Panoramas, etcetera

© Jerome Boccond-Gibod, Flickr

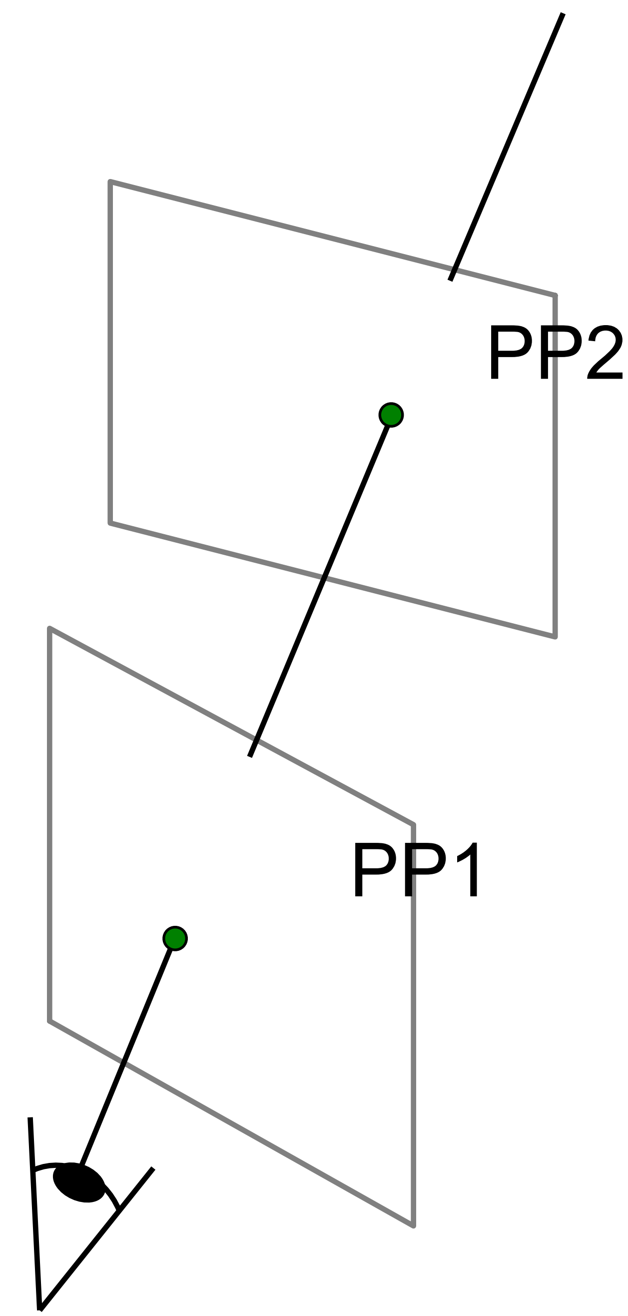
GIF-4105/7105 Photographie Algorithmique, Hiver 2018

Jean-François Lalonde

Merci à A. Efros, R. Szeliski, S. Seitz!

Homographies

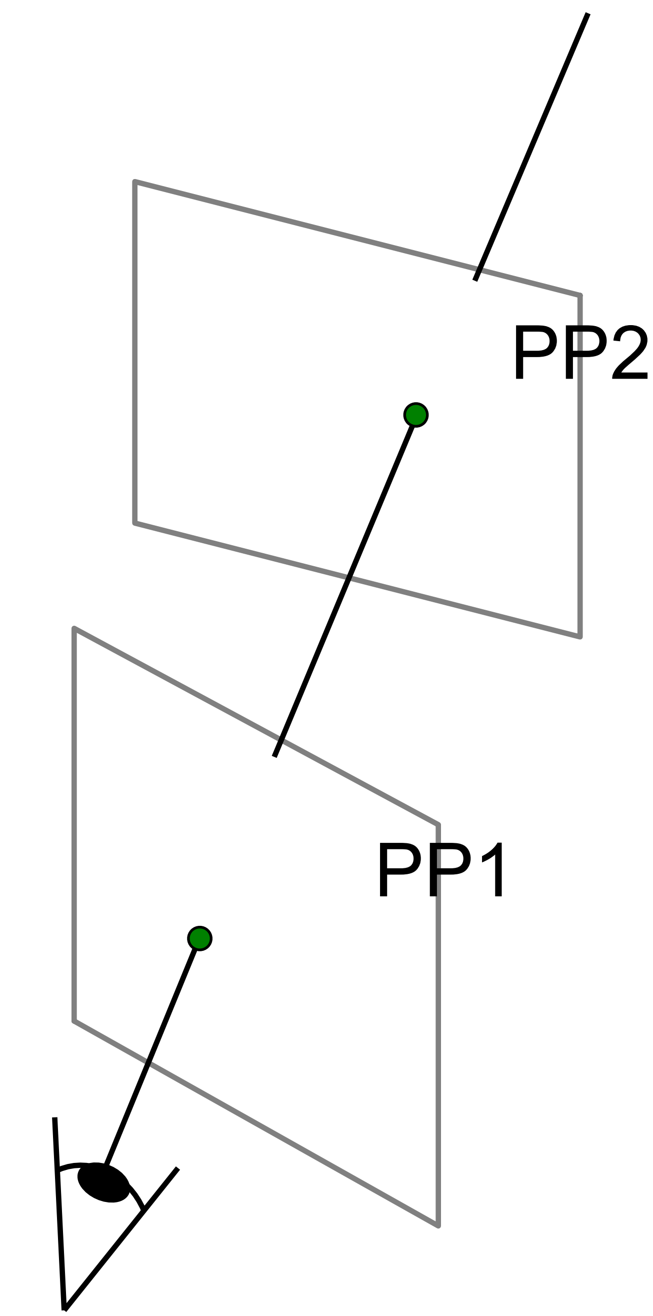
- Transformation entre deux caméras ayant le même centre de projection
- transformation entre deux plans (quadrilatères)
- on perd le parallélisme
- mais les droites sont préservées



Homographies

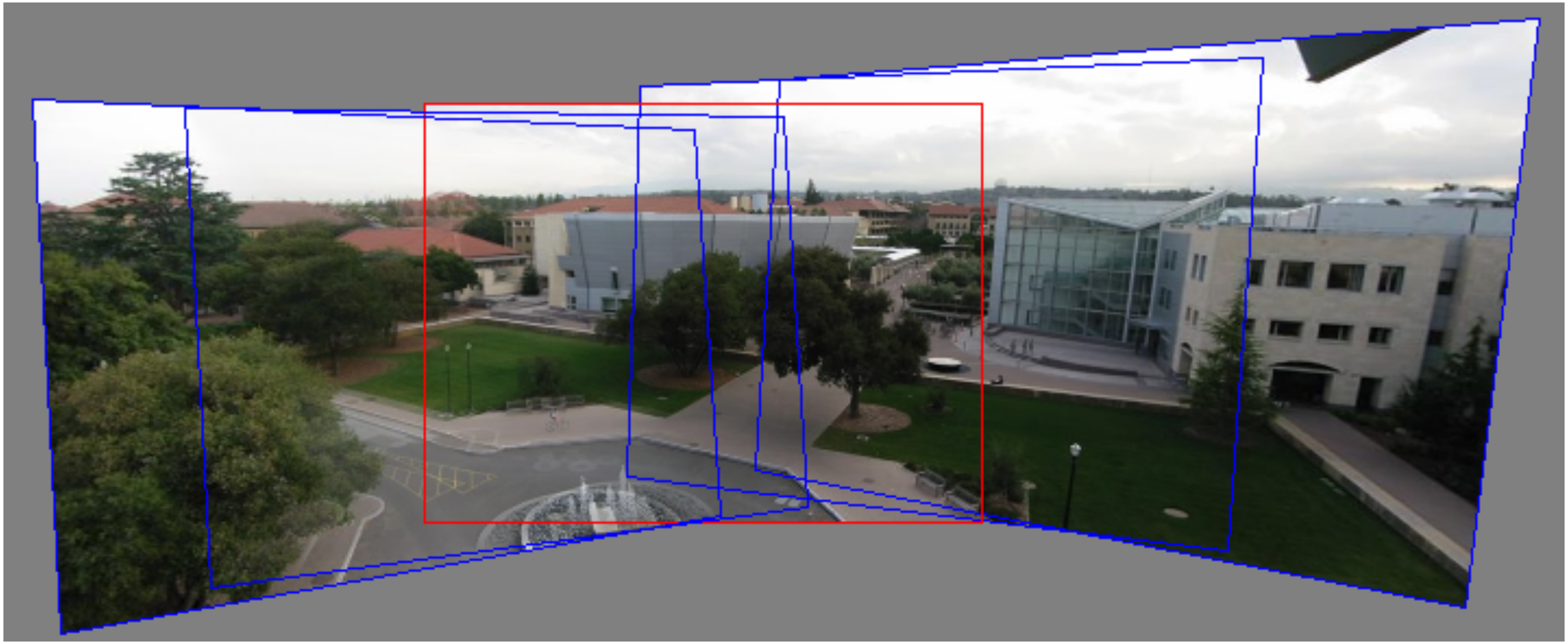
$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$p' = \mathbf{H}p$$

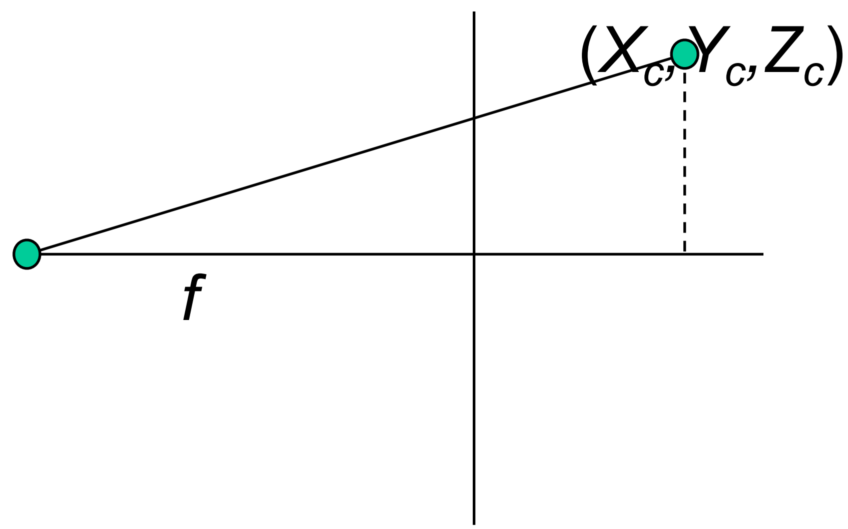


- Pour appliquer une homographie H
  - Calculer p' = Hp (en coordonnées homogènes)
  - Convertir p' en coordonnées dans l'image

Mosaïques de rotation



3D → 2D Projection de perspective



$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{K}} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Rotation 3D

1. Projeter de l'image vers le point 3D

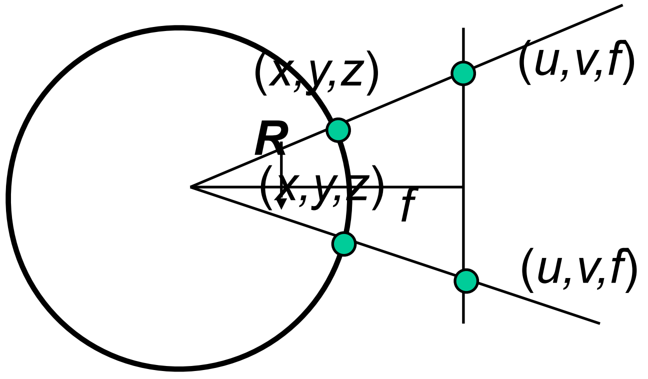
   $(x_0,y_0,z_0) \quad = (u_0-u_c, v_0-v_c, f)$

2. Appliquer la rotation

   $(x_1,y_1,z_1) \quad = \boldsymbol{R}_{01}\,(x_0,y_0,z_0)$

3. Reprojeter dans la nouvelle image

   $(u_1,v_1) = (fx_1/z_1+u_c, fy_1/z_1+v_c)$



Alors

$$\mathbf{H} = \mathbf{K}_0 \mathbf{R}_{01} \mathbf{K}_1^{-1}$$

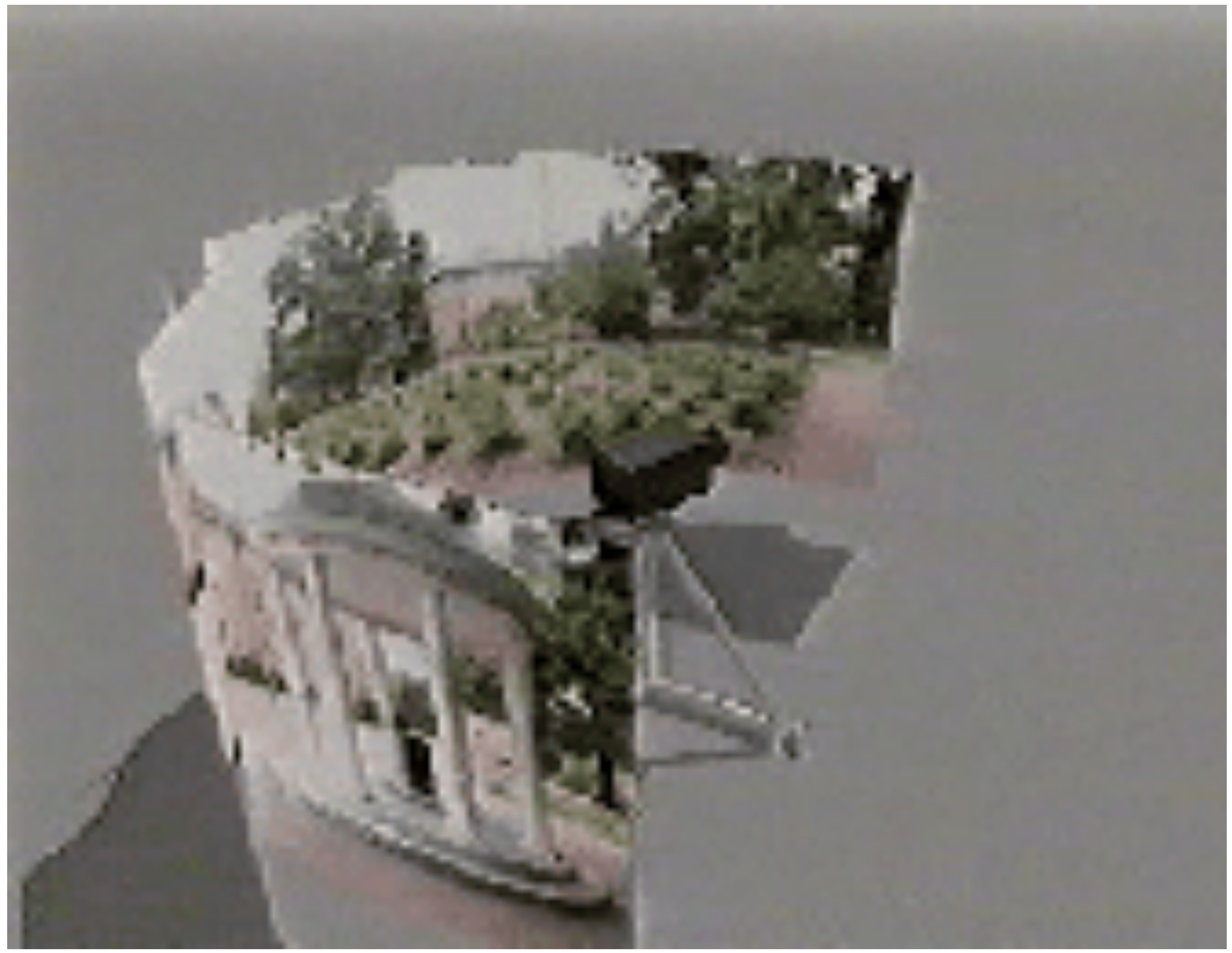
Notre homographie a alors :

3 DDL si la distance focale est connue

4 si elle est la même (et inconnue)

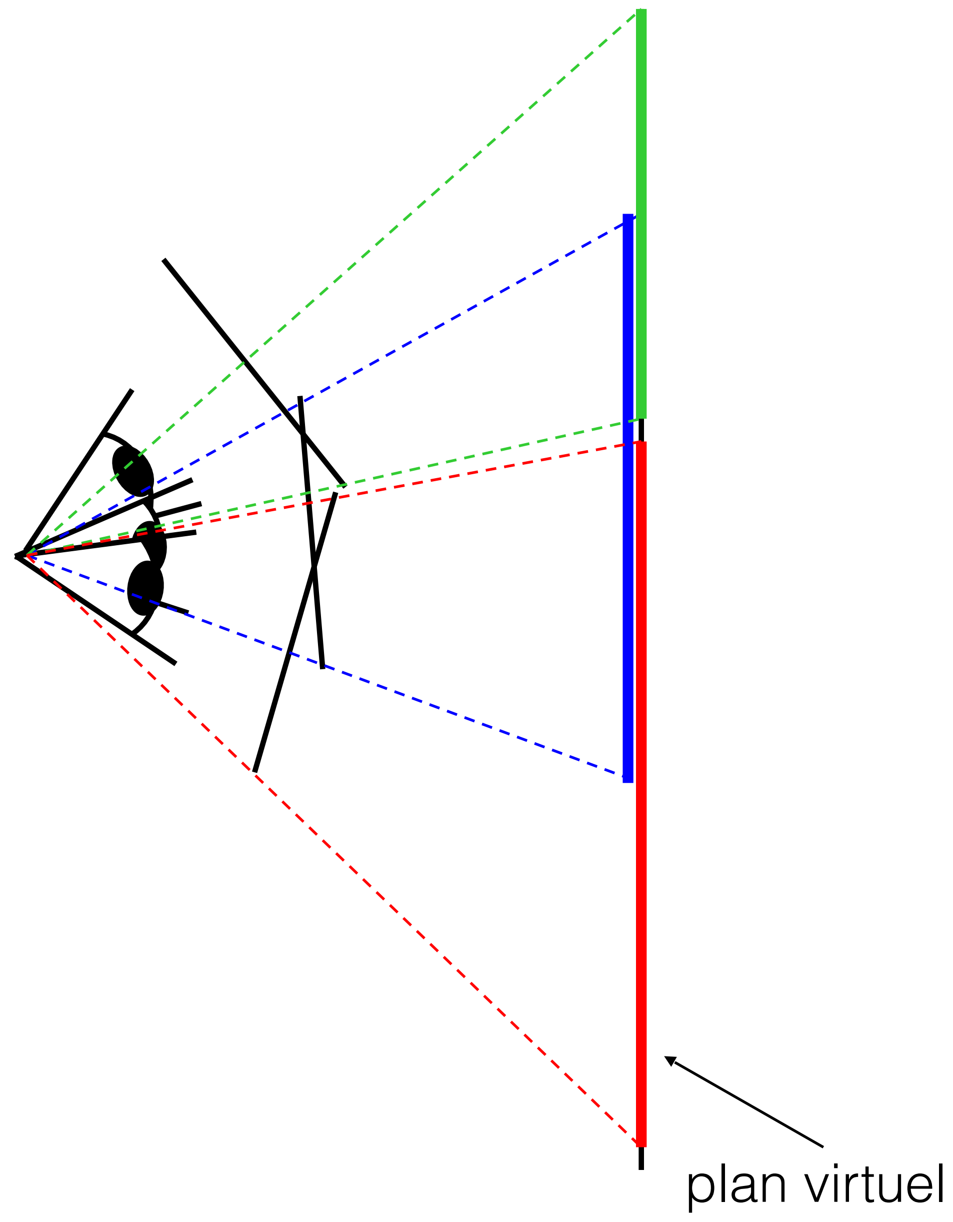
5 si elles sont différentes

Rotation autour de l’axe vertical



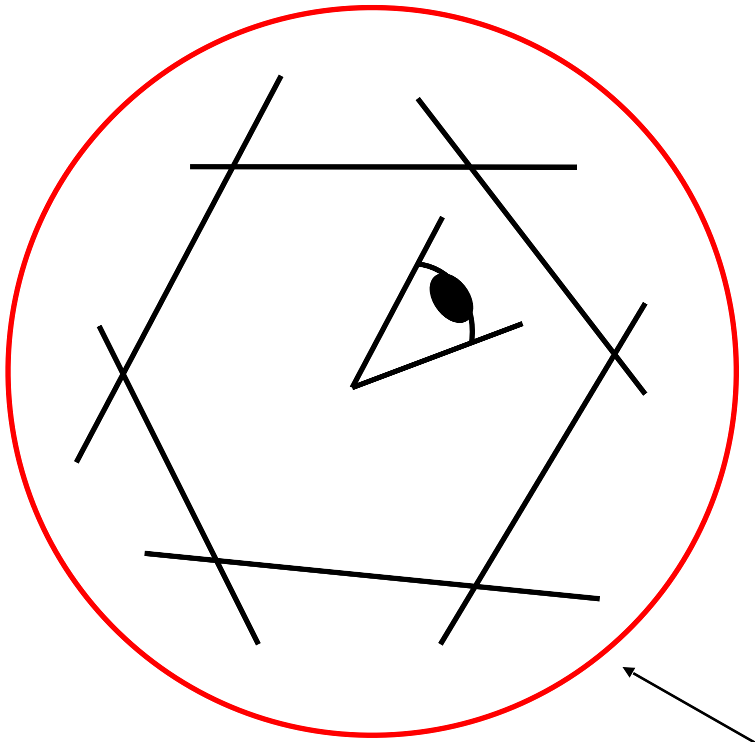
- Si notre caméra est sur un trépied
  - Quelle est la structure de H?

Projection sur un plan?



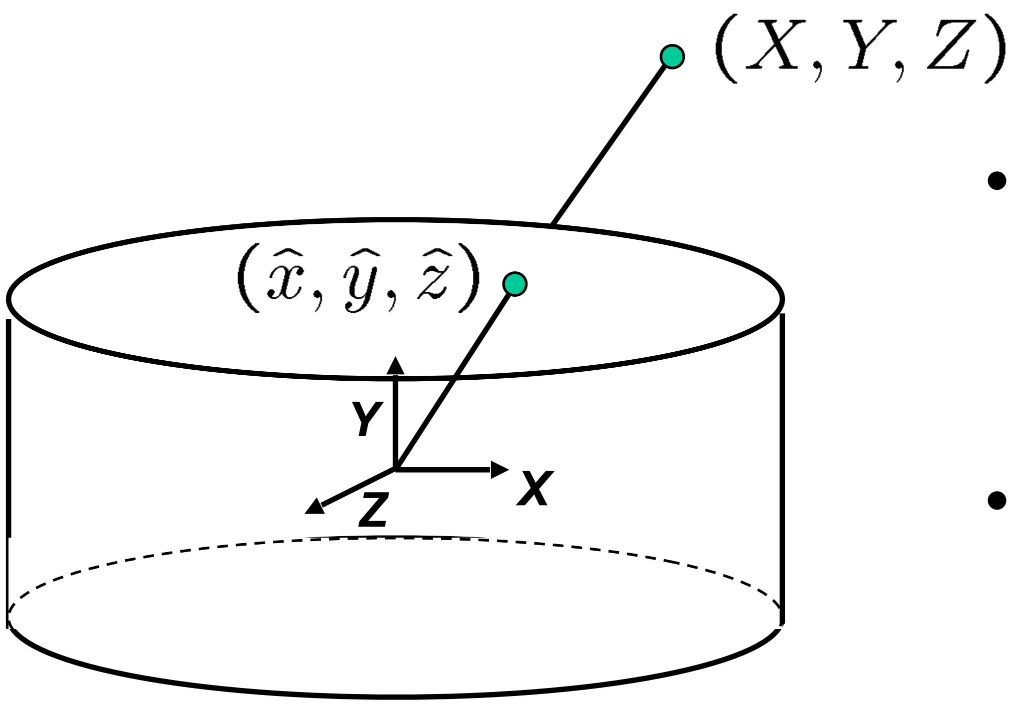
Panoramas complets

- Comment générer des panoramas 360°?



Cylindre de projection!

Projection cylindrique



cylindre unitaire

- Projeter point 3D (X,Y,Z) sur le cylindre

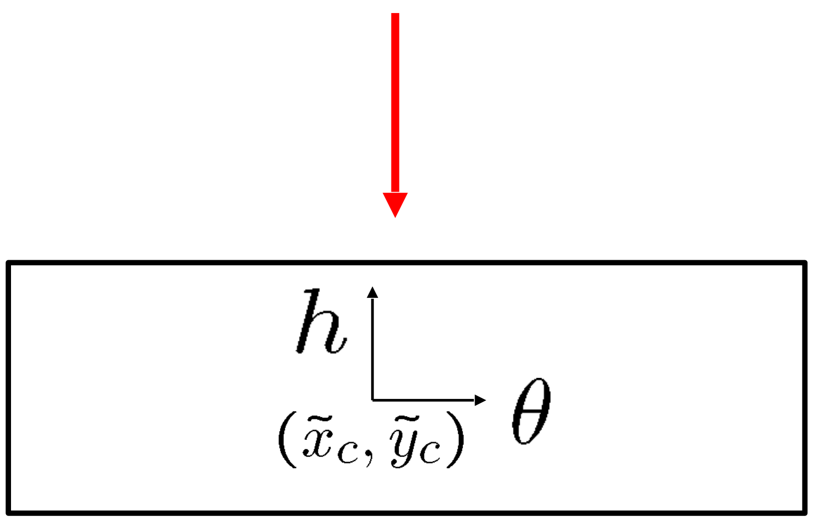
$$(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2+Z^2}}(X, Y, Z)$$

- Convertir en coordonnées cylindriques

$$(sin\theta, h, cos\theta) = (\hat{x}, \hat{y}, \hat{z})$$

- Convertir en coordonnées image (cylindre)

$$(\tilde{x}, \tilde{y}) = (f\theta, fh) + (\tilde{x}_c, \tilde{y}_c)$$



cylindre déroulé

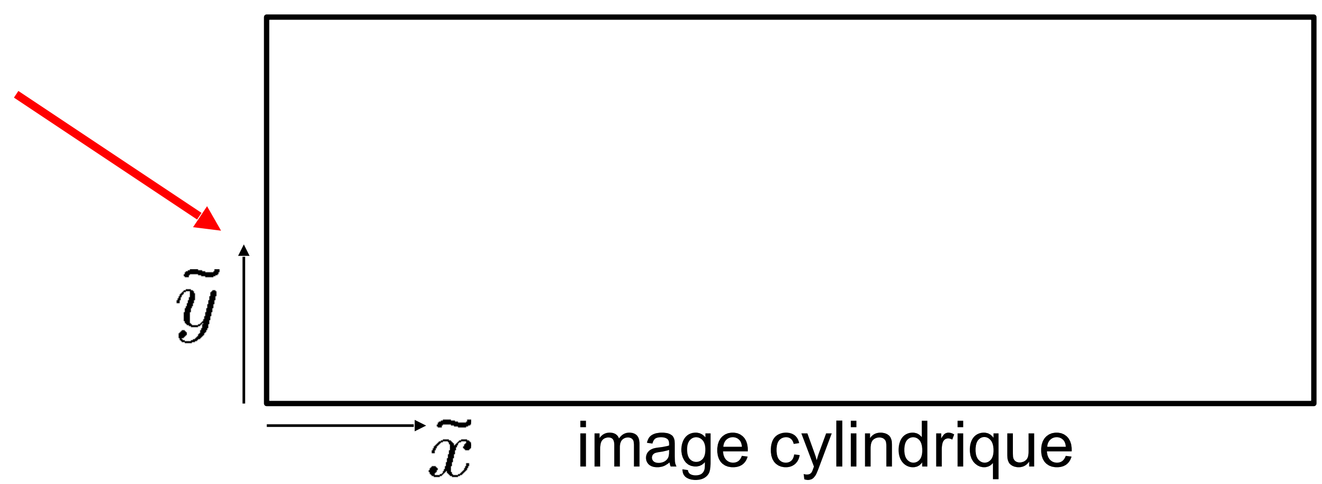
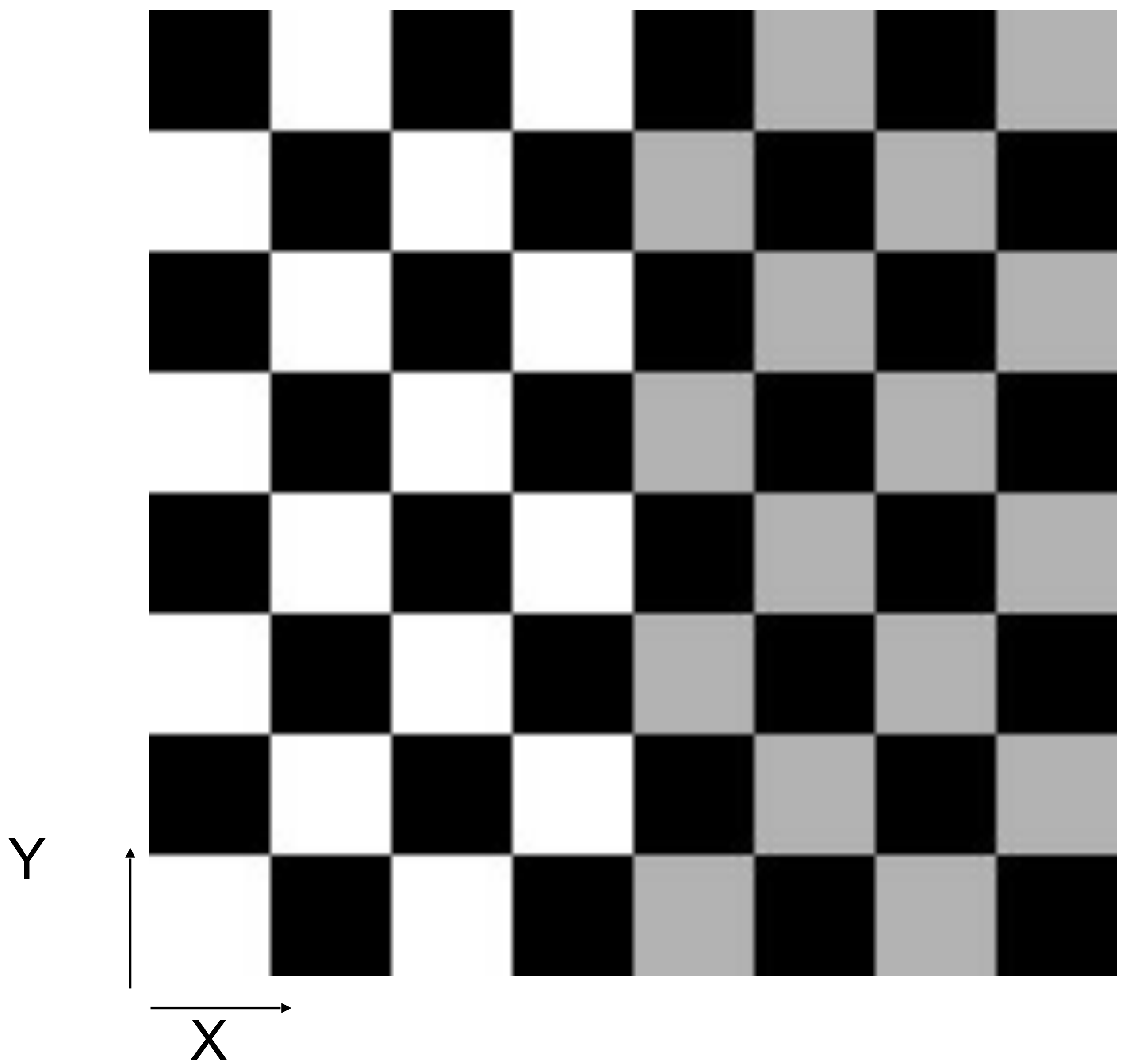
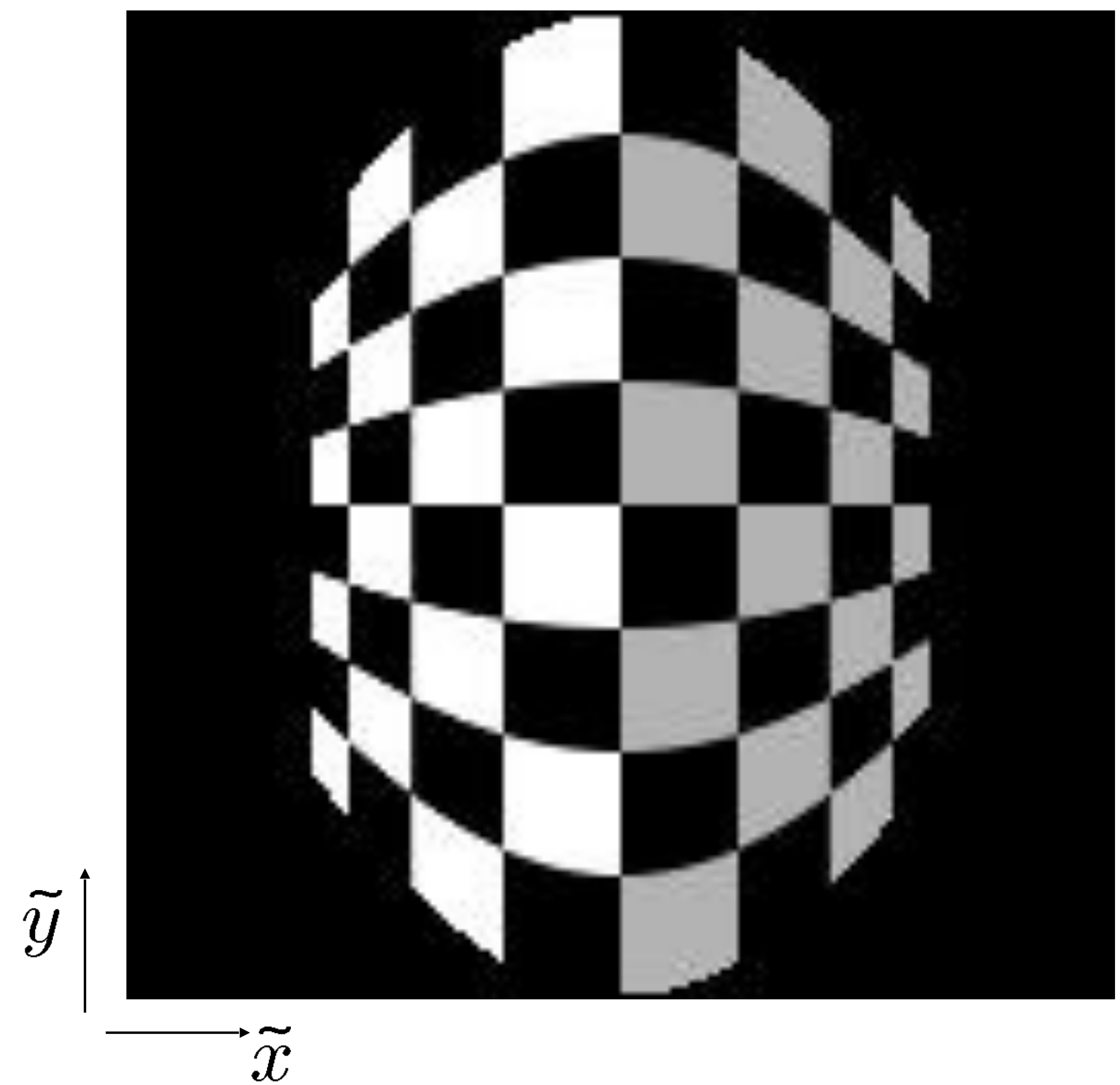


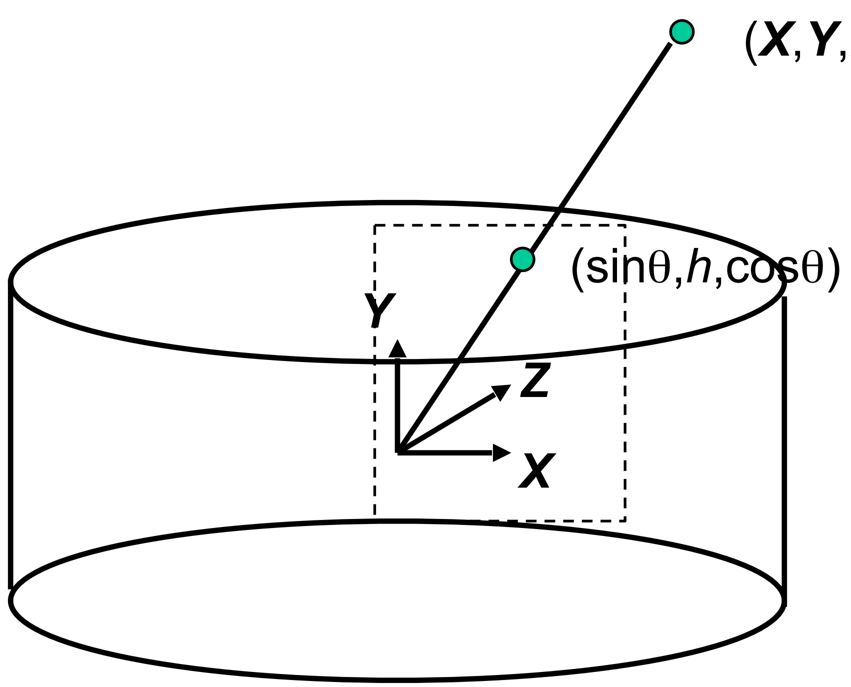
image cylindrique

Projection cylindrique



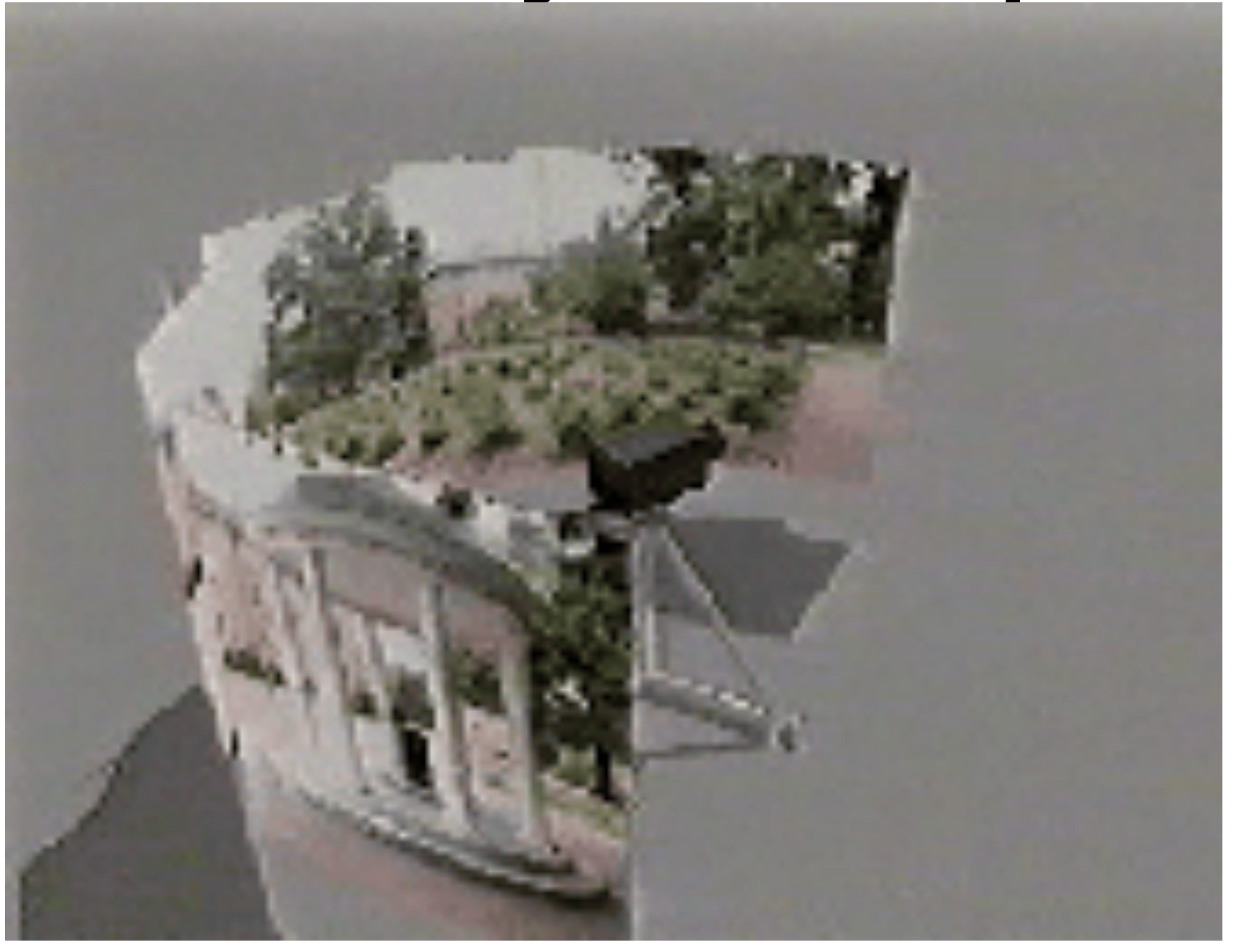


Projection cylindrique inverse



$$
\begin{aligned}
\theta &= (x_{cyl} - x_c)/f \\
h &= (y_{cyl} - y_c)/f \\
\hat{x} &= \sin\theta \\
\hat{y} &= h \\
\hat{z} &= \cos\theta \\
x &= f\hat{x}/\hat{z} + x_c \\
y &= f\hat{y}/\hat{z} + y_c
\end{aligned}
$$

Panoramas cylindriques



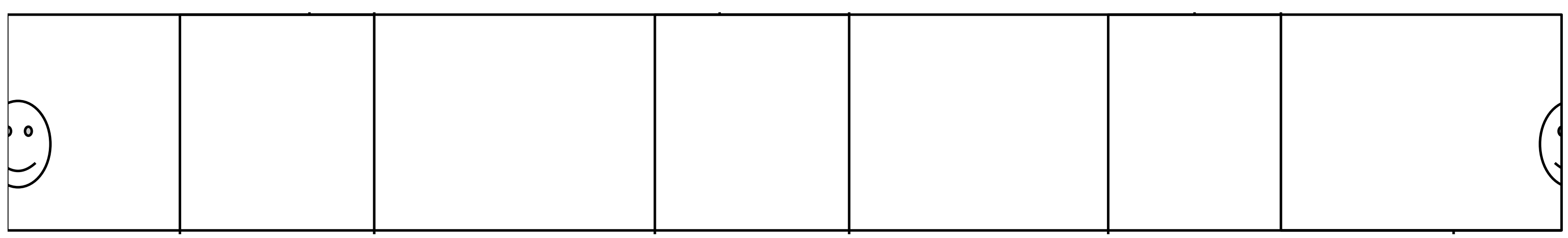
- Étapes (si l'on connaît les rotations)
    - Reprojeter les images sur un cylindre
    - Composer les images

Panoramas cylindriques

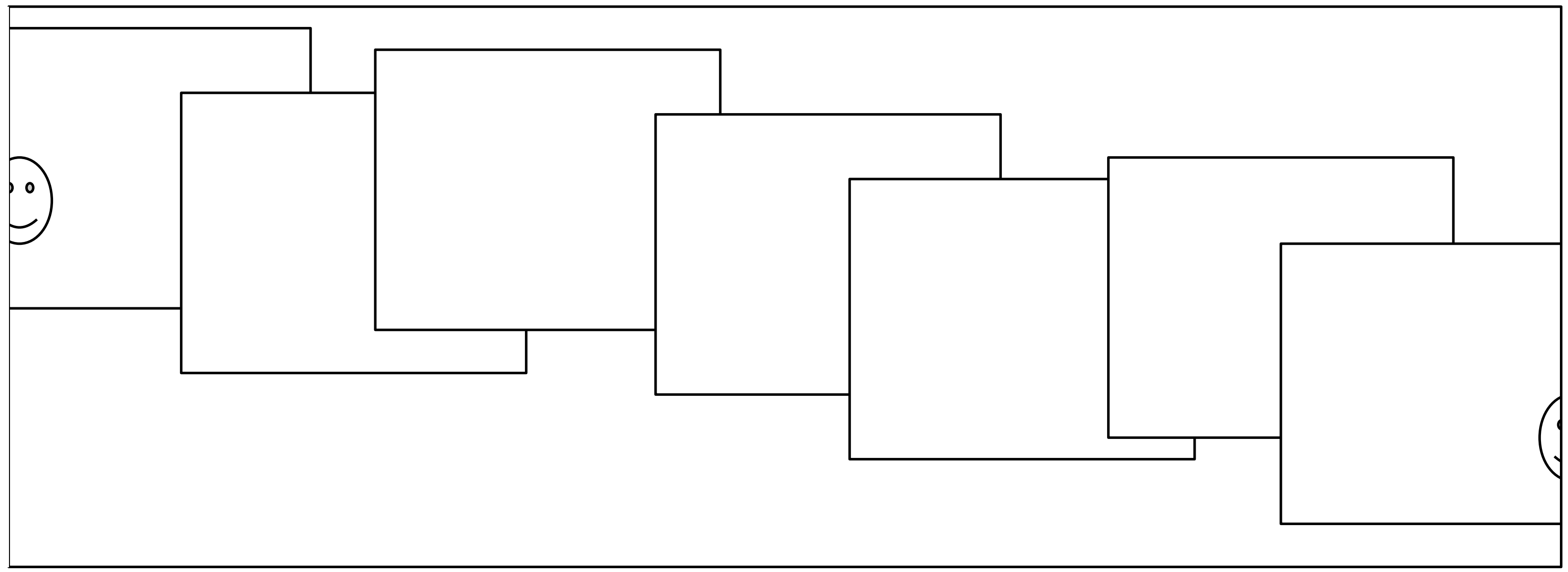
- Si l'on ne connaît pas la matrice de rotation?
  - Il faut la trouver…
    - Rotation de la caméra = translation du cylindre!

Créer le panorama

- Aligner les paires ensemble, composer, et rogner

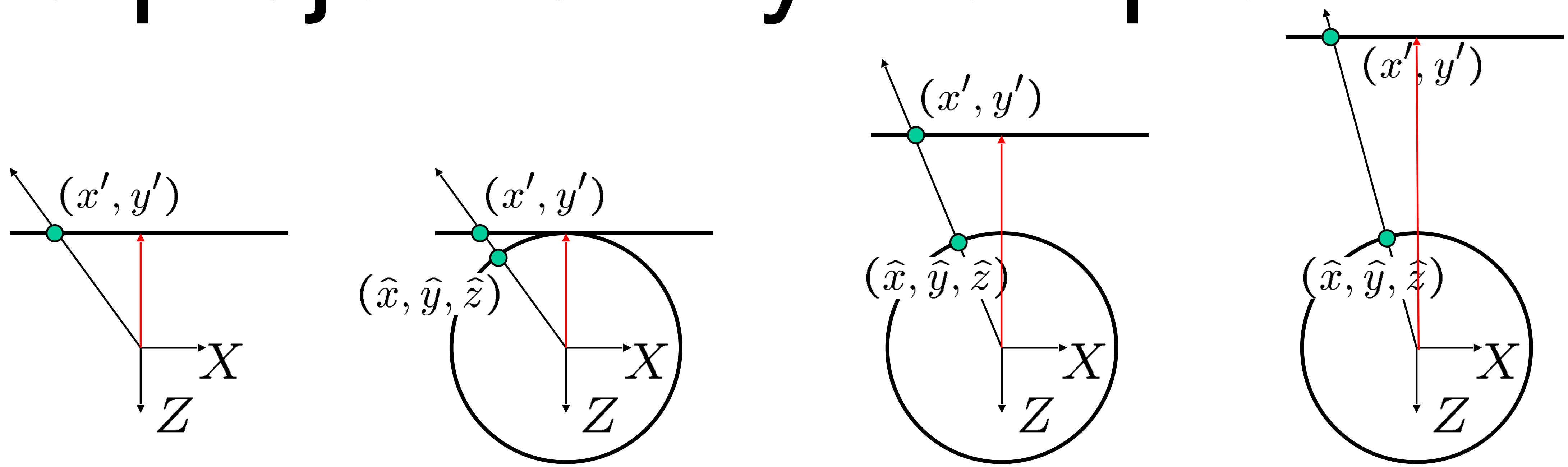


Problème: dérive



- Erreur verticale
  - calculer la correction de telle sorte que la somme = 0
- Erreur horizontale
  - ré-utiliser la première (ou dernière) image

Re-projection cylindrique



vue de haut

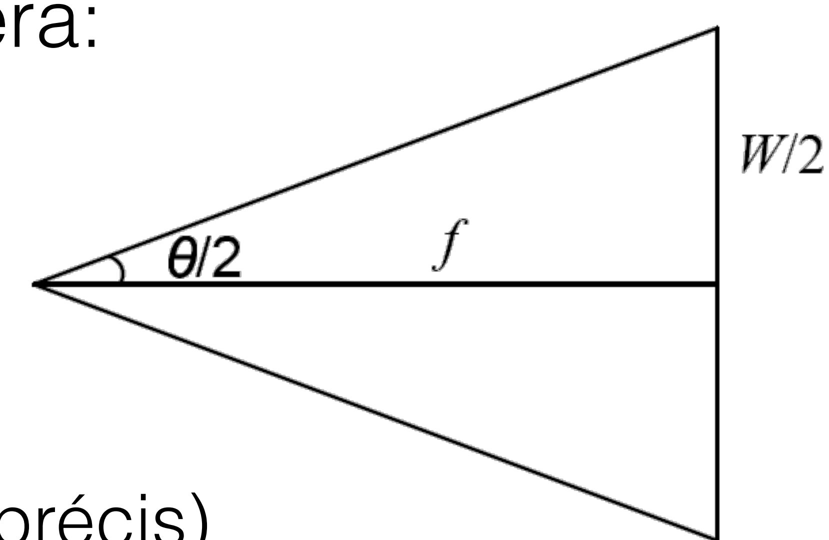
Le secret est dans la … distance focale

Image 384x300 **f = 180 (pixels)** **f = 280** **f = 380**

Panorama 360°

Notre amie la focale

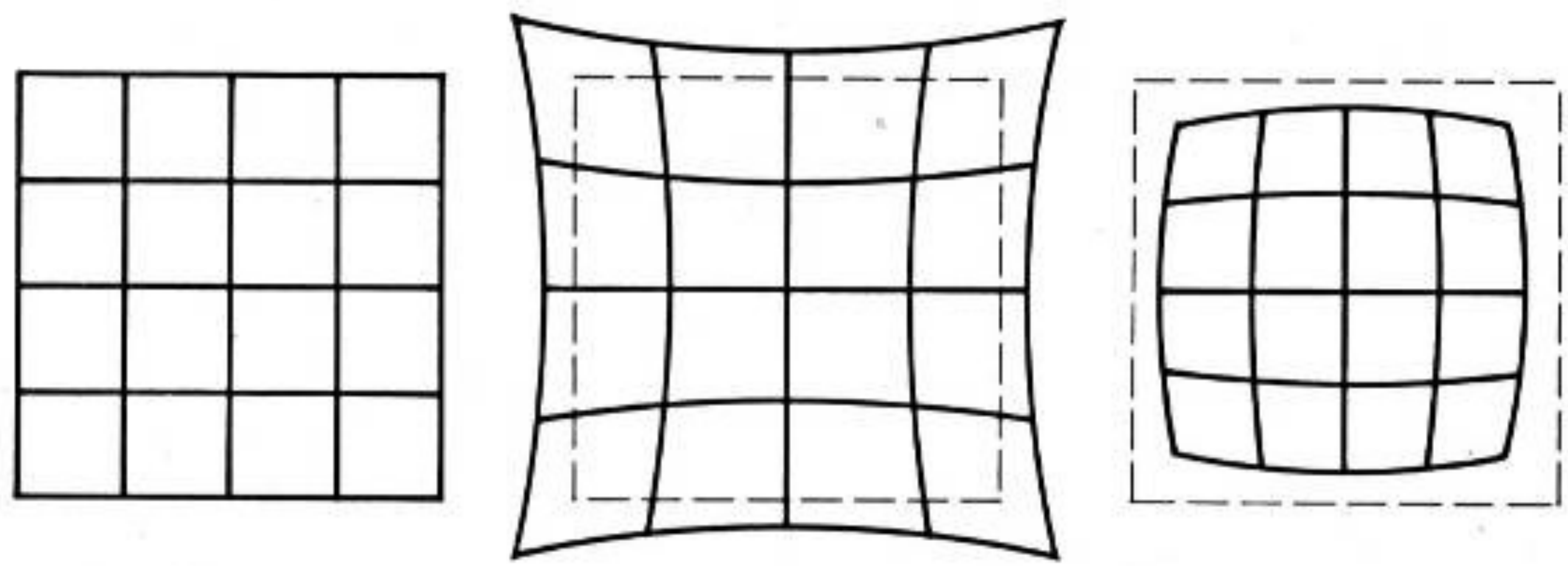
- La distance focale dépend de la caméra:



- On peut l'estimer:
  - à partir du champ de vue
  - de l'information dans l'EXIF (peut être imprécis)
  - en essayant plusieurs valeurs et garder celle qui aligne le panorama
  - en utilisant un objet 3D dont on connaît les dimensions
  - Etc.
- Il y a d'autres paramètres!
  - Centre optique, ratio des pixels, distorsions, etc.

Distorsion radiale

Distorsion radiale



Pas de distorsion "Pin cushion" "Barrel"

- Causée par lentilles imparfaites
- Encore une fois, plus important en bordure de l'image

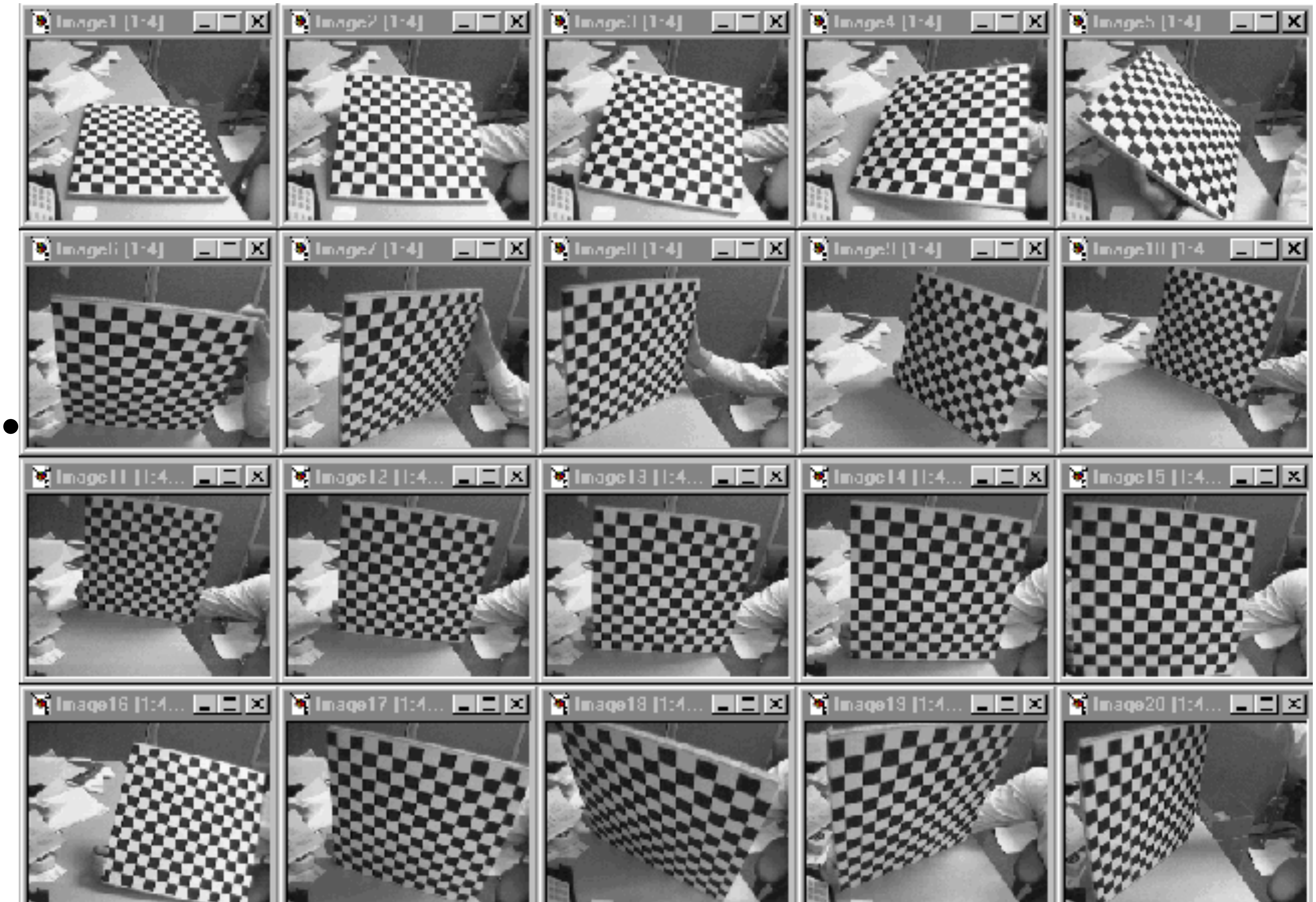
Estimer les paramètres de la caméra?

Intrinsèques Extrinsèques

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} \alpha & s & 0 & u_0 \\ 0 & \beta & 0 & v_0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{23} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

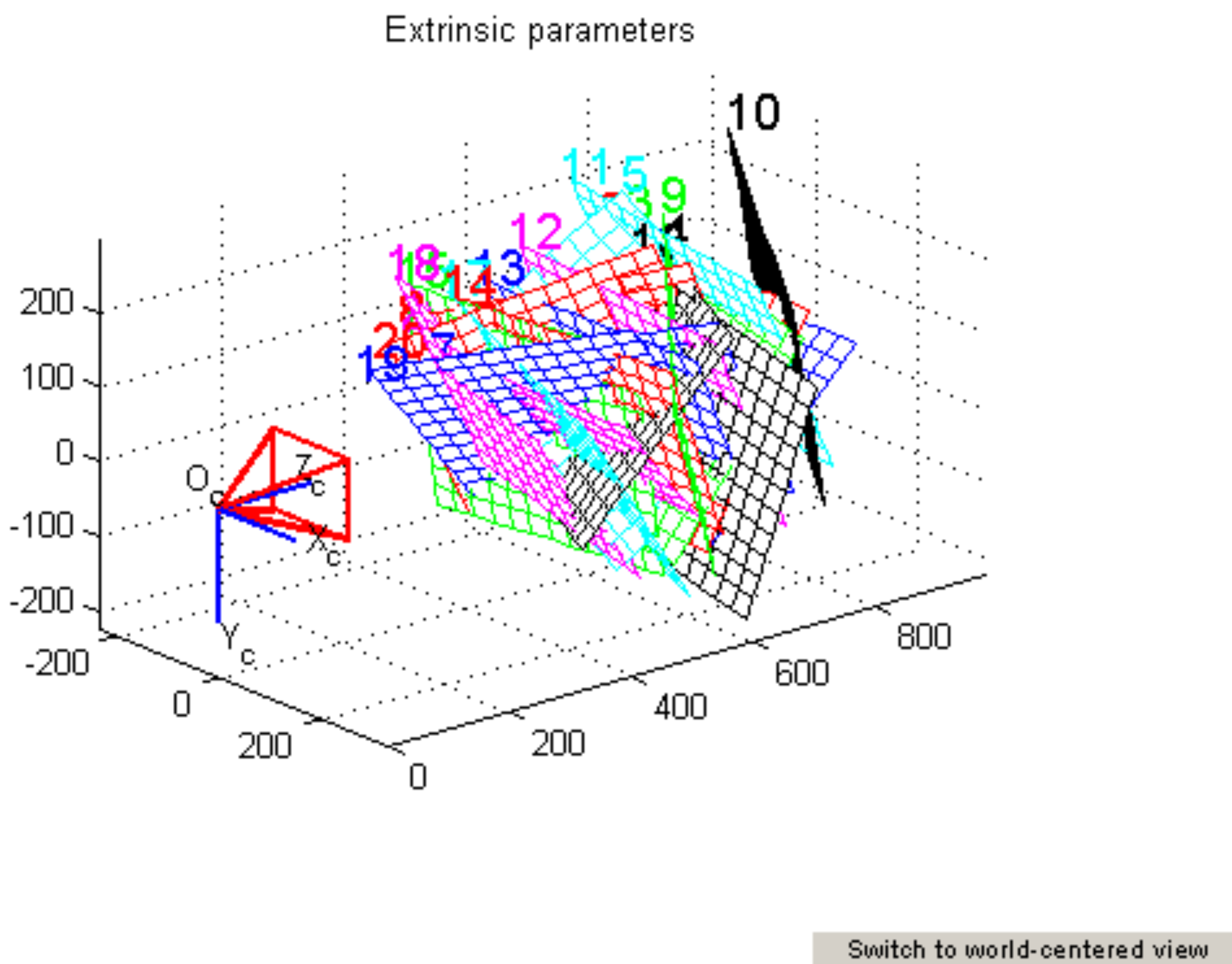
- Déterminer les paramètres de la caméra à partir d'objets 3D connus

Estimer les paramètres de la caméra



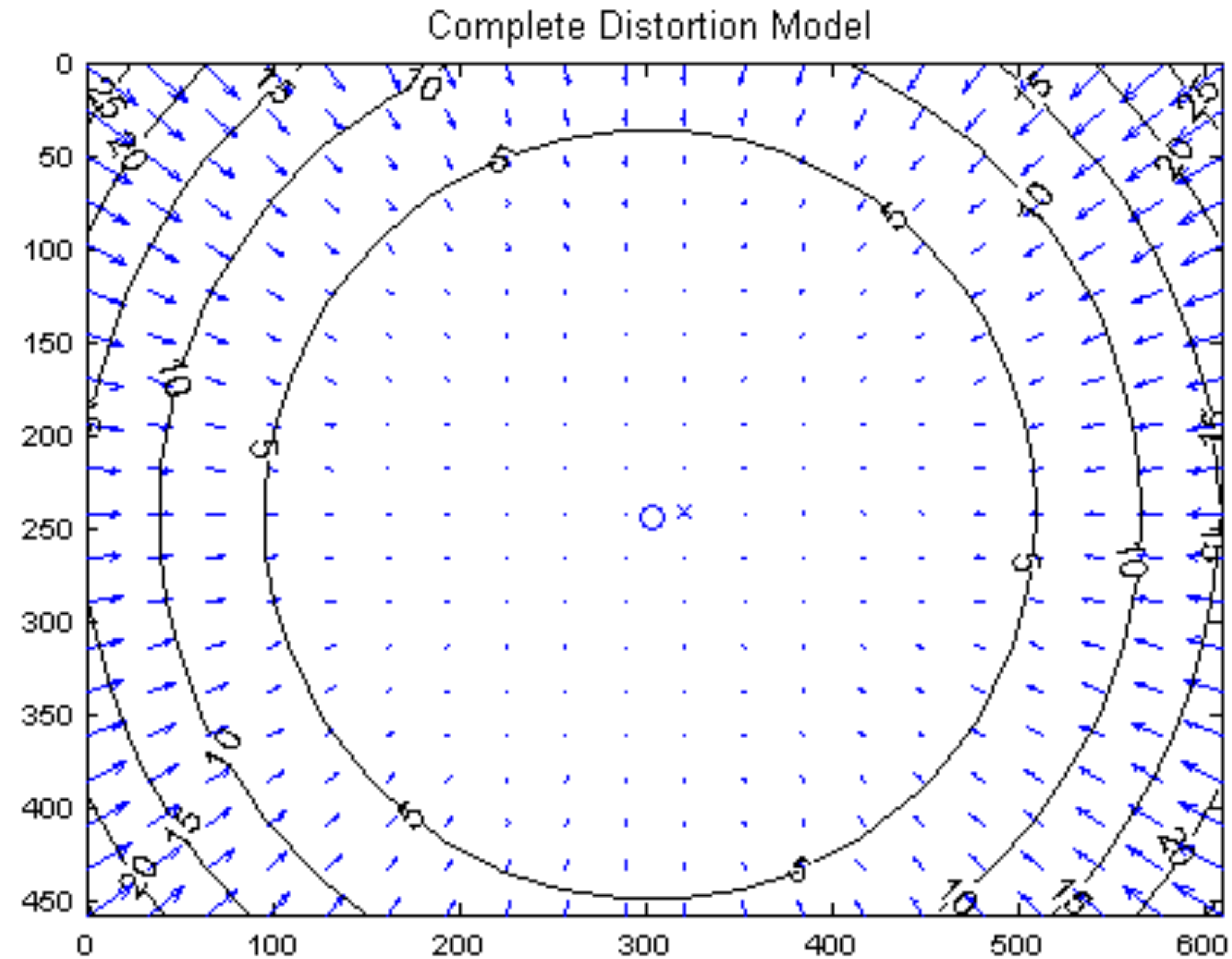
http://www.vision.caltech.edu/bouguetj/calib_doc/

Estimer les paramètres de la caméra



http://www.vision.caltech.edu/bouguetj/calib_doc/

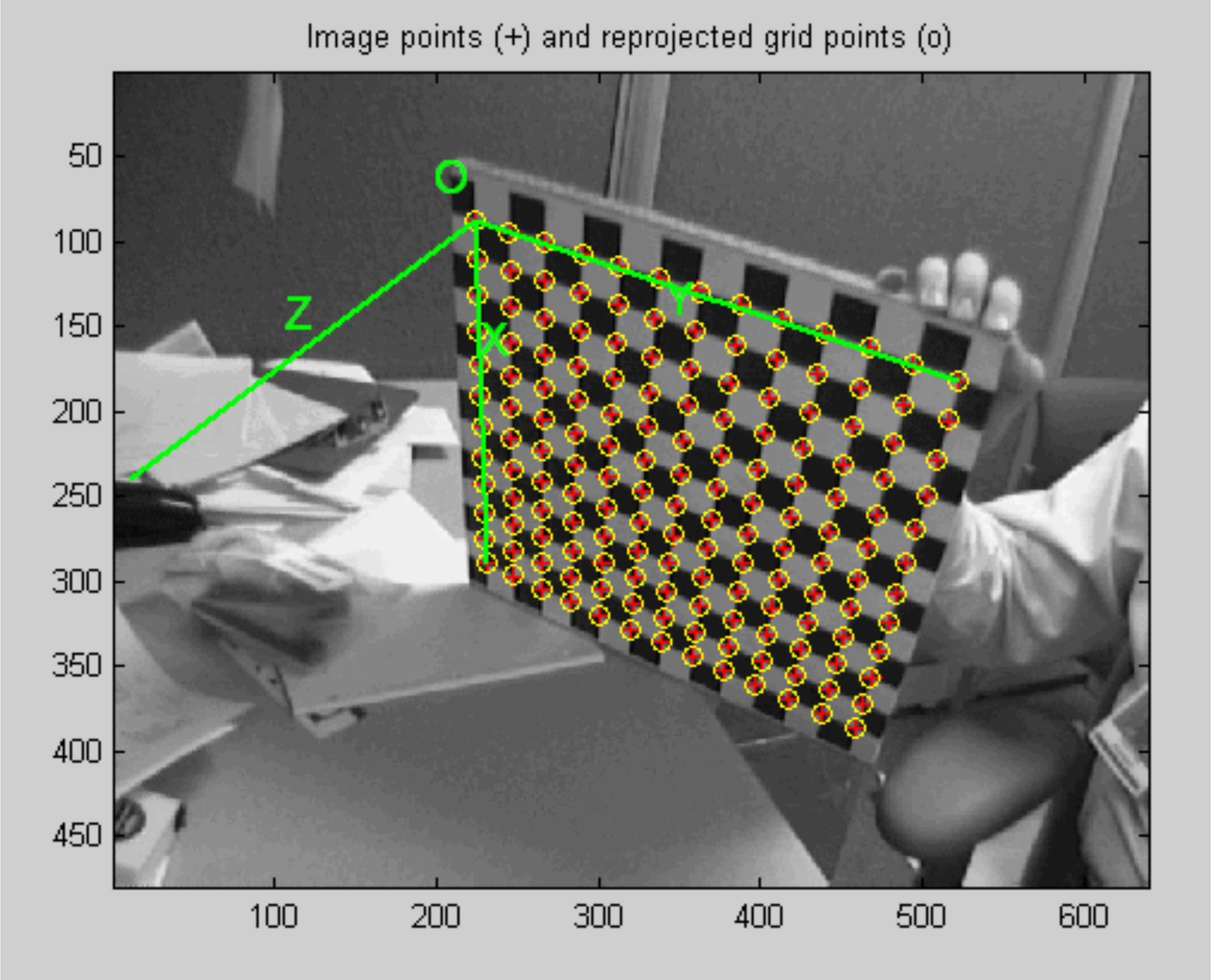
Estimer les paramètres de la caméra



Pixel error	= [0.1174, 0.1159]	
Focal Length	= (657.303, 657.744)	+/- [0.2849, 0.2894]
Principal Point	= (302.717, 242.334)	+/- [0.5912, 0.5571]
Skew	= 0.0004198	+/- 0.0001905
Radial coefficients	= (-0.2535, 0.1187, 0)	+/- [0.00231, 0.009418, 0]
Tangential coefficients	= (-0.0002789, 5.174e-005)	+/- [0.0001217, 0.0001208]

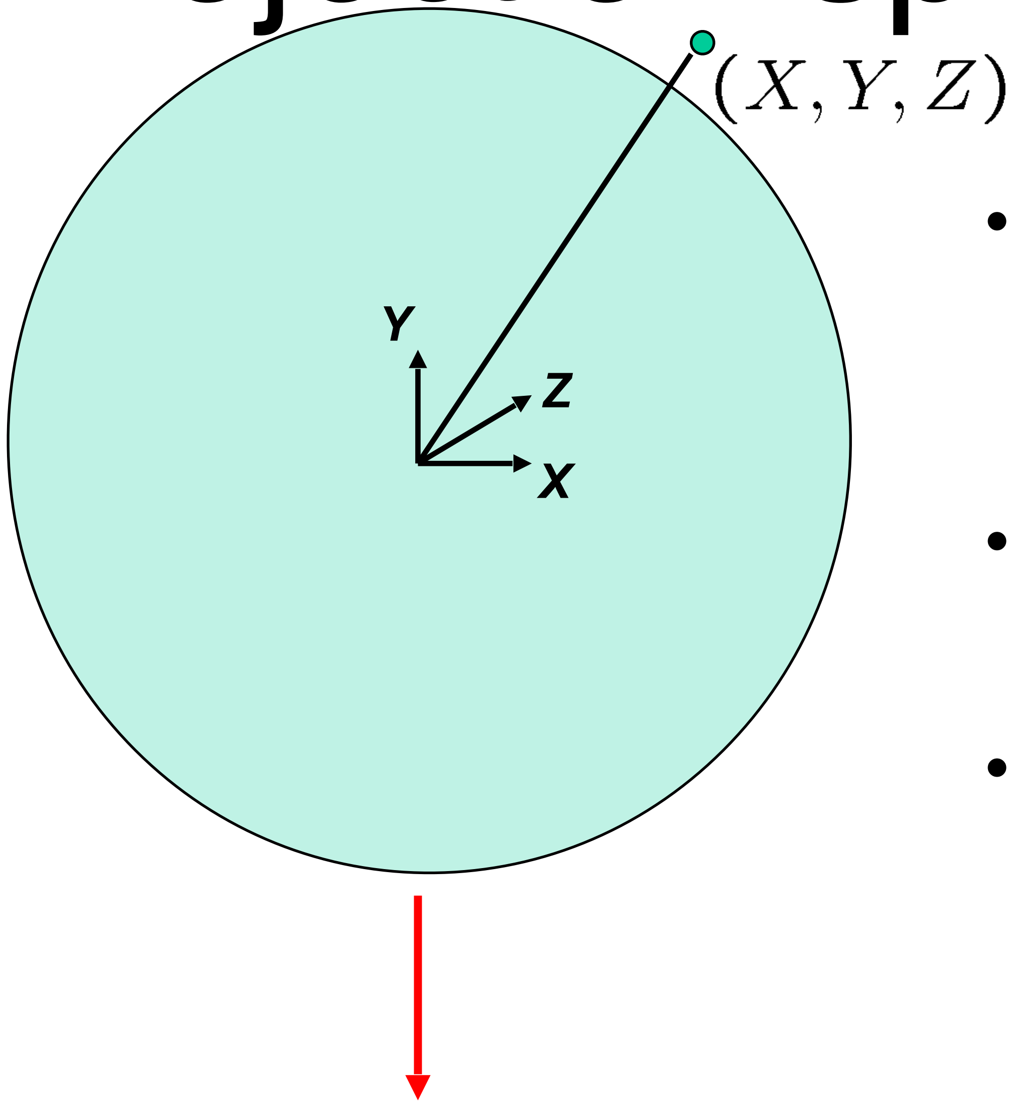
http://www.vision.caltech.edu/bouguetj/calib_doc/

Estimer les paramètres de la caméra



http://www.vision.caltech.edu/bouguetj/calib_doc/

Projection sphérique



- Projeter point 3D (X,Y,Z) sur la sphère

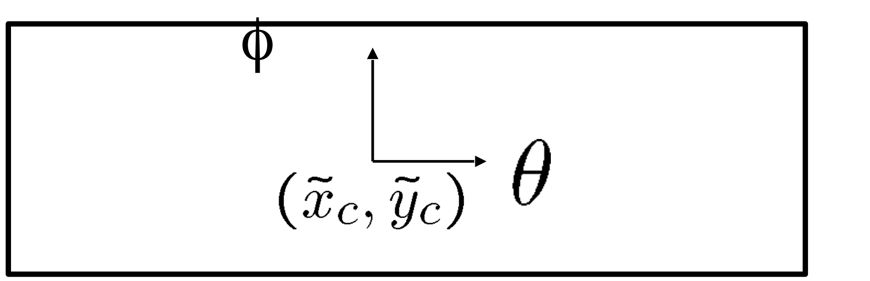
$$(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}}(X, Y, Z)$$

- Convertir en coordonnées sphériques

$$(\sin\theta \cos\phi, \sin\phi, \cos\theta \cos\phi) = (\hat{x}, \hat{y}, \hat{z})$$

- Convertir en coordonnées images

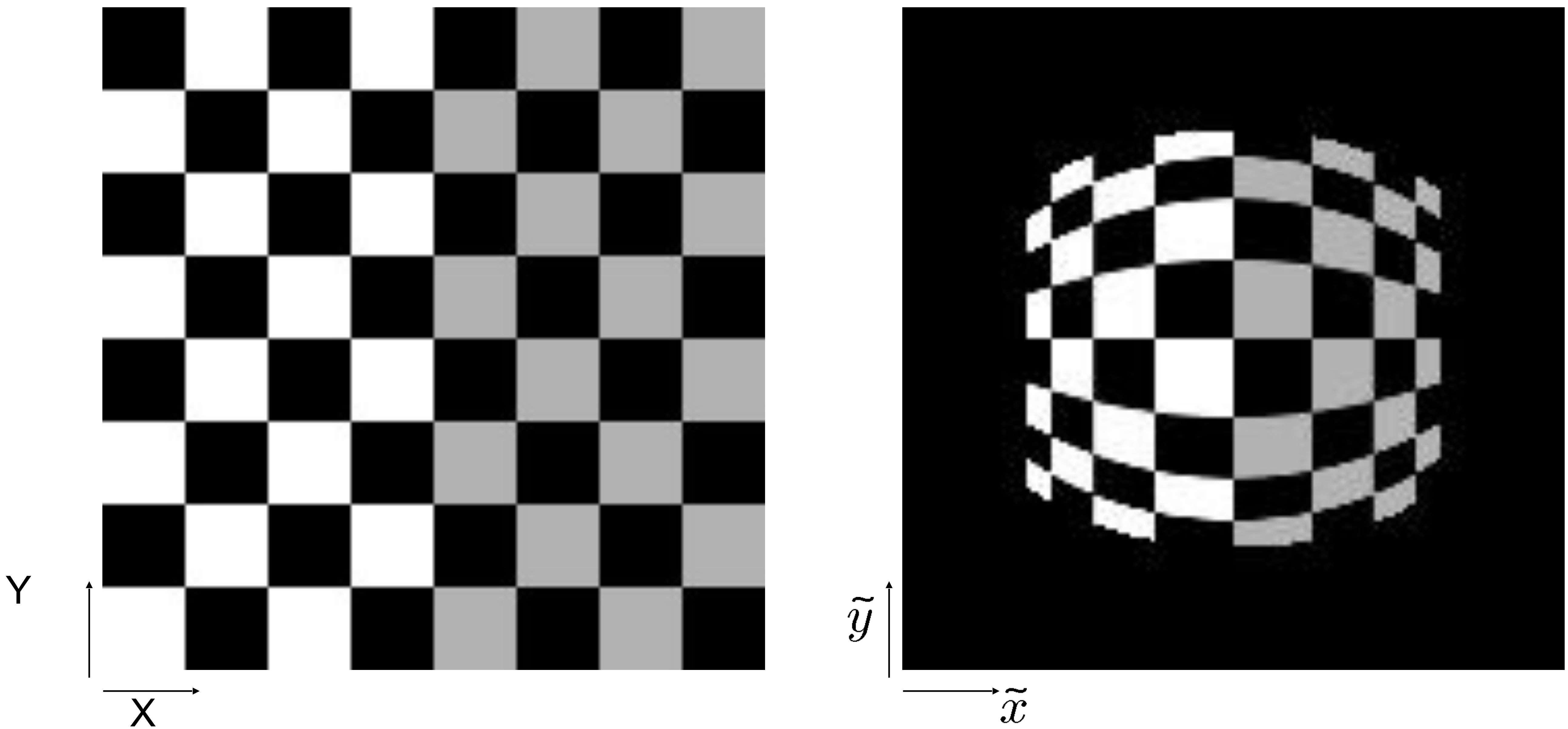
$$(\tilde{x}, \tilde{y}) = (f\theta, fh) + (\tilde{x}_c, \tilde{y}_c)$$



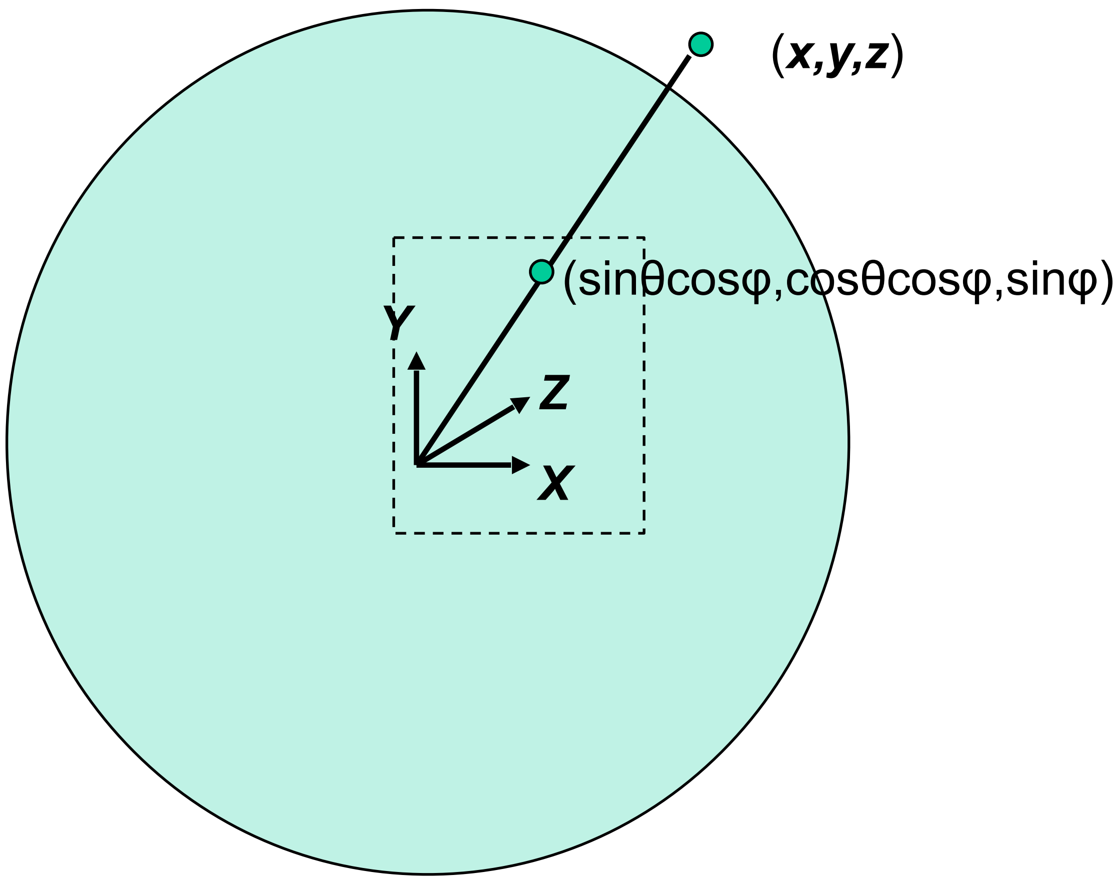
sphère déroulée

image sphérique

Projection sphérique



Projection sphérique inverse



$$\theta = (x_{sph} - x_c)/f$$
$$\varphi = (y_{sph} - y_c)/f$$
$$\hat{x} = \sin\theta \cos\varphi$$
$$\hat{y} = \sin\varphi$$
$$\hat{z} = \cos\theta\cos\varphi$$
$$x = f\hat{x}/\hat{z} + x_c$$
$$y = f\hat{y}/\hat{z} + y_c$$

Panorama complet

Autres projections

Autres projections

Demo!

- Hugin
  - http://hugin.sourceforge.net

Exemple: Reconnaître des panoramas

M. Brown et D. Lowe,
University of British Columbia

Pourquoi?

- Rotations 1D (θ)
  - Ordre des images = l'ordre des rotations

Pourquoi?

- Rotations 1D (θ)
  - Ordre des images = l’ordre des rotations

Pourquoi?

- Rotations 1D (θ)
  - Ordre des images = l'ordre des rotations

- Rotations 2D (θ)
  - Ordre des images ≠ l'ordre des rotations

Pourquoi?

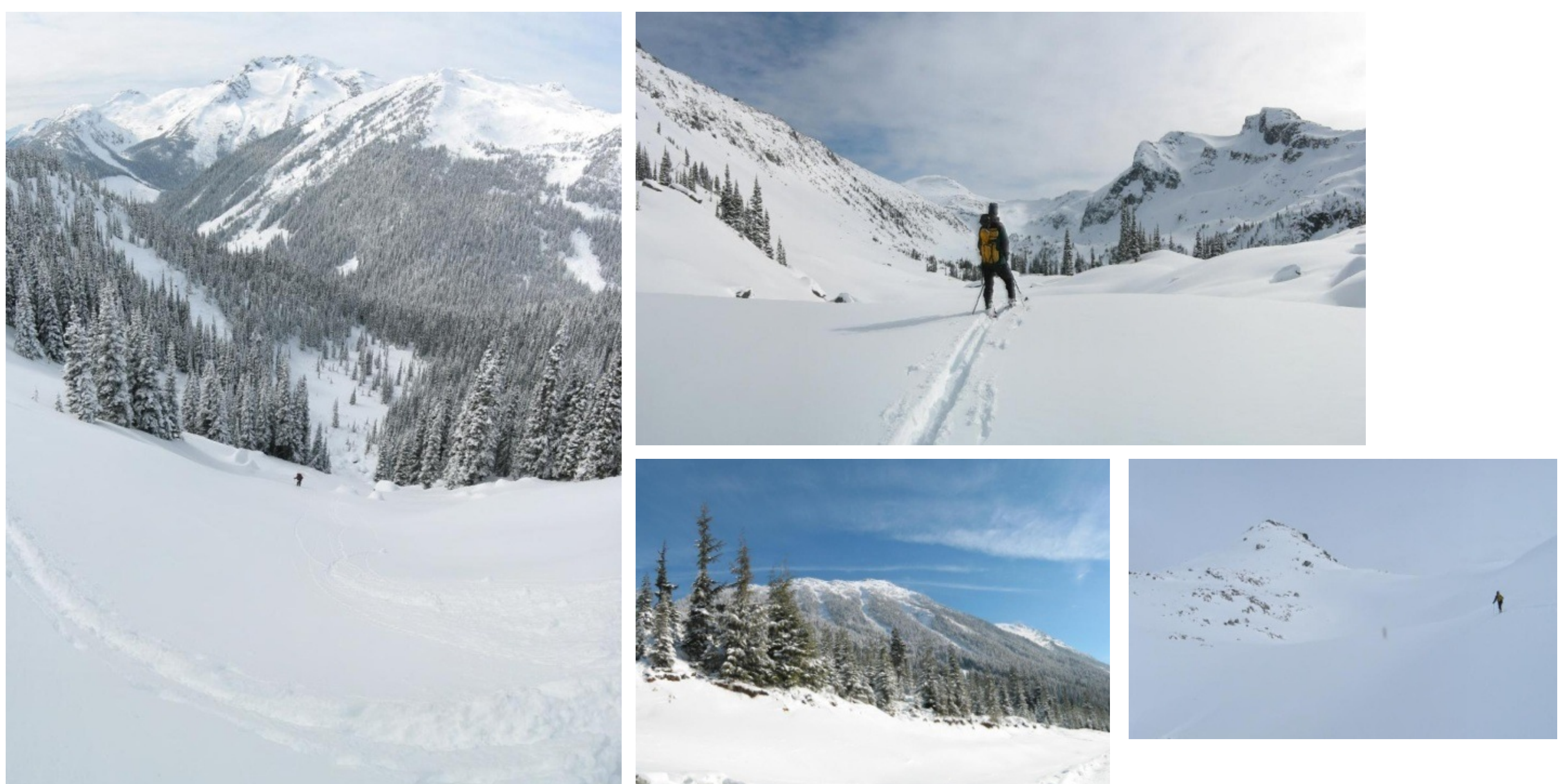
- Rotations 2D (θ)
  - Ordre des images ≠ l’ordre des rotations

Pourquoi?

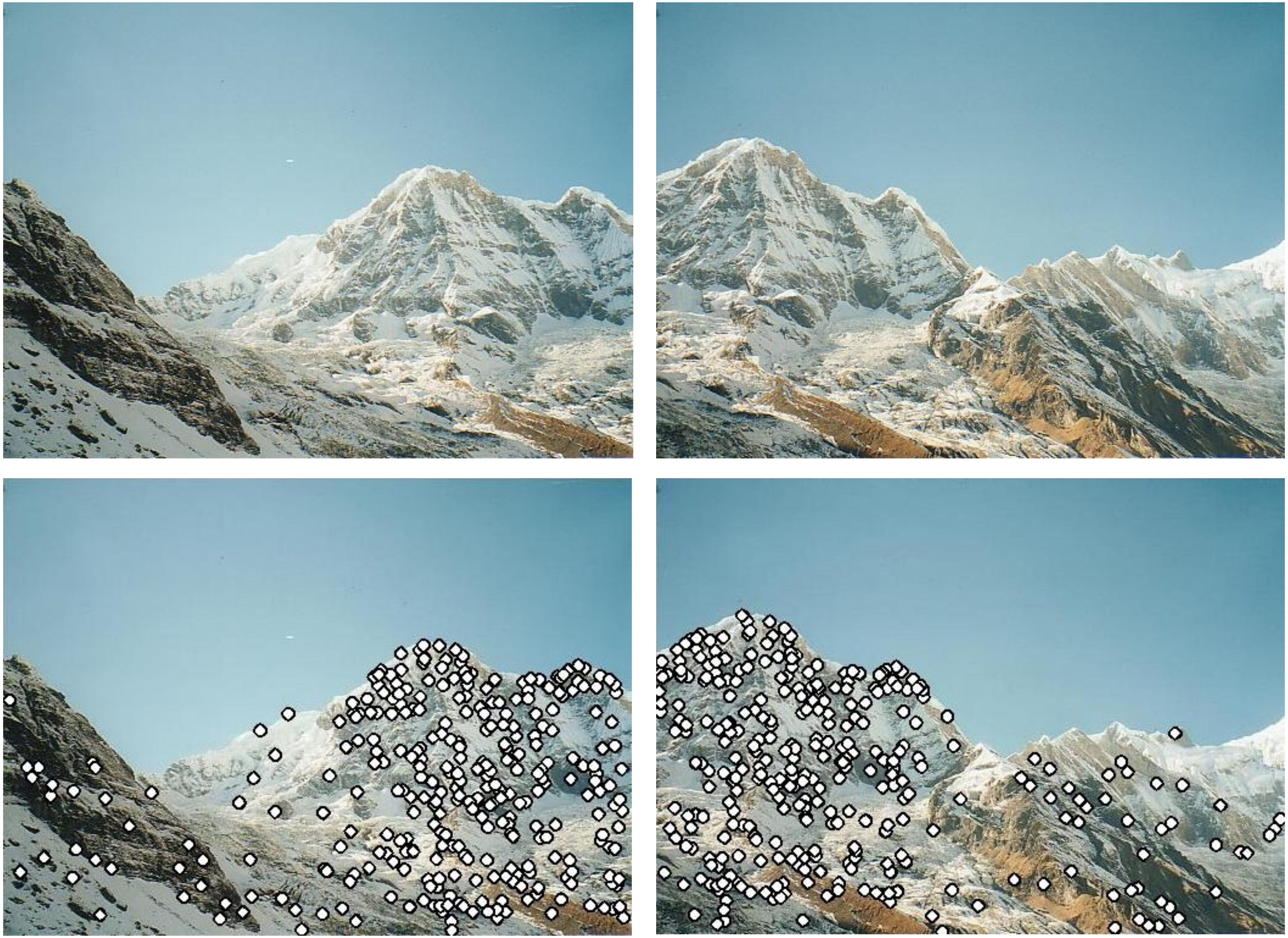
- Rotations 1D (θ)
  - Ordre des images = l'ordre des rotations

- Rotations 2D (θ)
  - Ordre des images ≠ l'ordre des rotations

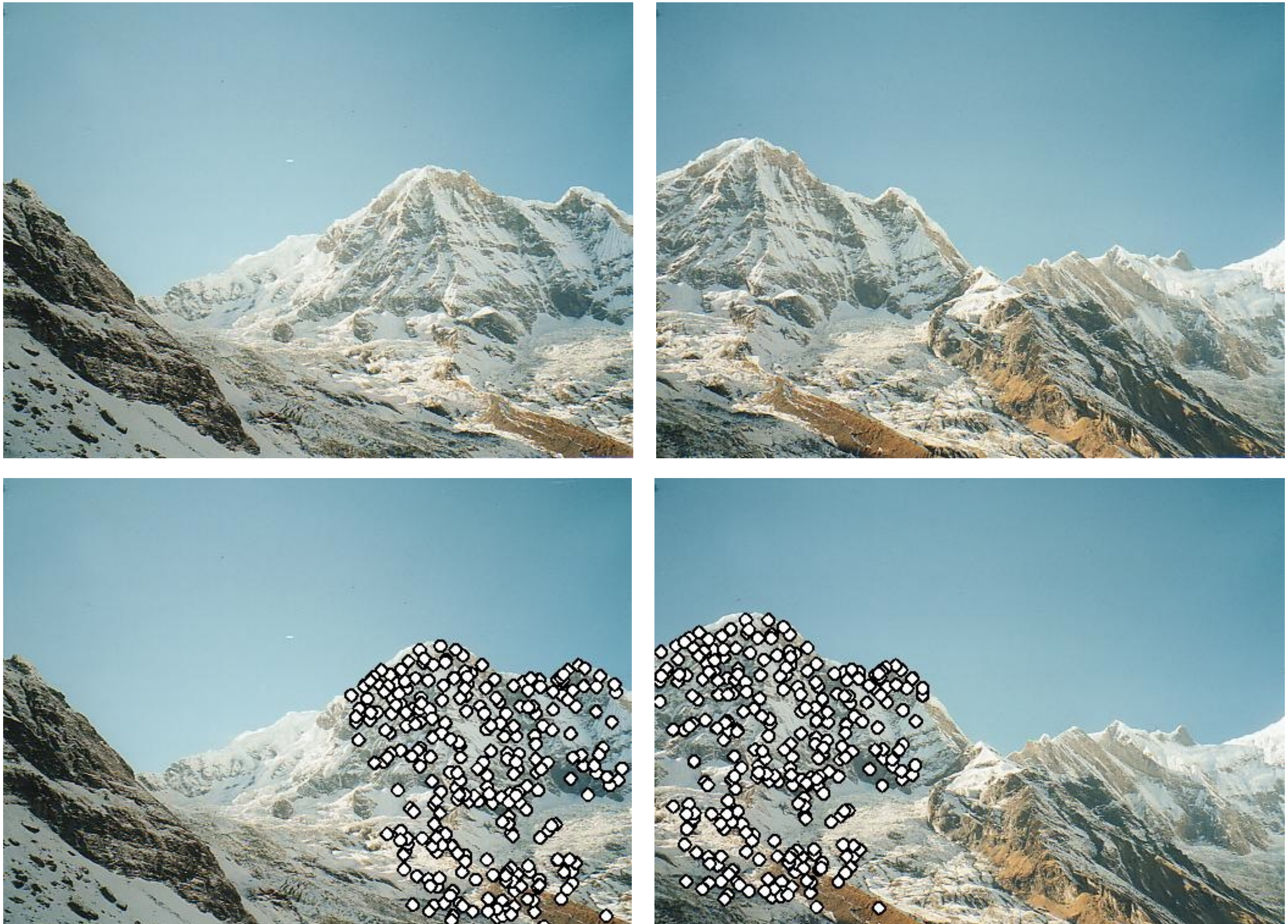
But



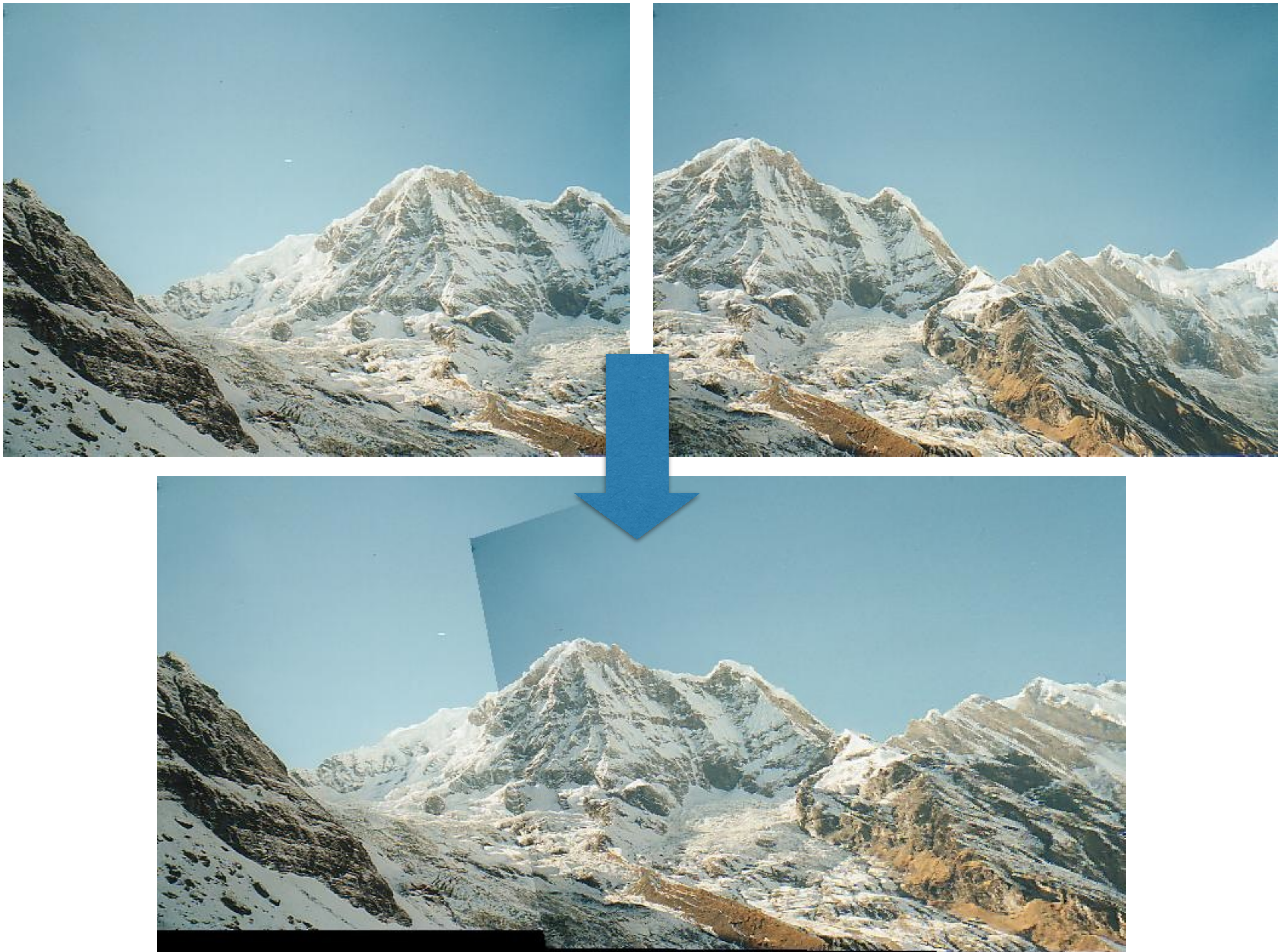
Calculer l’homographie avec RANSAC



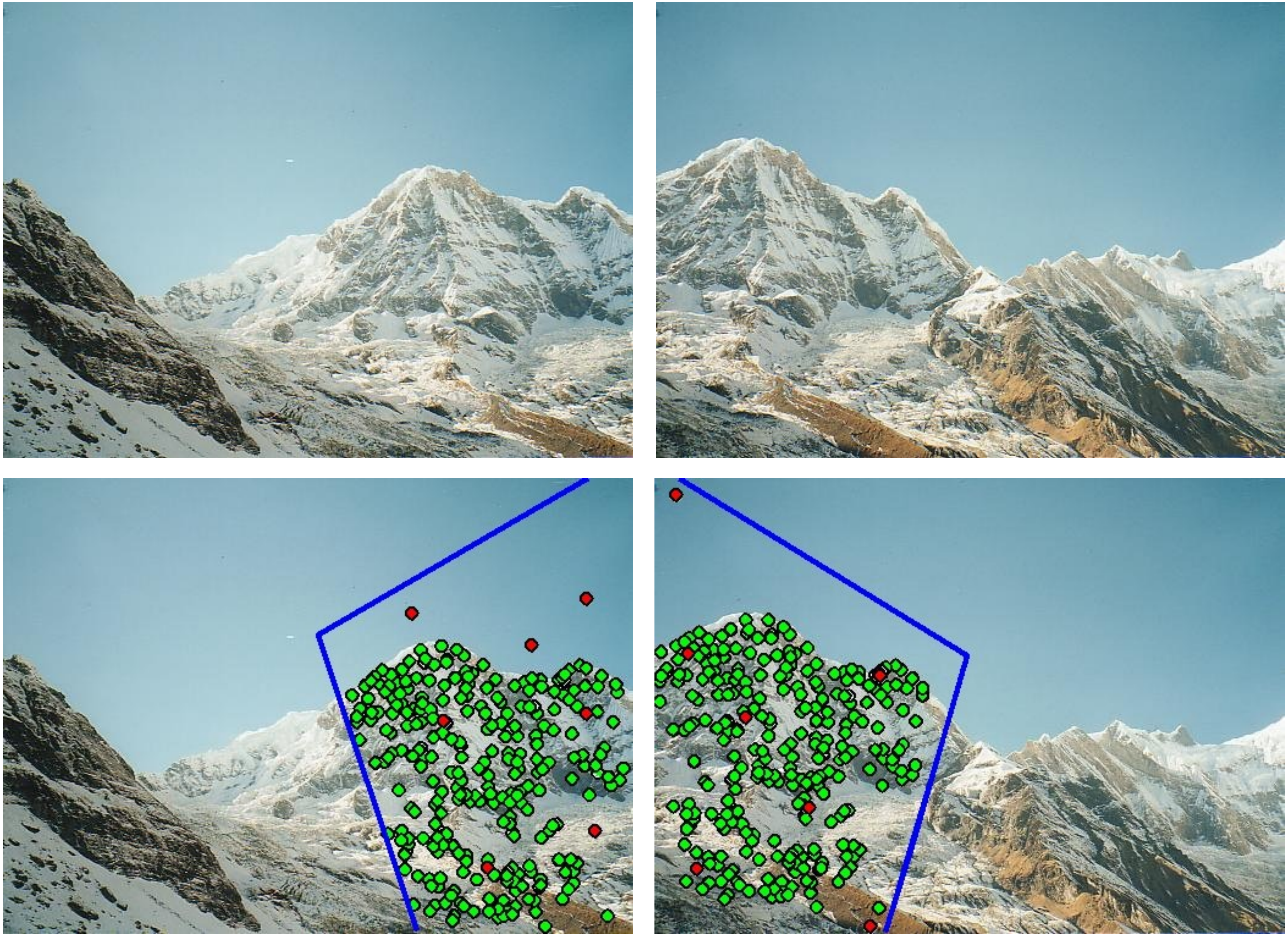
Calculer l'homographie avec RANSAC



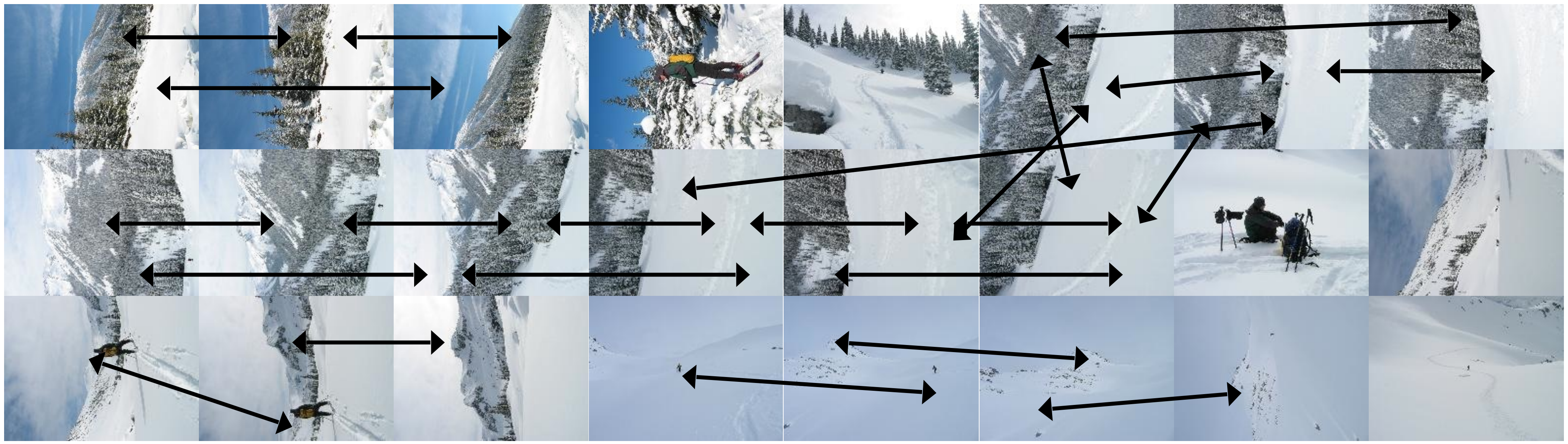
Calculer l’homographie avec RANSAC



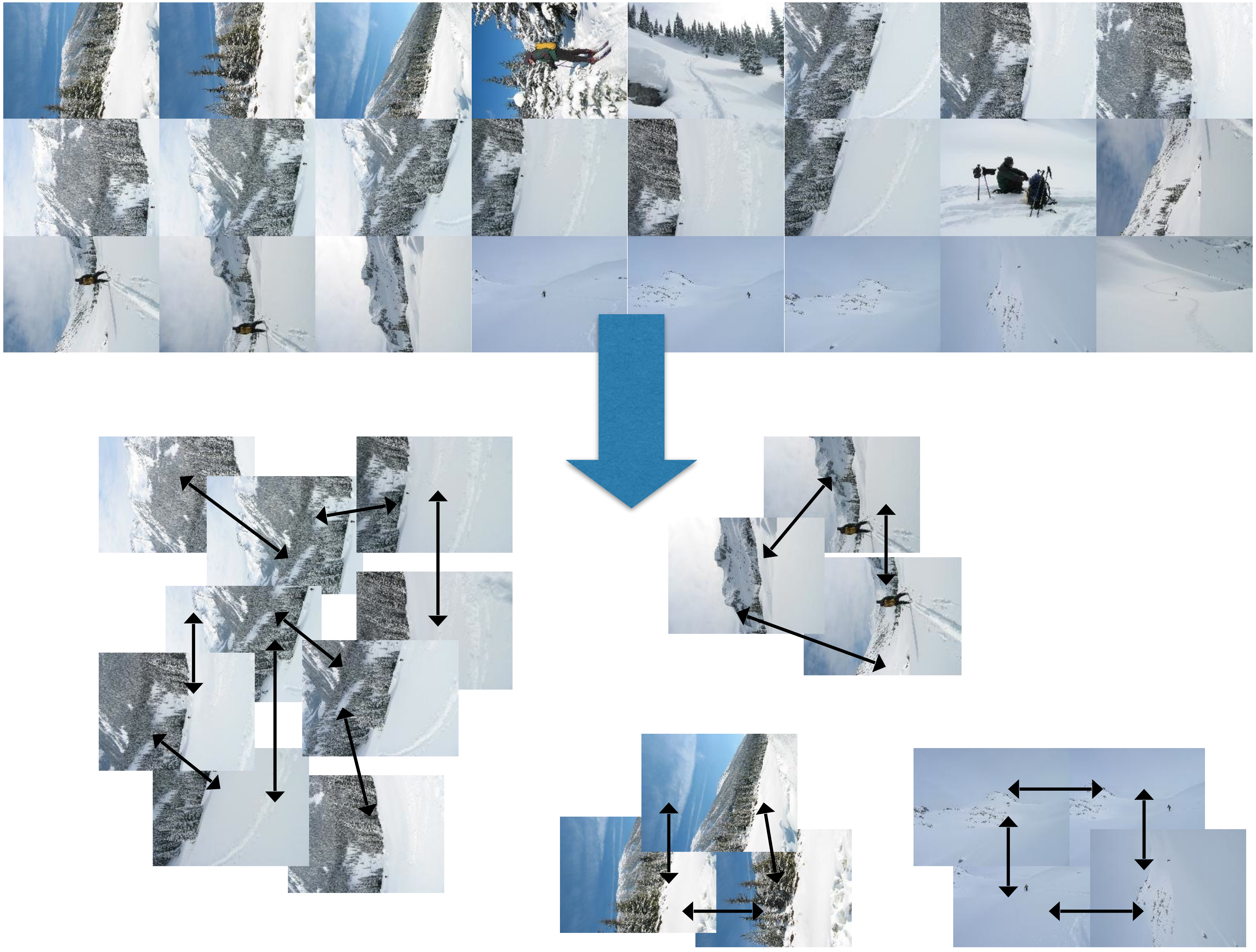
Modèle probabiliste pour vérification



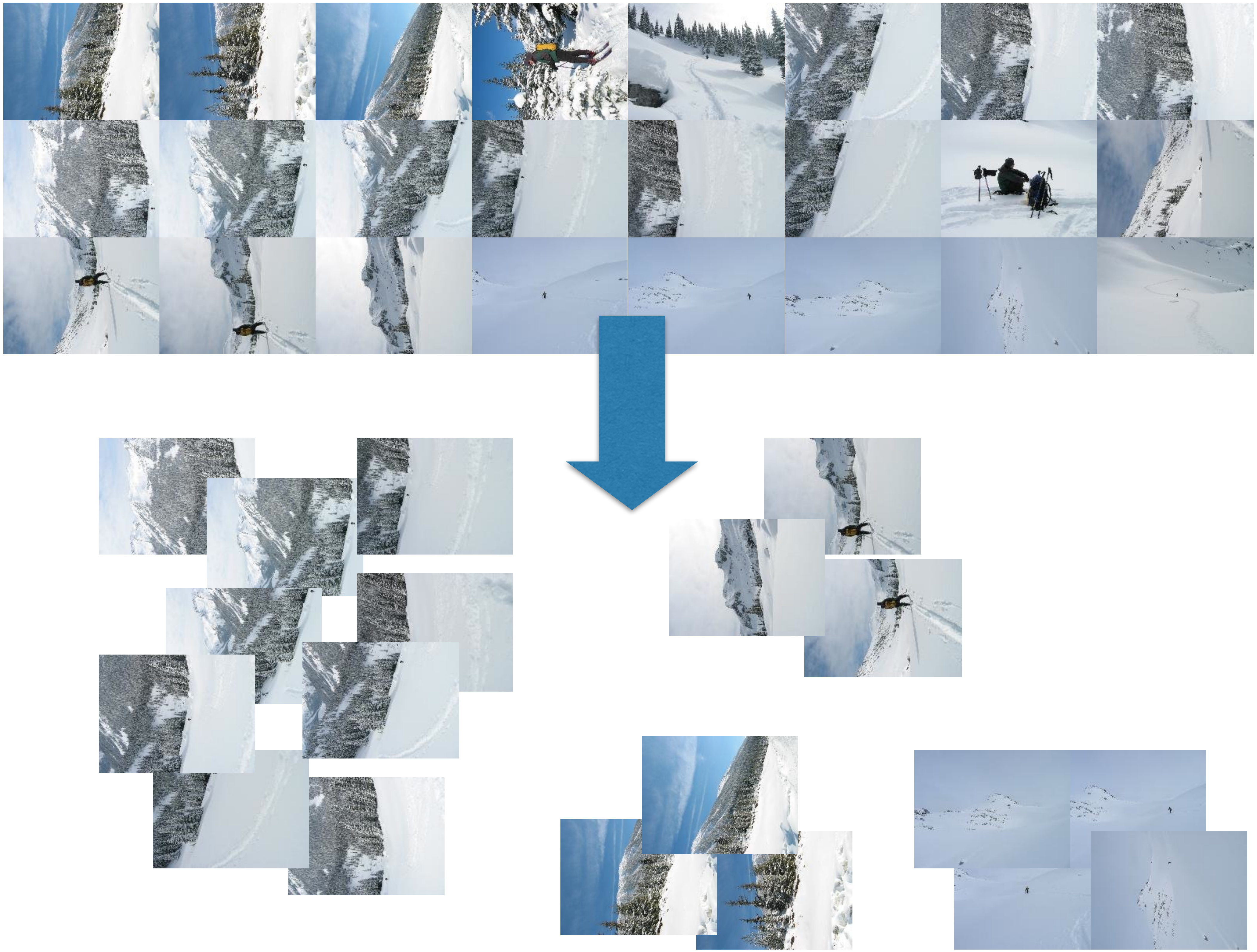
Trouver les panoramas



Trouver les panoramas



Trouver les panoramas



Trouver les panoramas

Résultats