution Class Reference

`#include <TesterEvolution.hpp>`

Inheritance diagram for TesterEvolution:



Public Methods

- **TesterEvolution** (**TstParam** &, double pt, double sg, double _v=0, double _s=0, int _tp=10, double _pm=0.01, double _pc=0.1)
- double **Fonction** (const Vecteur &x)

Protected Attributes

- **TstParam par**
- double **variance**
- double **saltpepper**
- double **tolpt**
- double **tolsg**
- **TableHO<TstParamNote> notes**
- **ImageGris image**
- **Contour compedges**

Constructor & Destructor Documentation

TesterEvolution::TesterEvolution (TstParam & *_p*, double *pt*, double *sg*, double *_v* = 0, double *_s* = 0, int *_tp* = 10, double *_pm* = 0.01, double *_pc* = 0.1) `[inline]`

Member Function Documentation

double TesterEvolution::Fonction (const Vecteur & *x*) `[inline]`

Member Data Documentation

Contour TesterEvolution::compedges `[protected]`

ImageGris TesterEvolution::image `[protected]`

TableHO< TstParamNote > TesterEvolution::notes<TstParamNote> `[protected]`

TstParam TesterEvolution::par `[protected]`

double TesterEvolution::saltpepper `[protected]`

double TesterEvolution::tolpt `[protected]`

double TesterEvolution::tolsg `[protected]`

double TesterEvolution::variance `[protected]`

The documentation for this class was generated from the following file:

- **TesterEvolution.hpp**

TstParam Class Reference

```
#include <TesterContour.hpp>
```

Public Attributes

- Chaine **entree_image**
- Chaine **entree_comparaison**
- Chaine **sortie_contours**
- Chaine **sortie_resultats**
- int **f**
- int **g**
- int **v**
- int **b**
- int **i**
- int **t**
- double **m**

Member Data Documentation

int TstParam::b

Chaine TstParam::entree_comparaison

Chaine TstParam::entree_image

int TstParam::f

int TstParam::g

int TstParam::i

double TstParam::m

Chaine TstParam::sortie_contours

Chaine TstParam::sortie_resultats

int TstParam::t

int TstParam::v

The documentation for this class was generated from the following file:

- **TesterContour.hpp**

TstParamNote Class Reference

`#include <TesterEvolution.hpp>`

Public Methods

- int **operator==** (const TstParamNote &tpn) const

Public Attributes

- **TstParam par**
- double **note**

Member Function Documentation

int TstParamNote::operator== (const TstParamNote & *tpn*) const `[inline]`

Member Data Documentation

double TstParamNote::note

TstParam TstParamNote::par

The documentation for this class was generated from the following file:

- **TesterEvolution.hpp**

Vardesc Struct Reference

```
#include <F2C.H>
```

Public Attributes

- char* **name**
- char* **addr**
- **ftnlen*** **dims**
- int **type**

Member Data Documentation

char * Vardesc::addr

ftnlen * Vardesc::dims

char * Vardesc::name

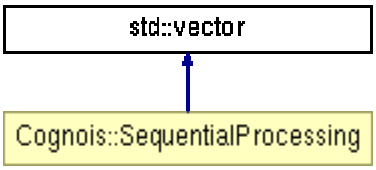
int Vardesc::type

The documentation for this struct was generated from the following file:

- **F2C.H**

std::vector Class Reference

Inheritance diagram for std::vector:



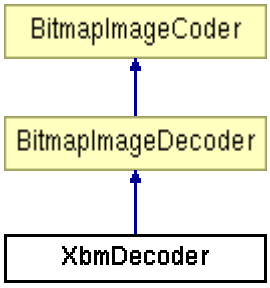
The documentation for this class was generated from the following file:

- **SequentialProcessing.h**

XbmDecoder Class Reference

`#include <xbmdecod.h>`

Inheritance diagram for XbmDecoder:



Public Methods

- **XbmDecoder** ()
- **XbmDecoder** (const XbmDecoder &)
- virtual **~XbmDecoder** ()
- XbmDecoder& **operator=** (const XbmDecoder &)
- void **ReadImage** (istream &, **BitmapImage** &)
- int **GetHotSpotX** () const
- int **GetHotSpotY** () const
- Chaine **GetImageName** () const

Private Methods

- void **Initialize** ()
- void **DoCopy** (const XbmDecoder &)
- int **NextChar** (istream &)
- int **NextToken** (istream &)

Private Attributes

- int **hot_spot_x**
- int **hot_spot_y**
- Chaine **image_name**
- int **last_char**
- unsigned int **lexical_value**

Constructor & Destructor Documentation

XbmDecoder::XbmDecoder ()

XbmDecoder::XbmDecoder (const XbmDecoder & *source*)

XbmDecoder::~XbmDecoder () `[virtual]`

Member Function Documentation

void XbmDecoder::DoCopy (const XbmDecoder & *source*) `[private]`

int XbmDecoder::GetHotSpotX () const

int XbmDecoder::GetHotSpotY () const

Chaine XbmDecoder::GetImageName () const

void XbmDecoder::Initialize () `[private]`

Reimplemented from **BitmapImageCoder** (*p.38*).

int XbmDecoder::NextChar (istream & *strm*) `[private]`

int XbmDecoder::NextToken (istream & *strm*) `[private]`

void XbmDecoder::ReadImage (istream & *strm*, BitmapImage & *image*) `[virtual]`

Reimplemented from **BitmapImageDecoder** (*p.39*).

XbmDecoder & XbmDecoder::operator= (const XbmDecoder & *source*)

Member Data Documentation

int XbmDecoder::hot_spot_x `[private]`

int XbmDecoder::hot_spot_y `[private]`

Chaine XbmDecoder::image_name `[private]`

int XbmDecoder::last_char `[private]`

unsigned int XbmDecoder::lexical_value `[private]`

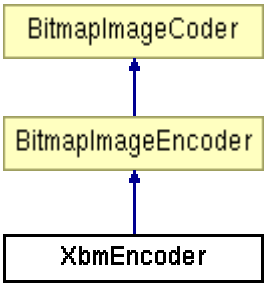
The documentation for this class was generated from the following files:

- **xbmdecod.h**
- **xbmdecod.cpp**

XbmEncoder Class Reference

`#include <xbmencod.h>`

Inheritance diagram for XbmEncoder:



Public Methods

- **XbmEncoder** ()
- **XbmEncoder** (const XbmEncoder &)
- virtual **~XbmEncoder** ()
- XbmEncoder& **operator=** (const XbmEncoder &)
- void **WriteImage** (ostream &, **BitmapImage** &)
- int **GetHotSpotX** () const
- void **SetHotSpotX** (int value)
- int **GetHotSpotY** () const
- void **SetHotSpotY** (int value)
- void **SetImageName** (const Chaine &)
- Chaine **GetImageName** () const

Private Methods

- void **Initialize** ()
- void **DoCopy** (const XbmEncoder &)

Private Attributes

- Chaine **image_name**
- int **hot_spot_x**
- int **hot_spot_y**

Constructor & Destructor Documentation

XbmEncoder::XbmEncoder ()

XbmEncoder::XbmEncoder (const XbmEncoder & *source*)

XbmEncoder::~XbmEncoder () `[virtual]`

Member Function Documentation

void XbmEncoder::DoCopy (const XbmEncoder & *source*) `[private]`

int XbmEncoder::GetHotSpotX () const

int XbmEncoder::GetHotSpotY () const

Chaine XbmEncoder::GetImageName () const

void XbmEncoder::Initialize () `[private]`

Reimplemented from **BitmapImageCoder** (*p.38*).

void XbmEncoder::SetHotSpotX (int *value*)

void XbmEncoder::SetHotSpotY (int *value*)

void XbmEncoder::SetImageName (const Chaine & *value*)

void XbmEncoder::WriteImage (ostream & *strm*, BitmapImage & *timage*) `[virtual]`

Reimplemented from **BitmapImageEncoder** (*p.40*).

XbmEncoder & XbmEncoder::operator= (const XbmEncoder & *source*)

Member Data Documentation

int XbmEncoder::hot_spot_x `[private]`

int XbmEncoder::hot_spot_y `[private]`

Chaine XbmEncoder::image_name `[private]`

The documentation for this class was generated from the following files:

- **xbmencod.h**
- **xbmencod.cpp**

FormatXPM::xpmColor Struct Reference

```
#include <FormatXPM.hpp>
```

Public Attributes

- **UBYTE4 indice**
- **CouleurRGB couleur**

Member Data Documentation

CouleurRGB FormatXPM::xpmColor::couleur

UBYTE4 FormatXPM::xpmColor::indice

The documentation for this struct was generated from the following file:

- **FormatXPM.hpp**

FormatXPM::xpmHeader Struct Reference

```
#include <FormatXPM.hpp>
```

Public Attributes

- **UBYTE4 format**
- **UBYTE4 width**
- **UBYTE4 height**
- **UBYTE4 ncolors**
- **UBYTE4 pixel**

Member Data Documentation

UBYTE4 FormatXPM::xpmHeader::format

UBYTE4 FormatXPM::xpmHeader::height

UBYTE4 FormatXPM::xpmHeader::ncolors

UBYTE4 FormatXPM::xpmHeader::pixel

UBYTE4 FormatXPM::xpmHeader::width

The documentation for this struct was generated from the following file:

- **FormatXPM.hpp**

FormatXWD::xwdColor Struct Reference

```
#include <FormatXWD.hpp>
```

Public Attributes

- **UBYTE4 number**
- **UBYTE2 red**
- **UBYTE2 green**
- **UBYTE2 blue**
- **UBYTE1 flags**
- **UBYTE1 pad**

Member Data Documentation

UBYTE2 FormatXWD::xwdColor::blue

UBYTE1 FormatXWD::xwdColor::flags

UBYTE2 FormatXWD::xwdColor::green

UBYTE4 FormatXWD::xwdColor::number

UBYTE1 FormatXWD::xwdColor::pad

UBYTE2 FormatXWD::xwdColor::red

The documentation for this struct was generated from the following file:

- **FormatXWD.hpp**

FormatXWD::xwdHeader Struct Reference

```
#include <FormatXWD.hpp>
```

Public Attributes

- **UBYTE4 hsize**
- **UBYTE4 version**
- **UBYTE4 format**
- **UBYTE4 depth**
- **UBYTE4 width**
- **UBYTE4 height**
- **UBYTE4 xoffset**
- **UBYTE4 byteorder**
- **UBYTE4 bitmapunit**
- **UBYTE4 bitorder**
- **UBYTE4 bitmappad**
- **UBYTE4 bitsperpixel**
- **UBYTE4 bytesperline**
- **UBYTE4 visualclass**
- **UBYTE4 zredmask**
- **UBYTE4 zgreenmask**
- **UBYTE4 zbluemask**
- **UBYTE4 bitsperrgb**
- **UBYTE4 colorentries**
- **UBYTE4 ncolors**
- **UBYTE4 windowwidth**
- **UBYTE4 windowheight**
- **BYTE4 windowx**
- **BYTE4 windowy**
- **UBYTE4 windowborder**

Member Data Documentation

UBYTE4 FormatXWD::xwdHeader::bitmappad

UBYTE4 FormatXWD::xwdHeader::bitmapunit

UBYTE4 FormatXWD::xwdHeader::bitorder

UBYTE4 FormatXWD::xwdHeader::bitsperpixel

UBYTE4 FormatXWD::xwdHeader::bitsperrgb

UBYTE4 FormatXWD::xwdHeader::byteorder

UBYTE4 FormatXWD::xwdHeader::bytesperline

UBYTE4 FormatXWD::xwdHeader::colorentries

UBYTE4 FormatXWD::xwdHeader::depth

UBYTE4 FormatXWD::xwdHeader::format

UBYTE4 FormatXWD::xwdHeader::height

UBYTE4 FormatXWD::xwdHeader::hsize

UBYTE4 FormatXWD::xwdHeader::ncolors

UBYTE4 FormatXWD::xwdHeader::version

UBYTE4 FormatXWD::xwdHeader::visualclass

UBYTE4 FormatXWD::xwdHeader::width

UBYTE4 FormatXWD::xwdHeader::windowborder

UBYTE4 FormatXWD::xwdHeader::windowheight

UBYTE4 FormatXWD::xwdHeader::windowwidth

BYTE4 FormatXWD::xwdHeader::windowx

BYTE4 FormatXWD::xwdHeader::windowy

UBYTE4 FormatXWD::xwdHeader::xoffset

UBYTE4 FormatXWD::xwdHeader::zbluemask

UBYTE4 FormatXWD::xwdHeader::zgreenmask

UBYTE4 FormatXWD::xwdHeader::zredmask

The documentation for this struct was generated from the following file:

- **FormatXWD.hpp**

Online Processing Library File Documentation

Acquisition.cpp File Reference

```
#include "Acquisition.h"
#include <iostream.h>
```

Defines

- #define **NO_FRAME**
- #define **METEOR_II_CL**

Functions

- long MFTYPE **GrabStartHookHandler** (long HookType, MIL_ID EventId, void MPTYPE *UserDataPtr)
  *DÉFINITION DES PROCÉDURES ////////////

Function Documentation

long MFTYPE FieldEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[static]`

long MFTYPE FieldEvenEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[static]`

long MFTYPE FieldOddEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[static]`

long MFTYPE FrameEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[static]`

long MFTYPE FrameStartHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[static]`

long MFTYPE GrabEndHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[static]`

long MFTYPE GrabStartHookHandler (long *HookType*, MIL_ID *EventId*, void MPTYPE * *UserDataPtr*) `[static]`

DÉFINITION DES PROCÉDURES ////////////

Acquisition.h File Reference

```
#include <mil.h>
#include <vector>
#include "AcquisitionImageListe.h"
```

Compounds

- class **Acquisition**

AcquisitionBalle.cpp File Reference

```
#include "AcquisitionBalle.h"
#include "Cognois.h"
```

AcquisitionBalle.h File Reference

```
#include "FrameSetClient.h"
#include "OnlineProcessing.h"
#include "ProcessingObject.h"
#include "cv.h"
#include "ipl.h"
#include "HighGUI.h"
#include "winbase.h"
#include "BackgroundSub.h"
#include "BlobExtractor.h"
#include "ColorAnalysis.h"
#include "FindBall.h"
#include "Filename.h"
#include "OpticalFlow.h"
#include "EdgeDetector.h"
#include "VariousFunctions.h"
#include "stdlib.h"
#include "string.h"
```

Namespaces

- namespace **Cognois**

Defines

- #define **COLOR_ANALYSIS**
- #define **SERVER_ON**

Define Documentation

#define COLOR_ANALYSIS

Value:

#define SERVER_ON

AcquisitionCible.cpp File Reference

```
#include "CalibrationClient.h"
#include "AcquisitionCible.h"
```

AcquisitionCible.h File Reference

```
#include "CalibrationClient.h"
#include "ProcessingObject.h"
#include <iostream>
#include "ExtractionDeCible.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::AcquisitionCible**
  Find a calibration pattern in an image.

AcquisitionImage.cpp File Reference

```
#include "AcquisitionImage.h"
```

AcquisitionImage.h File Reference

```
#include <mil.h>
#include <wtypes.h>
```

Compounds

- class **AcquisitionImage**

AcquisitionImageListe.cpp File Reference

```
#include "AcquisitionImageListe.h"
```

AcquisitionImageListe.h File Reference

```
#include "AcquisitionImage.h"
```

Compounds

- class **AcquisitionImageListe**

AcquisitionSurDisque.cpp File Reference

```
#include "AcquisitionSurDisque.h"
#include "Cognois.h"
```

AcquisitionSurDisque.h File Reference

```
#include "ProcessingObject.h"
#include <iostream>
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::AcquisitionSurDisque**
  Export grabbed images to files.

AcquisitionSurEcran.cpp File Reference

```
#include "AcquisitionSurEcran.h"
```

AcquisitionSurEcran.h File Reference

```
#include "ProcessingObject.h"
#include "MilSys.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::AcquisitionSurEcran**
  Display of grabbed images.

bitimage.cpp File Reference

```
#include "bitimage.h"
#include "grexcept.h"
```

Functions

- int **SQUARE** (int xx)
- unsigned int **RoundRow** (unsigned int width)

Function Documentation

unsigned int RoundRow (unsigned int *width*) `[inline, static]`

int SQUARE (int *xx*) `[inline, static]`

bitimage.h File Reference

```
#include <iostream>
#include <string>
#include "datatype.h"
```

Compounds

- class **BitmapImage**
- class **BitmapImageCoder**
- class **BitmapImageDecoder**
- class **BitmapImageEncoder**
- struct **BitmapImage::ColorArea**
- struct **BitmapImage::ColorMapEntry**
- struct **BitmapImage::ColorUsage**
- struct **BitmapImage::ColorUsageTable**

Typedefs

- typedef void (* **PROGRESSFUNCTION**)(**BitmapImageCoder** &coder, void *data, unsigned int currentpass, unsigned int passcount, unsigned int progress, bool &cancel)
- typedef void (* **IMAGEPROGRESSFUNCTION**)(**BitmapImage** &image, void *data, unsigned int currentpass, unsigned int passcount, unsigned int progress, bool &cancel)

Typedef Documentation

typedef void(* IMAGEPROGRESSFUNCTION)(BitmapImage &image, void *data, unsigned int currentpass, unsigned int passcount, unsigned int progress, bool &cancel)

typedef void(* PROGRESSFUNCTION)(BitmapImageCoder &coder, void *data, unsigned int currentpass, unsigned int passcount, unsigned int progress, bool &cancel)

BlobColoring.cpp File Reference

```
#include "BlobColoring.h"
```

Defines

- #define **NULL** 0
- #define **ERROR_INVALID_PARAMETER** 897234
- #define **ERROR_SUCCESS** 0
- #define **ERROR_NO_DATA** 234
- #define **ERROR_NOT_ENOUGH_MEMORY** 42

Define Documentation

#define ERROR_INVALID_PARAMETER 897234

#define ERROR_NOT_ENOUGH_MEMORY 42

#define ERROR_NO_DATA 234

#define ERROR_SUCCESS 0

#define NULL 0

BlobColoring.h File Reference

Compounds

- class **BlobColoring**
- struct **BlobInfo**

Defines

- #define **AFX_BLOBCOLORING_H__5E107B28_D49C_4885_A5D0_6FF9F4C086B4__INCLUDED_**

Define Documentation

#define AFX_BLOBCOLORING_H__5E107B28_D49C_4885_A5D0_6FF9F4C086B4__INCLUDED_

Value:

Calibration.cpp File Reference

```
#include "Calibration.h"
#include <fstream>
#include <stdlib.h>
#include <iostream>
#include <math.h>
#include "Erreur.hpp"
#include "Matrice.h"
#include <vector>
#include <algorithm>
```

Namespaces

- namespace **std**

Defines

- #define **MATLAB_HOME** "V:\\Donnees\\cognois\\echange\\"
- #define **MATLAB_COMMAND** "C:\\programs\\matlab6\\bin\\win32\\matlab"
- #define **ZHANG_HOME** "V:\\Usagers\\sdrouin\\Cognois\\Calibrage\\EasyCalib\\"
- #define **ZHANG_COMMAND** "EasyCalib"

Define Documentation

#define MATLAB_COMMAND "C:\\programs\\matlab6\\bin\\win32\\matlab"

#define MATLAB_HOME "V:\\Donnees\\cognois\\echange\\"

#define ZHANG_COMMAND "EasyCalib"

#define ZHANG_HOME "V:\\Usagers\\sdrouin\\Cognois\\Calibrage\\EasyCalib\\"

Calibration.h File Reference

```
#include "EspaceCalibrage.h"
#include "Cognois.h"
#include "Matrice.h"
#include "CameraModel.h"
```

Namespaces

- namespace **Cognois**

CalibrationClient.cpp File Reference

```
#include "CalibrationClient.h"
#include "SimpleXmlParser.h"
```

CalibrationClient.h File Reference

```
#include "FrameSetClient.h"
#include "CameraModel.h"
#include "EspaceCalibrage.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::CalibrationClient**
  Read and write calibration data on the server.

CameraClientController.cpp File Reference

```
#include "CameraClientController.h"
#include "Cognois.h"
#include <iostream>
```

CameraClientController.h File Reference

```
#include "FrameSetClient.h"
#include "OnlineProcessing.h"
#include <vector>
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::CameraClientController**
  Get the current mode from the server and use the appropriate processing object.

Defines

- #define **COGNOIS_CONTROL_NOOP** "NOOP"
- #define **COGNOIS_CONTROL_TERMINATE** "TERMINATE"

Define Documentation

#define COGNOIS_CONTROL_NOOP "NOOP"

#define COGNOIS_CONTROL_TERMINATE "TERMINATE"

CameraModel.cpp File Reference

```
#include "CameraModel.h"
#include "Calibration.h"
#include <fstream>
```

CameraModel.h File Reference

```
#include "Matrice.h"
#include "Cognois.h"
#include "EspaceCalibrage.h"
#include <iostream.h>
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::CameraModel**
  Calibration *(p.46) parameters of n cameras in a stereo environment.*

Defines

- #define **PC_ZHANG_0** 0x1000
  Multiple cameras, Zhang.

- #define **PC_HEIKKILA_0** 0x2000
  Multiple cameras, Heikkila.

- #define **PC_ZHANG_SINGLE_0** 0x3000
  Single camera with multiple pattern positions.

- #define **PC_HEIKKILA_SINGLE_0** 0x4000
  Single camera with multiple pattern positions.

- #define **PC_FILE_HEAD** 0x1234
  Header identifier of a calibration parameters file.

Define Documentation

#define PC_FILE_HEAD 0x1234

Header identifier of a calibration parameters file.

#define PC_HEIKKILA_0 0x2000

Multiple cameras, Heikkila.

#define PC_HEIKKILA_SINGLE_0 0x4000

Single camera with multiple pattern positions.

#define PC_ZHANG_0 0x1000

Multiple cameras, Zhang.

#define PC_ZHANG_SINGLE_0 0x3000

Single camera with multiple pattern positions.

ChaineUtile.hpp File Reference

`#include "Chaine.hpp"`

Functions

- int **ChaineToInt** (Chaine c)

Function Documentation

int ChaineToInt (Chaine *c*) `[inline]`

Cognois.h File Reference

```
#include "Exception.h"
```

Namespaces

- namespace **Cognois**

ColourSpaceConverter.cpp File Reference

```
#include "ColourSpaceConverter.h"
#include <math.h>
```

Defines

- #define **PI** 3.141592653589793
- #define **NUM_PRECISION_DOUBLE** 2.22e-16
- #define **ERROR_INVALID_PARAMETER** 897234
- #define **ERROR_SUCCESS** 0

Define Documentation

#define ERROR_INVALID_PARAMETER 897234

#define ERROR_SUCCESS 0

#define NUM_PRECISION_DOUBLE 2.22e-16

#define PI 3.141592653589793

ColourSpaceConverter.h File Reference

Compounds

- class **ColourSpaceConverter**

Defines

- #define **AFX_OLOURSPACECONVERTER_H__7CF04B29_751C_11D4_B2D2_00D0B7067F40__INCLUDED_**

Define Documentation

#define AFX_OLOURSPACECONVERTER_H__7CF04B29_751C_11D4_B2D2_00D0B7067F40__INCLUDED_

Value:

Contour.cpp File Reference

```
#include "Contour.hpp"
#include "Erreur.hpp"
#include "ListeTri.hpp"
#include "Table1D.hpp"
#include "ImageBinaire.hpp"
#include <math.h>
```

Defines

- #define **M_PI** 3.14159265359

Functions

- ostream& **operator**<< (ostream &out, const **ContourSegment** &sc)
- istream& **operator**>> (istream &in, **ContourSegment** &sc)
- ostream& **operator**<< (ostream &out, const **Contour** &c)
- istream& **operator**>> (istream &in, **Contour** &c)

Define Documentation

#define M_PI 3.14159265359

Function Documentation

ostream & operator<< (ostream & *out*, const Contour & *c*)

ostream & operator<< (ostream & *out*, const ContourSegment & *sc*)

istream & operator>> (istream & *in*, Contour & *c*)

istream & operator>> (istream & *in*, ContourSegment & *sc*)

Contour.hpp File Reference

```
#include "Point.hpp"
#include "Liste.hpp"
#include <iostream>
```

Compounds

- class **Contour**
- class **ContourSegment**
- struct **Contour::HoraudAngle**
- class **Contour::HoraudRelation**

CounterOCR.cpp File Reference

```
#include "FrameSetClient.h"
#include "CounterOCR.h"
#include "milocr.h"
#include <iostream>
#include <string>
#include <conio.h>
```

CounterOCR.h File Reference

```
#include "ProcessingObject.h"
#include "MilSys.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::CounterOCR**
  Optical character recognition of the 5 digits of the HP 5214L Preset Counter.

datatype.h File Reference

```
#include "Systeme.hpp"
```

Defines

- #define **DUMMY_RETURN**
- #define **true** 1
- #define **false** 0
- #define **NULL** 0

Typedefs

- typedef char **BYTE1**
- typedef unsigned char **UBYTE1**
- typedef short **BYTE2**
- typedef unsigned short **UBYTE2**
- typedef long **BYTE4**
- typedef unsigned long **UBYTE4**

Functions

- **UBYTE2 SystemToLittleEndian** (**UBYTE2** value)
- **UBYTE4 SystemToLittleEndian** (**UBYTE4** value)
- **UBYTE2 SystemToBigEndian** (**UBYTE2** value)
- **UBYTE4 SystemToBigEndian** (**UBYTE4** value)
- **UBYTE2 LittleEndianToSystem** (**UBYTE2** value)
- **UBYTE4 LittleEndianToSystem** (**UBYTE4** value)
- **UBYTE2 BigEndianToSystem** (**UBYTE2** value)
- **UBYTE4 BigEndianToSystem** (**UBYTE4** value)

Define Documentation

#define DUMMY_RETURN

Value:

#define NULL 0

#define false 0

#define true 1

Typedef Documentation

typedef char BYTE1

typedef short BYTE2

typedef long BYTE4

typedef unsigned char UBYTE1

typedef unsigned short UBYTE2

typedef unsigned long UBYTE4

Function Documentation

UBYTE4 BigEndianToSystem (UBYTE4 *value*) `[inline]`

UBYTE2 BigEndianToSystem (UBYTE2 *value*) `[inline]`

UBYTE4 LittleEndianToSystem (UBYTE4 *value*) `[inline]`

UBYTE2 LittleEndianToSystem (UBYTE2 *value*) `[inline]`

UBYTE4 SystemToBigEndian (UBYTE4 *value*) `[inline]`

UBYTE2 SystemToBigEndian (UBYTE2 *value*) `[inline]`

UBYTE4 SystemToLittleEndian (UBYTE4 *value*) `[inline]`

UBYTE2 SystemToLittleEndian (UBYTE2 *value*) `[inline]`

DPMPAR.cpp File Reference

```
#include "f2c.h"
```

Defines

- #define **dmach** ((**doublereal** *)&equiv_2)
- #define **minmag** ((**integer** *)&equiv_2 + 2)
- #define **maxmag** ((**integer** *)&equiv_2 + 4)
- #define **mcheps** ((**integer** *)&equiv_2)

Functions

- **doublereal dpmpar_** (**integer** *i)

Define Documentation

#define dmach ((doublereal *)&equiv_2)

#define maxmag ((integer *)&equiv_2 + 4)

#define mcheps ((integer *)&equiv_2)

#define minmag ((integer *)&equiv_2 + 2)

Function Documentation

doublereal dpmpar_ (integer * *i*)

ENORM.cpp File Reference

```
#include <math.h>
#include "f2c.h"
```

Functions

- **doublereal enorm_** (**integer** *n, **doublereal** *x)

Function Documentation

doublereal enorm_ (integer * *n*, doublereal * *x*)

Equivalence.hpp File Reference

```
#include "TableHO.hpp"
#include "datatype.h"
```

Compounds

- class **Equivalence**
- class **IndiceAlias**

Functions

- template<class T> ostream& **operator<<** (ostream &out, const **IndiceAlias**< T > &ia)
- istream& **operator>>** (istream &in, **IndiceAlias**< T > &ia)

Function Documentation

template<class T> ostream & operator<< (ostream & *out*, const IndiceAlias< T > & *ia*)

istream & operator>> (istream & *in*, IndiceAlias< T > & *ia*)

EspaceCalibrage.cpp File Reference

```
#include "EspaceCalibrage.h"
#include <iostream.h>
```

EspaceCalibrage.h File Reference

```
#include "Matrice.h"
#include <math.h>
#include <float.h>
#include <vector>
#include <iostream>
#include "Cognois.h"
```

Namespaces

- namespace **EspaceCalibrage**

Defines

- #define **CARRE**(x) (x)*(x)
- #define **DIST3D**(p1, p2) sqrt(CARRE(p1.x-p2.x)+CARRE(p1.y-p2.y)+CARRE(p1.z-p2.z))
- #define **DIST2D**(p1, p2) sqrt(CARRE(p1.x-p2.x)+CARRE(p1.y-p2.y))
- #define **MAX**(a, b) ((a) > (b) ? (a) : (b))
- #define **MIN**(a, b) ((a) < (b) ? (a) : (b))
- #define **SWAP**(a, b, c) c=a, a=b, b=c;
- #define **PI** 3.14159265359
- #define **POINT_OUT_OF_IMAGE** -1000
- #define **EPSILON** 4*FLT_EPSILON

Define Documentation

#define CARRE(x) (x)*(x)

#define DIST2D(p1, p2) sqrt(CARRE(p1.x-p2.x)+CARRE(p1.y-p2.y))

#define DIST3D(p1, p2) sqrt(CARRE(p1.x-p2.x)+CARRE(p1.y-p2.y)+CARRE(p1.z-p2.z))

#define EPSILON 4*FLT_EPSILON

#define MAX(a, b) ((a) > (b) ? (a) : (b))

#define MIN(a, b) ((a) < (b) ? (a) : (b))

#define PI 3.14159265359

#define POINT_OUT_OF_IMAGE -1000

#define SWAP(a, b, c) c=a, a=b, b=c;

EvaluationDeQualite.cpp File Reference

```
#include "EvaluationDeQualite.h"
#include "Calibration.h"
#include <vector>
```

EvaluationDeQualite.h File Reference

```
#include "CameraModel.h"
#include "EspaceCalibrage.h"
```

Namespaces

- namespace **Cognois**

Exception.h File Reference

```
#include <exception>
#include <iostream>
#include <string>
#include <strstream>
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::Exception**
  Exception *(p.121) class for COGNOIS.*

Defines

- #define **CognoisException**(inMessage) **Cognois::Exception**(inMessage,__FILE__,__LINE__)
  Macro to automatically include the file name and the line number to the error.

- #define **ITOA**(i, s, not_used) std::strstream ss; ss << i; ss >> s;

Define Documentation

#define CognoisException(inMessage) Cognois::Exception(inMessage,__FILE__,__LINE__)

Macro to automatically include the file name and the line number to the error.

#define ITOA(i, s, not_used) std::strstream ss; ss << i; ss >> s;

ExtractionDeCible.cpp File Reference

```
#include <algorithm>
#include <math.h>
#include "Picture.h"
#include "BlobColoring.h"
#include "Matrice.h"
#include "ExtractionDeCible.h"
#include "Statistics.h"
#include "Cognois.h"
#include "MilGarbageCollector.h"
```

Compounds

- struct **DirectMilAccess**
- struct **PointColore**

Defines

- #define **SEUIL_2_ITERATIONS**
- #define **UTILISE_HOMOGRAPHIE**
- #define **TOLERANCE_MIN** 2
- #define **TOLERANCE_MAX** 5
- #define **TOLERANCE_INC** 0.5
- #define **SEUIL_2_TOLERANCE** 1.2
- #define **DISPLAY_TOLERANCE** 0.6
- #define **MIN_DETECTE** 20
- #define **MIN_BLOB_DIST_TO_BORDER** 2
- #define **MIN_BLOB_SIZE** 5
- #define **NB_POINTS_REFERENCE** 4
- #define **DELTA_VIRTUEL** 3
- #define **MAX_RECONSTRUCTION_ERR** 0.45
- #define **HUE_GREEN_TO_BLUE** PI
- #define **HUE_BLUE_TO_RED** 5*PI/3
- #define **HUE_RED_TO_GREEN** PI/3
- #define **CROSS_SIZE** 10
- #define **INIT_BENCH**(start, finish, freq)
- #define **PERFORM_BENCH**(start, finish, freq, message)

Functions

- int **operator<** (const **PointColore** &p1, const **PointColore** &p2)

Variables

- **StatisticsR**<DataType> **stats**

Define Documentation

#define CROSS_SIZE 10

#define DELTA_VIRTUEL 3

#define DISPLAY_TOLERANCE 0.6

#define HUE_BLUE_TO_RED 5*PI/3

#define HUE_GREEN_TO_BLUE PI

#define HUE_RED_TO_GREEN PI/3

#define INIT_BENCH(start, finish, freq)

Value:

#define MAX_RECONSTRUCTION_ERR 0.45

#define MIN_BLOB_DIST_TO_BORDER 2

#define MIN_BLOB_SIZE 5

#define MIN_DETECTE 20

#define NB_POINTS_REFERENCE 4

#define PERFORM_BENCH(start, finish, freq, message)

Value:

#define SEUIL_2_ITERATIONS

#define SEUIL_2_TOLERANCE 1.2

#define TOLERANCE_INC 0.5

#define TOLERANCE_MAX 5

#define TOLERANCE_MIN 2

#define UTILISE_HOMOGRAPHIE

Function Documentation

int operator< (const PointColore & *p1*, const PointColore & *p2*) `[inline]`

Variable Documentation

StatisticsR< DataType > stats<DataType>

ExtractionDeCible.h File Reference

```
#include <mil.h>
#include "EspaceCalibrage.h"
#include "Cognois.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::ExtractionDeCible**
  Extraction des des points d'une cible de calibrage dans une image. La cible est formee // de cercles et son modele est connu.

F2C.H File Reference

Compounds

- struct **alist**
- struct **cilist**
- struct **cllist**
- struct **complex**
- struct **doublecomplex**
- struct **icilist**
- struct **inlist**
- union **Multitype**
- struct **Namelist**
- struct **olist**
- struct **Vardesc**

Defines

- #define **TRUE_** (1)
- #define **FALSE_** (0)
- #define **Extern** extern
- #define **VOID** void
- #define **abs**(x) ((x) >= 0 ? (x) : -(x))
- #define **dabs**(x) (**doublereal**)abs(x)
- #define **min**(a, b) ((a) <= (b) ? (a) : (b))
- #define **max**(a, b) ((a) >= (b) ? (a) : (b))
- #define **dmin**(a, b) (**doublereal**)min(a,b)
- #define **dmax**(a, b) (**doublereal**)max(a,b)
- #define **F2C_proc_par_types** 1
- #define **C_f** VOID
- #define **H_f** VOID
- #define **Z_f** VOID

Typedefs

- typedef long int **integer**
- typedef char* **address**
- typedef short int **shortint**
- typedef float **real**
- typedef double **doublereal**
- typedef long int **logical**
- typedef short int **shortlogical**
- typedef long **flag**
- typedef long **ftnlen**
- typedef long **ftnint**
- typedef union Multitype **Multitype**
- typedef long **Long**
- typedef struct Vardesc **Vardesc**
- typedef struct Namelist **Namelist**
- typedef int (* **U_fp**)()
- typedef **shortint** (* **J_fp**)()

- typedef **integer** (* **I_fp**)()
- typedef **real** (* **R_fp**)()
- typedef **doublereal** (* **D_fp**)(),(*E_fp)()
- typedef VOID (* **C_fp**)()
- typedef VOID (* **Z_fp**)()
- typedef **logical** (* **L_fp**)()
- typedef **shortlogical** (* **K_fp**)()
- typedef VOID (* **H_fp**)()
- typedef int (* **S_fp**)()
- typedef **doublereal E_f**

Functions

- int **lmdif_** (int(*fcn)(long *, long *, double *, double *, long *), long *m, long *n, double *x, double *fvec, double *ftol, double *xtol, double *gtol, long *maxfev, double *epsfcn, double *diag, long *mode, double *factor, long *nprint, long *info, long *nfev, double *fjac, long *ldfjac, long *ipvt, double *qtf, double *wa1, double *wa2, double *wa3, double *wa4)
- double **dpmpar_** (long *i)
- double **enorm_** (long *n, double *x)
- int **fdjac2_** (int(*fcn)(long *, long *, double *, double *, long *), long *m, long *n, double *x, double *fvec, double *fjac, long *ldfjac, long *iflag, double *epsfcn, double *wa)
- int **lmpar_** (long *n, double *r, long *ldr, long *ipvt, double *diag, double *qtb, double *delta, double *par, double *x, double *sdiag, double *wa1, double *wa2)
- int **qrfac_** (long *m, long *n, double *a, long *lda, long *pivot, long *ipvt, long *lipvt, double *rdiag, double *acnorm, double *wa)
- int **qrsolv_** (long *n, double *r, long *ldr, long *ipvt, double *diag, double *qtb, double *x, double *sdiag, double *wa)

Define Documentation

```
#define C_f  VOID

#define Extern  extern

#define F2C_proc_par_types  1

#define FALSE_  (0)

#define H_f  VOID

#define TRUE_  (1)

#define VOID  void

#define Z_f  VOID

#define abs(x)  ((x) >= 0 ? (x) : -(x))

#define dabs(x)  (doublereal)abs(x)

#define dmax(a, b)  (doublereal)max(a,b)

#define dmin(a, b)  (doublereal)min(a,b)

#define max(a, b)  ((a) >= (b) ? (a) : (b))

#define min(a, b)  ((a) <= (b) ? (a) : (b))
```

Typedef Documentation

typedef VOID(* C_fp)()

typedef doublereal(* D_fp)(),(*E_fp)()

typedef doublereal E_f

typedef VOID(* H_fp)()

typedef integer(* I_fp)()

typedef shortint(* J_fp)()

typedef shortlogical(* K_fp)()

typedef logical(* L_fp)()

typedef long Long

typedef union Multitype Multitype

typedef struct Namelist Namelist

typedef real(* R_fp)()

typedef int(* S_fp)()

typedef int(* U_fp)()

typedef struct Vardesc Vardesc

typedef VOID(* Z_fp)()

typedef char * address

typedef double doublereal

typedef long flag

typedef long ftnint

typedef long ftnlen

typedef long int integer

typedef long int logical

typedef float real

typedef short int shortint

typedef short int shortlogical

Function Documentation

double dpmpar_ (long * *i*)

double enorm_ (long * *n*, double * *x*)

int fdjac2_ (int(* *fcn*)(long *, long *, double *, double *, long *), long * *m*, long * *n*, double * *x*, double * *fvec*, double * *fjac*, long * *ldfjac*, long * *iflag*, double * *epsfcn*, double * *wa*)

int lmdif_ (int(* *fcn*)(long *, long *, double *, double *, long *), long * *m*, long * *n*, double * *x*, double * *fvec*, double * *ftol*, double * *xtol*, double * *gtol*, long * *maxfev*, double * *epsfcn*, double * *diag*, long * *mode*, double * *factor*, long * *nprint*, long * *info*, long * *nfev*, double * *fjac*, long * *ldfjac*, long * *ipvt*, double * *qtf*, double * *wa1*, double * *wa2*, double * *wa3*, double * *wa4*)

int lmpar_ (long * *n*, double * *r*, long * *ldr*, long * *ipvt*, double * *diag*, double * *qtb*, double * *delta*, double * *par*, double * *x*, double * *sdiag*, double * *wa1*, double * *wa2*)

int qrfac_ (long * *m*, long * *n*, double * *a*, long * *lda*, long * *pivot*, long * *ipvt*, long * *lipvt*, double * *rdiag*, double * *acnorm*, double * *wa*)

int qrsolv_ (long * *n*, double * *r*, long * *ldr*, long * *ipvt*, double * *diag*, double * *qtb*, double * *x*, double * *sdiag*, double * *wa*)

FDJAC2.cpp File Reference

```
#include <math.h>
#include "f2c.h"
```

Functions

- int **fdjac2_** (int(*fcn)(**integer** *, **integer** *, **doublereal** *, **doublereal** *, **integer** *), **integer** *m, **integer** *n, **doublereal** *x, **doublereal** *fvec, **doublereal** *fjac, **integer** *ldfjac, **integer** *iflag, **doublereal** *epsfcn, **doublereal** *wa)

Variables

- **integer c__1** = 1

Function Documentation

int fdjac2_ (int(* *fcn*)(integer *, integer *, doublereal *, doublereal *, integer *), integer * *m*, integer * *n*, doublereal * *x*, doublereal * *fvec*, doublereal * *fjac*, integer * *ldfjac*, integer * *iflag*, doublereal * *epsfcn*, doublereal * *wa*)

Variable Documentation

integer c__1 = 1 `[static]`

FormatBMP.cpp File Reference

```
#include "FormatBMP.hpp"
#include "ImageGris.hpp"
```

FormatBMP.hpp File Reference

```
#include "FormatImage.hpp"
```

Compounds

- struct **FormatBMP::bmpHeader**
- class **FormatBMP**

FormatContour.cpp File Reference

```
#include "FormatContour.hpp"
```

FormatContour.hpp File Reference

```
#include "FormatPortable.hpp"
#include "Contour.hpp"
```

Compounds

- class **FormatContour**

Defines

- #define **CTR_V1** "SD_CONTOUR_V1_072199"

Define Documentation

#define CTR_V1 "SD_CONTOUR_V1_072199"

FormatImage.hpp File Reference

```
#include "datatype.h"
#include <iostream>
#include "ImageStreamException.hpp"
#include "ImageRGB.hpp"
#include "ImageGris.hpp"
```

Compounds

- class **FormatImage**

FormatPBM.cpp File Reference

```
#include "FormatPBM.hpp"
```

FormatPBM.hpp File Reference

```
#include "FormatPortable.hpp"
```

Compounds

- class **FormatPBM**
- struct **FormatPBM::pbmHeader**

FormatPCX.cpp File Reference

```
#include "FormatPCX.hpp"
```

FormatPCX.hpp File Reference

```
#include "FormatImage.hpp"
```

Compounds

- class **FormatPCX**
- struct **FormatPCX::pcxHeader**

FormatPGM.cpp File Reference

```
#include "FormatPGM.hpp"
```

FormatPGM.hpp File Reference

```
#include "FormatPortable.hpp"
```

Compounds

- class **FormatPGM**
- struct **FormatPGM::pgmHeader**

FormatPortable.hpp File Reference

```
#include "FormatImage.hpp"
#include "ImageGris.hpp"
#include "Chaine.hpp"
```

Compounds

- class **FormatPortable**

Defines

- #define **COMMENT** '#'
- #define **NEWLINE** 10
- #define **BLANK** ' '
- #define **TAB** 15
- #define **CR** 13
- #define **PBM_ASCII** "P1"
- #define **PGM_ASCII** "P2"
- #define **PPM_ASCII** "P3"
- #define **PBM_RAW** "P4"
- #define **PGM_RAW** "P5"
- #define **PPM_RAW** "P6"

Define Documentation

```
#define BLANK  ' '

#define COMMENT  '#'

#define CR  13

#define NEWLINE  10

#define PBM_ASCII  "P1"

#define PBM_RAW  "P4"

#define PGM_ASCII  "P2"

#define PGM_RAW  "P5"

#define PPM_ASCII  "P3"

#define PPM_RAW  "P6"

#define TAB  15
```

FormatPPM.cpp File Reference

```
#include "FormatPPM.hpp"
```

FormatPPM.hpp File Reference

```
#include "FormatPortable.hpp"
```

Compounds

- class **FormatPPM**
- struct **FormatPPM::ppmHeader**

FormatSunRaster.cpp File Reference

```
#include "FormatSunRaster.hpp"
```

FormatSunRaster.hpp File Reference

```
#include "FormatImage.hpp"
#include "ImageGris.hpp"
```

Compounds

- class **FormatSunRaster**
- struct **FormatSunRaster::rasHeader**

Defines

- #define **RMAGIC** 0x59a66a95
- #define **ROLD** 0
- #define **RSTANDARD** 1
- #define **RRLENCODED** 2
- #define **RRGB** 3
- #define **RTIFF** 4
- #define **RIFF** 5
- #define **RNOMAP** 0
- #define **RRGBMAP** 1
- #define **RRAWMAP** 2
- #define **RESC** 128

Define Documentation

#define RESC 128

#define RIFF 5

#define RMAGIC 0x59a66a95

#define RNOMAP 0

#define ROLD 0

#define RRAWMAP 2

#define RRGB 3

#define RRGBMAP 1

#define RRLENCODED 2

#define RSTANDARD 1

#define RTIFF 4

FormatXPM.cpp File Reference

```
#include "FormatXPM.hpp"
#include "Table1D.hpp"
#include "ChaineUtile.hpp"
#include "Equivalence.hpp"
#include <ctype.h>
```

FormatXPM.hpp File Reference

`#include "FormatImage.hpp"`

Compounds

- class **FormatXPM**
- struct **FormatXPM::xpmColor**
- struct **FormatXPM::xpmHeader**

Defines

- #define **NoData** 0xFFFF

Define Documentation

const NoData 0xFFFF

FormatXWD.cpp File Reference

```
#include "FormatXWD.hpp"
```

FormatXWD.hpp File Reference

`#include "FormatImage.hpp"`

Compounds

- class **FormatXWD**
- struct **FormatXWD::xwdColor**
- struct **FormatXWD::xwdHeader**

Defines

- #define **X11WD_FILE_VERSION** 7

Define Documentation

#define X11WD_FILE_VERSION 7

FrameCounter.cpp File Reference

```
#include "FrameCounter.h"
```

FrameCounter.h File Reference

```
#include <wtypes.h>
#include <strstrea.h>
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::FrameCounter**
  Count grabbed frames according to their time stamp.

FrameSetClient.cpp File Reference

```
#include "FrameSetClient.h"
#include "Cognois.h"
#include <wtypes.h>
```

Defines

- #define **DEFAULT_MIN_POOLING_PERIOD** 2

Define Documentation

#define DEFAULT_MIN_POOLING_PERIOD 2

FrameSetClient.h File Reference

```
#include "GenericCognoisClient.h"
#include "SimpleXmlParser.h"
#include <wtypes.h>
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::FrameSetClient**
  Read and write blob information on the COGNOIS server.

gif.h File Reference

```
#include "datatype.h"
```

Compounds

- struct **GifApplicationBlock**
- struct **GifColorTableEntry**
- struct **GifGraphicControlBlock**
- struct **GifHeader**
- struct **GifImageDescriptor**
- struct **GifScreenDescriptor**

Enumerations

- enum **GifBlockType** { **GifTerminator** = 0x00, **GifImageSeparator** = 0x2C, **GifExtension** = 0x21, **GifCommentExtension** = 0xFE, **GifPlainTextExtension** = 0x01, **GifApplicationExtension** = 0xFF, **GifGraphicControlExtension** = 0xF9, **GifTrailer** = 0x3B }

Variables

- const int **GifBitsUsed** = 12
- const unsigned int **GifInputBufferSize** = 255
- const int **GifMaxColors** = 256

Enumeration Type Documentation

enum GifBlockType

Enumeration values:

GifTerminator
GifImageSeparator
GifExtension
GifCommentExtension
GifPlainTextExtension
GifApplicationExtension
GifGraphicControlExtension
GifTrailer

Variable Documentation

const int GifBitsUsed = 12

const unsigned int GifInputBufferSize = 255

const int GifMaxColors = 256

gifdecod.cpp File Reference

```
#include "gifdecod.h"
#include "gif.h"
```

Variables

- const int **TableSize** = 1 << **GifBitsUsed**
- const int **FixedArea** = 1 << CHAR_BIT

Variable Documentation

const int FixedArea = 1 << CHAR_BIT

const int TableSize = 1 << GifBitsUsed

gifdecod.h File Reference

```
#include <iostream>
#include "bitimage.h"
#include "gifexcep.h"
```

Compounds

- class **GifDecoder**
- struct **GifDecoder::LZ77String**

gifencod.cpp File Reference

```
#include "gif.h"
#include "gifencod.h"
```

Variables

- const int **CodeSize** = 9
- const int **ClearCode** = 256
- const int **EndCode** = 257
- const int **FirstCode** = 258
- const int **LastCode** = (1 << **CodeSize**) - 1

Variable Documentation

const int ClearCode = 256

const int CodeSize = 9

const int EndCode = 257

const int FirstCode = 258

const int LastCode = (1 << CodeSize) - 1

gifencod.h File Reference

```
#include "bitimage.h"
#include "gifexcep.h"
```

Compounds

- class **GifEncoder**

gifexcep.h File Reference

```
#include "grexcept.h"
#include "Chaine.hpp"
```

Compounds

- class **EGifCircularTree**
- class **EGifCorruptFile**
- class **EGifError**
- class **EGifFileReadError**
- class **EGifNoMoreImages**
- class **EGifNotAGifFile**

GrabInit.cpp File Reference

```
#include "GrabInit.h"
#include <iostream>
```

Functions

- int **InitGrabber** (**MILSys** &sys, int profile)
- void **FreeGrabber** (**MILSys** &sys)

Function Documentation

void FreeGrabber (MILSys & *sys*)

int InitGrabber (MILSys & *sys*, int *profile* = PRO_ECRAN)

GrabInit.h File Reference

```
#include <mil.h>
#include <wtypes.h>
#include "MILSys.h"
```

Defines

- #define **TRUE** 1
- #define **FALSE** 0
- #define **PRO_ECRAN** 0
- #define **PRO_METEOR2** 1

Functions

- int **InitGrabber** (**MILSys** &sys, int profile=PRO_ECRAN)
- void **FreeGrabber** (**MILSys** &sys)

Define Documentation

#define FALSE 0

#define PRO_ECRAN 0

#define PRO_METEOR2 1

#define TRUE 1

Function Documentation

void FreeGrabber (MILSys & *sys*)

int InitGrabber (MILSys & *sys*, int *profile* = PRO_ECRAN)

grexcept.h File Reference

```
#include "Erreur.hpp"
#include "Chaine.hpp"
```

Compounds

- class **EGraphicsAbort**
- class **EGraphicsException**
- class **EInvalidBitCount**
- class **ESubscriptOutOfRange**

Image.hpp File Reference

```
#include "Table1D.hpp"
#include "Table2D.hpp"
#include "Erreur.hpp"
#include "Random.hpp"
#include "Operateur.hpp"
#include "Point.hpp"
```

Compounds

- class **Image**

ImageBinaire.cpp File Reference

```
#include "ImageBinaire.hpp"
#include "ImageGradient.hpp"
#include "ListeTri.hpp"
#include "TablePile.hpp"
#include "Equivalence.hpp"
```

ImageBinaire.hpp File Reference

```
#include "ImageGris.hpp"
#include "Kernel.hpp"
#include "Liste.hpp"
```

Compounds

- class **Ensemble**
- class **ImageBinaire**
- class **Intervalle**

ImageGradient.cpp File Reference

```
#include "ImageGradient.hpp"
#include "ImageGris.hpp"
```

Defines

- #define **M_PI** 3.14159265359

Define Documentation

#define M_PI 3.14159265359

ImageGradient.hpp File Reference

```
#include "Image.hpp"
#include <math.h>
```

Compounds

- class **ImageGradient**

ImageGris.cpp File Reference

```
#include "ImageGris.hpp"
#include "ImageGradient.hpp"
#include "ImageRGB.hpp"
#include "ImageBinaire.hpp"
#include "Contour.hpp"
#include "MonceauBin.hpp"
#include "bitimage.h"
```

ImageGris.hpp File Reference

```
#include "Image.hpp"
#include "Liste.hpp"
```

Compounds

- class **ImageGris**
- class **ImageGris::PRep**

Defines

- #define **ON** 255
- #define **OFF** 0

Define Documentation

#define OFF 0

#define ON 255

ImageRGB.cpp File Reference

```
#include "ImageRGB.hpp"
#include "ImageGradient.hpp"
#include "ImageGris.hpp"
#include "bitimage.h"
#include "MonceauBin.hpp"
```

Defines

- #define **M_PI** 3.14159265359
- #define **PI_3** M_PI/3

Functions

- ostream& **operator**<< (ostream &out, const **CouleurRGB** &p)
- istream& **operator**>> (istream &in, **CouleurRGB** &p)
- ostream& **operator**<< (ostream &out, const **CouleurHSI** &p)
- istream& **operator**>> (istream &in, **CouleurHSI** &p)

Define Documentation

#define M_PI 3.14159265359

#define PI_3 M_PI/3

Function Documentation

ostream & operator<< (ostream & *out*, const CouleurHSI & *p*)

ostream & operator<< (ostream & *out*, const CouleurRGB & *p*)

istream & operator>> (istream & *in*, CouleurHSI & *p*)

istream & operator>> (istream & *in*, CouleurRGB & *p*)

ImageRGB.hpp File Reference

```
#include "Image.hpp"
#include "Liste.hpp"
#include <math.h>
```

Compounds

- class **CouleurHSI**
- class **CouleurRGB**
- class **ImageRGB**
- class **ImageRGB::PRep**

ImageStream.cpp File Reference

```
#include "ImageStream.hpp"
#include "datatype.h"
#include "bitimage.h"
#include "jpegdeco.h"
#include "jpegenco.h"
#include "gifdecod.h"
#include "gifencod.h"
#include "pngdecod.h"
#include "pngencod.h"
#include "xbmdecod.h"
#include "xbmencod.h"
#include "FormatSunRaster.hpp"
#include "FormatBMP.hpp"
#include "FormatPCX.hpp"
#include "FormatXWD.hpp"
#include "FormatPGM.hpp"
#include "FormatPBM.hpp"
#include "FormatPPM.hpp"
#include "FormatXPM.hpp"
#include "FormatContour.hpp"
```

ImageStream.hpp File Reference

```
#include <fstream>
#include "Chaine.hpp"
#include "ImageGris.hpp"
#include "ImageRGB.hpp"
#include "Contour.hpp"
#include "ImageStreamException.hpp"
```

Compounds

- class **ImageStream**

ImageStreamException.hpp File Reference

```
#include "Erreur.hpp"
#include <fstream>
```

Compounds

- class **ErreurFormat**
- class **ErreurFormatImage**
- class **ErreurFormatImageEcriture**
- class **ErreurFormatImageLecture**
- class **ErreurImage**

imagetyp.h File Reference

```
#include "bitimage.h"
```

Enumerations

- enum **ImageType** { **UnknownImage** = 0, **BmpImage** = 2, **JpegImage** = 3, **GifImage** = 5, **PngImage** = 7 }

Functions

- **ImageType GetStreamImageType** (std::istream &)
- **ImageType ReadImage** (std::istream &strm, **BitmapImage** &image, **PROGRESSFUNCTION** function, void *data)

Enumeration Type Documentation

enum ImageType

Enumeration values:

UnknownImage
BmpImage
JpegImage
GifImage
PngImage

Function Documentation

ImageType GetStreamImageType (std::istream &)

ImageType ReadImage (std::istream & *strm*, BitmapImage & *image*, PROGRESSFUNCTION *function*, void * *data*)

jfif.cpp File Reference

```
#include "jfif.h"
```

Functions

- int **Scaled** (double value)
- **JPEGSAMPLE YCbCrToR** (int yy, int, int cr)
- **JPEGSAMPLE YCbCrToG** (int yy, int cb, int cr)
- **JPEGSAMPLE YCbCrToB** (int yy, int cb, int)
- **JPEGSAMPLE RGBToY** (**JPEGSAMPLE** red, **JPEGSAMPLE** green, **JPEGSAMPLE** blue)
- **JPEGSAMPLE RGBToCb** (**JPEGSAMPLE** red, **JPEGSAMPLE** green, **JPEGSAMPLE** blue)
- **JPEGSAMPLE RGBToCr** (**JPEGSAMPLE** red, **JPEGSAMPLE** green, **JPEGSAMPLE** blue)

Variables

- const int **ScaleFactor** = 12
- const int **ScaleValue** = (1<<**ScaleFactor**)
- const int **Rounding** = (1<<(**ScaleFactor**-1))

Function Documentation

JPEGSAMPLE RGBToCb (JPEGSAMPLE *red*, JPEGSAMPLE *green*, JPEGSAMPLE *blue*)

JPEGSAMPLE RGBToCr (JPEGSAMPLE *red*, JPEGSAMPLE *green*, JPEGSAMPLE *blue*)

JPEGSAMPLE RGBToY (JPEGSAMPLE *red*, JPEGSAMPLE *green*, JPEGSAMPLE *blue*)

int Scaled (double *value*) `[inline, static]`

JPEGSAMPLE YCbCrToB (int *yy*, int *cb*, int *cr*)

JPEGSAMPLE YCbCrToG (int *yy*, int *cb*, int *cr*)

JPEGSAMPLE YCbCrToR (int *yy*, int *cb*, int *cr*)

Variable Documentation

const int Rounding = (1<<(ScaleFactor-1))

const int ScaleFactor = 12

const int ScaleValue = (1<<ScaleFactor)

jfif.h File Reference

`#include "jpeg.h"`

Compounds

- struct **JfifExtension**
- struct **JfifHeader**

Functions

- **JPEGSAMPLE YCbCrToR** (int yy, int cb, int cr)
- **JPEGSAMPLE YCbCrToG** (int yy, int cb, int cr)
- **JPEGSAMPLE YCbCrToB** (int yy, int cb, int cr)
- **JPEGSAMPLE RGBToY** (**JPEGSAMPLE** red, **JPEGSAMPLE** green, **JPEGSAMPLE** blue)
- **JPEGSAMPLE RGBToCb** (**JPEGSAMPLE** red, **JPEGSAMPLE** green, **JPEGSAMPLE** blue)
- **JPEGSAMPLE RGBToCr** (**JPEGSAMPLE** red, **JPEGSAMPLE** green, **JPEGSAMPLE** blue)

Function Documentation

JPEGSAMPLE RGBToCb (JPEGSAMPLE *red*, JPEGSAMPLE *green*, JPEGSAMPLE *blue*)

JPEGSAMPLE RGBToCr (JPEGSAMPLE *red*, JPEGSAMPLE *green*, JPEGSAMPLE *blue*)

JPEGSAMPLE RGBToY (JPEGSAMPLE *red*, JPEGSAMPLE *green*, JPEGSAMPLE *blue*)

JPEGSAMPLE YCbCrToB (int *yy*, int *cb*, int *cr*)

JPEGSAMPLE YCbCrToG (int *yy*, int *cb*, int *cr*)

JPEGSAMPLE YCbCrToR (int *yy*, int *cb*, int *cr*)

jpdecobk.h File Reference

```
#include "jpeg.h"
```

Typedefs

- typedef **BYTE2 JpegDecoderCoefficientBlock** [**JpegSampleWidth**][**JpegSampleWidth**]

Typedef Documentation

typedef BYTE2 JpegDecoderCoefficientBlock

jpdecomp.cpp File Reference

```
#include "jpdecomp.h"
#include "jpgexcep.h"
#include "jpdequan.h"
#include "jpegdeco.h"
#include "jpdecobk.h"
#include "jpeg.h"
#include "bitimage.h"
```

Functions

- int **Extend** (int vv, int tt)

Function Documentation

int Extend (int *vv*, int *tt*) `[inline, static]`

jpdecomp.h File Reference

```
#include "jpeg.h"
#include "jfif.h"
#include "jpdedu.h"
#include "jpdehuff.h"
#include "jpdecobk.h"
#include "bitimage.h"
```

Compounds

- class **JpegDecoderComponent**

jpdedu.cpp File Reference

```
#include <math.h>
#include "jpdedu.h"
```

Functions

- **JPEGSAMPLE SampleRange** (long value)
- long **Descale** (long value, int amount)

Variables

- const int **IntegerScale** = 6
- const long **IC4** = (long)((1<<**IntegerScale**) * cos (M_PI * 4.0/16.0))
- const long **ISEC2** = (long)((1<<(**IntegerScale** - 1)) / cos (M_PI * 2.0 / 16.0))
- const long **ISEC6** = (long)((1<<(**IntegerScale** - 1)) / cos (M_PI * 6.0 / 16.0))
- const double **FC4** = cos (M_PI * 4.0 / 16.0)
- const double **FSEC2** = 0.5 / cos (M_PI * 2.0 / 16.0)
- const double **FSEC6** = 0.5 / cos (M_PI * 6.0 / 16.0)

Function Documentation

long Descale (long *value*, int *amount*) `[inline, static]`

JPEGSAMPLE SampleRange (long *value*) `[inline, static]`

Variable Documentation

const double FC4 = cos (M_PI * 4.0 / 16.0)

const double FSEC2 = 0.5 / cos (M_PI * 2.0 / 16.0)

const double FSEC6 = 0.5 / cos (M_PI * 6.0 / 16.0)

const long IC4 = (long)((1<<IntegerScale) * cos (M_PI * 4.0/16.0))

const long ISEC2 = (long)((1<<(IntegerScale - 1)) / cos (M_PI * 2.0 / 16.0))

const long ISEC6 = (long)((1<<(IntegerScale - 1)) / cos (M_PI * 6.0 / 16.0))

const int IntegerScale = 6

jpdedu.h File Reference

```
#include <iostream>
#include "jpeg.h"
#include "jpdequan.h"
#include "jpdecobk.h"
```

Compounds

- class **JpegDecoderDataUnit**

Functions

- ostream& **operator**<< (ostream &, **JpegDecoderDataUnit** &)
- ostream& **operator**<< (ostream &strm, const **JpegDecoderDataUnit** &du)

Function Documentation

ostream & operator<< (ostream & *strm*, const JpegDecoderDataUnit & *du*) `[inline]`

ostream & operator<< (ostream &, JpegDecoderDataUnit &)

jpdehuff.cpp File Reference

```
#include "jpdehuff.h"
#include "jpegdeco.h"
#include "jpgexcep.h"
```

jpdehuff.h File Reference

```
#include "jpeg.h"
#include "bitimage.h"
```

Compounds

- class **JpegHuffmanDecoder**

jpdequan.cpp File Reference

```
#include <iostream>
#include "jpdequan.h"
#include "jpegdeco.h"
#include "jpgexcep.h"
```

Variables

- const double **floatscaling** [**JpegSampleWidth**][**JpegSampleWidth**]

Variable Documentation

const double floatscaling [static]

Initial value:

```
{
{ 0.125,                  0.09011997775086849627, 0.09567085809127244544,
0.1063037618459070632,  0.125,                  0.159094822571604233,
0.2309698831278216846, 0.4530637231764438333, },
{ 0.09011997775086849627, 0.0649728831185362593,  0.0689748448207357645,
0.07664074121909414394, 0.09011997775086849627, 0.1147009749634507608,
0.1665200058287998886, 0.3266407412190940884, },
{ 0.09567085809127244544, 0.0689748448207357645,  0.0732233047033631207,
0.08136137691302557096, 0.09567085809127244544, 0.1217659055464329343,
0.1767766952966368932, 0.3467599613305368256, },
{ 0.1063037618459070632,  0.07664074121909414394, 0.08136137691302557096,
0.09040391826073060355, 0.1063037618459070632,  0.135299025036549253,
0.1964237395967755595, 0.3852990250365491698, },
{ 0.125,                  0.09011997775086849627, 0.09567085809127244544,
0.1063037618459070632,  0.125,                  0.159094822571604233,
0.2309698831278216846, 0.4530637231764438333, },
{ 0.159094822571604233,   0.1147009749634507608,  0.1217659055464329343,
0.135299025036549253,   0.159094822571604233,   0.2024893005527218515,
0.2939689006048396558, 0.5766407412190940329, },
{ 0.2309698831278216846,  0.1665200058287998886,  0.1767766952966368932,
0.1964237395967755595,  0.2309698831278216846,  0.2939689006048396558,
0.4267766952966368654, 0.8371526015321518744, },
{ 0.4530637231764438333,  0.3266407412190940884,  0.3467599613305368256,
0.3852990250365491698,  0.4530637231764438333,  0.5766407412190940329,
0.8371526015321518744, 1.642133898068010689,  },
}
```

jpdequan.h File Reference

```
#include <iostream>
#include "jpeg.h"
#include "bitimage.h"
```

Compounds

- class **JpegDecoderQuantizationTable**

Functions

- ostream& **operator**<< (ostream &strm, const **JpegDecoderQuantizationTable** &du)

Function Documentation

ostream & operator<< (ostream & *strm*, const JpegDecoderQuantizationTable & *du*) [inline]

jpeg.cpp File Reference

`#include "jpeg.h"`

Variables

- const unsigned int **JpegZigZagInputOrderCodes** [**JpegSampleSize**]
- const unsigned int **JpegZigZagOutputOrderCodes** [**JpegSampleSize**]

Variable Documentation

const unsigned int JpegZigZagInputOrderCodes[JpegSampleSize]

Initial value:

```
{
   0,  1,  8, 16,  9,  2,  3, 10,
  17, 24, 32, 25, 18, 11,  4,  5,
  12, 19, 26, 33, 40, 48, 41, 34,
  27, 20, 13,  6,  7, 14, 21, 28,
  35, 42, 49, 56, 57, 50, 43, 36,
  29, 22, 15, 23, 30, 37, 44, 51,
  58, 59, 52, 45, 38, 31, 39, 46,
  53, 60, 61, 54, 47, 55, 62, 63
}
```

const unsigned int JpegZigZagOutputOrderCodes[JpegSampleSize]

Initial value:

```
{
   0,  1,  5,  6, 14, 15, 27, 28,
   2,  4,  7, 13, 16, 26, 29, 42,
   3,  8, 12, 17, 25, 30, 41, 43,
   9, 11, 18, 24, 31, 40, 44, 53,
  10, 19, 23, 32, 39, 45, 52, 54,
  20, 21, 33, 38, 46, 51, 55, 60,
  21, 34, 37, 47, 50, 56, 59, 61,
  35, 36, 48, 49, 57, 58, 62, 63
}
```

jpeg.h File Reference

```
#include "datatype.h"
#include "jpmarker.h"
```

Typedefs

- typedef **UBYTE1 JPEGSAMPLE**
- typedef **BYTE1 JPEGCOEFFICIENT**

Functions

- unsigned int **JpegZigZagInputOrder** (unsigned int item)
- unsigned int **JpegZigZagOutputOrder** (unsigned int item)

Variables

- const int **JpegMaxHuffmanTables** = 4
- const int **MaxQuantizationTables** = 4
- const int **JpegMaxComponentsPerFrame** = 255
- const int **JpegMaxComponentsPerScan** = 4
- const int **JpegMinSamplingFrequency** = 1
- const int **JpegMaxSamplingFrequency** = 4
- const int **JpegSampleWidth** = 8
- const int **JpegSampleSize** = **JpegSampleWidth** * **JpegSampleWidth**
- const int **JpegMinSampleValue** = 0
- const int **JpegMaxSampleValue** = 255
- const int **JpegMidpointSampleValue** = 128
- const unsigned int **JpegMaxHuffmanCodeLength** = 16
- const unsigned int **JpegMaxNumberOfHuffmanCodes** = 256
- const int **JpegMaxDataUnitsPerMCU** = 10
- const int **JpegMaxSuccessiveApproximation** = 13
- const int **JpegMax8BitQuantizationValue** = 255
- const int **JpegMinQuantizationValue** = 1
- const unsigned int **JpegZigZagInputOrderCodes** [**JpegSampleSize**]
- const unsigned int **JpegZigZagOutputOrderCodes** [**JpegSampleSize**]

Typedef Documentation

typedef BYTE1 JPEGCOEFFICIENT

typedef UBYTE1 JPEGSAMPLE

Function Documentation

unsigned int JpegZigZagInputOrder (unsigned int *item*) `[inline]`

unsigned int JpegZigZagOutputOrder (unsigned int *item*) `[inline]`

Variable Documentation

const int JpegMax8BitQuantizationValue = 255

const int JpegMaxComponentsPerFrame = 255

const int JpegMaxComponentsPerScan = 4

const int JpegMaxDataUnitsPerMCU = 10

const unsigned int JpegMaxHuffmanCodeLength = 16

const int JpegMaxHuffmanTables = 4

const unsigned int JpegMaxNumberOfHuffmanCodes = 256

const int JpegMaxSampleValue = 255

const int JpegMaxSamplingFrequency = 4

const int JpegMaxSuccessiveApproximation = 13

const int JpegMidpointSampleValue = 128

const int JpegMinQuantizationValue = 1

const int JpegMinSampleValue = 0

const int JpegMinSamplingFrequency = 1

const int JpegSampleSize = JpegSampleWidth * JpegSampleWidth

const int JpegSampleWidth = 8

const unsigned int JpegZigZagInputOrderCodes

Initial value:

```
{
```

```
   0,  1,  8, 16,  9,  2,  3, 10,
  17, 24, 32, 25, 18, 11,  4,  5,
  12, 19, 26, 33, 40, 48, 41, 34,
  27, 20, 13,  6,  7, 14, 21, 28,
  35, 42, 49, 56, 57, 50, 43, 36,
  29, 22, 15, 23, 30, 37, 44, 51,
  58, 59, 52, 45, 38, 31, 39, 46,
  53, 60, 61, 54, 47, 55, 62, 63
}
```

const unsigned int JpegZigZagOutputOrderCodes

Initial value:

```
{
   0,  1,  5,  6, 14, 15, 27, 28,
   2,  4,  7, 13, 16, 26, 29, 42,
   3,  8, 12, 17, 25, 30, 41, 43,
   9, 11, 18, 24, 31, 40, 44, 53,
  10, 19, 23, 32, 39, 45, 52, 54,
  20, 21, 33, 38, 46, 51, 55, 60,
  21, 34, 37, 47, 50, 56, 59, 61,
  35, 36, 48, 49, 57, 58, 62, 63
}
```

const int MaxQuantizationTables = 4

jpegdeco.cpp File Reference

```
#include "jpegdeco.h"
#include "jfif.h"
#include "jpdehuff.h"
#include "bitimage.h"
#include "jpdequan.h"
#include "jpdecomp.h"
```

jpegdeco.h File Reference

```
#include <iostream>
#include "datatype.h"
#include "jpgexcep.h"
#include "bitimage.h"
```

Compounds

- class **JpegDecoder**

jpegenco.cpp File Reference

```
#include <math.h>
#include <string.h>
#include "jpegenco.h"
#include "bitimage.h"
#include "jfif.h"
#include "jpenquan.h"
#include "jpenhuff.h"
```

Variables

- const unsigned int **default_luminance_table** [**JpegSampleSize**]
- const unsigned int **default_chrominance_table** [**JpegSampleSize**]

Variable Documentation

const unsigned int default_chrominance_table [static]

Initial value:

```
{
17, 18, 24, 47, 99, 99, 99, 99,
18, 21, 26, 66, 99, 99, 99, 99,
24, 26, 56, 99, 99, 99, 99, 99,
47, 66, 99, 99, 99, 99, 99, 99,
99, 99, 99, 99, 99, 99, 99, 99,
99, 99, 99, 99, 99, 99, 99, 99,
99, 99, 99, 99, 99, 99, 99, 99,
99, 99, 99, 99, 99, 99, 99, 99,
}
```

const unsigned int default_luminance_table [static]

Initial value:

```
{
16, 11, 10, 16,  24,  40,  51,  61,
12, 12, 14, 19,  26,  58,  60,  55,
14, 13, 16, 24,  40,  57,  69,  56,
14, 17, 22, 29,  51,  87,  80,  62,
18, 22, 37, 56,  68, 109, 103,  77,
24, 35, 55, 64,  81, 104, 113,  92,
49, 64, 78, 87, 103, 121, 120, 101,
72, 92, 95, 98, 112, 100, 103,  99,
}
```

jpegenco.h File Reference

```
#include "bitimage.h"
#include "jpencomp.h"
#include "jpgexcep.h"
#include "Chaine.hpp"
```

Compounds

- class **JpegEncoder**
- struct **JpegEncoder::Scan**

jpencobk.h File Reference

```
#include "jpeg.h"
```

Compounds

- class **JpegEncoderCoefficientBlock**

jpencomp.cpp File Reference

```
#include "jpegenco.h"
#include "jpencomp.h"
#include "jfif.h"
#include "jpencobk.h"
#include "jpendu.h"
#include "jpenhuff.h"
```

jpencomp.h File Reference

```
#include "bitimage.h"
#include "jpeg.h"
```

Compounds

- class **JpegEncoderComponent**

jpendu.cpp File Reference

```
#include <math.h>
#include "jpendu.h"
```

Variables

- const double **FC4** = cos (M_PI * 4.0 / 16.0)
- const double **FSEC2** = 0.5 / cos (M_PI * 2.0 / 16.0)
- const double **FSEC6** = 0.5 / cos (M_PI * 6.0 / 16.0)

Variable Documentation

const double FC4 = cos (M_PI * 4.0 / 16.0)

const double FSEC2 = 0.5 / cos (M_PI * 2.0 / 16.0)

const double FSEC6 = 0.5 / cos (M_PI * 6.0 / 16.0)

jpendu.h File Reference

```
#include "jpeg.h"
#include "jpencobk.h"
#include "jpenquan.h"
```

Compounds

- class **JpegEncoderDataUnit**

jpenhuff.cpp File Reference

```
#include "jpenhuff.h"
#include "jpegenco.h"
#include "jpgexcep.h"
```

jpenhuff.h File Reference

```
#include "jpeg.h"
```

Compounds

- class **JpegEncoderHuffmanTable**

jpenquan.cpp File Reference

```
#include "jpenquan.h"
#include "jpegenco.h"
#include "jpgexcep.h"
```

Variables

- const double **floatscaling** [**JpegSampleWidth**][**JpegSampleWidth**]

Variable Documentation

const double floatscaling [static]

Initial value:

```
{
{ 0.125,                  0.09011997775086849627, 0.09567085809127244544,
0.1063037618459070632,  0.125,                  0.159094822571604233,
0.2309698831278216846, 0.4530637231764438333, },
{ 0.09011997775086849627, 0.0649728831185362593,  0.0689748448207357645,
0.0766407412190941439 4, 0.09011997775086849627, 0.1147009749634507608,
0.1665200058287998886, 0.3266407412190940884, },
{ 0.09567085809127244544, 0.0689748448207357645,  0.0732233047033631207,
0.0813613769130255709 6, 0.09567085809127244544, 0.1217659055464329343,
0.1767766952966368932, 0.3467599613305368256, },
{ 0.1063037618459070632,  0.0766407412190941439 4, 0.0813613769130255709 6,
0.0904039182607306035 5, 0.1063037618459070632,  0.135299025036549253,
0.1964237395967755595, 0.3852990250365491698, },
{ 0.125,                  0.09011997775086849627, 0.09567085809127244544,
0.1063037618459070632,  0.125,                  0.159094822571604233,
0.2309698831278216846, 0.4530637231764438333, },
{ 0.159094822571604233,   0.1147009749634507608,  0.1217659055464329343,
0.135299025036549253,   0.159094822571604233,   0.2024893005527218515,
0.2939689006048396558, 0.5766407412190940329, },
{ 0.2309698831278216846,  0.1665200058287998886,  0.1767766952966368932,
0.1964237395967755595,  0.2309698831278216846,  0.2939689006048396558,
0.4267766952966368654, 0.8371526015321518744, },
{ 0.4530637231764438333,  0.3266407412190940884,  0.3467599613305368256,
0.3852990250365491698,  0.4530637231764438333,  0.5766407412190940329,
0.8371526015321518744, 1.642133898068010689,  },
}
```

jpenquan.h File Reference

```
#include "jpeg.h"
```

Compounds

- class **JpegEncoderQuantizationTable**

jpgexcep.h File Reference

```
#include "grexcept.h"
#include "Chaine.hpp"
```

Compounds

- class **EJpegAbort**
- class **EJpegBadData**
- class **EJpegBadOutputParameter**
- class **EJpegError**
- class **EJpegFatal**
- class **EJpegIndexOutOfRange**
- class **EJpegValueOutOfRange**

jpmarker.h File Reference

Enumerations

- enum **JpegMarkers** { **SOF0** = 0xC0, **SOF1** = 0xC1, **SOF2** = 0xC2, **SOF3** = 0xC3, **SOF5** = 0xC5, **SOF6** = 0xC6, **SOF7** = 0xC7, **SOF9** = 0xC9, **SOFA** = 0xCA, **SOFB** = 0xCB, **SOFD** = 0xCD, **SOFE** = 0xCE, **SOFF** = 0xCF, **DHT** = 0xC4, **DAC** = 0xCC, **RST0** = 0xD0, **RST1** = 0xD1, **RST2** = 0xD2, **RST3** = 0xD3, **RST4** = 0xD4, **RST5** = 0xD5, **RST6** = 0xD6, **RST7** = 0xD7, **SOI** = 0xD8, **EOI** = 0xD9, **SOS** = 0xDA, **DQT** = 0xDB, **DNL** = 0xDC, **DRI** = 0xDD, **DHP** = 0xDE, **EXP** = 0xDF, **APP0** = 0xE0, **APP1** = 0xE1, **APP2** = 0xE2, **APP3** = 0xE3, **APP4** = 0xE4, **APP5** = 0xE5, **APP6** = 0xE6, **APP7** = 0xE7, **APP8** = 0xE8, **APP9** = 0xE9, **APPA** = 0xEA, **APPB** = 0xEB, **APPC** = 0xEC, **APPD** = 0xED, **APPE** = 0xEE, **APPF** = 0xEF, **COM** = 0xFE, **SOB** = 0xFF }

Enumeration Type Documentation

enum JpegMarkers

Enumeration values:

SOF0
SOF1
SOF2
SOF3
SOF5
SOF6
SOF7
SOF9
SOFA
SOFB
SOFD
SOFE
SOFF
DHT
DAC
RST0
RST1
RST2
RST3
RST4
RST5
RST6
RST7
SOI
EOI
SOS
DQT
DNL
DRI
DHP
EXP
APP0

APP1
APP2
APP3
APP4
APP5
APP6
APP7
APP8
APP9
APPA
APPB
APPC
APPD
APPE
APPF
COM
SOB

Kernel.cpp File Reference

```
#include "Kernel.hpp"
```

Kernel.hpp File Reference

```
#include "Operateur.hpp"
```

Compounds

- class **Kernel**

ListeTri.hpp File Reference

```
#include "Liste.hpp"
```

Compounds

- class **ListeTri**

Functions

- template<class T> ostream& **operator<<** (ostream &_flot, const **ListeTri**< T > &_liste)
- istream& **operator>>** (istream &_flot, **ListeTri**< T > &_liste)

Function Documentation

template<class T> ostream & operator<< (ostream & *_flot*, const ListeTri< T > & *_liste*)

istream & operator>> (istream & *_flot*, ListeTri< T > & *_liste*)

LMDIF.cpp File Reference

```
#include <math.h>
#include "f2c.h"
```

Functions

- int **lmdif_** (int(*fcn)(**integer** *, **integer** *, **doublereal** *, **doublereal** *, **integer** *), **integer** *m, **integer** *n, **doublereal** *x, **doublereal** *fvec, **doublereal** *ftol, **doublereal** *xtol, **doublereal** *gtol, **integer** *maxfev, **doublereal** *epsfcn, **doublereal** *diag, **integer** *mode, **doublereal** *factor, **integer** *nprint, **integer** *info, **integer** *nfev, **doublereal** *fjac, **integer** *ldfjac, **integer** *ipvt, **doublereal** *qtf, **doublereal** *wa1, **doublereal** *wa2, **doublereal** *wa3, **doublereal** *wa4)

Variables

- **integer c__1** = 1

Function Documentation

int lmdif_ (int(*fcn*)(integer *, integer *, doublereal *, doublereal *, integer *), integer * *m*, integer * *n*, doublereal * *x*, doublereal * *fvec*, doublereal * *ftol*, doublereal * *xtol*, doublereal * *gtol*, integer * *maxfev*, doublereal * *epsfcn*, doublereal * *diag*, integer * *mode*, doublereal * *factor*, integer * *nprint*, integer * *info*, integer * *nfev*, doublereal * *fjac*, integer * *ldfjac*, integer * *ipvt*, doublereal * *qtf*, doublereal * *wa1*, doublereal * *wa2*, doublereal * *wa3*, doublereal * *wa4*)**

Variable Documentation

integer c__1 = 1 `[static]`

LMPAR.cpp File Reference

```
#include <math.h>
#include "f2c.h"
```

Functions

- int **lmpar_** (**integer** *n, **doublereal** *r, **integer** *ldr, **integer** *ipvt, **doublereal** *diag, **doublereal** *qtb, **doublereal** *delta, **doublereal** *par, **doublereal** *x, **doublereal** *sdiag, **doublereal** *wa1, **doublereal** *wa2)

Variables

- **integer c__2** = 2

Function Documentation

int lmpar_ (integer * *n*, doublereal * *r*, integer * *ldr*, integer * *ipvt*, doublereal * *diag*, doublereal * *qtb*, doublereal * *delta*, doublereal * *par*, doublereal * *x*, doublereal * *sdiag*, doublereal * *wa1*, doublereal * *wa2*)

Variable Documentation

integer c__2 = 2 `[static]`

MakeAVI.cpp File Reference

```
#include "MakeAVI.h"
#include <fstream>
#include <iostream.h>
```

MakeAVI.h File Reference

```
#include "ProcessingObject.h"
#include <vector>
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::MakeAVI**
  Export grabbed images to an AVI movie.

Matrice.cpp File Reference

```
#include "Matrice.h"
#include "MATRIX.h"
#include <math.h>
#include <fstream>
#include <vector>
#include <strstrea.h>
#include "Cognois.h"
#include <string>
#include <float.h>
```

Defines

- #define **sign**(a, b) (((b) >= 0.0) ? fabs(a) : -fabs(a))

Functions

- **Matrice operator *** (const double sc, const **Matrice** &m)
  Operateurs commutatifs.

Define Documentation

#define sign(a, b) (((b) >= 0.0) ? fabs(a) : -fabs(a))

Function Documentation

Matrice operator * (const double *double*, const Matrice & *m*)

Operateurs commutatifs.

Matrice.h File Reference

Compounds

- class **Matrice**
  Fonctions de base pour le calcul matriciel.

Functions

- **Matrice operator *** (const double, const **Matrice** &)
  Operateurs commutatifs.

Function Documentation

Matrice operator * (const *double*, const Matrice & *m*)

Operateurs commutatifs.

MATRIX.cpp File Reference

```
#include <stdio.h>
#include <math.h>
#include "matrix.h"
```

Defines

- #define **FALSE** 0
- #define **TRUE** 1
- #define **TOL_DET** 0.001
- #define **PERMBUFSIZE** 100

Functions

- void **print_mat** (**dmat** mat)
- **dmat newdmat** (int rs, int re, int cs, int ce, int *error)
- int **scamul** (**dmat** a, double b, **dmat** c)
- int **matmul** (**dmat** a, **dmat** b, **dmat** c)
- int **matadd** (**dmat** a, **dmat** b, **dmat** c)
- int **matcopy** (**dmat** A, **dmat** RSLT)
- int **transpose** (**dmat** A, **dmat** ATrans)
- double **matinvert** (**dmat** a)
- int **solve_system** (**dmat** M, **dmat** a, **dmat** b)

Define Documentation

#define FALSE 0

#define PERMBUFSIZE 100

#define TOL_DET 0.001

#define TRUE 1

Function Documentation

int matadd (dmat *a*, dmat *b*, dmat *c*)

int matcopy (dmat *A*, dmat *RSLT*)

double matinvert (dmat *a*)

int matmul (dmat *a*, dmat *b*, dmat *c*)

dmat newdmat (int *rs*, int *re*, int *cs*, int *ce*, int * *error*)

void print_mat (dmat *mat*)

int scamul (dmat *a*, double *b*, dmat *c*)

int solve_system (dmat *M*, dmat *a*, dmat *b*)

int transpose (dmat *A*, dmat *ATrans*)

MATRIX.H File Reference

```
#include <stdlib.h>
#include <errno.h>
```

Compounds

- struct **dmat**

Defines

- #define **freemat**(m) m.el+=m.lb1; free((m).mat_sto) ; free((m).el)

Functions

- **dmat newdmat** (int rs, int re, int cs, int ce, int *error)
- void **print_mat** (**dmat** mat)
- int **scamul** (**dmat** a, double b, **dmat** c)
- int **matmul** (**dmat** a, **dmat** b, **dmat** c)
- int **matadd** (**dmat** a, **dmat** b, **dmat** c)
- int **matcopy** (**dmat** A, **dmat** RSLT)
- int **transpose** (**dmat** A, **dmat** ATrans)
- double **matinvert** (**dmat** a)
- int **solve_system** (**dmat** M, **dmat** a, **dmat** b)

Define Documentation

#define freemat(m) m.el+=m.lb1; free((m).mat_sto) ; free((m).el)

Function Documentation

int matadd (dmat *a*, dmat *b*, dmat *c*)

int matcopy (dmat *A*, dmat *RSLT*)

double matinvert (dmat *a*)

int matmul (dmat *a*, dmat *b*, dmat *c*)

dmat newdmat (int *rs*, int *re*, int *cs*, int *ce*, int * *error*)

void print_mat (dmat *mat*)

int scamul (dmat *a*, double *b*, dmat *c*)

int solve_system (dmat *M*, dmat *a*, dmat *b*)

int transpose (dmat *A*, dmat *ATrans*)

MilGarbageCollector.cpp File Reference

```
#include "MilGarbageCollector.h"
#include <algorithm>
#include <iostream.h>
```

MilGarbageCollector.h File Reference

```
#include <mil.h>
#include <vector>
```

Namespaces

- namespace **Cognois**

MILSys.h File Reference

Compounds

- struct **MILSys**
  Mil system data.

MINPACK.cpp File Reference

```
#include "MINPACK.h"
#include "Erreur.hpp"
#include "F2C.H"
```

Defines

- #define **EPSFCN** 1.0E-16
- #define **X_TOL** 1.0E-7
- #define **F_TOL** 1.0E-5

Define Documentation

#define EPSFCN 1.0E-16

#define F_TOL 1.0E-5

#define X_TOL 1.0E-7

MINPACK.h File Reference

Compounds

- class **MINPACK**
  Levenberg-Marquardt algorithm to minimize the sum of the squares.

NoopToServer.h File Reference

```
#include "FrameSetClient.h"
#include "ProcessingObject.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::NoopToServer**
  Perform a sequence of nothing, but keep the server informed of the current frame ID.

OCRFontCreator.cpp File Reference

```
#include "OCRFontCreator.h"
#include "milocr.h"
#include <iostream>
#include <string>
#include <conio.h>
#include "cognois.h"
```

Defines

- #define **MAX_SAMPLES_PER_CHARACTER** 5

Define Documentation

#define MAX_SAMPLES_PER_CHARACTER 5

OCRFontCreator.h File Reference

```
#include "OnlineProcessing.h"
#include "ProcessingObject.h"
#include "GrabInit.h"
#include <vector>
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::OCRFontCreator**
  Optical character recognition of the 5 digits of the HP 5214L Preset Counter.

OfflineProcessing.cpp File Reference

```
#include "OfflineProcessing.h"
#include "SimpleXmlParser.h"
#include "GrabInit.h"
#include <winbase.h>
#include <conio.h>
#include "Cognois.h"
#include <fstream.h>
#include <iostream.h>
```

Defines

- #define **SYNCHRONIZATION**
  Force synchronization between the cameras.

- #define **C_CONTROL_BIT** 4
  CAMERA CONTROL BIT 2 : bit 4.

- #define **USER_IN_BIT** 0
  USER0 INPUT BIT 0: bit 0.

- #define **SYNC_SERVER_PERIOD** 100
  Resynchronization period with the server.

Functions

- DWORD WINAPI **OfflineThreadAcquisition** (LPVOID _lpParameter)
  Grabbing thread global function.

Define Documentation

#define C_CONTROL_BIT 4

CAMERA CONTROL BIT 2 : bit 4.

#define SYNCHRONIZATION

Force synchronization between the cameras.

#define SYNC_SERVER_PERIOD 100

Resynchronization period with the server.

#define USER_IN_BIT 0

USER0 INPUT BIT 0: bit 0.

Function Documentation

DWORD WINAPI OfflineThreadAcquisition (LPVOID *_lpParameter*)

Grabbing thread global function.

OfflineProcessing.h File Reference

```
#include "SynchClient.h"
#include <mil.h>
#include <wtypes.h>
#include "GrabInit.h"
#include "ProcessingObject.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::OfflineProcessing**
  Processing of images from AVI files.

OnlineProcessing.cpp File Reference

```
#include "OnlineProcessing.h"
#include "Cognois.h"
#include <iostream.h>
#include "Acquisition.h"
```

Defines

- #define **VERBOSE_SYNC**
  Print synchronization events on screen.

- #define **SYNCHRONIZATION**
  Force synchronization between the cameras.

- #define **READY_OUT_BIT** 4
  CAMERA CONTROL BIT 2 : bit 4; for process sync.

- #define **READY_IN_BIT** 0
  USER0 INPUT BIT 0: bit 0; for process sync.

- #define **COUNTING_OUT_BIT** 3
  CAMERA CONTROL BIT 1 : bit 3; for counting sync.

- #define **COUNTING_IN_BIT** 1
  USER0 INPUT BIT 1: bit 1; for counting sync.

- #define **setReady**(Sys, BitValue) Sys.EcrireBits(READY_OUT_BIT, BitValue)
- #define **getValid**(Sys, BitValue) Sys.LireBits(READY_IN_BIT, BitValue)
- #define **setCounting**(Sys, BitValue) Sys.EcrireBits(COUNTING_OUT_BIT, BitValue)
- #define **getCounting**(Sys, BitValue) Sys.LireBits(COUNTING_IN_BIT, BitValue)

Functions

- DWORD WINAPI **ThreadTraitement** (LPVOID _lpParameter)
- DWORD WINAPI **ThreadAcquisition** (LPVOID _lpParameter)
- DWORD WINAPI **ThreadSynchronization** (LPVOID _lpParameter)

Define Documentation

#define COUNTING_IN_BIT 1

USER0 INPUT BIT 1: bit 1; for counting sync.

#define COUNTING_OUT_BIT 3

CAMERA CONTROL BIT 1 : bit 3; for counting sync.

#define READY_IN_BIT 0

USER0 INPUT BIT 0: bit 0; for process sync.

#define READY_OUT_BIT 4

CAMERA CONTROL BIT 2 : bit 4; for process sync.

#define SYNCHRONIZATION

Force synchronization between the cameras.

#define VERBOSE_SYNC

Print synchronization events on screen.

#define getCounting(Sys, BitValue) Sys.LireBits(COUNTING_IN_BIT, BitValue)

#define getValid(Sys, BitValue) Sys.LireBits(READY_IN_BIT, BitValue)

#define setCounting(Sys, BitValue) Sys.EcrireBits(COUNTING_OUT_BIT, BitValue)

#define setReady(Sys, BitValue) Sys.EcrireBits(READY_OUT_BIT, BitValue)

Function Documentation

DWORD WINAPI ThreadAcquisition (LPVOID *_lpParameter*)

Parameters:

_lpParameter Grabbing thread global function

DWORD WINAPI ThreadSynchronization (LPVOID *_lpParameter*)

Parameters:

_lpParameter Synchronization thread

DWORD WINAPI ThreadTraitement (LPVOID *_lpParameter*)

Parameters:

_lpParameter Processing thread global function

OnlineProcessing.h File Reference

```
#include "SynchClient.h"
#include "Acquisition.h"
#include "ProcessingObject.h"
#include "FrameCounter.h"
#include "MilSys.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::OnlineProcessing**
  Online processing of images.

Operateur.cpp File Reference

```
#include "Operateur.hpp"
```

Operateur.hpp File Reference

```
#include "Table2D.hpp"
#include "Point.hpp"
#include <math.h>
```

Compounds

- class **Operateur**

Picture.cpp File Reference

```
#include "Picture.h"
#include <stdlib.h>
#include <time.h>
#include <wtypes.h>
```

Picture.h File Reference

`#include "ColourSpaceConverter.h"`

Compounds

- class **Picture**
- struct **RGBQUADsd**

Defines

- #define **AFX_PICTURE_H__DFC2EB67_37C7_11D4_BD19_00E0292D1826__INCLUDED_**
- #define **CS_GRAY8** 0x0000
- #define **CS_GRAY16** 0x0008
- #define **CS_COLOR** 0x0010

Typedefs

- typedef unsigned char **BYTE**

Define Documentation

#define AFX_PICTURE_H__DFC2EB67_37C7_11D4_BD19_00E0292D1826__INCLUDED_

Value:

#define CS_COLOR 0x0010

#define CS_GRAY16 0x0008

#define CS_GRAY8 0x0000

Typedef Documentation

typedef unsigned char BYTE

PlanarToPacked.cpp File Reference

```
#include "PlanarToPacked.h"
```

PlanarToPacked.h File Reference

```
#include "ProcessingObject.h"
#include "MILSys.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::PlanarToPacked**
  Transform a MIL planar buffer to a MIL packed buffer.

png.cpp File Reference

`#include "pngpvt.h"`

Variables

- const **UBYTE1 PngSignature** [**PngSignatureSize**]
- const **UBYTE1 PngLengthOrder** [**PngLengthSize**]

Variable Documentation

const UBYTE1 PngLengthOrder[PngLengthSize]

Initial value:

```
{
  16, 17, 18,  0,  8,
   7,  9,  6, 10,  5,
  11,  4, 12,  3, 13,
   2, 14,  1, 15,
}
```

const UBYTE1 PngSignature[PngSignatureSize]

Initial value:

```
{
  137, 80, 78, 71, 13, 10, 26, 10,
}
```

png.h File Reference

```
#include "bitimage.h"
#include "grexcept.h"
```

Compounds

- class **EPngError**

pngchksm.cpp File Reference

```
#include "pngchksm.h"
#include "datatype.h"
```

Functions

- void **MakeCrcTable** ()
- **UBYTE4 CRC32** (**UBYTE4** crc, void *buffer, unsigned int length)
- **UBYTE4 Adler** (**UBYTE4** adler, **UBYTE1** value)

Variables

- **UBYTE4 crc_table** [256]
- bool **table_made** = false

Function Documentation

UBYTE4 Adler (UBYTE4 *adler*, UBYTE1 *value*)

UBYTE4 CRC32 (UBYTE4 *crc*, void * *buffer*, unsigned int *length*)

void MakeCrcTable ()

Variable Documentation

UBYTE4 crc_table `[static]`

bool table_made = false `[static]`

pngchksm.h File Reference

```
#include "datatype.h"
```

Functions

- void **MakeCrcTable** ()
- **UBYTE4 CRC32** (**UBYTE4** crc, void *buffer, unsigned int length)
- **UBYTE4 Adler** (**UBYTE4** adler, **UBYTE1** value)

Function Documentation

UBYTE4 Adler (UBYTE4 *adler*, UBYTE1 *value*)

UBYTE4 CRC32 (UBYTE4 *crc*, void * *buffer*, unsigned int *length*)

void MakeCrcTable ()

pngdecod.cpp File Reference

```
#include <iomanip>
#include <limits.h>
#include "pngchksm.h"
#include "pngdecod.h"
#include "pngpvt.h"
#include "pnghuffd.h"
```

Defines

- #define **VERYVERBOSE**

Enumerations

- enum **CompressionType** { **Uncompressed** = 0, **FixedHuffmanCodes** = 1, **DynamicHuffmanCodes** = 2 }

Variables

- const unsigned int **Max8BitSampleValue** = 255
- const unsigned int **InterlacePasses** = 7

Define Documentation

#define VERYVERBOSE

Value:

Enumeration Type Documentation

enum CompressionType

Enumeration values:

Uncompressed
FixedHuffmanCodes
DynamicHuffmanCodes

Variable Documentation

const unsigned int InterlacePasses = 7

const unsigned int Max8BitSampleValue = 255

pngdecod.h File Reference

```
#include "bitimage.h"
#include "grexcept.h"
#include "png.h"
```

Compounds

- struct **PngDecoder::InterlaceInfo**
- class **PngDecoder**

pngencod.cpp File Reference

```
#include <limits.h>
#include "pngencod.h"
#include "pngpvt.h"
#include "pngchksm.h"
#include "pnghuffe.h"
```

Functions

- **Hash** (**UBYTE1** v1, **UBYTE1** v2, **UBYTE1** v3)
- void **DistanceToCode** (unsigned int distance, unsigned int &code, unsigned int &extra, unsigned int &value)
- void **LengthToCode** (unsigned int length, unsigned int &code, unsigned int &extra, unsigned int &value)

Variables

- const unsigned int **LookaheadSize** = (1 << 9)
- const unsigned int **LookaheadMask** = **LookaheadSize** - 1
- const unsigned int **HashBits** = 5
- const unsigned int **HashTableSize** = 1 << (3 * **HashBits**)
- const unsigned int **OutputBufferSize** = (1 << 13)
- const unsigned int **ENDIMAGE** = 0xFFFF

Function Documentation

void DistanceToCode (unsigned int *distance*, unsigned int & *code*, unsigned int & *extra*, unsigned int & *value*) `[static]`

Hash (UBYTE1 *v1*, UBYTE1 *v2*, UBYTE1 *v3*) `[inline, static]`

void LengthToCode (unsigned int *length*, unsigned int & *code*, unsigned int & *extra*, unsigned int & *value*) `[inline, static]`

Variable Documentation

const unsigned int ENDIMAGE = 0xFFFF

const unsigned int HashBits = 5

const unsigned int HashTableSize = 1 << (3 * HashBits)

const unsigned int LookaheadMask = LookaheadSize - 1

const unsigned int LookaheadSize = (1 << 9)

const unsigned int OutputBufferSize = (1 << 13)

pngencod.h File Reference

```
#include "png.h"
#include "bitimage.h"
#include "Chaine.hpp"
```

Compounds

- struct **PngEncoder::HashEntry**
- class **PngEncoder**

pnghuffd.cpp File Reference

```
#include <iostream>
#include "pnghuffd.h"
#include "pngdecod.h"
```

Variables

- const unsigned int **BADCODE** = 0xFFFFFFF

Variable Documentation

const unsigned int BADCODE = 0xFFFFFFF

pnghuffd.h File Reference

```
#include "datatype.h"
#include "pngpvt.h"
```

Compounds

- class **PngHuffmanDecoder**

pnghuffe.cpp File Reference

```
#include "pnghuffe.h"
#include "pngencod.h"
#include "pngpvt.h"
```

pnghuffe.h File Reference

```
#include "png.h"
```

Compounds

- class **PngHuffmanEncoder**

pngpvt.h File Reference

```
#include <string.h>
#include "datatype.h"
```

Compounds

- struct **PngChromaticitiesData**
- struct **PngImageHeader**
- struct **PngPixelDimensions**
- struct **PngTimeData**

Enumerations

- enum **PngColorType** { **Grayscale** = 0, **RGB** = 2, **Palette** = 3, **GrayscaleAlpha** = 4, **RGBAlpha** = 6 }
- enum **PngFilterType** { **FilterNone** = 0, **FilterSub** = 1, **FilterUp** = 2, **FilterAverage** = 3, **FilterPaeth** = 4 }

Functions

- const char* **Binary** (int vv, int ll)
- unsigned int **PaethPredictor** (**UBYTE1** left, **UBYTE1** above, **UBYTE1** upperleft)
- **UBYTE4 ChunkType** (const char type[5])

Variables

- const int **PngSignatureSize** = 8
- const **UBYTE1 PngSignature** [**PngSignatureSize**]
- const unsigned int **PngLengthSize** = 19
- const **UBYTE1 PngLengthOrder** [**PngLengthSize**]
- const unsigned int **PngMaxLengthCodes** = 288
- const unsigned int **PngMaxDistanceCodes** = 30
- const unsigned int **PngMaxNumberOfHuffmanCodes** = **PngMaxLengthCodes**
- const unsigned int **PngFirstLengthCode** = 257
- const unsigned int **PngEndCode** = 256
- const unsigned int **PngMaxLengthLengthCodeSize** = 7
- const unsigned int **PngMaxDistanceCodeSize** = 15
- const unsigned int **PngMaxLengthCodeSize** = 15
- const unsigned int **PngWindowSize** = (1 << 15)
- const unsigned int **PngWindowMask** = **PngWindowSize** - 1
- const unsigned int **PngLongestLength** = 258

Enumeration Type Documentation

enum PngColorType

Enumeration values:

Grayscale

RGB
Palette
GrayscaleAlpha
RGBAlpha

enum PngFilterType

Enumeration values:

FilterNone
FilterSub
FilterUp
FilterAverage
FilterPaeth

Function Documentation

const char * Binary (int *vv*, int *ll*) `[inline, static]`

UBYTE4 ChunkType (const char *type*[5]) `[inline]`

unsigned int PaethPredictor (UBYTE1 *left*, UBYTE1 *above*, UBYTE1 *upperleft*) `[inline, static]`

Variable Documentation

const unsigned int PngEndCode = 256

const unsigned int PngFirstLengthCode = 257

const UBYTE1 PngLengthOrder

Initial value:

```
{
  16, 17, 18,  0,  8,
   7,  9,  6, 10,  5,
  11,  4, 12,  3, 13,
   2, 14,  1, 15,
}
```

const unsigned int PngLengthSize = 19

const unsigned int PngLongestLength = 258

const unsigned int PngMaxDistanceCodeSize = 15

const unsigned int PngMaxDistanceCodes = 30

const unsigned int PngMaxLengthCodeSize = 15

const unsigned int PngMaxLengthCodes = 288

const unsigned int PngMaxLengthLengthCodeSize = 7

const unsigned int PngMaxNumberOfHuffmanCodes = PngMaxLengthCodes

const UBYTE1 PngSignature

Initial value:

```
{
  137, 80, 78, 71, 13, 10, 26, 10,
}
```

const int PngSignatureSize = 8

const unsigned int PngWindowMask = PngWindowSize - 1

const unsigned int PngWindowSize = (1 << 15)

Point.hpp File Reference

```
#include "Erreur.hpp"
#include "Chaine.hpp"
#include <iostream>
#include <math.h>
#include <ctype.h>
```

Compounds

- class **Point**

Functions

- ostream& **operator**<< (ostream &out, const **Point** &p)
- istream& **operator**>> (istream &in, **Point** &p)

Function Documentation

ostream & operator<< (std::ostream & *out*, const Point & *p*) `[inline]`

istream & operator>> (istream & *in*, Point & *p*) `[inline]`

ProcessingObject.h File Reference

```
#include <mil.h>
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::ProcessingObject**
  Virtual class for user-supplied processing objects.

QRFAC.cpp File Reference

```
#include <math.h>
#include "f2c.h"
```

Functions

- int **qrfac_** (**integer** *m, **integer** *n, **doublereal** *a, **integer** *lda, **logical** *pivot, **integer** *ipvt, **integer** *lipvt, **doublereal** *rdiag, **doublereal** *acnorm, **doublereal** *wa)

Variables

- **integer c__1** = 1

Function Documentation

int qrfac_ (integer * *m*, integer * *n*, doublereal * *a*, integer * *lda*, logical * *pivot*, integer * *ipvt*, integer * *lipvt*, doublereal * *rdiag*, doublereal * *acnorm*, doublereal * *wa*)

Variable Documentation

integer c__1 = 1 `[static]`

QRSOLV.cpp File Reference

```
#include <math.h>
#include "f2c.h"
```

Functions

- int **qrsolv_** (**integer** *n, **doublereal** *r, **integer** *ldr, **integer** *ipvt, **doublereal** *diag, **doublereal** *qtb, **doublereal** *x, **doublereal** *sdiag, **doublereal** *wa)

Function Documentation

int qrsolv_ (integer * *n*, doublereal * *r*, integer * *ldr*, integer * *ipvt*, doublereal * *diag*, doublereal * *qtb*, doublereal * *x*, doublereal * *sdiag*, doublereal * *wa*)

SCVolume.cpp File Reference

```
#include "SCVolume.h"
#include "Matrice.h"
#include "Cognois.h"
#include <iostream.h>
```

SCVolume.h File Reference

```
#include "CameraModel.h"
#include "Matrice.h"
#include <vector>
```

Namespaces

- namespace **Cognois**
- namespace **Cognois::SpaceCarving**

Compounds

- class **Cognois::SpaceCarving::SCVolume**
  Volume for space carving.

SequentialProcessing.h File Reference

```
#include <vector>
#include "ProcessingObject.h"
```

Namespaces

- namespace **Cognois**

Compounds

- class **Cognois::SequentialProcessing**
  Perform a sequence of image processing.

Statistics.h File Reference

```
#include <vector>
#include <algorithm>
```

Compounds

- class **StatisticsR**

Defines

- #define **ABS**(x) ((x)>0 ? (x):-(x))

Define Documentation

#define ABS(x) ((x)>0 ? (x):-(x))

Systeme.hpp File Reference

Defines

- #define **BIGENDIAN_SYSTEM** 0
- #define **FVDESSIN_SYSTEM** 0

Define Documentation

#define BIGENDIAN_SYSTEM 0

#define FVDESSIN_SYSTEM 0

TesterContour.hpp File Reference

```
#include "Chaine.hpp"
#include "Contour.hpp"
#include "Liste.hpp"
#include "ImageGris.hpp"
```

Compounds

- class **TesterContour**
- class **TstParam**

TesterEvolution.hpp File Reference

```
#include "AGParamBin.hpp"
#include "Contour.hpp"
#include "ImageGris.hpp"
#include "TesterContour.hpp"
#include "TableHO.hpp"
```

Compounds

- class **TesterEvolution**
- class **TstParamNote**

xbmdecod.cpp File Reference

```
#include <ctype.h>
#include "Chaine.hpp"
#include "xbmdecod.h"
#include "xbmpvt.h"
```

Enumerations

- enum **Token** { **Invalid**, **Define**, **Comma**, **Integer**, **Static**, **Char**, **Bits**, **LBrace**, **RBrace**, **Equal**, **LBracket**, **RBracket**, **HotX**, **HotY**, **Width**, **Height**, **Semicolon**, **Unsigned** }

Variables

- const **NoData** = 0xFFFF

Enumeration Type Documentation

enum Token

Enumeration values:

Invalid
Define
Comma
Integer
Static
Char
Bits
LBrace
RBrace
Equal
LBracket
RBracket
HotX
HotY
Width
Height
Semicolon
Unsigned

Variable Documentation

const NoData = 0xFFFF

xbmdecod.h File Reference

```
#include "bitimage.h"
#include "grexcept.h"
#include "Chaine.hpp"
```

Compounds

- class **EXbmInvalid**
- class **XbmDecoder**

xbmencod.cpp File Reference

```
#include <iomanip>
#include <ctype.h>
#include "xbmencod.h"
#include "xbmpvt.h"
```

xbmencod.h File Reference

```
#include "bitimage.h"
#include "grexcept.h"
#include "Chaine.hpp"
```

Compounds

- class **XbmEncoder**

xbmpvt.h File Reference

Functions

- **UBYTE1 RotateByte** (**UBYTE1** value)

Function Documentation

UBYTE1 RotateByte (UBYTE1 *value*) `[inline, static]`

Index

~Acquisition
  Acquisition, 11
~AcquisitionBalle
  Cognois::AcquisitionBalle, 17
~AcquisitionImage
  AcquisitionImage, 21
~AcquisitionImageListe
  AcquisitionImageListe, 23
~AcquisitionSurDisque
  Cognois::AcquisitionSurDisque, 26
~AcquisitionSurEcran
  Cognois::AcquisitionSurEcran, 28
~BitmapImage
  BitmapImage, 33
~BitmapImageCoder
  BitmapImageCoder, 38
~BitmapImageDecoder
  BitmapImageDecoder, 39
~BitmapImageEncoder
  BitmapImageEncoder, 40
~BlobColoring
  BlobColoring, 41
~ColourSpaceConverter
  ColourSpaceConverter, 71
~CounterOCR
  Cognois::CounterOCR, 85
~DirectMilAccess
  DirectMilAccess, 87
~FormatImage
  FormatImage, 132
~FrameCounter
  Cognois::FrameCounter, 152
~FrameSetClient
  Cognois::FrameSetClient, 158
~GifDecoder
  GifDecoder, 169
~GifEncoder
  GifEncoder, 174
~JpegDecoder
  JpegDecoder, 214
~JpegDecoderComponent
  JpegDecoderComponent, 221
~JpegDecoderDataUnit
  JpegDecoderDataUnit, 226
~JpegDecoderQuantizationTable
  JpegDecoderQuantizationTable, 228
~JpegEncoder
  JpegEncoder, 233
~JpegEncoderComponent
  JpegEncoderComponent, 242
~JpegEncoderDataUnit
  JpegEncoderDataUnit, 246
~JpegEncoderHuffmanTable
  JpegEncoderHuffmanTable, 248
~JpegEncoderQuantizationTable
  JpegEncoderQuantizationTable, 250
~JpegHuffmanDecoder
  JpegHuffmanDecoder, 252
~MakeAVI
  Cognois::MakeAVI, 261
~Matrice
  Matrice, 267
~MilGarbageCollector
  Cognois::MilGarbageCollector, 273
~OCRFontCreator
  Cognois::OCRFontCreator, 285
~OfflineProcessing
  Cognois::OfflineProcessing, 288
~OnlineProcessing
  Cognois::OnlineProcessing, 295
~Picture
  Picture, 308
~PlanarToPacked
  Cognois::PlanarToPacked, 312
~PngDecoder
  PngDecoder, 316
~PngEncoder
  PngEncoder, 326
~PngHuffmanDecoder
  PngHuffmanDecoder, 334
~PngHuffmanEncoder
  PngHuffmanEncoder, 335
~ProcessingObject
  Cognois::ProcessingObject, 350
~XbmDecoder
  XbmDecoder, 374
~XbmEncoder
  XbmEncoder, 376
Abouter
  Contour, 75
abs
  F2C.H, 436
ABS
  Statistics.h, 578
ac
  Cognois::AcquisitionBalle, 18
ac_table
  JpegDecoderComponent, 224
  JpegEncoderComponent, 245
ac_tables
  JpegDecoder, 218
  JpegEncoder, 237
ACOUTPUTFUNCTION
  JpegEncoderComponent, 241

Acquisition, 9
~Acquisition, 11
Acquisition, 10
Arreter, 12
Cognois::OfflineProcessing, 289
Demarrer, 12
EcrireBits, 12
Fermer, 12
FieldEndHookHandler, 12, 13
FieldEvenEndHookHandler, 12, 13
FieldOddEndHookHandler, 12, 13
FrameEndHookHandler, 12, 13
FrameStartHookHandler, 12, 13
GrabEndHookHandler, 12, 13
GrabStartHookHandler, 12, 13
LibererDerniereImage, 12
LireBits, 12
LireDimensions, 12
LireEvenement, 12
LireSysteme, 12
mCompteurDebut, 14
mCompteurFrequence, 14
mCount, 14
mEndGrabEvent, 14
mFieldEndHookSucceed, 14
mFieldEvenEndHookSucceed, 14
mFieldOddEndHookSucceed, 14
mFrameEndHookSucceed, 14
mFrameStartHookSucceed, 14
mGrabEnCourt, 14
mGrabEndHookSucceed, 14
mGrabStartHookSucceed, 14
mImageHauteur, 14
mImageLargeur, 14
mImageLongueur, 14
mMilAcces, 14
mMilApplication, 14
mMilBuffer, 14
mMilBufferAcces, 14
mMilBufferAttente, 14
mMilBufferBarre, 14
mMilBufferCourant, 14
mMilBufferDernier, 15
mMilBufferNombre, 15
mMilBufferProchainDernier, 15
mMilDigitizer, 15
mMilSysteme, 15
mOuvert, 15
mTemps, 15
Ouvrir, 12
ReserverDerniereImage, 12
Acquisition.cpp, 384
FieldEndHookHandler, 385
FieldEvenEndHookHandler, 385
FieldOddEndHookHandler, 385
FrameEndHookHandler, 385
FrameStartHookHandler, 385
GrabEndHookHandler, 385
GrabStartHookHandler, 385
METEOR_II_CL, 384
NO_FRAME, 384
Acquisition.h, 386
AcquisitionBalle
Cognois::AcquisitionBalle, 17
AcquisitionBalle.cpp, 387
AcquisitionBalle.h, 388
COLOR_ANALYSIS, 388
SERVER_ON, 388
AcquisitionCible
Cognois::AcquisitionCible, 19, 20
AcquisitionCible.cpp, 389
AcquisitionCible.h, 390
AcquisitionImage, 21
~AcquisitionImage, 21
AcquisitionImage, 21
Creer, 22
GetAdd, 22
GetId, 22
GetTempsDebut, 22
GetTempsFin, 22
mMilBuffer, 22
mMilBufferAdd, 22
mMilBufferBand, 22
mTempsDebut, 22
mTempsFin, 22
SetTempsDebut, 22
SetTempsFin, 22
AcquisitionImage.cpp, 391
AcquisitionImage.h, 392
AcquisitionImageListe, 23
~AcquisitionImageListe, 23
AcquisitionImageListe, 23
CreerListe, 24
EffacerListe, 24
GetAdd, 24
GetId, 24
GetTempsDebut, 24
GetTempsFin, 24
mImageHauteur, 24
mImageLargeur, 24
mListe, 24
mLongueur, 24
mMilSysteme, 24
SetTempsDebut, 24
SetTempsFin, 24
AcquisitionImageListe.cpp, 393
AcquisitionImageListe.h, 394
AcquisitionSurDisque
Cognois::AcquisitionSurDisque, 26
AcquisitionSurDisque.cpp, 395
AcquisitionSurDisque.h, 396
AcquisitionSurEcran

Cognois::AcquisitionSurEcran, 27
AcquisitionSurEcran.cpp, 397
AcquisitionSurEcran.h, 398
Activate
Cognois::ProcessingObject, 350
Cognois::SequentialProcessing, 358
AdaptiveTresholding
ImageBinaire, 191
AddColor
BitmapImage, 34
AddMode
Cognois::CameraClientController, 57
addr
Vardesc, 371
address
F2C.H, 437
Adler
pngchksm.cpp, 556
pngchksm.h, 557
adler_value
PngEncoder, 331
aerr
alist, 30
Afficher
Cognois::ExtractionDeCible, 125
Matrice, 267
AfficherDebug
Cognois::ExtractionDeCible, 125
AfficherFichier
Matrice, 267
AFX_BLOBCOLORING_H__5E107B28_D49C_4885_A5D0_6FF9F4C086B4__INCLUDED_
BlobColoring.h, 402
AFX_OLOURSPACECONVERTER_H__7CF04B29_751C_11D4_B2D2_00D0B7067F40__INCLUDED_
ColourSpaceConverter.h, 415
AFX_PICTURE_H__DFC2EB67_37C7_11D4_BD19_00E0292D1826__INCLUDED_
Picture.h, 551
AGParamBin, 29
AjusterSegments
Contour, 75
alist, 30
aerr, 30
aunit, 30
Allocate
Cognois::MilGarbageCollector, 273
AllocateComponentBuffers
JpegDecoderComponent, 222
AllocateDataBuffer
Picture, 309
allocatedBuffers
Cognois::MilGarbageCollector, 274
allocateFont
Cognois::OCRFontCreator, 285
AllocateReadBuffers
GifDecoder, 170
AllValues
DirectMilAccess, 87
APP0
jpmarker.h, 519
APP1
jpmarker.h, 520
APP2
jpmarker.h, 520
APP3
jpmarker.h, 520
APP4
jpmarker.h, 520
APP5
jpmarker.h, 520
APP6
jpmarker.h, 520
APP7
jpmarker.h, 520
APP8
jpmarker.h, 520
APP9
jpmarker.h, 520
APPA
jpmarker.h, 520
Appartient
Ensemble, 108
Intervalle, 208
APPB
jpmarker.h, 520
APPC
jpmarker.h, 520
APPD
jpmarker.h, 520
APPE
jpmarker.h, 520
APPF
jpmarker.h, 520
application_name
GifApplicationBlock, 166
ApplyThreshold
Picture, 309
array
Cognois::Ball, 31
Arreter
Acquisition, 12
AugmenterMatrice
Matrice, 268
aunit
alist, 30
authentication_code
GifApplicationBlock, 166
author_string
PngEncoder, 331

AutoThresholdEstimation
Cognois::ExtractionDeCible, 125
Picture, 309
AutoThresholding
Picture, 309
b
TstParam, 369
BackgrndColor
BlobColoring, 43
background
Cognois::AcquisitionBalle, 18
background_blue
PngDecoder, 320
background_color_index
GifScreenDescriptor, 179
background_gray
PngDecoder, 320
background_green
PngDecoder, 320
background_red
PngDecoder, 320
BADCODE
pnghuffd.cpp, 564
ball
Cognois::AcquisitionBalle, 18
Ball
Cognois, 4
beg_of_file
ImageStream, 201
Begin
Cognois::CameraClientController, 57
Cognois::OfflineProcessing, 288
Cognois::OnlineProcessing, 295
bem
Cognois::AcquisitionBalle, 18
bg
Cognois::AcquisitionBalle, 18
BIGENDIAN_SYSTEM
Systeme.hpp, 579
BigEndianToSystem
datatype.h, 421
Binarize
Cognois::ExtractionDeCible, 125
Binary
pngpvt.h, 569
bit_buffer
JpegEncoder, 237
PngDecoder, 320
bit_count
BitmapImage, 35
JpegEncoder, 237
bit_data
JpegDecoder, 218
bit_offset
GifDecoder, 172
PngDecoder, 320
bit_position
GifEncoder, 175
JpegDecoder, 218
PngEncoder, 331
bitbigendian
FormatXWD, 149
BitCount
BitmapImage, 34
bitdepth
PngImageHeader, 337
bitimage.cpp, 399
RoundRow, 399
SQUARE, 399
bitimage.h, 400
IMAGEPROGRESSFUNCTION, 400
PROGRESSFUNCTION, 400
BitmapImage, 32
~BitmapImage, 33
AddColor, 34
bit_count, 35
BitCount, 34
BitmapImage, 33
BlueOffset, 33
BytesPerRow, 34
CallProgressFunction, 34
Clear, 34
ClearData, 34
color_area_count, 35
color_areas, 35
color_count, 35
color_map, 35
color_usage, 35
ColorCount, 34
ColorMap, 34
CreateColor, 34
DoCopy, 34
EightBitQuantization, 34
FindColor, 34
FindColorUsage, 34
FreeColorQuantizationData, 34
GetRGB, 34
GreenOffset, 33
Height, 34
image_data, 35
image_height, 35
image_width, 35
ImageData, 34
Initialize, 34
LargestColorRange, 34
operator[], 35
operator=, 35
progress_data, 35
progress_function, 35
QuantizedColor, 35
QuantizeSourceImage, 34
RedOffset, 33

row_width, 35
RowRounding, 33
SetProgressFunction, 35
SetSize, 35
SplitAreaInHalf, 35
Width, 35
BitmapImage::ColorArea, 67
color_count, 67
color_values, 67
high, 67
low, 67
pixel_count, 67
BitmapImage::ColorMapEntry, 68
blue, 68
green, 68
red, 68
BitmapImage::ColorUsage, 69
colors, 69
next, 69
usage, 69
BitmapImage::ColorUsageTable, 70
color_count, 70
lists, 70
BitmapImageCoder, 37
~BitmapImageCoder, 38
BitmapImageCoder, 38
DoCopy, 38
Initialize, 38
operator=, 38
progress_data, 38
progress_function, 38
SetProgressFunction, 38
UpdateImage, 38
BitmapImageDecoder, 39
~BitmapImageDecoder, 39
ReadImage, 39
BitmapImageEncoder, 40
~BitmapImageEncoder, 40
WriteImage, 40
bitmappad
FormatXWD::xwdHeader, 382
bitmapunit
FormatXWD::xwdHeader, 382
bitorder
FormatXWD::xwdHeader, 382
Bits
FormatXPM, 144
xbmdecod.cpp, 582
bitsperpixel
FormatPCX::pcxHeader, 305
FormatXWD::xwdHeader, 382
bitsperrgb
FormatXWD::xwdHeader, 382
BLANK
FormatPortable.hpp, 452
Blob
BlobColoring, 42
BlobColoring, 41
~BlobColoring, 41
BackgrndColor, 43
Blob, 42
BlobColoring, 41
CollectBlobInfo, 42
FillOffsetTable, 42
GetNbBlob, 42
InPict, 43
NbBlob, 43
NgbrStack, 43
OffsetTable, 43
OffsetTableX, 43
OffsetTableY, 43
OutPict, 43
PHeight, 43
PositionStack, 43
PositionXStack, 43
PositionYStack, 43
Process, 42
Propagate, 42
PSize, 43
PWidth, 43
BlobColoring.cpp, 401
ERROR_INVALID_PARAMETER, 401
ERROR_NO_DATA, 401
ERROR_NOT_ENOUGH_MEMORY, 401
ERROR_SUCCESS, 401
NULL, 401
BlobColoring.h, 402
AFX_BLOBCOLORING_H__5E107B28_D49C_4885_A5D0_6FF9F4C086B4__INCLUDED_, 402
blobCount
Cognois::AcquisitionBalle, 18
BlobCount
Cognois::FrameSetClient, 158
BlobInfo, 44
CentreX, 44
CentreY, 44
MaxX, 44
MaxY, 44
MinX, 44
MinY, 44
NbPixels, 44
block_buffer
PngEncoder, 331
block_buffer_count
PngEncoder, 331
block_buffer_size
PngEncoder, 331
block_size
GifApplicationBlock, 166
GifGraphicControlBlock, 176
blue

BitmapImage::ColorMapEntry, 68
FormatXWD::xwdColor, 380
GifColorTableEntry, 167
Blue
CouleurHSI, 80
BlueOffset
BitmapImage, 33
blueValue
CouleurRGB, 82
BlueValue
DirectMilAccess, 87
bluex
PngChromaticitiesData, 313
bluey
PngChromaticitiesData, 313
BMP
ImageStream, 199
BmpImage
imagetyp.h, 488
Boundary
ImageBinaire, 191
BRGB
FormatBMP, 128
BRLE4
FormatBMP, 128
BRLE8
FormatBMP, 128
BruitFlipBits
Image, 186
BruitGaussien
Image, 186
BruitImpulsion
Image, 186
BruitSaltPepper
Image, 186
BuildScaledTables
JpegDecoderQuantizationTable, 228
JpegEncoderQuantizationTable, 251
BuildTable
JpegEncoderHuffmanTable, 249
PngHuffmanEncoder, 336
BYTE
Picture.h, 551
byte_offset
GifDecoder, 172
PngDecoder, 320
byte_position
PngEncoder, 331
BYTE1
datatype.h, 421
BYTE2
datatype.h, 421
BYTE4
datatype.h, 421
bytebigendian
FormatXWD, 149
byteorder
FormatXWD::xwdHeader, 382
bytesperline
FormatPCX::pcxHeader, 305
FormatXWD::xwdHeader, 382
BytesPerRow
BitmapImage, 34
c
Multitype, 280
c__1
FDJAC2.cpp, 439
LMDIF.cpp, 524
QRFAC.cpp, 573
c__2
LMPAR.cpp, 525
C_CONTROL_BIT
OfflineProcessing.cpp, 542
C_f
F2C.H, 436
C_fp
F2C.H, 437
CalculateDuDimensions
JpegEncoderComponent, 243
CalculateMcuDimensions
JpegDecoder, 214
JpegEncoder, 234
CalculerAngles
EspaceCalibrage, 7
CalculerFondamentales
Cognois::CameraModel, 61
CalculerMatrice
EspaceCalibrage, 7
CalculerPosition
Operateur, 300
CalculerRt
Cognois::Calibration, 47
CalculerXYZ
Cognois::CameraModel, 62
Calibration, 49
Cognois::CameraModel, 64
Calibration.cpp, 403
MATLAB_COMMAND, 403
MATLAB_HOME, 403
ZHANG_COMMAND, 403
ZHANG_HOME, 403
Calibration.h, 404
CalibrationClient
Cognois::CalibrationClient, 51
Cognois::CameraModel, 64
CalibrationClient.cpp, 405
CalibrationClient.h, 406
CallProgressFunction
BitmapImage, 34
GifDecoder, 170
GifEncoder, 175
JpegDecoder, 214

JpegEncoder, 234
PngDecoder, 317
PngEncoder, 327
CameraClientController
Cognois::CameraClientController, 57
CameraClientController.cpp, 407
CameraClientController.h, 408
COGNOIS_CONTROL_NOOP, 408
COGNOIS_CONTROL_TERMINATE, 408
CameraModel
Cognois::CameraModel, 61
CameraModel.cpp, 409
CameraModel.h, 410
PC_FILE_HEAD, 410
PC_HEIKKILA_0, 410
PC_HEIKKILA_SINGLE_0, 411
PC_ZHANG_0, 411
PC_ZHANG_SINGLE_0, 411
CameraToImage
Cognois::CameraModel, 62
Canny
ImageGris, 195
CARRE
EspaceCalibrage.h, 426
Kernel, 255
carveInitialVolume
Cognois::SpaceCarving::SCVolume, 356
CbComponent
JpegEncoder, 232
centerX
Cognois::Ball, 31
centerY
Cognois::Ball, 31
CentreX
BlobInfo, 44
CentreY
BlobInfo, 44
cerr
cllist, 66
ChaineToInt
ChaineUtile.hpp, 412
ChaineUtile.hpp, 412
ChaineToInt, 412
Char
FormatXPM, 144
xbmdecod.cpp, 582
CheckAcTable
JpegDecoderComponent, 222
CheckAdler
PngDecoder, 317
CheckDcTable
JpegDecoderComponent, 222
CheckQuantizationTable
JpegDecoderComponent, 222
Chercher
Equivalence, 110
Chessboard
Point, 341
chrominance_quanttbl
JpegEncoder, 237
chunk_crc
PngDecoder, 320
chunk_data
PngDecoder, 320
chunk_data_size
PngDecoder, 320
chunk_length
PngDecoder, 320
chunk_type
PngDecoder, 320
PngEncoder, 331
ChunkType
pngpvt.h, 569
cibleCourante
Cognois::ExtractionDeCible, 127
ciblesParDefaut
Cognois::ExtractionDeCible, 127
ciend
cilist, 65
cierr
cilist, 65
cifmt
cilist, 65
cilist, 65
ciend, 65
cierr, 65
cifmt, 65
cirec, 65
ciunit, 65
cirec
cilist, 65
CityBlock
Point, 341
ciunit
cilist, 65
Clear
BitmapImage, 34
Picture, 309
ClearCode
gifencod.cpp, 470
ClearData
BitmapImage, 34
cllist, 66
cerr, 66
csta, 66
cunit, 66
Close
ImageBinaire, 191
code_count
GifEncoder, 175
code_size
GifDecoder, 172

CodeSize
gifencod.cpp, 470
coefficient_blocks
JpegDecoderComponent, 224
Cognois, 2
Ball, 4
Cognois.h, 413
Cognois::AcquisitionBalle, 16
~AcquisitionBalle, 17
ac, 18
AcquisitionBalle, 17
background, 18
ball, 18
bem, 18
bg, 18
blobCount, 18
compteur, 18
facteurX, 18
facteurY, 18
frequency, 18
imageDest, 18
imageTraitee, 18
mask, 18
mBlocCount, 18
mIterator, 18
of, 18
Process, 17
start, 18
stop, 18
transmission, 18
Cognois::AcquisitionCible, 19
AcquisitionCible, 19, 20
mCognoisServer, 20
mPatternFinder, 20
Process, 20
Cognois::AcquisitionSurDisque, 25
~AcquisitionSurDisque, 26
AcquisitionSurDisque, 26
compteur, 26
fileName, 26
Process, 26
queueLimit, 26
Cognois::AcquisitionSurEcran, 27
~AcquisitionSurEcran, 28
AcquisitionSurEcran, 27
MilBufferDisplay, 28
MilSys, 28
mMilDisplay, 28
Process, 28
Cognois::Ball, 31
array, 31
centerX, 31
centerY, 31
height, 31
offsetX, 31
offsetY, 31
width, 31
Cognois::Calibration, 46
CalculerRt, 47
CombinerRT, 47
Heikkila, 47
Homographie, 47
HomographyToTransformation, 47
Zhang, 47
Cognois::CalibrationClient, 50
CalibrationClient, 51
Get3dPoint, 52
GetCalibrationParameters, 52
GetCalibrationPoints, 52
GetLatestPoints, 52, 53
GetNewPoints, 53
GetNumberOfCameraOnline, 53
GetReal2DPoints, 53
Send3dPoint, 53
SendCalibrationParameters, 54
SendPoints, 54
Cognois::CameraClientController, 56
AddMode, 57
Begin, 57
CameraClientController, 57
ListModes, 57
mCurrentMode, 58
mGrabber, 58
mModes, 58
mProcessors, 58
mServer, 58
Processor, 57
Cognois::CameraModel, 59
CalculerFondamentales, 61
CalculerXYZ, 62
Calibration, 64
CalibrationClient, 64
CameraModel, 61
CameraToImage, 62
Dedistorsion, 62
Distorsion, 62
DroiteEpipolaire, 62
EcrireParametres, 62
EcrireParametresExtrinseques, 62
EcrireParametresIntrinseques, 62
Fij, 64
Fij_valid, 64
GetParametresExtrinseques, 63
GetParametresIntrinseques, 63
ImageToCamera, 63
MatriceEssentielle, 63
MatriceFondamentale, 63
MatriceIntrinseque, 63
MatriceProjection, 63
MatriceRotation, 63
MatriceTranslation, 63
MultiCam2SingleCam, 63

operator<<, 64
operator>>, 64
parExt, 64
parInt, 64
Projeter, 63
Rt, 63
SetParametresIntrinseques, 63
SingleCam2MultiCam, 63
Size, 64
TransformInverse, 64
ValiderCorrespondance, 64
version, 64
Version, 64
VMatrices, 61
Cognois::CounterOCR, 84
~CounterOCR, 85
CounterOCR, 85
mBotX, 85
mBotY, 85
mCognoisServer, 85
mCounterImage, 85
mFont, 85
mGrayImage, 85
mOCRResult, 85
mTopX, 85
mTopY, 85
Process, 85
Cognois::EvaluationDeQualite, 117
QualiteExtraction, 117
QualiteReconstruction, 117
Cognois::Exception, 121
Exception, 121
Message, 122
mMessage, 122
what, 122
Cognois::ExtractionDeCible, 123
Afficher, 125
AfficherDebug, 125
AutoThresholdEstimation, 125
Binarize, 125
cibleCourante, 127
ciblesParDefaut, 127
ConvertMilToPicture, 125
ConvertToGray, 125
distMinDefaut, 127
ExtractionDeCible, 124
ExtrairePoints, 125
GetBlobs, 125
GetCircles, 125
Iden4Coins, 125
Intersection, 125
Mapping, 125
Matching, 125
MilImageDisp, 127
MilImageWork, 127
mImageSize, 127
mLastDisplay, 127
mLastPosition, 127
mLastSize, 127
ObtenirPointsCoins, 125
ObtenirPointsRGGG, 125, 126
Ordonner, 126
OrdonnerCouleur, 126
RGBtoHSI, 126
Seuils, 126
SortMatrix, 126
Statistiques, 126
SurCible, 126
TrouverPointRouge, 126
TrouverPointsRougeVert, 126
useCibleParDefaut, 127
Cognois::FrameCounter, 150
~FrameCounter, 152
ElapsedMilliseconds, 152
FrameCounter, 152
GetFrameInformation, 152
mFrameRate, 154
mLastFrameIds, 154
mLastTimeStamps, 154
mLeastRecentFrameIndex, 154
mMostRecentFrameIndex, 154
mPerformanceFrequency, 154
mQueueLength, 154
mStream, 154
mTimeAtBeginning, 154
mTransientState, 154
mVerbose, 154
NextIndex, 152
NextLeastRecentFrameIndex, 153
NextMostRecentFrameIndex, 153
operator=, 153
Reset, 153
SetVerbose, 153
TimeSinceLastFrame, 153
Cognois::FrameSetClient, 155
~FrameSetClient, 158
BlobCount, 158
CreateFrame, 158
DestroyFrame, 158
Frame, 158
FrameCount, 159
FrameSetClient, 158
Get2dPoint, 159
GetAngle, 159
GetImageSize, 159
GetLatestFrameSet, 159
GetMode, 160
GetNewFrameSet, 160
mCameraClientInitialized, 162
mFrame, 162
MinPoolingPeriod, 160
mLastPooling, 162

mLatestFrameId, 162
mMinPoolingPeriod, 162
mPerformanceFrequency, 162
mXmlData, 163
operator=, 162
ResetServer, 160
Send2dPoint, 160
SendAngle, 161
SendFrame, 161
SendImageSize, 161
SendMode, 161
WaitToPool, 162
XmlData, 162
Cognois::MakeAVI, 260
~MakeAVI, 261
MakeAVI, 261
mAviFileName, 262
mFileName, 262
mFrameIds, 262
mImageContents, 262
mImageCounter, 262
mImageLimit, 262
mSizeBand, 262
mSizeX, 262
mSizeY, 262
mTimeStamps, 262
Process, 261
SaveData, 261
Cognois::MilGarbageCollector, 272
~MilGarbageCollector, 273
Allocate, 273
allocatedBuffers, 274
FreeAllBuffers, 273
MbufAlloc2d, 273
MbufAllocColor, 273
MbufChild2d, 273
MbufChildColor, 274
MbufFree, 274
Cognois::NoopToServer, 282
mCognoisServer, 283
NoopToServer, 282
Process, 282
Cognois::OCRFontCreator, 284
~OCRFontCreator, 285
allocateFont, 285
getCharacterSamplesMean, 285
mBotX, 285
mBotY, 285
mCharacters, 286
mCharacterSamples, 286
mCounterImage, 286
mFont, 286
mFontFilename, 286
mFontIsAllocated, 286
mGrayImage, 286
mMilSys, 286
mOnlineProcessing, 286
mTopX, 286
mTopY, 286
OCRFontCreator, 285
Process, 285
Cognois::OfflineProcessing, 287
~OfflineProcessing, 288
Acquisition, 289
Begin, 288
End, 289
LoopMe, 289
mAVIFile, 289
milBuffer, 289
MILSys, 289
mMilSys, 289
mSyncServer, 289
mTimeFile, 289
OfflineProcessing, 288
processor, 289
ResetEvents, 289
SetProcessor, 289
ThreadAcquisition, 289
threadGrabbingHasEnded, 290
Cognois::OnlineProcessing, 292
~OnlineProcessing, 295
Begin, 295
End, 295
IsGrabbing, 295
LoopMe, 295
mAcquisition, 296
mArretTraitement, 296
mEventBeginProcessing, 296
mEventEndProcessing, 296
mEventGrabEnd, 296
mEventImageGrabbed, 296
mEventProcessingHasEnded, 296
mEventRequestProcessing, 296
mEventResetFrameId, 296
mEventThreadGrabbingHasEnded, 296
mEventThreadProcessingHasEnded, 296
mEventThreadSynchronizationHasEnded, 296
mFrameCounter, 296
mFrameIdProcessing, 296
mGrabber, 296
MILSys, 295
mImageBuffer, 297
mInitialFrameId, 297
mMilSys, 297
mMutexImageBuffer, 297
mNoValidFrameYet, 297
mProcessor, 297
mResetCompleted, 297
mResetPeriod, 297
mSynchro, 297
mSyncServer, 297
mTimeStampProcessing, 297

OnlineProcessing, 294
ResetEvents, 295
SetProcessor, 295
ThreadAcquisition, 295
ThreadSynchronization, 295
ThreadTraitement, 295
Cognois::PlanarToPacked, 311
~PlanarToPacked, 312
milBufferPacked, 312
PlanarToPacked, 311
Process, 312
Cognois::ProcessingObject, 349
~ProcessingObject, 350
Activate, 350
Deactivate, 350
Process, 350
Cognois::SequentialProcessing, 358
Activate, 358
Deactivate, 358
Process, 359
Cognois::SpaceCarving, 5
Cognois::SpaceCarving::SCVolume, 355
carveInitialVolume, 356
eType1, 355
eType2, 355
getInitialVolumeVertices, 356
mCamera, 357
mImage, 357
mVoxelSize, 357
SCVolume, 356
Type, 355
COGNOIS_CONTROL_NOOP
CameraClientController.h, 408
COGNOIS_CONTROL_TERMINATE
CameraClientController.h, 408
CognoisException
Exception.h, 429
col
Point, 342
col_interval
PngDecoder::InterlaceInfo, 207
CollectBlobInfo
BlobColoring, 42
COLOR_ANALYSIS
AcquisitionBalle.h, 388
color_area_count
BitmapImage, 35
color_areas
BitmapImage, 35
color_count
BitmapImage, 35
BitmapImage::ColorArea, 67
BitmapImage::ColorUsageTable, 70
color_map
BitmapImage, 35
color_usage
BitmapImage, 35
color_values
BitmapImage::ColorArea, 67
ColorCount
BitmapImage, 34
colorentries
FormatXWD::xwdHeader, 382
ColorMap
BitmapImage, 34
colorplanes
FormatPCX::pcxHeader, 305
colors
BitmapImage::ColorUsage, 69
colortype
PngImageHeader, 337
ColourSpaceConverter, 71
~ColourSpaceConverter, 71
ColourSpaceConverter, 71
HSV2RGB, 71
pi_div_3, 71
pi_x_2, 71
RGB2HSV, 71
ColourSpaceConverter.cpp, 414
ERROR_INVALID_PARAMETER, 414
ERROR_SUCCESS, 414
NUM_PRECISION_DOUBLE, 414
PI, 414
ColourSpaceConverter.h, 415
AFX_OLOURSPACECONVERTER_H__7C F04B29_751C_11D4_B2D2_00D0B7067F 40__INCLUDED_, 415
COM
jpmarker.h, 520
CombinerRT
Cognois::Calibration, 47
Comma
FormatXPM, 144
xbmdecod.cpp, 582
COMMENT
FormatPortable.hpp, 452
comment_string
JpegEncoder, 237
PngEncoder, 331
compedges
TesterContour, 366
TesterEvolution, 368
complex, 73
i, 73
r, 73
component_count
JpegDecoder, 218
component_id
JpegDecoderComponent, 224
component_indices
JpegDecoder, 218
component_mask

JpegEncoder::Scan, 354
ComponentLabeling
ImageBinaire, 191
COMPONENTPASSFUNCTION
JpegEncoderComponent, 241
components
JpegDecoder, 218
compression
FormatPCX::pcxHeader, 305
compression_level
PngEncoder, 331
CompressionLevel
PngEncoder, 326
compressionmethod
PngImageHeader, 337
compressionType
FormatBMP::bmpHeader, 45
CompressionType
pngdecod.cpp, 558
compteur
Cognois::AcquisitionBalle, 18
Cognois::AcquisitionSurDisque, 26
Contour, 74
Abouter, 75
AjusterSegments, 75
Contour, 75
EcrireSegments, 75
EdgeLinking, 75
Filtrer, 75
HoraudCurveDetection, 75
HoraudCurveSuivant, 75
HoraudJunctionDetection, 75
intersections, 76
operator<<, 76
operator>>, 76
RegrouperUnitaires, 75
SegmenterCourbes, 75
SegmentsDroite, 75
TrouverIntersections, 75
Vider, 75
Contour.cpp, 416
M_PI, 416
operator<<, 416
operator>>, 416
Contour.hpp, 417
Contour::HoraudAngle, 181
l1, 181
l2, 181
q, 181
Contour::HoraudRelation, 182
ContourSegment, 77
DiffOrientation, 78
Distance, 78
DroiteMaxErreur, 78
EcrireExtremites, 78
EquationDroite, 78
Fin, 78
Intersection, 78
LongueurDroite, 78
operator<<, 78
operator>>, 78
Orientation, 78
Origine, 78
Separer, 78
ConvertMilToPicture
Cognois::ExtractionDeCible, 125
ConvertToGray
Cognois::ExtractionDeCible, 125
ConvertToGrayScale
Picture, 309
copy_count
PngDecoder, 320
copy_position
PngDecoder, 320
CopyInterlacedRowToImage
PngDecoder, 317
CopyNoninterlacedRowToImage
PngDecoder, 317
copyright_string
PngEncoder, 331
couleur
FormatXPM::xpmColor, 378
CouleurHSI, 79
Blue, 80
CouleurHSI, 79
getChar, 80
getInt, 80
Green, 80
hue, 80
intensite, 80
operator unsigned long, 80
operator<<, 80
operator=, 80
operator>>, 80
Red, 80
saturation, 80
Valider, 80
CouleurRGB, 81
blueValue, 82
CouleurRGB, 81
getChar, 82
getInt, 82
greenValue, 82
Hue, 82
Intensite, 82
operator-, 82
operator unsigned long, 82
operator+, 82
operator+=, 82
operator<<, 82
operator=, 82
operator-=, 82

operator>>, 82
redValue, 82
Saturation, 82
CounterOCR
Cognois::CounterOCR, 85
CounterOCR.cpp, 418
CounterOCR.h, 419
COUNTING_IN_BIT
OnlineProcessing.cpp, 545
COUNTING_OUT_BIT
OnlineProcessing.cpp, 545
CountPassesForProgressReporting
JpegEncoder, 234
CR
FormatPortable.hpp, 452
crc_table
pngchksm.cpp, 556
CRC32
pngchksm.cpp, 556
pngchksm.h, 557
CrComponent
JpegEncoder, 232
CreateColor
BitmapImage, 34
CreateFrame
Cognois::FrameSetClient, 158
CreateQuantizationTables
JpegEncoder, 234
Creer
AcquisitionImage, 22
CreerListe
AcquisitionImageListe, 24
CROIX
Kernel, 255
Cross
Matrice, 268
CROSS_SIZE
ExtractionDeCible.cpp, 431
CrossColor16
Picture, 310
CrossColor8
Picture, 310
CrossColorRGB
Picture, 310
CS_COLOR
Picture.h, 551
CS_GRAY16
Picture.h, 551
CS_GRAY8
Picture.h, 551
csta
cllist, 66
CTR
ImageStream, 199
CTR_V1
FormatContour.hpp, 443
cunit
cllist, 66
current_block_size
GifDecoder, 172
current_column
GifDecoder, 172
current_filter
PngEncoder, 331
current_image
GifDecoder, 172
GifEncoder, 175
JpegDecoder, 218
JpegEncoder, 237
PngDecoder, 320
PngEncoder, 331
current_pass
JpegEncoder, 237
current_row
GifDecoder, 172
current_row_buffer
PngDecoder, 320
current_scan
JpegDecoder, 218
d
Multitype, 280
D_fp
F2C.H, 437
dabs
F2C.H, 436
DAC
jpmarker.h, 519
data
JpegEncoderCoefficientBlock, 239
JpegEncoderDataUnit, 247
data_buffer
GifEncoder, 175
data_read
PngDecoder, 320
data_units
JpegDecoderComponent, 224
data_values
JpegDecoderQuantizationTable, 228
JpegEncoderQuantizationTable, 251
Data16
Picture, 310
Data8
Picture, 310
DataRGB
Picture, 310
DataType
EspaceCalibrage, 7
datatype.h, 420
BigEndianToSystem, 421
BYTE1, 421
BYTE2, 421
BYTE4, 421

DUMMY_RETURN, 420
false, 420
LittleEndianToSystem, 421
NULL, 420
SystemToBigEndian, 421
SystemToLittleEndian, 421
true, 420
UBYTE1, 421
UBYTE2, 421
UBYTE4, 421
DataUnitCols
JpegEncoderComponent, 243
DataUnitRows
JpegEncoderComponent, 243
DATAUNITVALUE
JpegEncoderDataUnit, 246
day
PngTimeData, 339
dc_table
JpegDecoderComponent, 224
JpegEncoderComponent, 245
dc_tables
JpegDecoder, 218
JpegEncoder, 237
DCOUTPUTFUNCTION
JpegEncoderComponent, 241
dct_coefficients
JpegEncoderComponent, 245
Deactivate
Cognois::ProcessingObject, 350
Cognois::SequentialProcessing, 358
Decode
JpegHuffmanDecoder, 253
PngHuffmanDecoder, 334
DecodeAcFirst
JpegDecoderComponent, 222
DecodeAcRefine
JpegDecoderComponent, 222
DecodeByte
PngDecoder, 317
DecodeCompressedByte
PngDecoder, 317
DecodeDcFirst
JpegDecoderComponent, 222
DecodeDcRefine
JpegDecoderComponent, 222
DecodeLiteralByte
PngDecoder, 317
DecodeSequential
JpegDecoderComponent, 222
DecomposeUnit
FormatXWD, 148
Dedistorsion
Cognois::CameraModel, 62
default_chrominance_table
jpegenco.cpp, 507
default_luminance_table
jpegenco.cpp, 507
DEFAULT_MIN_POOLING_PERIOD
FrameSetClient.cpp, 464
DefaultCompression
PngEncoder, 326
defCarre
Kernel, 255
defCroix
Kernel, 255
defGaussienne
Operateur, 301
defGaussienneH
Operateur, 301
defGaussienneV
Operateur, 301
defIdentite
Operateur, 301
Define
FormatXPM, 144
xbmdecod.cpp, 582
Defined
JpegDecoderQuantizationTable, 228
JpegHuffmanDecoder, 253
defLaplacien
Operateur, 301
defMoyenne
Operateur, 301
defPrewittH
Operateur, 301
defPrewittV
Operateur, 301
defRobertsH
Operateur, 301
defRobertsV
Operateur, 301
defSobelH
Operateur, 301
defSobelV
Operateur, 301
defT
Kernel, 255
defType
Kernel, 255
Operateur, 301
delay_time
GifGraphicControlBlock, 176
DELTA
FormatBMP, 128
DELTA_VIRTUEL
ExtractionDeCible.cpp, 431
Demarrer
Acquisition, 12
depth
FormatSunRaster::rasHeader, 352
FormatXWD::xwdHeader, 382

des
IndiceAlias, 203
Descale
jpdedu.cpp, 495
description_string
PngEncoder, 331
DestroyFrame
Cognois::FrameSetClient, 158
Det
Matrice, 268
DHP
jpmarker.h, 519
DHT
jpmarker.h, 519
DiffOrientation
ContourSegment, 78
Dilate
ImageBinaire, 191
dims
Vardesc, 371
DirectC
FormatXWD, 147
direction
ImageGradient, 193
DirectionGradient
ImageGradient, 193
DirectMilAccess, 87
~DirectMilAccess, 87
AllValues, 87
BlueValue, 87
DirectMilAccess, 87
GreenValue, 87
MilBluePtr, 88
MilGreenPtr, 88
MilImageBlue, 88
MilImageGreen, 88
MilImageRed, 88
MilImageWorkPitch, 88
MilRedPtr, 88
RedValue, 87
disclaimer_string
PngEncoder, 331
DISPLAY_TOLERANCE
ExtractionDeCible.cpp, 431
DIST2D
EspaceCalibrage.h, 426
DIST3D
EspaceCalibrage.h, 426
Distance
ContourSegment, 78
distance_table
PngDecoder, 320
PngEncoder, 331
DistanceToCode
pngencod.cpp, 561
distMinDefaut
Cognois::ExtractionDeCible, 127
Distorsion
Cognois::CameraModel, 62
dmach
DPMPAR.cpp, 422
dmat, 89
el, 89
lb1, 89
lb2, 89
mat_sto, 89
ub1, 89
ub2, 89
dmax
F2C.H, 436
dmin
F2C.H, 436
DNL
jpmarker.h, 519
DoCopy
BitmapImage, 34
BitmapImageCoder, 38
GifDecoder, 170
GifEncoder, 175
JpegDecoder, 214
JpegEncoder, 234
PngDecoder, 317
PngEncoder, 327
XbmDecoder, 374
XbmEncoder, 377
doublecomplex, 90
i, 90
r, 90
doublereal
F2C.H, 437
DoubleTresholding
ImageBinaire, 191
DoWrite
PngEncoder, 327
DPMPAR.cpp, 422
dmach, 422
dpmpar_, 422
maxmag, 422
mcheps, 422
minmag, 422
dpmpar_
DPMPAR.cpp, 422
F2C.H, 438
DQT
jpmarker.h, 519
DRI
jpmarker.h, 519
DroiteEpipolaire
Cognois::CameraModel, 62
DroiteMaxErreur
ContourSegment, 78
du_cols

JpegDecoderComponent, 224
JpegEncoderComponent, 245
du_rows
JpegDecoderComponent, 224
JpegEncoderComponent, 245
DUMMY_RETURN
datatype.h, 420
Dump
JpegHuffmanDecoder, 253
PngHuffmanDecoder, 334
DynamicHuffmanCodes
pngdecod.cpp, 558
E_f
F2C.H, 437
EcrireBits
Acquisition, 12
EcrireBMP
ImageStream, 200
EcrireContour
ImageStream, 200
EcrireExtremites
ContourSegment, 78
EcrireGIF
ImageStream, 200
EcrireJPG
ImageStream, 200
EcrireParametres
Cognois::CameraModel, 62
EcrireParametresExtrinseques
Cognois::CameraModel, 62
EcrireParametresIntrinseques
Cognois::CameraModel, 62
EcrirePBM
ImageStream, 200
EcrirePGM
ImageStream, 200
EcrirePNG
ImageStream, 200
EcrirePPM
ImageStream, 200
EcrireRaster
ImageStream, 200
EcrireSegments
Contour, 75
EcrireXBM
ImageStream, 200
EcrireXWD
ImageStream, 200
EdgeLinking
Contour, 75
eff
IndiceAlias, 203
EffacerListe
AcquisitionImageListe, 24
EffectuerTest
TesterContour, 365
EGifCircularTree, 91
EGifCircularTree, 91
EGifCorruptFile, 92
EGifCorruptFile, 92
EGifError, 93
EGifError, 93
EGifFileReadError, 94
EGifFileReadError, 94
EGifNoMoreImages, 95
EGifNoMoreImages, 95
EGifNotAGifFile, 96
EGifNotAGifFile, 96
EGraphicsAbort, 97
EGraphicsAbort, 97
EGraphicsException, 98
EGraphicsException, 99
operator=, 99
ehufco
JpegEncoderHuffmanTable, 249
PngHuffmanEncoder, 336
ehufsi
JpegEncoderHuffmanTable, 249
PngHuffmanEncoder, 336
EightBitQuantization
BitmapImage, 34
EInvalidBitCount, 100
EInvalidBitCount, 100
EJpegAbort, 101
EJpegAbort, 101
EJpegBadData, 102
EJpegBadData, 102
EJpegBadOutputParameter, 103
EJpegBadOutputParameter, 103
EJpegError, 104
EJpegError, 104
EJpegFatal, 105
EJpegFatal, 105
EJpegIndexOutOfRange, 106
EJpegIndexOutOfRange, 106
EJpegValueOutOfRange, 107
EJpegValueOutOfRange, 107
el
dmat, 89
Matrice, 271
ElapsedMilliseconds
Cognois::FrameCounter, 152
Encode
JpegEncoderHuffmanTable, 249
PngHuffmanEncoder, 336
EncodeSequential
JpegEncoderComponent, 243
End
Cognois::OfflineProcessing, 289
Cognois::OnlineProcessing, 295
end_code
GifDecoder, 172

end_read
PngDecoder, 320
EndCode
gifencod.cpp, 470
EndianToSystem
FormatXWD, 148
ENDIMAGE
pngencod.cpp, 562
ENORM.cpp, 423
enorm_, 423
enorm_
ENORM.cpp, 423
F2C.H, 438
Ensemble, 108
Appartient, 108
Ensemble, 108
Entre
Intervalle, 208
entree_comparaison
TstParam, 369
entree_image
TstParam, 369
EOB
FormatBMP, 128
eob_run
JpegDecoderComponent, 224
JpegEncoderComponent, 245
eob_start_du_col
JpegEncoderComponent, 245
eob_start_du_row
JpegEncoderComponent, 245
eob_start_position
JpegEncoderComponent, 245
EOI
jpmarker.h, 519
eoi_found
JpegDecoder, 218
EOL
FormatBMP, 128
EPngError, 109
EPngError, 109
EPSFCN
MINPACK.cpp, 537
EPSILON
EspaceCalibrage.h, 426
Equal
FormatXPM, 145
Picture, 309
xbmdecod.cpp, 582
EquationDroite
ContourSegment, 78
Equivalence, 110
Chercher, 110
Inserer, 110
Equivalence.hpp, 424
operator<<, 424
operator>>, 424
Erode
ImageBinaire, 191
ErreurFormat, 111
ErreurFormat, 111
ErreurFormatImage, 112
ErreurFormatImage, 112
ErreurFormatImageEcriture, 113
ErreurFormatImageEcriture, 113
ErreurFormatImageLecture, 114
ErreurFormatImageLecture, 114
ErreurImage, 115
ErreurImage, 115
us, 115
ERROR_INVALID_PARAMETER
BlobColoring.cpp, 401
ColourSpaceConverter.cpp, 414
ERROR_NO_DATA
BlobColoring.cpp, 401
ERROR_NOT_ENOUGH_MEMORY
BlobColoring.cpp, 401
ERROR_SUCCESS
BlobColoring.cpp, 401
ColourSpaceConverter.cpp, 414
EspaceCalibrage, 6
CalculerAngles, 7
CalculerMatrice, 7
DataType, 7
MatriceProduitVectoriel, 7
MDoubles, 7
MPoints, 7
MPointsXYZ, 7
operator<<, 7
operator==, 7
VDoubles, 7
VPoints, 7
VPointsXYZ, 7
EspaceCalibrage.cpp, 425
EspaceCalibrage.h, 426
CARRE, 426
DIST2D, 426
DIST3D, 426
EPSILON, 426
MAX, 426
MIN, 426
PI, 426
POINT_OUT_OF_IMAGE, 426
SWAP, 426
EspaceCalibrage::Point, 343
EstHorsImage, 343
x, 343
y, 343
EspaceCalibrage::PointXYZ, 345
x, 345
y, 345
z, 345

Est
Intervalle, 208
EstHorsImage
EspaceCalibrage::Point, 343
EstVide
Matrice, 268
ESubscriptOutOfRange, 116
ESubscriptOutOfRange, 116
EtatPixel
Kernel, 254
eType1
Cognois::SpaceCarving::SCVolume, 355
eType2
Cognois::SpaceCarving::SCVolume, 355
Euclidean
Point, 341
evaluation_erreur
MINPACK, 278
EvaluationDeQualite.cpp, 427
EvaluationDeQualite.h, 428
EXbmInvalid, 118
EXbmInvalid, 118
Exception, 119
Cognois::Exception, 121
Exception.h, 429
CognoisException, 429
ITOA, 429
EXP
jpmarker.h, 519
Expand
ImageBinaire, 191
expected_restart
JpegDecoder, 218
Extend
jpdecomp.cpp, 493
extension_code
JfifExtension, 210
Extern
F2C.H, 436
ExtractionDeCible
Cognois::ExtractionDeCible, 124
ExtractionDeCible.cpp, 430
CROSS_SIZE, 431
DELTA_VIRTUEL, 431
DISPLAY_TOLERANCE, 431
HUE_BLUE_TO_RED, 431
HUE_GREEN_TO_BLUE, 431
HUE_RED_TO_GREEN, 431
INIT_BENCH, 431
MAX_RECONSTRUCTION_ERR, 431
MIN_BLOB_DIST_TO_BORDER, 431
MIN_BLOB_SIZE, 431
MIN_DETECTE, 431
NB_POINTS_REFERENCE, 431
operator<, 432
PERFORM_BENCH, 431
SEUIL_2_ITERATIONS, 431
SEUIL_2_TOLERANCE, 431
stats, 432
TOLERANCE_INC, 431
TOLERANCE_MAX, 431
TOLERANCE_MIN, 431
UTILISE_HOMOGRAPHIE, 431
ExtractionDeCible.h, 433
ExtrairePoints
Cognois::ExtractionDeCible, 125
f
TstParam, 369
F_TOL
MINPACK.cpp, 537
F2C.H, 434
abs, 436
address, 437
C_f, 436
C_fp, 437
D_fp, 437
dabs, 436
dmax, 436
dmin, 436
doublereal, 437
dpmpar_, 438
E_f, 437
enorm_, 438
Extern, 436
F2C_proc_par_types, 436
FALSE_, 436
fdjac2_, 438
flag, 437
ftnint, 437
ftnlen, 437
H_f, 436
H_fp, 437
I_fp, 437
integer, 437
J_fp, 437
K_fp, 437
L_fp, 437
lmdif_, 438
lmpar_, 438
logical, 437
Long, 437
max, 436
min, 436
Multitype, 437
Namelist, 437
qrfac_, 438
qrsolv_, 438
R_fp, 437
real, 438
S_fp, 437
shortint, 438
shortlogical, 438

TRUE_, 436
U_fp, 437
Vardesc, 437
VOID, 436
Z_f, 436
Z_fp, 437
F2C_proc_par_types
F2C.H, 436
facteur
Operateur, 302
Facteur
Operateur, 300
facteurX
Cognois::AcquisitionBalle, 18
facteurY
Cognois::AcquisitionBalle, 18
false
datatype.h, 420
FALSE
GrabInit.h, 474
MATRIX.cpp, 530
FALSE_
F2C.H, 436
FastCompression
PngEncoder, 326
FastestCompression
PngEncoder, 326
FC4
jpdedu.cpp, 495
jpendu.cpp, 512
FDJAC2.cpp, 439
c__1, 439
fdjac2_, 439
fdjac2_
F2C.H, 438
FDJAC2.cpp, 439
Fermer
Acquisition, 12
FieldEndHookHandler
Acquisition, 12, 13
Acquisition.cpp, 385
FieldEvenEndHookHandler
Acquisition, 12, 13
Acquisition.cpp, 385
FieldOddEndHookHandler
Acquisition, 12, 13
Acquisition.cpp, 385
Fij
Cognois::CameraModel, 64
Fij_valid
Cognois::CameraModel, 64
file_gamma
PngDecoder, 320
file_signature
GifHeader, 177
file_version
GifHeader, 177
fileName
Cognois::AcquisitionSurDisque, 26
FillBuffer
PngEncoder, 327
filler
FormatPCX::pcxHeader, 305
FillOffsetTable
BlobColoring, 42
filter_buffers
PngEncoder, 331
filter_mask
PngEncoder, 331
filter_width
PngEncoder, 331
FilterAverage
pngpvt.h, 569
FilterBufferCount
PngEncoder, 325
filtermethod
PngImageHeader, 337
FilterNone
pngpvt.h, 569
FilterPaeth
pngpvt.h, 569
FilterRow
PngDecoder, 317
PngEncoder, 327
FilterSub
pngpvt.h, 569
FilterUp
pngpvt.h, 569
FiltreConservateur
Image, 186
FiltreGaussien
Image, 186
FiltreMaxMin
Image, 186
FiltreMediane
Image, 186
FiltreMoyenne
Image, 186
Filtrer
Contour, 75
FiltreUnsharp
Image, 186
Fin
ContourSegment, 78
final_data_set
PngDecoder, 320
FindColor
BitmapImage, 34
FindColorUsage
BitmapImage, 34
FindComponentsInScan
JpegEncoder, 234

FindLengthCodes
PngEncoder, 327
FirstAcData
JpegEncoder, 234
FirstCode
gifencod.cpp, 470
FixedArea
gifdecod.cpp, 468
FixedHuffmanCodes
pngdecod.cpp, 558
flag
F2C.H, 437
flags
FormatXWD::xwdColor, 380
float_scaling
JpegDecoderQuantizationTable, 228
JpegEncoderQuantizationTable, 251
FloatInverseDCT
JpegDecoderDataUnit, 226
floatscaling
jpdequan.cpp, 499
jpenquan.cpp, 516
FlushBitBuffer
JpegEncoder, 234
PngEncoder, 327
Fonction
TesterEvolution, 367
format
FormatXPM::xpmHeader, 379
FormatXWD::xwdHeader, 382
ImageStream, 201
Format
FormatXPM, 145
FormatBMP, 128
BRGB, 128
BRLE4, 128
BRLE8, 128
DELTA, 128
EOB, 128
EOL, 128
ReadImage, 128
WriteImage, 129
FormatBMP.cpp, 440
FormatBMP.hpp, 441
FormatBMP::bmpHeader, 45
compressionType, 45
height, 45
horizontalResolution, 45
imageSize, 45
nbBitsPerPixel, 45
nbColorIndexes, 45
nbImportantColorIndexes, 45
verticalResolution, 45
width, 45
FormatContour, 130
ReadImage, 130
ReadV1, 130
WriteImage, 130
FormatContour.cpp, 442
FormatContour.hpp, 443
CTR_V1, 443
FormatImage, 132
~FormatImage, 132
FormatImage, 132
GetVerbose, 132
ReadImage, 132
SetVerbose, 133
verbose_flag, 133
WriteImage, 133
FormatImage.hpp, 444
FormatPBM, 134
hd, 135
ReadASCII, 134
ReadImage, 134
ReadRAW, 134
WriteImage, 134, 135
FormatPBM.cpp, 445
FormatPBM.hpp, 446
FormatPBM::pbmHeader, 303
height, 303
width, 303
FormatPCX, 136
PCOLOR, 136
PCX16c, 137
PCX16c_pal, 137
PCX16Mc, 137
PCX256c_pal, 137
PCXMONO, 137
PGSCALE, 136
PMAGIC, 136
PRLE, 136
ReadImage, 137
WriteImage, 137
FormatPCX.cpp, 447
FormatPCX.hpp, 448
FormatPCX::pcxHeader, 304
bitsperpixel, 305
bytesperline, 305
colorplanes, 305
compression, 305
filler, 305
horesolution, 305
palette, 305
palettetype, 305
reserved, 305
signature, 305
veresolution, 305
version, 305
xmax, 305
xmin, 305
ymax, 305
ymin, 305

FormatPGM, 138
hd, 139
ReadASCII, 138
ReadImage, 138
ReadRAW, 138
WriteImage, 138, 139
FormatPGM.cpp, 449
FormatPGM.hpp, 450
FormatPGM::pgmHeader, 306
height, 306
maxvalue, 306
width, 306
FormatPortable, 140
NextInformation, 140
NextLine, 140
FormatPortable.hpp, 451
BLANK, 452
COMMENT, 452
CR, 452
NEWLINE, 452
PBM_ASCII, 452
PBM_RAW, 452
PGM_ASCII, 452
PGM_RAW, 452
PPM_ASCII, 452
PPM_RAW, 452
TAB, 452
FormatPPM, 141
hd, 142
ReadASCII, 141
ReadImage, 141
ReadRAW, 141
WriteImage, 141
FormatPPM.cpp, 453
FormatPPM.hpp, 454
FormatPPM::ppmHeader, 346
height, 346
maxvalue, 346
width, 346
FormatSunRaster, 143
ReadImage, 143
WriteImage, 143
FormatSunRaster.cpp, 455
FormatSunRaster.hpp, 456
RESC, 457
RIFF, 457
RMAGIC, 457
RNOMAP, 457
ROLD, 457
RRAWMAP, 457
RRGB, 457
RRGBMAP, 457
RRLENCODED, 457
RSTANDARD, 457
RTIFF, 457
FormatSunRaster::rasHeader, 352
depth, 352
height, 352
length, 352
magic, 352
maplength, 352
maptype, 352
type, 352
width, 352
FormatXPM, 144
Bits, 144
Char, 144
Comma, 144
Define, 144
Equal, 145
Format, 145
Guillemet, 145
Height, 145
Integer, 144
Invalid, 144
last_char, 145
LBrace, 144
LBracket, 145
lexical_value, 145
Mono, 145
NColors, 145
NextChar, 145
NextColor, 145
NextPixel, 145
NextToken, 145
openstring, 145
Pixel, 145
Pointeur, 145
RBrace, 145
RBracket, 145
ReadImage, 145
Semicolon, 145
Slash, 145
Static, 144
Token, 144
Unsigned, 145
Width, 145
WriteImage, 145
FormatXPM.cpp, 458
FormatXPM.hpp, 459
NoData, 459
FormatXPM::xpmColor, 378
couleur, 378
indice, 378
FormatXPM::xpmHeader, 379
format, 379
height, 379
ncolors, 379
pixel, 379
width, 379
FormatXWD, 147
bitbigendian, 149

bytebigendian, 149
DecomposeUnit, 148
DirectC, 147
EndianToSystem, 148
LSBFirst, 148
MSBFirst, 148
pixmask, 149
PseudoC, 147
ReadImage, 148
ScaleG, 147
StaticC, 147
StaticG, 147
SystemToEndian, 148
TrueC, 147
WriteImage, 148
XYPixmap, 148
ZPixmap, 148
FormatXWD.cpp, 460
FormatXWD.hpp, 461
X11WD_FILE_VERSION, 461
FormatXWD::xwdColor, 380
blue, 380
flags, 380
green, 380
number, 380
pad, 380
red, 380
FormatXWD::xwdHeader, 381
bitmappad, 382
bitmapunit, 382
bitorder, 382
bitsperpixel, 382
bitsperrgb, 382
byteorder, 382
bytesperline, 382
colorentries, 382
depth, 382
format, 382
height, 382
hsize, 382
ncolors, 382
version, 382
visualclass, 382
width, 382
windowborder, 382
windowheight, 382
windowwidth, 382
windowx, 382
windowy, 382
xoffset, 382
zbluemask, 382
zgreenmask, 382
zredmask, 383
ForwardDct
JpegEncoderDataUnit, 247
Frame
Cognois::FrameSetClient, 158
frame_height
JpegDecoder, 218
JpegEncoder, 237
frame_type
JpegDecoder, 218
frame_width
JpegDecoder, 218
JpegEncoder, 237
FrameCount
Cognois::FrameSetClient, 159
FrameCounter
Cognois::FrameCounter, 152
FrameCounter.cpp, 462
FrameCounter.h, 463
FrameEndHookHandler
Acquisition, 12, 13
Acquisition.cpp, 385
FrameHeight
JpegDecoder, 214
FrameSetClient
Cognois::FrameSetClient, 158
FrameSetClient.cpp, 464
DEFAULT_MIN_POOLING_PERIOD, 464
FrameSetClient.h, 465
FrameStartHookHandler
Acquisition, 12, 13
Acquisition.cpp, 385
FrameWidth
JpegDecoder, 214
FreeAllBuffers
Cognois::MilGarbageCollector, 273
FreeAllocatedResources
JpegDecoder, 214
FreeBuffers
PngEncoder, 327
FreeColorQuantizationData
BitmapImage, 34
FreeComponentBuffers
JpegDecoderComponent, 222
FreeData
PngDecoder, 317
FreeDynamicStorage
JpegEncoderComponent, 243
FreeGrabber
GrabInit.cpp, 473
GrabInit.h, 474
freemat
MATRIX.H, 532
FreeReadBuffers
GifDecoder, 170
frequencies
JpegEncoderHuffmanTable, 249
PngHuffmanEncoder, 336
frequency
Cognois::AcquisitionBalle, 18

FSEC2
jpdedu.cpp, 495
jpendu.cpp, 512
FSEC6
jpdedu.cpp, 495
jpendu.cpp, 512
fstream, 164
ftnint
F2C.H, 437
ftnlen
F2C.H, 437
FVDESSIN_SYSTEM
Systeme.hpp, 579
g
TstParam, 369
GatherAcData
JpegEncoderComponent, 243
GatherDcData
JpegEncoderComponent, 243
GatherLengthCounts
PngEncoder, 327
Gaussienne
Operateur, 300
GAUSSIENNE
Operateur, 299
GAUSSIENNEH
Operateur, 299
GAUSSIENNEV
Operateur, 299
GenericCognoisClient, 165
GenInv
Matrice, 268
Get2dPoint
Cognois::FrameSetClient, 159
Get3dPoint
Cognois::CalibrationClient, 52
GetAdd
AcquisitionImage, 22
AcquisitionImageListe, 24
GetAngle
Cognois::FrameSetClient, 159
GetAuthor
PngEncoder, 327
GetBits
PngDecoder, 317
GetBlobs
Cognois::ExtractionDeCible, 125
GetBlockSize
PngEncoder, 327
GetCalibrationParameters
Cognois::CalibrationClient, 52
GetCalibrationPoints
Cognois::CalibrationClient, 52
getChar
CouleurHSI, 80
CouleurRGB, 82
getCharacterSamplesMean
Cognois::OCRFontCreator, 285
GetCircles
Cognois::ExtractionDeCible, 125
GetComment
JpegEncoder, 234
PngEncoder, 327
GetCompressionLevel
PngEncoder, 327
GetCopyright
PngEncoder, 327
getCouleur
ImageRGB, 197
getCounting
OnlineProcessing.cpp, 546
GetDataPtr
Picture, 309
GetDescription
PngEncoder, 327
GetDisclaimer
PngEncoder, 327
GetExtension
ImageStream, 200
GetFrameInformation
Cognois::FrameCounter, 152
GetGrayscale
JpegEncoder, 234
GetHeight
Picture, 309
GetHistogram
Picture, 309
GetHorizontalFrequency
JpegEncoderComponent, 243
GetHotSpotX
XbmDecoder, 374
XbmEncoder, 377
GetHotSpotY
XbmDecoder, 374
XbmEncoder, 377
GetId
AcquisitionImage, 22
AcquisitionImageListe, 24
GetIDATByte
PngDecoder, 317
GetImageByte
GifDecoder, 170
GetImageName
XbmDecoder, 374
XbmEncoder, 377
GetImageSize
Cognois::FrameSetClient, 159
getInitialVolumeVertices
Cognois::SpaceCarving::SCVolume, 356
getInt
CouleurHSI, 80
CouleurRGB, 82

GetLastError
Picture, 309
GetLatestFrameSet
Cognois::FrameSetClient, 159
GetLatestPoints
Cognois::CalibrationClient, 52, 53
GetMode
Cognois::FrameSetClient, 160
GetNbBlob
BlobColoring, 42
GetNewFrameSet
Cognois::FrameSetClient, 160
GetNewPoints
Cognois::CalibrationClient, 53
GetNumberOfCameraOnline
Cognois::CalibrationClient, 53
GetParametresExtrinseques
Cognois::CameraModel, 63
GetParametresIntrinseques
Cognois::CameraModel, 63
GetProgressive
JpegEncoder, 234
GetQuality
JpegEncoder, 234
GetReadMultipleImages
GifDecoder, 170
GetReal2DPoints
Cognois::CalibrationClient, 53
GetRGB
BitmapImage, 34
GetRowsPerRestart
JpegEncoder, 234
GetSamplingFrequency
JpegEncoder, 234
GetScanAttributes
JpegEncoder, 234
GetScanCount
JpegDecoder, 214
GetSize
Picture, 309
GetSoftware
PngEncoder, 327
GetSource
PngEncoder, 327
GetStreamImageType
imagetyp.h, 488
GetTempsDebut
AcquisitionImage, 22
AcquisitionImageListe, 24
GetTempsFin
AcquisitionImage, 22
AcquisitionImageListe, 24
GetTitle
PngEncoder, 327
GetType
Picture, 309
GetUseFilters
PngEncoder, 327
getValid
OnlineProcessing.cpp, 546
GetVerbose
FormatImage, 132
GifDecoder, 170
ImageStream, 200
JpegDecoder, 214
PngDecoder, 317
GetVerboseFlag
GifDecoder, 170
GetVerticalFrequency
JpegEncoderComponent, 243
GetWarning
PngEncoder, 327
GetWidth
Picture, 309
GIF
ImageStream, 199
gif.h, 466
GifApplicationExtension, 466
GifBitsUsed, 467
GifBlockType, 466
GifCommentExtension, 466
GifExtension, 466
GifGraphicControlExtension, 466
GifImageSeparator, 466
GifInputBufferSize, 467
GifMaxColors, 467
GifPlainTextExtension, 466
GifTerminator, 466
GifTrailer, 466
GifApplicationBlock, 166
application_name, 166
authentication_code, 166
block_size, 166
GifApplicationExtension
gif.h, 466
GifBitsUsed
gif.h, 467
GifBlockType
gif.h, 466
GifColorTableEntry, 167
blue, 167
green, 167
red, 167
GifCommentExtension
gif.h, 466
gifdecod.cpp, 468
FixedArea, 468
TableSize, 468
gifdecod.h, 469
GifDecoder, 168
~GifDecoder, 169
AllocateReadBuffers, 170

bit_offset, 172
byte_offset, 172
CallProgressFunction, 170
code_size, 172
current_block_size, 172
current_column, 172
current_image, 172
current_row, 172
DoCopy, 170
end_code, 172
FreeReadBuffers, 170
GetImageByte, 170
GetReadMultipleImages, 170
GetVerbose, 170
GetVerboseFlag, 170
GifDecoder, 169
global_color_count, 172
global_color_table, 172
image_height, 172
image_left, 172
image_top, 172
image_width, 172
ImageLeft, 170
ImageTop, 170
Initialize, 170
InitializeLocalData, 170
input_buffer, 172
input_stream, 172
interlace_group, 172
interlaced_image, 172
length_distance_table, 172
LoadBuffer, 170
lz_window, 172
more_images, 172
MoreImages, 170
NextBit, 170
NextCode, 170
operator=, 171
read_multiple_images, 172
ReadCompressedData, 170
ReadImage, 170
ReadNextImage, 171
ReadStreamHeader, 171
ReadToNextImage, 171
Reset, 171
rows_done, 173
screen_height, 173
screen_width, 173
ScreenHeight, 171
ScreenWidth, 171
SetReadMultipleImages, 171
SetVerbose, 171
SetVerboseFlag, 171
stack_pointer, 173
StoreByte, 171
UpdateImage, 171
verbose_flag, 173
GifDecoder::LZ77String, 259
length, 259
offset, 259
gifencod.cpp, 470
ClearCode, 470
CodeSize, 470
EndCode, 470
FirstCode, 470
LastCode, 470
gifencod.h, 471
GifEncoder, 174
~GifEncoder, 174
bit_position, 175
CallProgressFunction, 175
code_count, 175
current_image, 175
data_buffer, 175
DoCopy, 175
GifEncoder, 174
Initialize, 175
operator=, 175
output_stream, 175
OutputBits, 175
OutputCode, 175
quantized_image, 175
WriteImage, 175
gifexcep.h, 472
GifExtension
gif.h, 466
GifGraphicControlBlock, 176
block_size, 176
delay_time, 176
packedfield, 176
transparent_color, 176
GifGraphicControlExtension
gif.h, 466
GifHeader, 177
file_signature, 177
file_version, 177
GifImage
imagetyp.h, 488
GifImageDescriptor, 178
image_height, 178
image_width, 178
left_position, 178
packedfield, 178
top_position, 178
GifImageSeparator
gif.h, 466
GifInputBufferSize
gif.h, 467
GifMaxColors
gif.h, 467
GifPlainTextExtension
gif.h, 466

GifScreenDescriptor, 179
background_color_index, 179
logical_screen_height, 179
logical_screen_width, 179
packedfield, 179
pixel_aspect_ratio, 179
GifTerminator
gif.h, 466
GifTrailer
gif.h, 466
global_color_count
GifDecoder, 172
global_color_table
GifDecoder, 172
GrabEndHookHandler
Acquisition, 12, 13
Acquisition.cpp, 385
GrabInit.cpp, 473
FreeGrabber, 473
InitGrabber, 473
GrabInit.h, 474
FALSE, 474
FreeGrabber, 474
InitGrabber, 474
PRO_ECRAN, 474
PRO_METEOR2, 474
TRUE, 474
GrabStartHookHandler
Acquisition, 12, 13
Acquisition.cpp, 385
Gradient
ImageGradient, 193
GradientMaxima
ImageGradient, 193
GradXY
ImageGradient, 193
GraphLine
Picture, 309
gray_scale
JpegEncoder, 237
Grayscale
pngpvt.h, 568
GrayscaleAlpha
pngpvt.h, 569
GrayscaleConvert
JpegDecoderComponent, 222
green
BitmapImage::ColorMapEntry, 68
FormatXWD::xwdColor, 380
GifColorTableEntry, 167
Green
CouleurHSI, 80
GreenOffset
BitmapImage, 33
greenValue
CouleurRGB, 82
GreenValue
DirectMilAccess, 87
greenx
PngChromaticitiesData, 313
greeny
PngChromaticitiesData, 313
grexcept.h, 475
Guillemet
FormatXPM, 145
h
Multitype, 280
H_f
F2C.H, 436
H_fp
F2C.H, 437
h_frequency
JpegEncoderComponent, 245
h_period
JpegEncoderComponent, 245
h_sampling
JpegDecoderComponent, 224
Harris
ImageGris, 195
ImageRGB, 196
Hash
pngencod.cpp, 561
hash_table
PngEncoder, 331
hash_values
PngEncoder, 331
HashBits
pngencod.cpp, 562
HashTableSize
pngencod.cpp, 562
HashValue
PngEncoder, 328
hd
FormatPBM, 135
FormatPGM, 139
FormatPPM, 142
header_read
PngDecoder, 320
height
Cognois::Ball, 31
FormatBMP::bmpHeader, 45
FormatPBM::pbmHeader, 303
FormatPGM::pgmHeader, 306
FormatPPM::ppmHeader, 346
FormatSunRaster::rasHeader, 352
FormatXPM::xpmHeader, 379
FormatXWD::xwdHeader, 382
PngImageHeader, 337
Height
BitmapImage, 34
FormatXPM, 145
Image, 186

Picture, 310
xbmdecod.cpp, 582
Heikkila
Cognois::Calibration, 47
high
BitmapImage::ColorArea, 67
HistoEqual
Image, 186
ImageRGB, 196
Histogramme
Image, 186
ImageRGB, 197
HistoScale
Image, 186
Homographie
Cognois::Calibration, 47
HomographyToTransformation
Cognois::Calibration, 47
Horaud
ImageGris, 195
HoraudCurveDetection
Contour, 75
HoraudCurveSuivant
Contour, 75
HoraudJunctionDetection
Contour, 75
horesolution
FormatPCX::pcxHeader, 305
horizontal_frequency
JpegDecoderComponent, 224
HorizontalFrequency
JpegDecoderComponent, 222
horizontalResolution
FormatBMP::bmpHeader, 45
hot_spot_x
XbmDecoder, 375
XbmEncoder, 377
hot_spot_y
XbmDecoder, 375
XbmEncoder, 377
HotX
xbmdecod.cpp, 582
HotY
xbmdecod.cpp, 582
hour
PngTimeData, 339
hsize
FormatXWD::xwdHeader, 382
HSV2RGB
ColourSpaceConverter, 71
hue
CouleurHSI, 80
PointColore, 344
Hue
CouleurRGB, 82
HUE_BLUE_TO_RED
ExtractionDeCible.cpp, 431
HUE_GREEN_TO_BLUE
ExtractionDeCible.cpp, 431
HUE_RED_TO_GREEN
ExtractionDeCible.cpp, 431
huff_bits
JpegEncoderHuffmanTable, 249
huff_values
JpegEncoderHuffmanTable, 249
JpegHuffmanDecoder, 253
PngHuffmanDecoder, 334
i
complex, 73
doublecomplex, 90
Multitype, 280
TstParam, 369
I_fp
F2C.H, 437
IC4
jpdedu.cpp, 495
iciend
icilist, 183
icierr
icilist, 183
icifmt
icilist, 183
icilist, 183
iciend, 183
icierr, 183
icifmt, 183
icirlen, 183
icirnum, 183
iciunit, 183
icirlen
icilist, 183
icirnum
icilist, 183
iciunit
icilist, 183
idct_function
JpegDecoderDataUnit, 226
IDctFunction
JpegDecoderDataUnit, 225
Iden4Coins
Cognois::ExtractionDeCible, 125
identifier
JfifExtension, 210
JfifHeader, 211
IDENTITE
Matrice, 267
Operateur, 299
image
TesterEvolution, 368
Image, 184
BruitFlipBits, 186
BruitGaussien, 186

BruitImpulsion, 186
BruitSaltPepper, 186
FiltreConservateur, 186
FiltreGaussien, 186
FiltreMaxMin, 186
FiltreMediane, 186
FiltreMoyenne, 186
FiltreUnsharp, 186
Height, 186
HistoEqual, 186
Histogramme, 186
HistoScale, 186
Image, 185
Length, 187
Max, 187
MaxMin, 187
maxVal, 189
MaxValue, 187
Min, 187
minVal, 189
MinValue, 187
ObtenirMiroir, 187
operator-, 188
operator &, 187
operator &=, 187
operator *, 187
operator *=, 187
operator const T *, 187
operator(), 187, 188
operator/, 188
operator/=, 188
operator^, 188
operator^=, 188
operator|, 188
operator|=, 188
operator~, 188
operator+, 188
operator+=, 188
operator=, 188
operator-=, 188
RegionGet, 187
RegionPut, 187
Width, 187
Image.hpp, 476
image_col
PngEncoder, 331
image_color_type
PngDecoder, 321
image_components
JpegEncoder, 237
image_compression_method
PngDecoder, 321
image_data
BitmapImage, 35
image_depth
PngDecoder, 321
image_end
PngEncoder, 332
image_filter_method
PngDecoder, 321
image_height
BitmapImage, 35
GifDecoder, 172
GifImageDescriptor, 178
PngDecoder, 321
image_interlace_method
PngDecoder, 321
image_left
GifDecoder, 172
image_name
XbmDecoder, 375
XbmEncoder, 377
image_quality
JpegEncoder, 237
image_row
PngEncoder, 332
image_scans
JpegEncoder, 237
image_top
GifDecoder, 172
image_width
BitmapImage, 35
GifDecoder, 172
GifImageDescriptor, 178
PngDecoder, 321
ImageBinaire, 190
AdaptiveTresholding, 191
Boundary, 191
Close, 191
ComponentLabeling, 191
Dilate, 191
DoubleTresholding, 191
Erode, 191
Expand, 191
IterativeTresholding, 191
MAXMIN, 191
MEDIANE, 191
MOYENNE, 190
Open, 191
Shrink, 191
Skeleton, 191
Statistique, 190
Treshold, 191
ZeroCrossing, 191
ImageBinaire.cpp, 477
ImageBinaire.hpp, 478
ImageData
BitmapImage, 34
imageDest
Cognois::AcquisitionBalle, 18
ImageDimX
MILSys, 276

ImageDimY
MILSys, 276
ImageGradient, 192
direction, 193
DirectionGradient, 193
Gradient, 193
GradientMaxima, 193
GradXY, 193
ImageGradient, 192
ModuleGradient, 193
operator=, 193
ImageGradient.cpp, 479
M_PI, 479
ImageGradient.hpp, 480
ImageGris, 194
Canny, 195
Harris, 195
Horaud, 195
ImageGris, 194
LoG, 195
operator BitmapImage, 195
operator=, 195
ImageGris.cpp, 481
ImageGris.hpp, 482
OFF, 482
ON, 482
ImageGris::PRep, 348
operator<, 348
operator==, 348
operator>, 348
pt, 348
rep, 348
ImageLeft
GifDecoder, 170
imageorg
TesterContour, 366
IMAGEPROGRESSFUNCTION
bitimage.h, 400
ImageRGB, 196
getCouleur, 197
Harris, 196
HistoEqual, 196
Histogramme, 197
ImageRGB, 196
operator BitmapImage, 197
operator=, 197
ImageRGB.cpp, 483
M_PI, 483
operator<<, 483
operator>>, 483
PI_3, 483
ImageRGB.hpp, 484
ImageRGB::PRep, 347
operator<, 347
operator==, 347
operator>, 347
pt, 347
rep, 347
ImageScale
MILSys, 276
imageSize
FormatBMP::bmpHeader, 45
ImageStream, 198
beg_of_file, 201
BMP, 199
CTR, 199
EcrireBMP, 200
EcrireContour, 200
EcrireGIF, 200
EcrireJPG, 200
EcrirePBM, 200
EcrirePGM, 200
EcrirePNG, 200
EcrirePPM, 200
EcrireRaster, 200
EcrireXBM, 200
EcrireXWD, 200
format, 201
GetExtension, 200
GetVerbose, 200
GIF, 199
ImageStream, 199
imgFormat, 199
JPG, 199
LireBMP, 200
LireContour, 200
LireGIF, 200
LireJPG, 200
LirePBM, 200
LirePCX, 200
LirePGM, 200
LirePNG, 201
LirePPM, 201
LireRaster, 201
LireXBM, 201
LireXPM, 201
LireXWD, 201
mode, 201
NbFormats, 199
operator<<, 201
operator>>, 201
PBM, 199
PCX, 199
PGM, 199
PNG, 199
PPM, 199
RAS, 199
SetVerbose, 201
verbose_flag, 201
XBM, 199
XPM, 199
XWD, 199

ImageStream.cpp, 485
ImageStream.hpp, 486
ImageStreamException.hpp, 487
ImageToCamera
  Cognois::CameraModel, 63
ImageTop
  GifDecoder, 170
imageTraitee
  Cognois::AcquisitionBalle, 18
imagetyp.h, 488
  BmpImage, 488
  GetStreamImageType, 488
  GifImage, 488
  ImageType, 488
  JpegImage, 488
  PngImage, 488
  ReadImage, 488
  UnknownImage, 488
ImageType
  imagetyp.h, 488
imgFormat
  ImageStream, 199
inacc
  inlist, 205
inacclen
  inlist, 205
inblank
  inlist, 205
inblanklen
  inlist, 205
IncrementFrequency
  JpegEncoderHuffmanTable, 249
  PngHuffmanEncoder, 336
index
  PngEncoder::HashEntry, 180
indice
  FormatXPM::xpmColor, 378
  PointColore, 344
IndiceAlias, 202
  des, 203
  eff, 203
  IndiceAlias, 202
  operator<<, 203
  operator==, 202
  operator>>, 203
indir
  inlist, 205
indirlen
  inlist, 205
inerr
  inlist, 205
inex
  inlist, 205
infile
  inlist, 205
infilen
  inlist, 205
infmt
  inlist, 205
infmtlen
  inlist, 205
inform
  inlist, 205
informlen
  inlist, 205
INIT_BENCH
  ExtractionDeCible.cpp, 431
InitGrabber
  GrabInit.cpp, 473
  GrabInit.h, 474
Initialize
  BitmapImage, 34
  BitmapImageCoder, 38
  GifDecoder, 170
  GifEncoder, 175
  JpegDecoder, 214
  JpegEncoder, 234
  PngDecoder, 317
  PngEncoder, 328
  XbmDecoder, 374
  XbmEncoder, 377
InitializeHashTable
  PngEncoder, 329
InitializeLocalData
  GifDecoder, 170
InitializeLZWindow
  PngEncoder, 329
inlist, 204
  inacc, 205
  inacclen, 205
  inblank, 205
  inblanklen, 205
  indir, 205
  indirlen, 205
  inerr, 205
  inex, 205
  infile, 205
  infilen, 205
  infmt, 205
  infmtlen, 205
  inform, 205
  informlen, 205
  inname, 205
  innamed, 205
  innamlen, 205
  innrec, 205
  innum, 205
  inopen, 205
  inrecl, 205
  inseq, 205
  inseqlen, 205
  inunf, 205

inunflen, 206
inunit, 206
inname
inlist, 205
innamed
inlist, 205
innamlen
inlist, 205
innrec
inlist, 205
innum
inlist, 205
inopen
inlist, 205
InPict
BlobColoring, 43
input_buffer
GifDecoder, 172
input_stream
GifDecoder, 172
JpegDecoder, 218
PngDecoder, 321
inrecl
inlist, 205
inseq
inlist, 205
inseqlen
inlist, 205
Inserer
Equivalence, 110
ListeTri, 258
integer
F2C.H, 437
Integer
FormatXPM, 144
xbmdecod.cpp, 582
integer_scaling
JpegDecoderQuantizationTable, 228
IntegerInverseDCT
JpegDecoderDataUnit, 226
IntegerScale
jpdedu.cpp, 495
intensite
CouleurHSI, 80
Intensite
CouleurRGB, 82
intensity
PointColore, 344
interlace_group
GifDecoder, 172
interlace_pass
PngDecoder, 321
interlace_values
PngDecoder, 321
interlaced_image
GifDecoder, 172
interlacemethod
PngImageHeader, 337
InterlacePasses
pngdecod.cpp, 559
InterleavedPass
JpegEncoder, 235
Intersection
Cognois::ExtractionDeCible, 125
ContourSegment, 78
intersections
Contour, 76
Intervalle, 208
Appartient, 208
Entre, 208
Est, 208
Intervalle, 208
max, 209
min, 209
PlusGrand, 208
PlusPetit, 208
inunf
inlist, 205
inunflen
inlist, 206
inunit
inlist, 206
Inv
Matrice, 268
Invalid
FormatXPM, 144
xbmdecod.cpp, 582
InverseDCT
JpegDecoderDataUnit, 226
ISEC2
jpdedu.cpp, 495
ISEC6
jpdedu.cpp, 495
IsGrabbing
Cognois::OnlineProcessing, 295
IsProgressive
JpegDecoder, 215
IterativeTresholding
ImageBinaire, 191
ITOA
Exception.h, 429
J_fp
F2C.H, 437
jfif.cpp, 489
RGBToCb, 489
RGBToCr, 489
RGBToY, 489
Rounding, 490
Scaled, 489
ScaleFactor, 490
ScaleValue, 490
YCbCrToB, 489

YCbCrToG, 489
YCbCrToR, 489
jfif.h, 491
RGBToCb, 491
RGBToCr, 491
RGBToY, 491
YCbCrToB, 491
YCbCrToG, 491
YCbCrToR, 491
JfifExtension, 210
extension_code, 210
identifier, 210
JpegThumbnail, 210
length, 210
OneByteThumbnail, 210
ThreeByteThumbnail, 210
JfifHeader, 211
identifier, 211
length, 211
units, 211
version, 211
xdensity, 211
xthumbnail, 211
ydensity, 211
ythumbnail, 211
jpdecobk.h, 492
JpegDecoderCoefficientBlock, 492
jpdecomp.cpp, 493
Extend, 493
jpdecomp.h, 494
jpdedu.cpp, 495
Descale, 495
FC4, 495
FSEC2, 495
FSEC6, 495
IC4, 495
IntegerScale, 495
ISEC2, 495
ISEC6, 495
SampleRange, 495
jpdedu.h, 496
operator<<, 496
jpdehuff.cpp, 497
jpdehuff.h, 498
jpdequan.cpp, 499
floatscaling, 499
jpdequan.h, 500
operator<<, 500
jpeg.cpp, 501
JpegZigZagInputOrderCodes, 501
JpegZigZagOutputOrderCodes, 501
jpeg.h, 502
JPEGCOEFFICIENT, 502
JpegMax8BitQuantizationValue, 503
JpegMaxComponentsPerFrame, 503
JpegMaxComponentsPerScan, 503
JpegMaxDataUnitsPerMCU, 503
JpegMaxHuffmanCodeLength, 503
JpegMaxHuffmanTables, 503
JpegMaxNumberOfHuffmanCodes, 503
JpegMaxSampleValue, 503
JpegMaxSamplingFrequency, 503
JpegMaxSuccessiveApproximation, 503
JpegMidpointSampleValue, 503
JpegMinQuantizationValue, 503
JpegMinSampleValue, 503
JpegMinSamplingFrequency, 503
JPEGSAMPLE, 502
JpegSampleSize, 503
JpegSampleWidth, 503
JpegZigZagInputOrder, 503
JpegZigZagInputOrderCodes, 503
JpegZigZagOutputOrder, 503
JpegZigZagOutputOrderCodes, 504
MaxQuantizationTables, 504
jpeg_encoder
JpegEncoderComponent, 245
JPEGCOEFFICIENT
jpeg.h, 502
jpegdeco.cpp, 505
jpegdeco.h, 506
JpegDecoder, 212
~JpegDecoder, 214
ac_tables, 218
bit_data, 218
bit_position, 218
CalculateMcuDimensions, 214
CallProgressFunction, 214
component_count, 218
component_indices, 218
components, 218
current_image, 218
current_scan, 218
dc_tables, 218
DoCopy, 214
eoi_found, 218
expected_restart, 218
frame_height, 218
frame_type, 218
frame_width, 218
FrameHeight, 214
FrameWidth, 214
FreeAllocatedResources, 214
GetScanCount, 214
GetVerbose, 214
Initialize, 214
input_stream, 218
IsProgressive, 215
JpegDecoder, 214
JpegDecoderComponent, 217
JpegDecoderQuantizationTable, 217
JpegHuffmanDecoder, 217

max_horizontal_frequency, 218
max_vertical_frequency, 218
MaxHFrequency, 215
MaxVFrequency, 215
mcu_cols, 218
mcu_height, 218
mcu_rows, 218
mcu_width, 218
McuCols, 215
McuRows, 215
NextBit, 215
operator=, 216
ProcessRestartMarker, 215
progressive_frame, 218
quantization_tables, 219
ReadAcFirst, 215
ReadAcRefine, 215
ReadApplication, 215
ReadByte, 215
ReadDcFirst, 215
ReadDcRefine, 215
ReadHuffmanTable, 215
ReadImage, 215
ReadMarker, 216
ReadProgressiveScanData, 216
ReadQuantization, 216
ReadRestartInterval, 216
ReadSequentialInterleavedScan, 216
ReadSequentialNonInterleavedScan, 216
ReadSequentialScanData, 216
ReadStartOfFrame, 216
ReadStartOfScan, 216
ReadStreamHeader, 216
ReadWord, 216
Receive, 216
RefineAcCoefficient, 216
ResetDcDifferences, 216
restart_interval, 219
RestartInterval, 216
scan_component_count, 219
scan_components, 219
scan_count, 219
ScanIsInterleaved, 216
SetVerbose, 216
sof_found, 219
strict_jfif, 219
UpdateImage, 216
verbose_flag, 219
JpegDecoderCoefficientBlock
jpdecobk.h, 492
JpegDecoderComponent, 220
~JpegDecoderComponent, 221
ac_table, 224
AllocateComponentBuffers, 222
CheckAcTable, 222
CheckDcTable, 222
CheckQuantizationTable, 222
coefficient_blocks, 224
component_id, 224
data_units, 224
dc_table, 224
DecodeAcFirst, 222
DecodeAcRefine, 222
DecodeDcFirst, 222
DecodeDcRefine, 222
DecodeSequential, 222
du_cols, 224
du_rows, 224
eob_run, 224
FreeComponentBuffers, 222
GrayscaleConvert, 222
h_sampling, 224
horizontal_frequency, 224
HorizontalFrequency, 222
JpegDecoder, 217
JpegDecoderComponent, 221
last_dc_value, 224
noninterleaved_cols, 224
noninterleaved_rows, 224
NoninterleavedCols, 222
NoninterleavedRows, 222
operator=, 223
ProgressiveInverseDct, 222
quantization_table, 224
ResetDcDifference, 222
RGBConvert, 222
SetHuffmanTables, 223
SetQuantizationTable, 223
Upsample, 223
upsample_data, 224
v_sampling, 224
vertical_frequency, 224
VerticalFrequency, 223
JpegDecoderDataUnit, 225
~JpegDecoderDataUnit, 226
FloatInverseDCT, 226
idct_function, 226
IDctFunction, 225
IntegerInverseDCT, 226
InverseDCT, 226
JpegDecoderDataUnit, 226
JpegDecoderQuantizationTable, 228
operator[], 226
operator=, 226
Print, 226
values, 226
JpegDecoderQuantizationTable, 227
~JpegDecoderQuantizationTable, 228
BuildScaledTables, 228
data_values, 228
Defined, 228
float_scaling, 228

integer_scaling, 228
JpegDecoder, 217
JpegDecoderDataUnit, 228
JpegDecoderQuantizationTable, 228
operator=, 228
Print, 228
QuantizationIntegerScale, 227
ReadTable, 228
table_defined, 228
jpegenco.cpp, 507
default_chrominance_table, 507
default_luminance_table, 507
jpegenco.h, 508
JpegEncoder, 230
~JpegEncoder, 233
ac_tables, 237
bit_buffer, 237
bit_count, 237
CalculateMcuDimensions, 234
CallProgressFunction, 234
CbComponent, 232
chrominance_quanttbl, 237
comment_string, 237
CountPassesForProgressReporting, 234
CrComponent, 232
CreateQuantizationTables, 234
current_image, 237
current_pass, 237
dc_tables, 237
DoCopy, 234
FindComponentsInScan, 234
FirstAcData, 234
FlushBitBuffer, 234
frame_height, 237
frame_width, 237
GetComment, 234
GetGrayscale, 234
GetProgressive, 234
GetQuality, 234
GetRowsPerRestart, 234
GetSamplingFrequency, 234
GetScanAttributes, 234
gray_scale, 237
image_components, 237
image_quality, 237
image_scans, 237
Initialize, 234
InterleavedPass, 235
JpegEncoder, 233
JpegEncoderComponent, 236
JpegEncoderHuffmanTable, 236
luminance_quanttbl, 237
max_horizontal_frequency, 237
max_vertical_frequency, 237
MaxComponents, 232
MaxScans, 232
mcu_cols, 237
mcu_rows, 237
NoninterleavedPass, 235
operator=, 236
output_stream, 237
OutputBits, 235
OutputByte, 235
OutputJfifHeader, 235
OutputMarker, 235
OutputRestartInterval, 235
OutputWord, 235
PrintAcFirst, 235
PrintAcRefine, 235
PrintComment, 235
PrintDcFirst, 235
PrintDcRefine, 235
PrintHuffmanTables, 235
PrintProgressiveFrame, 235
PrintProgressiveScan, 235
PrintQuantizationTables, 235
PrintSequentialFrame, 235
PrintSequentialScan, 235
progressive_mode, 237
RefineAcData, 235
ResetDcValues, 236
restart_interval, 237
rows_per_restart, 238
scan_component_count, 238
scan_components, 238
scan_count, 238
SetComment, 236
SetGrayscale, 236
SetProgressive, 236
SetQuality, 236
SetRowsPerRestart, 236
SetSamplingFrequency, 236
SetScanAttributes, 236
total_passes, 238
ValidateParameters, 236
WriteImage, 236
YComponent, 232
JpegEncoder::Scan, 354
component_mask, 354
spectral_selection_end, 354
spectral_selection_start, 354
successive_approximation, 354
successive_approximation_high, 354
successive_approximation_low, 354
JpegEncoderCoefficientBlock, 239
data, 239
operator[], 239
ZigZagInput, 239
JpegEncoderComponent, 240
~JpegEncoderComponent, 242
ac_table, 245
ACOUTPUTFUNCTION, 241

CalculateDuDimensions, 243
COMPONENTPASSFUNCTION, 241
DataUnitCols, 243
DataUnitRows, 243
dc_table, 245
DCOUTPUTFUNCTION, 241
dct_coefficients, 245
du_cols, 245
du_rows, 245
EncodeSequential, 243
eob_run, 245
eob_start_du_col, 245
eob_start_du_row, 245
eob_start_position, 245
FreeDynamicStorage, 243
GatherAcData, 243
GatherDcData, 243
GetHorizontalFrequency, 243
GetVerticalFrequency, 243
h_frequency, 245
h_period, 245
jpeg_encoder, 245
JpegEncoder, 236
JpegEncoderComponent, 242
last_dc_value, 245
operator=, 244
PrintAcData, 243
PrintDcData, 243
PrintEobRun, 243
PrintRefineEobRun, 243
ProgressiveAcFirst, 243
ProgressiveAcRefine, 243
ProgressiveDcFirst, 243
ProgressiveDcRefine, 243
quantization_table, 245
ResetDcDifference, 243
ResetEobRun, 244
RGBTOYCBCRFUNCTION, 241
Sample1to1Component, 244
SampleCbComponent, 244
SampleCrComponent, 244
SampleNtoNComponent, 244
SampleYComponent, 244
SetHorizontalFrequency, 244
SetHuffmanTables, 244
SetQuantizationTable, 244
SetVerticalFrequency, 244
v_frequency, 245
v_period, 245
JpegEncoderDataUnit, 246
~JpegEncoderDataUnit, 246
data, 247
DATAUNITVALUE, 246
ForwardDct, 247
JpegEncoderDataUnit, 246
JpegEncoderQuantizationTable, 251
operator[], 247
operator=, 247
ZigZagInput, 247
JpegEncoderHuffmanTable, 248
~JpegEncoderHuffmanTable, 248
BuildTable, 249
ehufco, 249
ehufsi, 249
Encode, 249
frequencies, 249
huff_bits, 249
huff_values, 249
IncrementFrequency, 249
JpegEncoder, 236
JpegEncoderHuffmanTable, 248
MakeTable, 249
operator=, 249
OutputSize, 249
PrintTable, 249
Reset, 249
sizes_found, 249
JpegEncoderQuantizationTable, 250
~JpegEncoderQuantizationTable, 250
BuildScaledTables, 251
data_values, 251
float_scaling, 251
JpegEncoderDataUnit, 251
JpegEncoderQuantizationTable, 250
operator[], 251
operator=, 251
JpegHuffmanDecoder, 252
~JpegHuffmanDecoder, 252
Decode, 253
Defined, 253
Dump, 253
huff_values, 253
JpegDecoder, 217
JpegHuffmanDecoder, 252
MakeTable, 253
maxcode, 253
mincode, 253
operator=, 253
ReadTable, 253
table_defined, 253
valptr, 253
JpegImage
imagetyp.h, 488
JpegMarkers
jpmarker.h, 519
JpegMax8BitQuantizationValue
jpeg.h, 503
JpegMaxComponentsPerFrame
jpeg.h, 503
JpegMaxComponentsPerScan
jpeg.h, 503
JpegMaxDataUnitsPerMCU

jpeg.h, 503
JpegMaxHuffmanCodeLength
jpeg.h, 503
JpegMaxHuffmanTables
jpeg.h, 503
JpegMaxNumberOfHuffmanCodes
jpeg.h, 503
JpegMaxSampleValue
jpeg.h, 503
JpegMaxSamplingFrequency
jpeg.h, 503
JpegMaxSuccessiveApproximation
jpeg.h, 503
JpegMidpointSampleValue
jpeg.h, 503
JpegMinQuantizationValue
jpeg.h, 503
JpegMinSampleValue
jpeg.h, 503
JpegMinSamplingFrequency
jpeg.h, 503
JPEGSAMPLE
jpeg.h, 502
JpegSampleSize
jpeg.h, 503
JpegSampleWidth
jpeg.h, 503
JpegThumbnail
JfifExtension, 210
JpegZigZagInputOrder
jpeg.h, 503
JpegZigZagInputOrderCodes
jpeg.cpp, 501
jpeg.h, 503
JpegZigZagOutputOrder
jpeg.h, 503
JpegZigZagOutputOrderCodes
jpeg.cpp, 501
jpeg.h, 504
jpencobk.h, 509
jpencomp.cpp, 510
jpencomp.h, 511
jpendu.cpp, 512
FC4, 512
FSEC2, 512
FSEC6, 512
jpendu.h, 513
jpenhuff.cpp, 514
jpenhuff.h, 515
jpenquan.cpp, 516
floatscaling, 516
jpenquan.h, 517
JPG
ImageStream, 199
jpgexcep.h, 518
jpmarker.h, 519
APP0, 519
APP1, 520
APP2, 520
APP3, 520
APP4, 520
APP5, 520
APP6, 520
APP7, 520
APP8, 520
APP9, 520
APPA, 520
APPB, 520
APPC, 520
APPD, 520
APPE, 520
APPF, 520
COM, 520
DAC, 519
DHP, 519
DHT, 519
DNL, 519
DQT, 519
DRI, 519
EOI, 519
EXP, 519
JpegMarkers, 519
RST0, 519
RST1, 519
RST2, 519
RST3, 519
RST4, 519
RST5, 519
RST6, 519
RST7, 519
SOB, 520
SOF0, 519
SOF1, 519
SOF2, 519
SOF3, 519
SOF5, 519
SOF6, 519
SOF7, 519
SOF9, 519
SOFA, 519
SOFB, 519
SOFD, 519
SOFE, 519
SOFF, 519
SOI, 519
SOS, 519
K_fp
F2C.H, 437
Kernel, 254
CARRE, 255
CROIX, 255
defCarre, 255

defCroix, 255
defT, 255
defType, 255
EtatPixel, 254
Kernel, 255
kerneltype, 255
KernelType, 255
NbKernelType, 255
P_OFF, 254
P_ON, 254
P_X, 254
T, 255
Type, 255
Kernel.cpp, 521
Kernel.hpp, 522
kerneltype
Kernel, 255
KernelType
Kernel, 255
L_fp
F2C.H, 437
l1
Contour::HoraudAngle, 181
l2
Contour::HoraudAngle, 181
LAPLACIEN
Operateur, 299
LargestColorRange
BitmapImage, 34
last_char
FormatXPM, 145
XbmDecoder, 375
last_dc_value
JpegDecoderComponent, 224
JpegEncoderComponent, 245
LastCode
gifencod.cpp, 470
LastError
Picture, 310
lb1
dmat, 89
lb2
dmat, 89
LBrace
FormatXPM, 144
xbmdecod.cpp, 582
LBracket
FormatXPM, 145
xbmdecod.cpp, 582
left_position
GifImageDescriptor, 178
length
FormatSunRaster::rasHeader, 352
GifDecoder::LZ77String, 259
JfifExtension, 210
JfifHeader, 211
Length
Image, 187
length_distance_table
GifDecoder, 172
length_length_table
PngEncoder, 332
length_table
PngEncoder, 332
LENGTHFUNCTION
PngEncoder, 325
LengthToCode
pngencod.cpp, 561
lexical_value
FormatXPM, 145
XbmDecoder, 375
LibererDerniereImage
Acquisition, 12
ligne
Point, 342
LireBits
Acquisition, 12
LireBMP
ImageStream, 200
LireContour
ImageStream, 200
LireDimensions
Acquisition, 12
LireEvenement
Acquisition, 12
LireGIF
ImageStream, 200
LireJPG
ImageStream, 200
LirePBM
ImageStream, 200
LirePCX
ImageStream, 200
LirePGM
ImageStream, 200
LirePNG
ImageStream, 201
LirePPM
ImageStream, 201
LireRaster
ImageStream, 201
LireSysteme
Acquisition, 12
LireXBM
ImageStream, 201
LireXPM
ImageStream, 201
LireXWD
ImageStream, 201
Liste, 257
ListeTri, 258
Inserer, 258

Trier, 258
ListeTri.hpp, 523
operator<<, 523
operator>>, 523
ListModes
Cognois::CameraClientController, 57
lists
BitmapImage::ColorUsageTable, 70
literal_count
PngDecoder, 322
literal_mode
PngDecoder, 322
literal_table
PngDecoder, 322
LittleEndianToSystem
datatype.h, 421
lmdif
MINPACK, 279
LMDIF.cpp, 524
c__1, 524
lmdif_, 524
lmdif_
F2C.H, 438
LMDIF.cpp, 524
LMPAR.cpp, 525
c__2, 525
lmpar_, 525
lmpar_
F2C.H, 438
LMPAR.cpp, 525
LoadBuffer
GifDecoder, 170
LoG
ImageGris, 195
logical
F2C.H, 437
logical_screen_height
GifScreenDescriptor, 179
logical_screen_width
GifScreenDescriptor, 179
Long
F2C.H, 437
LongestMatch
PngEncoder, 329
LongueurDroite
ContourSegment, 78
lookahead_buffer
PngEncoder, 332
lookahead_position
PngEncoder, 332
LookaheadMask
pngencod.cpp, 562
LookaheadSize
pngencod.cpp, 562
LoopMe
Cognois::OfflineProcessing, 289
Cognois::OnlineProcessing, 295
low
BitmapImage::ColorArea, 67
LSBFirst
FormatXWD, 148
luminance_quanttbl
JpegEncoder, 237
lz_position
PngEncoder, 332
lz_window
GifDecoder, 172
PngDecoder, 322
PngEncoder, 332
m
TstParam, 369
M_PI
Contour.cpp, 416
ImageGradient.cpp, 479
ImageRGB.cpp, 483
mAcquisition
Cognois::OnlineProcessing, 296
magic
FormatSunRaster::rasHeader, 352
MakeAVI
Cognois::MakeAVI, 261
MakeAVI.cpp, 526
MakeAVI.h, 527
MakeCrcTable
pngchksm.cpp, 556
pngchksm.h, 557
MakeTable
JpegEncoderHuffmanTable, 249
JpegHuffmanDecoder, 253
PngHuffmanDecoder, 334
PngHuffmanEncoder, 336
maplength
FormatSunRaster::rasHeader, 352
Mapping
Cognois::ExtractionDeCible, 125
maptype
FormatSunRaster::rasHeader, 352
mArretTraitement
Cognois::OnlineProcessing, 296
mask
Cognois::AcquisitionBalle, 18
mat
Matrice, 271
mat_sto
dmat, 89
matadd
MATRIX.cpp, 531
MATRIX.H, 533
Matching
Cognois::ExtractionDeCible, 125
matcopy
MATRIX.cpp, 531

MATRIX.H, 533
matinvert
MATRIX.cpp, 531
MATRIX.H, 533
MATLAB_COMMAND
Calibration.cpp, 403
MATLAB_HOME
Calibration.cpp, 403
MatlabOutput
Matrice, 268
matmul
MATRIX.cpp, 531
MATRIX.H, 533
Matrice, 263
~Matrice, 267
Afficher, 267
AfficherFichier, 267
AugmenterMatrice, 268
Cross, 268
Det, 268
el, 271
EstVide, 268
GenInv, 268
IDENTITE, 267
Inv, 268
mat, 271
MatlabOutput, 268
Matrice, 267
Max, 268
MaxCol, 268
MaxLigne, 268
Min, 268
MinCol, 268
MinLigne, 268
NbCols, 268
NbLignes, 268
Norm, 269
operator-, 269, 270
operator *, 269
operator *=, 269
operator!=, 269
operator(), 269
operator/, 270
operator/=, 270
operator|, 270
operator|=, 270
operator~, 270
operator+, 269
operator+=, 269
operator=, 270
operator-=, 270
operator==, 270
Resize, 269
SubMatrix, 269
SVD, 269
Type, 267
UNS, 267
ZEROS, 267
Matrice.cpp, 528
operator *, 528
sign, 528
Matrice.h, 529
operator *, 529
MatriceEssentielle
Cognois::CameraModel, 63
MatriceFondamentale
Cognois::CameraModel, 63
MatriceIntrinseque
Cognois::CameraModel, 63
MatriceProduitVectoriel
EspaceCalibrage, 7
MatriceProjection
Cognois::CameraModel, 63
MatriceRotation
Cognois::CameraModel, 63
MatriceTranslation
Cognois::CameraModel, 63
MATRIX.cpp, 530
FALSE, 530
matadd, 531
matcopy, 531
matinvert, 531
matmul, 531
newdmat, 531
PERMBUFSIZE, 530
print_mat, 531
scamul, 531
solve_system, 531
TOL_DET, 530
transpose, 531
TRUE, 530
MATRIX.H, 532
freemat, 532
matadd, 533
matcopy, 533
matinvert, 533
matmul, 533
newdmat, 533
print_mat, 533
scamul, 533
solve_system, 533
transpose, 533
mAVIFile
Cognois::OfflineProcessing, 289
mAviFileName
Cognois::MakeAVI, 262
max
F2C.H, 436
Intervalle, 209
Max
Image, 187
Matrice, 268

StatisticsR, 360
MAX
EspaceCalibrage.h, 426
max_horizontal_frequency
JpegDecoder, 218
JpegEncoder, 237
MAX_RECONSTRUCTION_ERR
ExtractionDeCible.cpp, 431
MAX_SAMPLES_PER_CHARACTER
OCRFontCreator.cpp, 540
max_vertical_frequency
JpegDecoder, 218
JpegEncoder, 237
Max8BitSampleValue
pngdecod.cpp, 559
maxcode
JpegHuffmanDecoder, 253
PngHuffmanDecoder, 334
MaxCol
Matrice, 268
MaxComponents
JpegEncoder, 232
MaxHFrequency
JpegDecoder, 215
MaxHuffmanCodeLength
PngHuffmanDecoder, 333
MaximumCompression
PngEncoder, 326
MaxLigne
Matrice, 268
maxmag
DPMPAR.cpp, 422
MaxMin
Image, 187
MAXMIN
ImageBinaire, 191
MaxQuantizationTables
jpeg.h, 504
MaxScans
JpegEncoder, 232
maxVal
Image, 189
maxvalue
FormatPGM::pgmHeader, 306
FormatPPM::ppmHeader, 346
MaxValue
Image, 187
MaxVFrequency
JpegDecoder, 215
MaxX
BlobInfo, 44
MaxY
BlobInfo, 44
mBlocCount
Cognois::AcquisitionBalle, 18
mBotX
Cognois::CounterOCR, 85
Cognois::OCRFontCreator, 285
mBotY
Cognois::CounterOCR, 85
Cognois::OCRFontCreator, 285
MbufAlloc2d
Cognois::MilGarbageCollector, 273
MbufAllocColor
Cognois::MilGarbageCollector, 273
MbufChild2d
Cognois::MilGarbageCollector, 273
MbufChildColor
Cognois::MilGarbageCollector, 274
MbufFree
Cognois::MilGarbageCollector, 274
mCamera
Cognois::SpaceCarving::SCVolume, 357
mCameraClientInitialized
Cognois::FrameSetClient, 162
mCharacters
Cognois::OCRFontCreator, 286
mCharacterSamples
Cognois::OCRFontCreator, 286
mcheps
DPMPAR.cpp, 422
mCognoisServer
Cognois::AcquisitionCible, 20
Cognois::CounterOCR, 85
Cognois::NoopToServer, 283
mCompteurDebut
Acquisition, 14
mCompteurFrequence
Acquisition, 14
mCount
Acquisition, 14
mCounterImage
Cognois::CounterOCR, 85
Cognois::OCRFontCreator, 286
mcu_cols
JpegDecoder, 218
JpegEncoder, 237
mcu_height
JpegDecoder, 218
mcu_rows
JpegDecoder, 218
JpegEncoder, 237
mcu_width
JpegDecoder, 218
McuCols
JpegDecoder, 215
McuRows
JpegDecoder, 215
mCurrentMode
Cognois::CameraClientController, 58
MDoubles
EspaceCalibrage, 7

Mean
StatisticsR, 360
Median
StatisticsR, 360
MedianDeviation
StatisticsR, 360
MEDIANE
ImageBinaire, 191
mEndGrabEvent
Acquisition, 14
Message
Cognois::Exception, 122
METEOR_II_CL
Acquisition.cpp, 384
mEventBeginProcessing
Cognois::OnlineProcessing, 296
mEventEndProcessing
Cognois::OnlineProcessing, 296
mEventGrabEnd
Cognois::OnlineProcessing, 296
mEventImageGrabbed
Cognois::OnlineProcessing, 296
mEventProcessingHasEnded
Cognois::OnlineProcessing, 296
mEventRequestProcessing
Cognois::OnlineProcessing, 296
mEventResetFrameId
Cognois::OnlineProcessing, 296
mEventThreadGrabbingHasEnded
Cognois::OnlineProcessing, 296
mEventThreadProcessingHasEnded
Cognois::OnlineProcessing, 296
mEventThreadSynchronizationHasEnded
Cognois::OnlineProcessing, 296
mFieldEndHookSucceed
Acquisition, 14
mFieldEvenEndHookSucceed
Acquisition, 14
mFieldOddEndHookSucceed
Acquisition, 14
mFileName
Cognois::MakeAVI, 262
mFont
Cognois::CounterOCR, 85
Cognois::OCRFontCreator, 286
mFontFilename
Cognois::OCRFontCreator, 286
mFontIsAllocated
Cognois::OCRFontCreator, 286
mFrame
Cognois::FrameSetClient, 162
mFrameCounter
Cognois::OnlineProcessing, 296
mFrameEndHookSucceed
Acquisition, 14
mFrameIdProcessing
Cognois::OnlineProcessing, 296
mFrameIds
Cognois::MakeAVI, 262
mFrameRate
Cognois::FrameCounter, 154
mFrameStartHookSucceed
Acquisition, 14
mGrabber
Cognois::CameraClientController, 58
Cognois::OnlineProcessing, 296
mGrabEnCourt
Acquisition, 14
mGrabEndHookSucceed
Acquisition, 14
mGrabStartHookSucceed
Acquisition, 14
mGrayImage
Cognois::CounterOCR, 85
Cognois::OCRFontCreator, 286
MilApplication
MILSys, 276
MilBluePtr
DirectMilAccess, 88
milBuffer
Cognois::OfflineProcessing, 289
MilBufferDisplay
Cognois::AcquisitionSurEcran, 28
milBufferPacked
Cognois::PlanarToPacked, 312
MilDigitizer
MILSys, 276
MilDisplay
MILSys, 276
MilGarbageCollector, 275
MilGarbageCollector.cpp, 534
MilGarbageCollector.h, 535
MilGreenPtr
DirectMilAccess, 88
MilImageBlue
DirectMilAccess, 88
MilImageDisp
Cognois::ExtractionDeCible, 127
MilImageGreen
DirectMilAccess, 88
MilImageRed
DirectMilAccess, 88
MilImageWork
Cognois::ExtractionDeCible, 127
MilImageWorkPitch
DirectMilAccess, 88
MilRedPtr
DirectMilAccess, 88
MilSys
Cognois::AcquisitionSurEcran, 28
MILSys, 276
Cognois::OfflineProcessing, 289

Cognois::OnlineProcessing, 295
ImageDimX, 276
ImageDimY, 276
ImageScale, 276
MilApplication, 276
MilDigitizer, 276
MilDisplay, 276
MilSystem, 276
NbColors, 276
MILSys.h, 536
MilSystem
MILSys, 276
mImage
Cognois::SpaceCarving::SCVolume, 357
mImageBuffer
Cognois::OnlineProcessing, 297
mImageContents
Cognois::MakeAVI, 262
mImageCounter
Cognois::MakeAVI, 262
mImageHauteur
Acquisition, 14
AcquisitionImageListe, 24
mImageLargeur
Acquisition, 14
AcquisitionImageListe, 24
mImageLimit
Cognois::MakeAVI, 262
mImageLongueur
Acquisition, 14
mImageSize
Cognois::ExtractionDeCible, 127
min
F2C.H, 436
Intervalle, 209
Min
Image, 187
Matrice, 268
MIN
EspaceCalibrage.h, 426
MIN_BLOB_DIST_TO_BORDER
ExtractionDeCible.cpp, 431
MIN_BLOB_SIZE
ExtractionDeCible.cpp, 431
MIN_DETECTE
ExtractionDeCible.cpp, 431
mincode
JpegHuffmanDecoder, 253
PngHuffmanDecoder, 334
MinCol
Matrice, 268
mInitialFrameId
Cognois::OnlineProcessing, 297
MinLigne
Matrice, 268
minmag
DPMPAR.cpp, 422
MINPACK, 278
evaluation_erreur, 278
lmdif, 279
user_constants, 279
user_fcn, 279
MINPACK.cpp, 537
EPSFCN, 537
F_TOL, 537
X_TOL, 537
MINPACK.h, 538
MinPoolingPeriod
Cognois::FrameSetClient, 160
minute
PngTimeData, 339
minVal
Image, 189
MinValue
Image, 187
MinX
BlobInfo, 44
MinY
BlobInfo, 44
mIterator
Cognois::AcquisitionBalle, 18
mLastDisplay
Cognois::ExtractionDeCible, 127
mLastFrameIds
Cognois::FrameCounter, 154
mLastPooling
Cognois::FrameSetClient, 162
mLastPosition
Cognois::ExtractionDeCible, 127
mLastSize
Cognois::ExtractionDeCible, 127
mLastTimeStamps
Cognois::FrameCounter, 154
mLatestFrameId
Cognois::FrameSetClient, 162
mLeastRecentFrameIndex
Cognois::FrameCounter, 154
mListe
AcquisitionImageListe, 24
mLongueur
AcquisitionImageListe, 24
mMessage
Cognois::Exception, 122
mMilAcces
Acquisition, 14
mMilApplication
Acquisition, 14
mMilBuffer
Acquisition, 14
AcquisitionImage, 22
mMilBufferAcces
Acquisition, 14

mMilBufferAdd
AcquisitionImage, 22
mMilBufferAttente
Acquisition, 14
mMilBufferBand
AcquisitionImage, 22
mMilBufferBarre
Acquisition, 14
mMilBufferCourant
Acquisition, 14
mMilBufferDernier
Acquisition, 15
mMilBufferNombre
Acquisition, 15
mMilBufferProchainDernier
Acquisition, 15
mMilDigitizer
Acquisition, 15
mMilDisplay
Cognois::AcquisitionSurEcran, 28
mMilSys
Cognois::OCRFontCreator, 286
Cognois::OfflineProcessing, 289
Cognois::OnlineProcessing, 297
mMilSysteme
Acquisition, 15
AcquisitionImageListe, 24
mMinPoolingPeriod
Cognois::FrameSetClient, 162
mModes
Cognois::CameraClientController, 58
mMostRecentFrameIndex
Cognois::FrameCounter, 154
mMutexImageBuffer
Cognois::OnlineProcessing, 297
mNoValidFrameYet
Cognois::OnlineProcessing, 297
mOCRResult
Cognois::CounterOCR, 85
mode
ImageStream, 201
ModuleGradient
ImageGradient, 193
mOnlineProcessing
Cognois::OCRFontCreator, 286
Mono
FormatXPM, 145
month
PngTimeData, 339
more_images
GifDecoder, 172
MoreImages
GifDecoder, 170
mOuvert
Acquisition, 15
MoveHashEntry
PngEncoder, 329
MOYENNE
ImageBinaire, 190
Operateur, 299
mPatternFinder
Cognois::AcquisitionCible, 20
mPerformanceFrequency
Cognois::FrameCounter, 154
Cognois::FrameSetClient, 162
MPoints
EspaceCalibrage, 7
MPointsXYZ
EspaceCalibrage, 7
mProcessor
Cognois::OnlineProcessing, 297
mProcessors
Cognois::CameraClientController, 58
mQueueLength
Cognois::FrameCounter, 154
mResetCompleted
Cognois::OnlineProcessing, 297
mResetPeriod
Cognois::OnlineProcessing, 297
MSBFirst
FormatXWD, 148
mServer
Cognois::CameraClientController, 58
mSizeBand
Cognois::MakeAVI, 262
mSizeX
Cognois::MakeAVI, 262
mSizeY
Cognois::MakeAVI, 262
mStream
Cognois::FrameCounter, 154
mSynchro
Cognois::OnlineProcessing, 297
mSyncServer
Cognois::OfflineProcessing, 289
Cognois::OnlineProcessing, 297
mTemps
Acquisition, 15
mTempsDebut
AcquisitionImage, 22
mTempsFin
AcquisitionImage, 22
mTimeAtBeginning
Cognois::FrameCounter, 154
mTimeFile
Cognois::OfflineProcessing, 289
mTimeStampProcessing
Cognois::OnlineProcessing, 297
mTimeStamps
Cognois::MakeAVI, 262
mTopX
Cognois::CounterOCR, 85

Cognois::OCRFontCreator, 286
mTopY
Cognois::CounterOCR, 85
Cognois::OCRFontCreator, 286
mTransientState
Cognois::FrameCounter, 154
muet
TesterContour, 366
Muet
TesterContour, 365
MultiCam2SingleCam
Cognois::CameraModel, 63
Multitype, 280
c, 280
d, 280
F2C.H, 437
h, 280
i, 280
r, 280
z, 280
mVerbose
Cognois::FrameCounter, 154
mVoxelSize
Cognois::SpaceCarving::SCVolume, 357
mXmlData
Cognois::FrameSetClient, 163
name
Namelist, 281
Vardesc, 371
Namelist, 281
F2C.H, 437
name, 281
nvars, 281
vars, 281
NB_POINTS_REFERENCE
ExtractionDeCible.cpp, 431
nbBitsPerPixel
FormatBMP::bmpHeader, 45
NbBlob
BlobColoring, 43
nbColorIndexes
FormatBMP::bmpHeader, 45
NbColors
MILSys, 276
NbCols
Matrice, 268
NbFormats
ImageStream, 199
nbImportantColorIndexes
FormatBMP::bmpHeader, 45
NbKernelType
Kernel, 255
NbLignes
Matrice, 268
NbPixels
BlobInfo, 44
ncolors
FormatXPM::xpmHeader, 379
FormatXWD::xwdHeader, 382
NColors
FormatXPM, 145
newdmat
MATRIX.cpp, 531
MATRIX.H, 533
NEWLINE
FormatPortable.hpp, 452
next
BitmapImage::ColorUsage, 69
PngEncoder::HashEntry, 180
NextBit
GifDecoder, 170
JpegDecoder, 215
NextChar
FormatXPM, 145
XbmDecoder, 374
NextCode
GifDecoder, 170
NextColor
FormatXPM, 145
NextIndex
Cognois::FrameCounter, 152
NextInformation
FormatPortable, 140
NextLeastRecentFrameIndex
Cognois::FrameCounter, 153
NextLine
FormatPortable, 140
NextMostRecentFrameIndex
Cognois::FrameCounter, 153
NextPixel
FormatXPM, 145
NextToken
FormatXPM, 145
XbmDecoder, 374
NgbrStack
BlobColoring, 43
NO_FRAME
Acquisition.cpp, 384
NoData
FormatXPM.hpp, 459
xbmdecod.cpp, 582
noninterleaved_cols
JpegDecoderComponent, 224
noninterleaved_rows
JpegDecoderComponent, 224
NoninterleavedCols
JpegDecoderComponent, 222
NoninterleavedPass
JpegEncoder, 235
NoninterleavedRows
JpegDecoderComponent, 222
NoopToServer

Cognois::NoopToServer, 282
NoopToServer.h, 539
Norm
Matrice, 269
note
TstParamNote, 370
Note
TesterContour, 365
NotePoints
TesterContour, 365
notes
TesterEvolution, 368
NoteSegments
TesterContour, 365
NULL
BlobColoring.cpp, 401
datatype.h, 420
NUM_PRECISION_DOUBLE
ColourSpaceConverter.cpp, 414
number
FormatXWD::xwdColor, 380
nvars
Namelist, 281
oacc
olist, 291
oblnk
olist, 291
ObtenirMiroir
Image, 187
ObtenirPointsCoins
Cognois::ExtractionDeCible, 125
ObtenirPointsRGGG
Cognois::ExtractionDeCible, 125, 126
OCRFontCreator
Cognois::OCRFontCreator, 285
OCRFontCreator.cpp, 540
MAX_SAMPLES_PER_CHARACTER, 540
OCRFontCreator.h, 541
oerr
olist, 291
of
Cognois::AcquisitionBalle, 18
OFF
ImageGris.hpp, 482
OfflineProcessing
Cognois::OfflineProcessing, 288
OfflineProcessing.cpp, 542
C_CONTROL_BIT, 542
OfflineThreadAcquisition, 543
SYNC_SERVER_PERIOD, 542
SYNCHRONIZATION, 542
USER_IN_BIT, 542
OfflineProcessing.h, 544
OfflineThreadAcquisition
OfflineProcessing.cpp, 543
offset
GifDecoder::LZ77String, 259
OffsetTable
BlobColoring, 43
OffsetTableX
BlobColoring, 43
OffsetTableY
BlobColoring, 43
offsetX
Cognois::Ball, 31
offsetY
Cognois::Ball, 31
ofm
olist, 291
ofnm
olist, 291
ofnmlen
olist, 291
olist, 291
oacc, 291
oblnk, 291
oerr, 291
ofm, 291
ofnm, 291
ofnmlen, 291
orl, 291
osta, 291
ounit, 291
ON
ImageGris.hpp, 482
OneByteThumbnail
JfifExtension, 210
OnlineProcessing
Cognois::OnlineProcessing, 294
OnlineProcessing.cpp, 545
COUNTING_IN_BIT, 545
COUNTING_OUT_BIT, 545
getCounting, 546
getValid, 546
READY_IN_BIT, 546
READY_OUT_BIT, 546
setCounting, 546
setReady, 546
SYNCHRONIZATION, 546
ThreadAcquisition, 546
ThreadSynchronization, 546
ThreadTraitement, 546
VERBOSE_SYNC, 546
OnlineProcessing.h, 547
Open
ImageBinaire, 191
openstring
FormatXPM, 145
Operateur, 298
CalculerPosition, 300
defGaussienne, 301
defGaussienneH, 301

defGaussienneV, 301
defIdentite, 301
defLaplacien, 301
defMoyenne, 301
defPrewittH, 301
defPrewittV, 301
defRobertsH, 301
defRobertsV, 301
defSobelH, 301
defSobelV, 301
defType, 301
facteur, 302
Facteur, 300
Gaussienne, 300
GAUSSIENNE, 299
GAUSSIENNEH, 299
GAUSSIENNEV, 299
IDENTITE, 299
LAPLACIEN, 299
MOYENNE, 299
Operateur, 299
OperateurType, 299
operator const int *, 301
org, 302
Origine, 300
PREWITT, 299
PREWITTH, 299
PREWITTV, 299
ROBERTS, 299
ROBERTSH, 299
ROBERTSV, 299
SOBEL, 299
SOBELH, 299
SOBELV, 299
type, 302
Type, 300
UNKNOWN, 299
ValiderPosition, 301
xd, 302
xg, 302
yb, 302
yh, 302
Operateur.cpp, 548
Operateur.hpp, 549
OperateurType
Operateur, 299
operator-
CouleurRGB, 82
Image, 188
Matrice, 269, 270
operator &
Image, 187
operator &=
Image, 187
operator *
Image, 187
Matrice, 269
Matrice.cpp, 528
Matrice.h, 529
operator *=
Image, 187
Matrice, 269
operator BitmapImage
ImageGris, 195
ImageRGB, 197
operator const int *
Operateur, 301
operator const T *
Image, 187
operator unsigned long
CouleurHSI, 80
CouleurRGB, 82
operator!=
Matrice, 269
Point, 341
operator()
Image, 187, 188
Matrice, 269
operator/
Image, 188
Matrice, 270
operator/=
Image, 188
Matrice, 270
operator[]
BitmapImage, 35
JpegDecoderDataUnit, 226
JpegEncoderCoefficientBlock, 239
JpegEncoderDataUnit, 247
JpegEncoderQuantizationTable, 251
operator^
Image, 188
operator^=
Image, 188
operator|
Image, 188
Matrice, 270
operator|=
Image, 188
Matrice, 270
operator~
Image, 188
Matrice, 270
operator+
CouleurRGB, 82
Image, 188
Matrice, 269
operator+=
CouleurRGB, 82
Image, 188
Matrice, 269
operator<

ExtractionDeCible.cpp, 432
ImageGris::PRep, 348
ImageRGB::PRep, 347
Point, 341
operator<<
Cognois::CameraModel, 64
Contour, 76
Contour.cpp, 416
ContourSegment, 78
CouleurHSI, 80
CouleurRGB, 82
Equivalence.hpp, 424
EspaceCalibrage, 7
ImageRGB.cpp, 483
ImageStream, 201
IndiceAlias, 203
jpdedu.h, 496
jpdequan.h, 500
ListeTri.hpp, 523
Point, 341
Point.hpp, 571
operator<=
Point, 341
operator=
BitmapImage, 35
BitmapImageCoder, 38
Cognois::FrameCounter, 153
Cognois::FrameSetClient, 162
CouleurHSI, 80
CouleurRGB, 82
EGraphicsException, 99
GifDecoder, 171
GifEncoder, 175
Image, 188
ImageGradient, 193
ImageGris, 195
ImageRGB, 197
JpegDecoder, 216
JpegDecoderComponent, 223
JpegDecoderDataUnit, 226
JpegDecoderQuantizationTable, 228
JpegEncoder, 236
JpegEncoderComponent, 244
JpegEncoderDataUnit, 247
JpegEncoderHuffmanTable, 249
JpegEncoderQuantizationTable, 251
JpegHuffmanDecoder, 253
Matrice, 270
Picture, 310
PngDecoder, 319
PngEncoder, 330
PngHuffmanDecoder, 334
PngHuffmanEncoder, 336
XbmDecoder, 374
XbmEncoder, 377
operator-=
CouleurRGB, 82
Image, 188
Matrice, 270
operator==
EspaceCalibrage, 7
ImageGris::PRep, 348
ImageRGB::PRep, 347
IndiceAlias, 202
Matrice, 270
Point, 341
TstParamNote, 370
operator>
ImageGris::PRep, 348
ImageRGB::PRep, 347
Point, 341
operator>=
Point, 341
operator>>
Cognois::CameraModel, 64
Contour, 76
Contour.cpp, 416
ContourSegment, 78
CouleurHSI, 80
CouleurRGB, 82
Equivalence.hpp, 424
ImageRGB.cpp, 483
ImageStream, 201
IndiceAlias, 203
ListeTri.hpp, 523
Point, 341
Point.hpp, 571
Ordonner
Cognois::ExtractionDeCible, 126
OrdonnerCouleur
Cognois::ExtractionDeCible, 126
org
Operateur, 302
Orientation
ContourSegment, 78
Origine
ContourSegment, 78
Operateur, 300
orl
olist, 291
osta
olist, 291
ounit
olist, 291
OutPict
BlobColoring, 43
output_buffer
PngEncoder, 332
output_stream
GifEncoder, 175
JpegEncoder, 237
PngEncoder, 332

OutputAdler
PngEncoder, 329
OutputBits
GifEncoder, 175
JpegEncoder, 235
OutputBlock
PngEncoder, 329
OutputBlockData
PngEncoder, 329
OutputBufferSize
pngencod.cpp, 562
OutputByte
JpegEncoder, 235
PngEncoder, 329
OutputCode
GifEncoder, 175
OutputDataBits
PngEncoder, 329
OutputDeflateHeader
PngEncoder, 329
OutputJfifHeader
JpegEncoder, 235
OutputLengthCounts
PngEncoder, 329
OutputMarker
JpegEncoder, 235
OutputRestartInterval
JpegEncoder, 235
OutputSize
JpegEncoderHuffmanTable, 249
OutputWord
JpegEncoder, 235
OutputZLibHeader
PngEncoder, 329
Ouvrir
Acquisition, 12
P_OFF
Kernel, 254
P_ON
Kernel, 254
P_X
Kernel, 254
pac
TesterContour, 366
packedfield
GifGraphicControlBlock, 176
GifImageDescriptor, 178
GifScreenDescriptor, 179
pad
FormatXWD::xwdColor, 380
pae
TesterContour, 366
PaethPredictor
pngpvt.h, 569
palette
FormatPCX::pcxHeader, 305
Palette
pngpvt.h, 569
palette_read
PngDecoder, 322
palette_size
PngDecoder, 322
palettetype
FormatPCX::pcxHeader, 305
par
TesterEvolution, 368
TstParamNote, 370
param
TesterContour, 366
parExt
Cognois::CameraModel, 64
parInt
Cognois::CameraModel, 64
pat
TesterContour, 366
PBM
ImageStream, 199
PBM_ASCII
FormatPortable.hpp, 452
PBM_RAW
FormatPortable.hpp, 452
PC_FILE_HEAD
CameraModel.h, 410
PC_HEIKKILA_0
CameraModel.h, 410
PC_HEIKKILA_SINGLE_0
CameraModel.h, 411
PC_ZHANG_0
CameraModel.h, 411
PC_ZHANG_SINGLE_0
CameraModel.h, 411
PCOLOR
FormatPCX, 136
PCX
ImageStream, 199
PCX16c
FormatPCX, 137
PCX16c_pal
FormatPCX, 137
PCX16Mc
FormatPCX, 137
PCX256c_pal
FormatPCX, 137
PCXMONO
FormatPCX, 137
pdm
TesterContour, 366
PERFORM_BENCH
ExtractionDeCible.cpp, 431
PERMBUFSIZE
MATRIX.cpp, 530
PGM

ImageStream, 199
PGM_ASCII
FormatPortable.hpp, 452
PGM_RAW
FormatPortable.hpp, 452
PGSCALE
FormatPCX, 136
PHeight
BlobColoring, 43
PI
ColourSpaceConverter.cpp, 414
EspaceCalibrage.h, 426
PI_3
ImageRGB.cpp, 483
pi_div_3
ColourSpaceConverter, 71
pi_x_2
ColourSpaceConverter, 71
Picture, 307
~Picture, 308
AllocateDataBuffer, 309
ApplyThreshold, 309
AutoThresholdEstimation, 309
AutoThresholding, 309
Clear, 309
ConvertToGrayScale, 309
CrossColor16, 310
CrossColor8, 310
CrossColorRGB, 310
Data16, 310
Data8, 310
DataRGB, 310
Equal, 309
GetDataPtr, 309
GetHeight, 309
GetHistogram, 309
GetLastError, 309
GetSize, 309
GetType, 309
GetWidth, 309
GraphLine, 309
Height, 310
LastError, 310
operator=, 310
Picture, 308
PutCross, 309
RepresentFloatData, 309
RepresentIndexedData, 309
SetCrossColor, 309, 310
SetHeight, 310
SetType, 310
SetWidth, 310
Size, 310
Type, 310
Width, 310
Picture.cpp, 550
Picture.h, 551
AFX_PICTURE_H__DFC2EB67_37C7_11D4_BD19_00E0292D1826__INCLUDED_, 551
BYTE, 551
CS_COLOR, 551
CS_GRAY16, 551
CS_GRAY8, 551
pixel
FormatXPM::xpmHeader, 379
Pixel
FormatXPM, 145
pixel_aspect_ratio
GifScreenDescriptor, 179
pixel_count
BitmapImage::ColorArea, 67
pixelsx
PngPixelDimensions, 338
pixelsy
PngPixelDimensions, 338
pixmask
FormatXWD, 149
PlanarToPacked
Cognois::PlanarToPacked, 311
PlanarToPacked.cpp, 552
PlanarToPacked.h, 553
PlusGrand
Intervalle, 208
PlusPetit
Intervalle, 208
PMAGIC
FormatPCX, 136
PNG
ImageStream, 199
png.cpp, 554
PngLengthOrder, 554
PngSignature, 554
png.h, 555
pngchksm.cpp, 556
Adler, 556
crc_table, 556
CRC32, 556
MakeCrcTable, 556
table_made, 556
pngchksm.h, 557
Adler, 557
CRC32, 557
MakeCrcTable, 557
PngChromaticitiesData, 313
bluex, 313
bluey, 313
greenx, 313
greeny, 313
redx, 313
redy, 313
whitepointx, 313

whitepointy, 313
PngColorType
pngpvt.h, 568
pngdecod.cpp, 558
CompressionType, 558
DynamicHuffmanCodes, 558
FixedHuffmanCodes, 558
InterlacePasses, 559
Max8BitSampleValue, 559
Uncompressed, 558
VERYVERBOSE, 558
pngdecod.h, 560
PngDecoder, 314
~PngDecoder, 316
background_blue, 320
background_gray, 320
background_green, 320
background_red, 320
bit_buffer, 320
bit_offset, 320
byte_offset, 320
CallProgressFunction, 317
CheckAdler, 317
chunk_crc, 320
chunk_data, 320
chunk_data_size, 320
chunk_length, 320
chunk_type, 320
copy_count, 320
copy_position, 320
CopyInterlacedRowToImage, 317
CopyNoninterlacedRowToImage, 317
current_image, 320
current_row_buffer, 320
data_read, 320
DecodeByte, 317
DecodeCompressedByte, 317
DecodeLiteralByte, 317
distance_table, 320
DoCopy, 317
end_read, 320
file_gamma, 320
FilterRow, 317
final_data_set, 320
FreeData, 317
GetBits, 317
GetIDATByte, 317
GetVerbose, 317
header_read, 320
image_color_type, 321
image_compression_method, 321
image_depth, 321
image_filter_method, 321
image_height, 321
image_interlace_method, 321
image_width, 321
Initialize, 317
input_stream, 321
interlace_pass, 321
interlace_values, 321
literal_count, 322
literal_mode, 322
literal_table, 322
lz_window, 322
operator=, 319
palette_read, 322
palette_size, 322
PngDecoder, 316
PngHuffmanDecoder, 319
ProcessBackground, 318
ProcessChromaticities, 318
ProcessCompressedText, 318
ProcessData, 318
ProcessGamma, 318
ProcessHeader, 318
ProcessHistogram, 318
ProcessImageTime, 318
ProcessPalette, 318
ProcessPhysicalPixelDimensions, 318
ProcessSignificantBits, 318
ProcessTextualData, 318
ProcessTransparency, 318
ReadChunk, 318
ReadImage, 318
reading_pixel_data, 322
ReadInterlaced, 319
ReadLengths, 319
ReadNoninterlaced, 319
ReadPixelData, 319
ReadSignature, 319
row_buffer_width, 322
row_buffers, 322
SetVerbose, 319
StartNewDataSet, 319
stream_adler, 322
verbose_flag, 322
window_position, 322
PngDecoder::InterlaceInfo, 207
col_interval, 207
row_interval, 207
start_col, 207
start_row, 207
pngencod.cpp, 561
DistanceToCode, 561
ENDIMAGE, 562
Hash, 561
HashBits, 562
HashTableSize, 562
LengthToCode, 561
LookaheadMask, 562
LookaheadSize, 562
OutputBufferSize, 562

pngencod.h, 563
PngEncoder, 323
~PngEncoder, 326
adler_value, 331
author_string, 331
bit_position, 331
block_buffer, 331
block_buffer_count, 331
block_buffer_size, 331
byte_position, 331
CallProgressFunction, 327
chunk_type, 331
comment_string, 331
compression_level, 331
CompressionLevel, 326
copyright_string, 331
current_filter, 331
current_image, 331
DefaultCompression, 326
description_string, 331
disclaimer_string, 331
distance_table, 331
DoCopy, 327
DoWrite, 327
FastCompression, 326
FastestCompression, 326
FillBuffer, 327
filter_buffers, 331
filter_mask, 331
filter_width, 331
FilterBufferCount, 325
FilterRow, 327
FindLengthCodes, 327
FlushBitBuffer, 327
FreeBuffers, 327
GatherLengthCounts, 327
GetAuthor, 327
GetBlockSize, 327
GetComment, 327
GetCompressionLevel, 327
GetCopyright, 327
GetDescription, 327
GetDisclaimer, 327
GetSoftware, 327
GetSource, 327
GetTitle, 327
GetUseFilters, 327
GetWarning, 327
hash_table, 331
hash_values, 331
HashValue, 328
image_col, 331
image_end, 332
image_row, 332
Initialize, 328
InitializeHashTable, 329
InitializeLZWindow, 329
length_length_table, 332
length_table, 332
LENGTHFUNCTION, 325
LongestMatch, 329
lookahead_buffer, 332
lookahead_position, 332
lz_position, 332
lz_window, 332
MaximumCompression, 326
MoveHashEntry, 329
operator=, 330
output_buffer, 332
output_stream, 332
OutputAdler, 329
OutputBlock, 329
OutputBlockData, 329
OutputByte, 329
OutputDataBits, 329
OutputDeflateHeader, 329
OutputLengthCounts, 329
OutputZLibHeader, 329
PngEncoder, 326
PngHuffmanEncoder, 336
ProcessImageData, 329
row_width, 332
search_limit, 332
SetAuthor, 329
SetBlockSize, 329
SetComment, 329
SetCompressionLevel, 329
SetCopyright, 329
SetDescription, 329
SetDisclaimer, 329
SetSoftware, 329
SetSource, 329
SetTitle, 329
SetUseFilters, 330
SetWarning, 330
software_string, 332
source_string, 332
StartChunk, 330
title_string, 332
warning_string, 332
WriteCurrentChunk, 330
WriteImage, 330
WriteImageData, 330
WriteText, 330
WriteTextBlocks, 330
PngEncoder::HashEntry, 180
index, 180
next, 180
previous, 180
PngEndCode
pngpvt.h, 569
PngFilterType

pngpvt.h, 569
PngFirstLengthCode
pngpvt.h, 569
pnghuffd.cpp, 564
BADCODE, 564
pnghuffd.h, 565
pnghuffe.cpp, 566
pnghuffe.h, 567
PngHuffmanDecoder, 333
~PngHuffmanDecoder, 334
Decode, 334
Dump, 334
huff_values, 334
MakeTable, 334
maxcode, 334
MaxHuffmanCodeLength, 333
mincode, 334
operator=, 334
PngDecoder, 319
PngHuffmanDecoder, 334
valptr, 334
value_count, 334
PngHuffmanEncoder, 335
~PngHuffmanEncoder, 335
BuildTable, 336
ehufco, 336
ehufsi, 336
Encode, 336
frequencies, 336
IncrementFrequency, 336
MakeTable, 336
operator=, 336
PngEncoder, 336
PngHuffmanEncoder, 335
Reset, 336
PngImage
imagetyp.h, 488
PngImageHeader, 337
bitdepth, 337
colortype, 337
compressionmethod, 337
filtermethod, 337
height, 337
interlacemethod, 337
width, 337
PngLengthOrder
png.cpp, 554
pngpvt.h, 569
PngLengthSize
pngpvt.h, 570
PngLongestLength
pngpvt.h, 570
PngMaxDistanceCodes
pngpvt.h, 570
PngMaxDistanceCodeSize
pngpvt.h, 570
PngMaxLengthCodes
pngpvt.h, 570
PngMaxLengthCodeSize
pngpvt.h, 570
PngMaxLengthLengthCodeSize
pngpvt.h, 570
PngMaxNumberOfHuffmanCodes
pngpvt.h, 570
PngPixelDimensions, 338
pixelsx, 338
pixelsy, 338
unit, 338
pngpvt.h, 568
Binary, 569
ChunkType, 569
FilterAverage, 569
FilterNone, 569
FilterPaeth, 569
FilterSub, 569
FilterUp, 569
Grayscale, 568
GrayscaleAlpha, 569
PaethPredictor, 569
Palette, 569
PngColorType, 568
PngEndCode, 569
PngFilterType, 569
PngFirstLengthCode, 569
PngLengthOrder, 569
PngLengthSize, 570
PngLongestLength, 570
PngMaxDistanceCodes, 570
PngMaxDistanceCodeSize, 570
PngMaxLengthCodes, 570
PngMaxLengthCodeSize, 570
PngMaxLengthLengthCodeSize, 570
PngMaxNumberOfHuffmanCodes, 570
PngSignature, 570
PngSignatureSize, 570
PngWindowMask, 570
PngWindowSize, 570
RGB, 569
RGBAlpha, 569
PngSignature
png.cpp, 554
pngpvt.h, 570
PngSignatureSize
pngpvt.h, 570
PngTimeData, 339
day, 339
hour, 339
minute, 339
month, 339
second, 339
year, 339
PngWindowMask

pngpvt.h, 570
PngWindowSize
pngpvt.h, 570
Point, 340
Chessboard, 341
CityBlock, 341
col, 342
Euclidean, 341
ligne, 342
operator!=, 341
operator<, 341
operator<<, 341
operator<=, 341
operator==, 341
operator>, 341
operator>=, 341
operator>>, 341
Point, 340
X, 341
Y, 341
Point.hpp, 571
operator<<, 571
operator>>, 571
POINT_OUT_OF_IMAGE
EspaceCalibrage.h, 426
PointColore, 344
hue, 344
indice, 344
intensity, 344
position, 344
saturation, 344
Pointeur
FormatXPM, 145
position
PointColore, 344
PositionStack
BlobColoring, 43
PositionXStack
BlobColoring, 43
PositionYStack
BlobColoring, 43
PPM
ImageStream, 199
PPM_ASCII
FormatPortable.hpp, 452
PPM_RAW
FormatPortable.hpp, 452
previous
PngEncoder::HashEntry, 180
PREWITT
Operateur, 299
PREWITTH
Operateur, 299
PREWITTV
Operateur, 299
Print
JpegDecoderDataUnit, 226
JpegDecoderQuantizationTable, 228
print_mat
MATRIX.cpp, 531
MATRIX.H, 533
PrintAcData
JpegEncoderComponent, 243
PrintAcFirst
JpegEncoder, 235
PrintAcRefine
JpegEncoder, 235
PrintComment
JpegEncoder, 235
PrintDcData
JpegEncoderComponent, 243
PrintDcFirst
JpegEncoder, 235
PrintDcRefine
JpegEncoder, 235
PrintEobRun
JpegEncoderComponent, 243
PrintHuffmanTables
JpegEncoder, 235
PrintProgressiveFrame
JpegEncoder, 235
PrintProgressiveScan
JpegEncoder, 235
PrintQuantizationTables
JpegEncoder, 235
PrintRefineEobRun
JpegEncoderComponent, 243
PrintSequentialFrame
JpegEncoder, 235
PrintSequentialScan
JpegEncoder, 235
PrintTable
JpegEncoderHuffmanTable, 249
PRLE
FormatPCX, 136
PRO_ECRAN
GrabInit.h, 474
PRO_METEOR2
GrabInit.h, 474
Process
BlobColoring, 42
Cognois::AcquisitionBalle, 17
Cognois::AcquisitionCible, 20
Cognois::AcquisitionSurDisque, 26
Cognois::AcquisitionSurEcran, 28
Cognois::CounterOCR, 85
Cognois::MakeAVI, 261
Cognois::NoopToServer, 282
Cognois::OCRFontCreator, 285
Cognois::PlanarToPacked, 312
Cognois::ProcessingObject, 350
Cognois::SequentialProcessing, 359

ProcessBackground
PngDecoder, 318
ProcessChromaticities
PngDecoder, 318
ProcessCompressedText
PngDecoder, 318
ProcessData
PngDecoder, 318
ProcessGamma
PngDecoder, 318
ProcessHeader
PngDecoder, 318
ProcessHistogram
PngDecoder, 318
ProcessImageData
PngEncoder, 329
ProcessImageTime
PngDecoder, 318
ProcessingObject.h, 572
processor
Cognois::OfflineProcessing, 289
Processor
Cognois::CameraClientController, 57
ProcessPalette
PngDecoder, 318
ProcessPhysicalPixelDimensions
PngDecoder, 318
ProcessRestartMarker
JpegDecoder, 215
ProcessSignificantBits
PngDecoder, 318
ProcessTextualData
PngDecoder, 318
ProcessTransparency
PngDecoder, 318
progress_data
BitmapImage, 35
BitmapImageCoder, 38
progress_function
BitmapImage, 35
BitmapImageCoder, 38
PROGRESSFUNCTION
bitimage.h, 400
progressive_frame
JpegDecoder, 218
progressive_mode
JpegEncoder, 237
ProgressiveAcFirst
JpegEncoderComponent, 243
ProgressiveAcRefine
JpegEncoderComponent, 243
ProgressiveDcFirst
JpegEncoderComponent, 243
ProgressiveDcRefine
JpegEncoderComponent, 243
ProgressiveInverseDct
JpegDecoderComponent, 222
Projeter
Cognois::CameraModel, 63
Propagate
BlobColoring, 42
PseudoC
FormatXWD, 147
PSize
BlobColoring, 43
pt
ImageGris::PRep, 348
ImageRGB::PRep, 347
PutCross
Picture, 309
PWidth
BlobColoring, 43
q
Contour::HoraudAngle, 181
QRFAC.cpp, 573
c__1, 573
qrfac_, 573
qrfac_
F2C.H, 438
QRFAC.cpp, 573
QRSOLV.cpp, 574
qrsolv_, 574
qrsolv_
F2C.H, 438
QRSOLV.cpp, 574
QualiteExtraction
Cognois::EvaluationDeQualite, 117
QualiteReconstruction
Cognois::EvaluationDeQualite, 117
quantization_table
JpegDecoderComponent, 224
JpegEncoderComponent, 245
quantization_tables
JpegDecoder, 219
QuantizationIntegerScale
JpegDecoderQuantizationTable, 227
quantized_image
GifEncoder, 175
QuantizedColor
BitmapImage, 35
QuantizeSourceImage
BitmapImage, 34
queueLimit
Cognois::AcquisitionSurDisque, 26
r
complex, 73
doublecomplex, 90
Multitype, 280
R_fp
F2C.H, 437
Rapport
TesterContour, 365

RAS
ImageStream, 199
RBrace
FormatXPM, 145
xbmdecod.cpp, 582
RBracket
FormatXPM, 145
xbmdecod.cpp, 582
read_multiple_images
GifDecoder, 172
ReadAcFirst
JpegDecoder, 215
ReadAcRefine
JpegDecoder, 215
ReadApplication
JpegDecoder, 215
ReadASCII
FormatPBM, 134
FormatPGM, 138
FormatPPM, 141
ReadByte
JpegDecoder, 215
ReadChunk
PngDecoder, 318
ReadCompressedData
GifDecoder, 170
ReadDcFirst
JpegDecoder, 215
ReadDcRefine
JpegDecoder, 215
ReadHuffmanTable
JpegDecoder, 215
ReadImage
BitmapImageDecoder, 39
FormatBMP, 128
FormatContour, 130
FormatImage, 132
FormatPBM, 134
FormatPCX, 137
FormatPGM, 138
FormatPPM, 141
FormatSunRaster, 143
FormatXPM, 145
FormatXWD, 148
GifDecoder, 170
imagetyp.h, 488
JpegDecoder, 215
PngDecoder, 318
XbmDecoder, 374
reading_pixel_data
PngDecoder, 322
ReadInterlaced
PngDecoder, 319
ReadLengths
PngDecoder, 319
ReadMarker
JpegDecoder, 216
ReadNextImage
GifDecoder, 171
ReadNoninterlaced
PngDecoder, 319
ReadPixelData
PngDecoder, 319
ReadProgressiveScanData
JpegDecoder, 216
ReadQuantization
JpegDecoder, 216
ReadRAW
FormatPBM, 134
FormatPGM, 138
FormatPPM, 141
ReadRestartInterval
JpegDecoder, 216
ReadSequentialInterleavedScan
JpegDecoder, 216
ReadSequentialNonInterleavedScan
JpegDecoder, 216
ReadSequentialScanData
JpegDecoder, 216
ReadSignature
PngDecoder, 319
ReadStartOfFrame
JpegDecoder, 216
ReadStartOfScan
JpegDecoder, 216
ReadStreamHeader
GifDecoder, 171
JpegDecoder, 216
ReadTable
JpegDecoderQuantizationTable, 228
JpegHuffmanDecoder, 253
ReadToNextImage
GifDecoder, 171
ReadV1
FormatContour, 130
ReadWord
JpegDecoder, 216
READY_IN_BIT
OnlineProcessing.cpp, 546
READY_OUT_BIT
OnlineProcessing.cpp, 546
real
F2C.H, 438
Receive
JpegDecoder, 216
red
BitmapImage::ColorMapEntry, 68
FormatXWD::xwdColor, 380
GifColorTableEntry, 167
Red
CouleurHSI, 80
RedOffset

BitmapImage, 33
redValue
CouleurRGB, 82
RedValue
DirectMilAccess, 87
redx
PngChromaticitiesData, 313
redy
PngChromaticitiesData, 313
RefineAcCoefficient
JpegDecoder, 216
RefineAcData
JpegEncoder, 235
RegionGet
Image, 187
RegionPut
Image, 187
RegrouperUnitaires
Contour, 75
rep
ImageGris::PRep, 348
ImageRGB::PRep, 347
RepresentFloatData
Picture, 309
RepresentIndexedData
Picture, 309
RESC
FormatSunRaster.hpp, 457
reserved
FormatPCX::pcxHeader, 305
ReserverDerniereImage
Acquisition, 12
Reset
Cognois::FrameCounter, 153
GifDecoder, 171
JpegEncoderHuffmanTable, 249
PngHuffmanEncoder, 336
ResetDcDifference
JpegDecoderComponent, 222
JpegEncoderComponent, 243
ResetDcDifferences
JpegDecoder, 216
ResetDcValues
JpegEncoder, 236
ResetEobRun
JpegEncoderComponent, 244
ResetEvents
Cognois::OfflineProcessing, 289
Cognois::OnlineProcessing, 295
ResetServer
Cognois::FrameSetClient, 160
Resize
Matrice, 269
restart_interval
JpegDecoder, 219
JpegEncoder, 237
RestartInterval
JpegDecoder, 216
RGB
pngpvt.h, 569
RGB2HSV
ColourSpaceConverter, 71
RGBAlpha
pngpvt.h, 569
rgbBlue
RGBQUADsd, 353
RGBConvert
JpegDecoderComponent, 222
rgbGreen
RGBQUADsd, 353
RGBQUADsd, 353
rgbBlue, 353
rgbGreen, 353
rgbRed, 353
rgbReserved, 353
rgbRed
RGBQUADsd, 353
rgbReserved
RGBQUADsd, 353
RGBToCb
jfif.cpp, 489
jfif.h, 491
RGBToCr
jfif.cpp, 489
jfif.h, 491
RGBtoHSI
Cognois::ExtractionDeCible, 126
RGBToY
jfif.cpp, 489
jfif.h, 491
RGBTOYCBCRFUNCTION
JpegEncoderComponent, 241
RIFF
FormatSunRaster.hpp, 457
RMAGIC
FormatSunRaster.hpp, 457
RNOMAP
FormatSunRaster.hpp, 457
ROBERTS
Operateur, 299
ROBERTSH
Operateur, 299
ROBERTSV
Operateur, 299
RobustStandardDeviation
StatisticsR, 360
ROLD
FormatSunRaster.hpp, 457
RotateByte
xbmpvt.h, 586
Rounding
jfif.cpp, 490

RoundRow
bitimage.cpp, 399
row_buffer_width
PngDecoder, 322
row_buffers
PngDecoder, 322
row_interval
PngDecoder::InterlaceInfo, 207
row_width
BitmapImage, 35
PngEncoder, 332
RowRounding
BitmapImage, 33
rows_done
GifDecoder, 173
rows_per_restart
JpegEncoder, 238
RRAWMAP
FormatSunRaster.hpp, 457
RRGB
FormatSunRaster.hpp, 457
RRGBMAP
FormatSunRaster.hpp, 457
RRLENCODED
FormatSunRaster.hpp, 457
RST0
jpmarker.h, 519
RST1
jpmarker.h, 519
RST2
jpmarker.h, 519
RST3
jpmarker.h, 519
RST4
jpmarker.h, 519
RST5
jpmarker.h, 519
RST6
jpmarker.h, 519
RST7
jpmarker.h, 519
RSTANDARD
FormatSunRaster.hpp, 457
Rt
Cognois::CameraModel, 63
RTIFF
FormatSunRaster.hpp, 457
S_fp
F2C.H, 437
sac
TesterContour, 366
sae
TesterContour, 366
saltpepper
TesterEvolution, 368
Sample1to1Component
JpegEncoderComponent, 244
SampleCbComponent
JpegEncoderComponent, 244
SampleCrComponent
JpegEncoderComponent, 244
SampleNtoNComponent
JpegEncoderComponent, 244
SampleRange
jpdedu.cpp, 495
SampleYComponent
JpegEncoderComponent, 244
sat
TesterContour, 366
saturation
CouleurHSI, 80
PointColore, 344
Saturation
CouleurRGB, 82
SaveData
Cognois::MakeAVI, 261
Scaled
jfif.cpp, 489
ScaleFactor
jfif.cpp, 490
ScaleG
FormatXWD, 147
ScaleValue
jfif.cpp, 490
scamul
MATRIX.cpp, 531
MATRIX.H, 533
scan_component_count
JpegDecoder, 219
JpegEncoder, 238
scan_components
JpegDecoder, 219
JpegEncoder, 238
scan_count
JpegDecoder, 219
JpegEncoder, 238
ScanIsInterleaved
JpegDecoder, 216
screen_height
GifDecoder, 173
screen_width
GifDecoder, 173
ScreenHeight
GifDecoder, 171
ScreenWidth
GifDecoder, 171
SCVolume
Cognois::SpaceCarving::SCVolume, 356
SCVolume.cpp, 575
SCVolume.h, 576
sdm
TesterContour, 366

search_limit
PngEncoder, 332
second
PngTimeData, 339
SegmenterCourbes
Contour, 75
SegmentsDroite
Contour, 75
Semicolon
FormatXPM, 145
xbmdecod.cpp, 582
Send2dPoint
Cognois::FrameSetClient, 160
Send3dPoint
Cognois::CalibrationClient, 53
SendAngle
Cognois::FrameSetClient, 161
SendCalibrationParameters
Cognois::CalibrationClient, 54
SendFrame
Cognois::FrameSetClient, 161
SendImageSize
Cognois::FrameSetClient, 161
SendMode
Cognois::FrameSetClient, 161
SendPoints
Cognois::CalibrationClient, 54
Separer
ContourSegment, 78
SequentialProcessing.h, 577
SERVER_ON
AcquisitionBalle.h, 388
SetAuthor
PngEncoder, 329
SetBlockSize
PngEncoder, 329
SetComment
JpegEncoder, 236
PngEncoder, 329
SetCompressionLevel
PngEncoder, 329
SetCopyright
PngEncoder, 329
setCounting
OnlineProcessing.cpp, 546
SetCrossColor
Picture, 309, 310
SetDescription
PngEncoder, 329
SetDisclaimer
PngEncoder, 329
SetGrayscale
JpegEncoder, 236
SetHeight
Picture, 310
SetHorizontalFrequency
JpegEncoderComponent, 244
SetHotSpotX
XbmEncoder, 377
SetHotSpotY
XbmEncoder, 377
SetHuffmanTables
JpegDecoderComponent, 223
JpegEncoderComponent, 244
SetImageName
XbmEncoder, 377
SetParametresIntrinseques
Cognois::CameraModel, 63
SetProcessor
Cognois::OfflineProcessing, 289
Cognois::OnlineProcessing, 295
SetProgressFunction
BitmapImage, 35
BitmapImageCoder, 38
SetProgressive
JpegEncoder, 236
SetQuality
JpegEncoder, 236
SetQuantizationTable
JpegDecoderComponent, 223
JpegEncoderComponent, 244
SetReadMultipleImages
GifDecoder, 171
setReady
OnlineProcessing.cpp, 546
SetRowsPerRestart
JpegEncoder, 236
SetSamplingFrequency
JpegEncoder, 236
SetScanAttributes
JpegEncoder, 236
SetSize
BitmapImage, 35
SetSoftware
PngEncoder, 329
SetSource
PngEncoder, 329
SetTempsDebut
AcquisitionImage, 22
AcquisitionImageListe, 24
SetTempsFin
AcquisitionImage, 22
AcquisitionImageListe, 24
SetTitle
PngEncoder, 329
SetType
Picture, 310
SetUseFilters
PngEncoder, 330
SetVerbose
Cognois::FrameCounter, 153
FormatImage, 133

GifDecoder, 171
ImageStream, 201
JpegDecoder, 216
PngDecoder, 319
SetVerboseFlag
GifDecoder, 171
SetVerticalFrequency
JpegEncoderComponent, 244
SetWarning
PngEncoder, 330
SetWidth
Picture, 310
SEUIL_2_ITERATIONS
ExtractionDeCible.cpp, 431
SEUIL_2_TOLERANCE
ExtractionDeCible.cpp, 431
Seuils
Cognois::ExtractionDeCible, 126
shortint
F2C.H, 438
shortlogical
F2C.H, 438
Shrink
ImageBinaire, 191
sign
Matrice.cpp, 528
signature
FormatPCX::pcxHeader, 305
SingleCam2MultiCam
Cognois::CameraModel, 63
Size
Cognois::CameraModel, 64
Picture, 310
sizes_found
JpegEncoderHuffmanTable, 249
Skeleton
ImageBinaire, 191
Slash
FormatXPM, 145
slm
TesterContour, 366
SOB
jpmarker.h, 520
SOBEL
Operateur, 299
SOBELH
Operateur, 299
SOBELV
Operateur, 299
sof_found
JpegDecoder, 219
SOF0
jpmarker.h, 519
SOF1
jpmarker.h, 519
SOF2
jpmarker.h, 519
SOF3
jpmarker.h, 519
SOF5
jpmarker.h, 519
SOF6
jpmarker.h, 519
SOF7
jpmarker.h, 519
SOF9
jpmarker.h, 519
SOFA
jpmarker.h, 519
SOFB
jpmarker.h, 519
SOFD
jpmarker.h, 519
SOFE
jpmarker.h, 519
SOFF
jpmarker.h, 519
software_string
PngEncoder, 332
SOI
jpmarker.h, 519
solve_system
MATRIX.cpp, 531
MATRIX.H, 533
sortie_contours
TstParam, 369
sortie_resultats
TstParam, 369
SortMatrix
Cognois::ExtractionDeCible, 126
SOS
jpmarker.h, 519
source_string
PngEncoder, 332
spectral_selection_end
JpegEncoder::Scan, 354
spectral_selection_start
JpegEncoder::Scan, 354
SplitAreaInHalf
BitmapImage, 35
SQUARE
bitimage.cpp, 399
stack_pointer
GifDecoder, 173
StandardDeviation
StatisticsR, 360
start
Cognois::AcquisitionBalle, 18
start_col
PngDecoder::InterlaceInfo, 207
start_row
PngDecoder::InterlaceInfo, 207

StartChunk
PngEncoder, 330
StartNewDataSet
PngDecoder, 319
Static
FormatXPM, 144
xbmdecod.cpp, 582
StaticC
FormatXWD, 147
StaticG
FormatXWD, 147
Statistics.h, 578
ABS, 578
StatisticsR, 360
Max, 360
Mean, 360
Median, 360
MedianDeviation, 360
RobustStandardDeviation, 360
StandardDeviation, 360
Statistique
ImageBinaire, 190
Statistiques
Cognois::ExtractionDeCible, 126
stats
ExtractionDeCible.cpp, 432
std, 8
std::exception, 120
std::vector, 372
stop
Cognois::AcquisitionBalle, 18
StoreByte
GifDecoder, 171
stream_adler
PngDecoder, 322
strict_jfif
JpegDecoder, 219
SubMatrix
Matrice, 269
successive_approximation
JpegEncoder::Scan, 354
successive_approximation_high
JpegEncoder::Scan, 354
successive_approximation_low
JpegEncoder::Scan, 354
SurCible
Cognois::ExtractionDeCible, 126
SVD
Matrice, 269
SWAP
EspaceCalibrage.h, 426
SYNC_SERVER_PERIOD
OfflineProcessing.cpp, 542
SYNCHRONIZATION
OfflineProcessing.cpp, 542
OnlineProcessing.cpp, 546
Systeme.hpp, 579
BIGENDIAN_SYSTEM, 579
FVDESSIN_SYSTEM, 579
SystemToBigEndian
datatype.h, 421
SystemToEndian
FormatXWD, 148
SystemToLittleEndian
datatype.h, 421
t
TstParam, 369
T
Kernel, 255
TAB
FormatPortable.hpp, 452
table_defined
JpegDecoderQuantizationTable, 228
JpegHuffmanDecoder, 253
table_made
pngchksm.cpp, 556
Table2D, 362
TableHO, 363
TableSize
gifdecod.cpp, 468
TesterContour, 364
compedges, 366
EffectuerTest, 365
imageorg, 366
muet, 366
Muet, 365
Note, 365
NotePoints, 365
NoteSegments, 365
pac, 366
pae, 366
param, 366
pat, 366
pdm, 366
Rapport, 365
sac, 366
sae, 366
sat, 366
sdm, 366
slm, 366
TesterContour, 364
TesterPoints, 365
TesterSegments, 365
tolpoints, 366
tolsegments, 366
TesterContour.hpp, 580
TesterEvolution, 367
compedges, 368
Fonction, 367
image, 368
notes, 368
par, 368

saltpepper, 368
TesterEvolution, 367
tolpt, 368
tolsg, 368
variance, 368
TesterEvolution.hpp, 581
TesterPoints
TesterContour, 365
TesterSegments
TesterContour, 365
ThreadAcquisition
Cognois::OfflineProcessing, 289
Cognois::OnlineProcessing, 295
OnlineProcessing.cpp, 546
threadGrabbingHasEnded
Cognois::OfflineProcessing, 290
ThreadSynchronization
Cognois::OnlineProcessing, 295
OnlineProcessing.cpp, 546
ThreadTraitement
Cognois::OnlineProcessing, 295
OnlineProcessing.cpp, 546
ThreeByteThumbnail
JfifExtension, 210
TimeSinceLastFrame
Cognois::FrameCounter, 153
title_string
PngEncoder, 332
Token
FormatXPM, 144
xbmdecod.cpp, 582
TOL_DET
MATRIX.cpp, 530
TOLERANCE_INC
ExtractionDeCible.cpp, 431
TOLERANCE_MAX
ExtractionDeCible.cpp, 431
TOLERANCE_MIN
ExtractionDeCible.cpp, 431
tolpoints
TesterContour, 366
tolpt
TesterEvolution, 368
tolsegments
TesterContour, 366
tolsg
TesterEvolution, 368
top_position
GifImageDescriptor, 178
total_passes
JpegEncoder, 238
TransformInverse
Cognois::CameraModel, 64
transmission
Cognois::AcquisitionBalle, 18
transparent_color
GifGraphicControlBlock, 176
transpose
MATRIX.cpp, 531
MATRIX.H, 533
Treshold
ImageBinaire, 191
Trier
ListeTri, 258
TrouverIntersections
Contour, 75
TrouverPointRouge
Cognois::ExtractionDeCible, 126
TrouverPointsRougeVert
Cognois::ExtractionDeCible, 126
true
datatype.h, 420
TRUE
GrabInit.h, 474
MATRIX.cpp, 530
TRUE_
F2C.H, 436
TrueC
FormatXWD, 147
TstParam, 369
b, 369
entree_comparaison, 369
entree_image, 369
f, 369
g, 369
i, 369
m, 369
sortie_contours, 369
sortie_resultats, 369
t, 369
v, 369
TstParamNote, 370
note, 370
operator==, 370
par, 370
type
FormatSunRaster::rasHeader, 352
Operateur, 302
Vardesc, 371
Type
Cognois::SpaceCarving::SCVolume, 355
Kernel, 255
Matrice, 267
Operateur, 300
Picture, 310
U_fp
F2C.H, 437
ub1
dmat, 89
ub2
dmat, 89
UBYTE1

datatype.h, 421
UBYTE2
datatype.h, 421
UBYTE4
datatype.h, 421
Uncompressed
pngdecod.cpp, 558
unit
PngPixelDimensions, 338
units
JfifHeader, 211
UNKNOWN
Operateur, 299
UnknownImage
imagetyp.h, 488
UNS
Matrice, 267
Unsigned
FormatXPM, 145
xbmdecod.cpp, 582
UpdateImage
BitmapImageCoder, 38
GifDecoder, 171
JpegDecoder, 216
Upsample
JpegDecoderComponent, 223
upsample_data
JpegDecoderComponent, 224
us
ErreurImage, 115
usage
BitmapImage::ColorUsage, 69
useCibleParDefaut
Cognois::ExtractionDeCible, 127
user_constants
MINPACK, 279
user_fcn
MINPACK, 279
USER_IN_BIT
OfflineProcessing.cpp, 542
UTILISE_HOMOGRAPHIE
ExtractionDeCible.cpp, 431
v
TstParam, 369
v_frequency
JpegEncoderComponent, 245
v_period
JpegEncoderComponent, 245
v_sampling
JpegDecoderComponent, 224
ValidateParameters
JpegEncoder, 236
Valider
CouleurHSI, 80
ValiderCorrespondance
Cognois::CameraModel, 64
ValiderPosition
Operateur, 301
valptr
JpegHuffmanDecoder, 253
PngHuffmanDecoder, 334
value_count
PngHuffmanDecoder, 334
values
JpegDecoderDataUnit, 226
Vardesc, 371
addr, 371
dims, 371
F2C.H, 437
name, 371
type, 371
variance
TesterEvolution, 368
vars
Namelist, 281
VDoubles
EspaceCalibrage, 7
verbose_flag
FormatImage, 133
GifDecoder, 173
ImageStream, 201
JpegDecoder, 219
PngDecoder, 322
VERBOSE_SYNC
OnlineProcessing.cpp, 546
veresolution
FormatPCX::pcxHeader, 305
version
Cognois::CameraModel, 64
FormatPCX::pcxHeader, 305
FormatXWD::xwdHeader, 382
JfifHeader, 211
Version
Cognois::CameraModel, 64
vertical_frequency
JpegDecoderComponent, 224
VerticalFrequency
JpegDecoderComponent, 223
verticalResolution
FormatBMP::bmpHeader, 45
VERYVERBOSE
pngdecod.cpp, 558
Vider
Contour, 75
visualclass
FormatXWD::xwdHeader, 382
VMatrices
Cognois::CameraModel, 61
VOID
F2C.H, 436
VPoints
EspaceCalibrage, 7

VPointsXYZ
EspaceCalibrage, 7
WaitToPool
Cognois::FrameSetClient, 162
warning_string
PngEncoder, 332
what
Cognois::Exception, 122
whitepointx
PngChromaticitiesData, 313
whitepointy
PngChromaticitiesData, 313
width
Cognois::Ball, 31
FormatBMP::bmpHeader, 45
FormatPBM::pbmHeader, 303
FormatPGM::pgmHeader, 306
FormatPPM::ppmHeader, 346
FormatSunRaster::rasHeader, 352
FormatXPM::xpmHeader, 379
FormatXWD::xwdHeader, 382
PngImageHeader, 337
Width
BitmapImage, 35
FormatXPM, 145
Image, 187
Picture, 310
xbmdecod.cpp, 582
window_position
PngDecoder, 322
windowborder
FormatXWD::xwdHeader, 382
windowheight
FormatXWD::xwdHeader, 382
windowwidth
FormatXWD::xwdHeader, 382
windowx
FormatXWD::xwdHeader, 382
windowy
FormatXWD::xwdHeader, 382
WriteCurrentChunk
PngEncoder, 330
WriteImage
BitmapImageEncoder, 40
FormatBMP, 129
FormatContour, 130
FormatImage, 133
FormatPBM, 134, 135
FormatPCX, 137
FormatPGM, 138, 139
FormatPPM, 141
FormatSunRaster, 143
FormatXPM, 145
FormatXWD, 148
GifEncoder, 175
JpegEncoder, 236
PngEncoder, 330
XbmEncoder, 377
WriteImageData
PngEncoder, 330
WriteText
PngEncoder, 330
WriteTextBlocks
PngEncoder, 330
x
EspaceCalibrage::Point, 343
EspaceCalibrage::PointXYZ, 345
X
Point, 341
X_TOL
MINPACK.cpp, 537
X11WD_FILE_VERSION
FormatXWD.hpp, 461
XBM
ImageStream, 199
xbmdecod.cpp, 582
Bits, 582
Char, 582
Comma, 582
Define, 582
Equal, 582
Height, 582
HotX, 582
HotY, 582
Integer, 582
Invalid, 582
LBrace, 582
LBracket, 582
NoData, 582
RBrace, 582
RBracket, 582
Semicolon, 582
Static, 582
Token, 582
Unsigned, 582
Width, 582
xbmdecod.h, 583
XbmDecoder, 373
~XbmDecoder, 374
DoCopy, 374
GetHotSpotX, 374
GetHotSpotY, 374
GetImageName, 374
hot_spot_x, 375
hot_spot_y, 375
image_name, 375
Initialize, 374
last_char, 375
lexical_value, 375
NextChar, 374
NextToken, 374
operator=, 374

ReadImage, 374
XbmDecoder, 374
xbmencod.cpp, 584
xbmencod.h, 585
XbmEncoder, 376
~XbmEncoder, 376
DoCopy, 377
GetHotSpotX, 377
GetHotSpotY, 377
GetImageName, 377
hot_spot_x, 377
hot_spot_y, 377
image_name, 377
Initialize, 377
operator=, 377
SetHotSpotX, 377
SetHotSpotY, 377
SetImageName, 377
WriteImage, 377
XbmEncoder, 376
xbmpvt.h, 586
RotateByte, 586
xd
Operateur, 302
xdensity
JfifHeader, 211
xg
Operateur, 302
xmax
FormatPCX::pcxHeader, 305
xmin
FormatPCX::pcxHeader, 305
XmlData
Cognois::FrameSetClient, 162
xoffset
FormatXWD::xwdHeader, 382
XPM
ImageStream, 199
xthumbnail
JfifHeader, 211
XWD
ImageStream, 199
XYPixmap
FormatXWD, 148
y
EspaceCalibrage::Point, 343
EspaceCalibrage::PointXYZ, 345
Y
Point, 341
yb
Operateur, 302
YCbCrToB
jfif.cpp, 489
jfif.h, 491
YCbCrToG
jfif.cpp, 489
jfif.h, 491
YCbCrToR
jfif.cpp, 489
jfif.h, 491
YComponent
JpegEncoder, 232
ydensity
JfifHeader, 211
year
PngTimeData, 339
yh
Operateur, 302
ymax
FormatPCX::pcxHeader, 305
ymin
FormatPCX::pcxHeader, 305
ythumbnail
JfifHeader, 211
z
EspaceCalibrage::PointXYZ, 345
Multitype, 280
Z_f
F2C.H, 436
Z_fp
F2C.H, 437
zbluemask
FormatXWD::xwdHeader, 382
ZeroCrossing
ImageBinaire, 191
ZEROS
Matrice, 267
zgreenmask
FormatXWD::xwdHeader, 382
Zhang
Cognois::Calibration, 47
ZHANG_COMMAND
Calibration.cpp, 403
ZHANG_HOME
Calibration.cpp, 403
ZigZagInput
JpegEncoderCoefficientBlock, 239
JpegEncoderDataUnit, 247
ZPixmap
FormatXWD, 148
zredmask
FormatXWD::xwdHeader, 383