Efficient camera motion and 3D recovery using an inertial sensor

Martin Labrie
Computer Vision and Systems Laboratory
Laval University, Québec, Canada
mlabrie@neptec.com

Patrick Hébert
Computer Vision and Systems Laboratory
Laval University, Québec, Canada
hebert@gel.ulaval.ca

Abstract

This paper presents a system for 3D reconstruction using a camera combined with an inertial sensor. The system mainly exploits the orientation obtained from the inertial sensor in order to accelerate and improve the matching process between wide baseline images. The orientation further contributes to incremental 3D reconstruction of a set of feature points from linear equation systems. The processing can be performed online while using consecutive groups of three images overlapping each other. Classic or incremental bundle adjustment is applied to improve the quality of the model. Test validation has been performed on object and camera centric sequences.

1 Introduction

3D Reconstruction from multiple images has been a widely studied field in computer vision over the last twenty years. This process is also known as shape from motion (SfM) and relies on basic photogrammetry principles for obtaining 3D information from a set of 2D images. The relative positions of a set of physical points can be calculated up to a scale factor if their projections are identified and matched in different calibrated images. Calibrated means that the positions of the cameras as well as their intrinsic parameters are known. Usually, calibrating the images is made part of the 3D reconstruction problem or at least, when intrinsic parameters are known and fixed, the positions of the cameras are computed. Although recent works have focused more on 3D reconstruction from long video sequences, either closed or open, and very large sets of images, one fundamental problem is still to identify matches between images. Several systems have been proposed [1, 2, 3, 4, 5]; usually either non-linear optimization is applied or constraints are imposed to the projection model or movement of the camera between images.

In this paper, we investigate the use of an inertial sensor attached to a camera for 3D reconstruction over a sequence of images. It is assumed that the intrinsic parameters of the camera are known and fixed. A few studies have been conducted on this combination in the fields of mobile robotics and augmented reality [6, 7, 8, 9]. Our work is mostly inspired by the more recent research of Okatani and Deguchi who demonstrated in [10] that the orientation provided by an inertial sensor could be efficiently used to calculate the translation between two camera positions. This paper goes further by exploiting this combination to estimate the set of translations between frames in a sequence of images. Following a similar strategy adopted in [11, 12], triplets of images are used as sub-sequences and are linked together in a chain (1-2-3, 2-3-4, 3-4-5, etc.) in order to produce a 3D model from the entire sequence, as new images are captured.

The contribution here is to build on [10] for developing a 3D reconstruction system that accepts as input triplets of images taken freely in space. This makes it possible to develop and to apply a linear type of sequential 3D reconstruction algorithm: the positions of the camera can be estimated by solving systems of linear equations followed by the computation of the 3D points using simple triangulation. Moreover, since an estimate of the camera relative orientation is provided, feature point matching is simplified. The 3D reconstruction process can then be computed online and the 3D model is updated incrementally as new images are taken. It is our aim to avoid the necessity of processing the entire sequence offline. A system for 3D reconstruction over a sequence of images using an inertial sensor has already been introduced in [13]. However, that system relies on tracked features over a video sequence. The downside of this is that tracking usually breaks down too often so 3D reconstruction over long sequences cannot be easily achieved. Here, we aim to match snapshot images with arbitrary motion - wide baseline.

The development of inertial measurement units (IMUs) has been thrusted up by the needs in the automotive, aerospace and military domains. In recent years, those sensors have become smaller and cheaper so it is now possible to integrate them alongside the camera [6, 9]. It is also

worth noting that new sophisticated optical gyroscopes are already available. Their high accuracy and low drift will make them excellent devices when they will be more accessible.

Section 2 describes how the matching process between image pairs and triplets benefit from a rotation estimate. The 3D model reconstruction algorithm is described in section 3. The system setup is described in Section 4. It precedes reconstruction results where both object and environment centric sequences have been modeled.

2 Feature matching

2.1 Feature detection

The first stage of the matching process is the selection of feature points in each image. It is expected that the selected feature points correspond to physical points in the scene, that are re-observable from a different viewpoint. In each image, these points must have a signature neighbourhood that characterizes them so that they can be discriminated from other selected feature points. In this paper, the set of points that are compositing the 3D model is limited to the matched feature points between images. This is referred to as a sparse model as opposed to obtaining a dense set of points.

Edge contour points with high curvature in image, namely corners, are good feature candidates. They are even better when their neighbourhood depicts discriminative information with respect to other selected corners. Based on the structure tensor, the well known Harris corner detector is used here to select a set of features in each image. In an image, the neighbourhood region of a corner can be represented by more or less sophisticated descriptors [14]. We use a simple descriptor consisting of the local image patch around the detected corner. However, the corners are localized at subpixel resolution using a variation of the original Forstner's operator [15].

In the following expression,

$$H(x,y) = Det(J) - \phi Trace(J)^2, \tag{1}$$

$Det(J)$ is the determinant of J, the gradient autocorrelation matrix, and ϕ is a constant parameter fixed here at 0.04. $H(x,y)$ is an indicator of the presence of a corner. When the entire image has been processed, local maximums of $H(x,y)$ are registered as feature points. Typically, the maximum number of feature points is limited to 500 per image.

2.2 Matching features

A good feature point in an image represents the projection of a unique physical point. If more than one image is captured of that same physical point then one must match its projections in the images. The set of all those correspondences established between images will ultimately lead to the 3D positions of the physical points as well as to the poses of the camera.

2.2.1 Between image pairs

A first set of matched points is obtained by evaluating the similarity between the image regions surrounding feature points. The normalized correlation score is calculated for each possible pair, considering the neighborhood W (typically 5×5) centered at each detected feature point. All those scores are cumulated in a correlation matrix as potential matches if:

1. The score is higher than a predetermined similarity threshold (typically 0.7).
2. The score is the highest in its associated row and column (symmetry constraint).

Every pair that meets with these two conditions composes the set of initial matches. Nevertheless, the similarity constraint is still generally insufficient since false matches are typically found in the retained candidates. Although improved descriptors may limit their numbers, there is still no guarantee of uniqueness. Imposing the epipolar constraint during the matching process further restricts the number of matches. It imposes the correspondent of a given feature point in one image to stay along a line in the second image. To obtain the point-line relationship between two images, one will estimate the fundamental matrix F.

Several linear and non-linear optimization methods have been introduced in order to estimate F from two sets of matched points between images [2]. These methods require a minimum number of 7 matches and are often sensitive to noise. This may also lead to a very large number of combinations to explore even after applying a similarity threshold. Interestingly, if one takes advantage of the orientation provided by an inertial sensor along with the intrinsic camera parameters, the minimum number of matches then drops to 2. Actually, the only unknown is now the translation vector $T_{3\times1}$ that is computed up to a scale factor between two camera positions. Then, since each matched feature point provides one equation, 2 points suffice [10]. F can then be estimated through the essential matrix E which expands to $([T]_X R)$. $[T]_X$ denotes the (3×3) skew symmetric matrix encoding the translation,

$$[T]_X = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}, \tag{2}$$

so that $[T]_X m = t \times m$ for any 3×1 vectors t and m. The matrix R is the rotation matrix. The fundamental matrix can

be developed as follows

$$F = A^{-T}([T]_X R)A^{-1}, \tag{3}$$

where A is the 3×3 matrix of the intrinsics. It is worth noting that this model equation subsumes that nonlinear distortion was removed from the image beforehand. The small number of matches is more suitable for a RANSAC type algorithm [16] that is used to estimate F between every image pair. Furthermore since its rank is intrinsically 2, the estimate is more stable and robust (in wide sense) even in the case of a planar scene. Actually, R is of full rank and $[T]_X$ is of rank 2. All points that are coherent with this estimate are then processed to obtain the final least square estimate of F.

We have now obtained the rotation and translation between both cameras. The translation is however only known up to a scale factor and we will have to impose a single factor for the whole sequence of images.

2.2.2 Between image triplets

Even after imposing the epipolar constraint, matching between two images is still relatively difficult since the correspondent of a point can be found anywhere along the epipolar line in the second image. That is why we are using a third image to impose an additional constraint to the matching process and thus reduce the number of false matches. To do so, the fundamental matrix is estimated for each pair (1-2, 2-3 and 1-3) before obtaining an estimate of the trifocal tensor [17]. A match between 2 images is validated by computing its similarity factor with the corresponding point in the third image, which is at the intersection of both epipolar lines estimated from the two matched points. The triple matches are further validated using the compound scale of the translation T_{13} between the first and the third image. This will be discussed in more detail in section 3.1.

3 3D model construction

The general principle of a structure from motion algorithm is applied. By knowing the position of each snapshot, one can triangulate every single matched feature point and estimate its position in 3D space. All we are missing is a coherent scale factor for the translation between images. Actually, this estimation was performed in a different coordinate system for each image pair so we must transpose them into a common system. For that purpose, one approach consists in processing the sequence by triplets of images that overlap in the following way [11]: 1-2-3, 2-3-4, 3-4-5, etc. Every single triplet is thus related to the previous one by a common rotation and translation. So by processing every triplet one after the other, one can relate every camera position to the coordinate system of the first triplet in the sequence. The downside of this approach is that a certain number of physical points have to be seen in 3 consecutive images. This fact has to be taken into account when collecting images.

3.1 A single triplet of images

3.1.1 Orientation

Since the IMU provides an orientation with respect to an intrinsic reference, the orientation of each snapshot can be obtained by compositing the instantaneous rotation matrix with the inverse of the rotation matrix at the first position in the sequence. That leads to the rotation matrix,

$$R_{1i} = R_i R_1^T, \tag{4}$$

where R_i is the orientation obtained from the IMU, that is associated with image i for $i = 1, 2, 3$ of the triplet and R_{1i} is the relative orientation between images i and 1.

3.1.2 Translation

Since the translation vector obtained between each pair of images is known up to a scale factor, we must relate the third image to the first image so that all three images share the same coordinate system. In order to do so, we use the feature points identified and matched in the 3 images. Then, we estimate the translation T_{13} that minimizes, in the third image, the projection error of 3D points X triangulated from the first two images. The equation for the projection is

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} (X' - T_{13}), \tag{5}$$

where

$$X' = R_{13}X. \tag{6}$$

In Equation 5, α, β and (u_0, v_0) are respectively the intrinsic scale factors and the principal point. These values are the intrinsic parameters of matrix A. Equation 5 can then be used to obtain 2 linear equations in the form $MT_{13} = b$,

$$x = \frac{u}{w} \Leftrightarrow \begin{bmatrix} \alpha \\ 0 \\ u_0 \\ -x \end{bmatrix}^T T_{13} = \alpha X'_x - x X'_z + u_0 X'_z \tag{7}$$

$$y = \frac{v}{w} \Leftrightarrow \begin{bmatrix} 0 \\ \beta \\ v_0 \\ -y \end{bmatrix}^T T_{13} = \beta X'_y - y X'_z + v_0 X'_z \tag{8}$$

where (x, y) is the corresponding feature point in the third image. Since the translation vector T_{13} holds 3 degrees of freedom, a minimum of two matches are necessary. Again, a RANSAC [16] type algorithm is used to find a good estimate.

3.2 A sequence of overlapping image triplets

The orientation of any image snapshot in the sequence is estimated using Equation 4. For computing the translation, we need to transform the current triplet of images into the coordinate system of the first triplet in the sequence. For example, in a sequence of 4 images, we will link the second triplet (images 2-3-4) to the first triplet (images 1-2-3) without imposing a point to be viewed in the four images. In order to do so, a similar procedure to the one described in section 3.1 is applied. However, for triangulating the 3D points X, we use the computed translations T_{12} and T_{13} from the previous triplet (in this example triplet 1) instead of using the relative position of the first two images in the new triplet (in this example images 2 and 3). By doing so, the triangulation of the 3D points is computed in the same coordinate system as that of the previous triplet, which is related to the first image of the sequence. The translation associated with image 4 is then obtained in the same coordinate system, that is T_{14}. Applying this procedure incrementally, one can link every single new image to the first image's coordinate system.

3.3 3D model reconstruction

The structure of the 3D model representation is image based. This approach means that the list of observed features are linked through consecutive image triplets. A dynamic array is used for that purpose; it evolves as new images come in. The array is first initialized with matches found in the first triplet. It is then updated for any new input image by comparing new matches with those related to the last image in the array. A physical point can thus be seen in more than 3 consecutive images. However, the model structure does not allow any physical point that would disappear and reappear later on in the sequence, to be identified per se without reprocessing the list. When that situation occurs, we simply create a new 3D point identity that will be located nearly at the same position. The position and orientation of each new image are estimated as soon as they come in, so the model is updated incrementally by simply triangulating the matches found in the dynamic array. It is also possible to optimize the 3D model using a bundle adjustment after each new image or at the end of the entire sequence.

4 Experimental setup

For performing the experiments, a webcam (logitech QuickCam) was coupled with a 3DM-G inertial sensor from Microstrain (see Figure 1). This sensor is built on micro electro mechanical sensor (MEMS) technology to produce angle sensors with both static and dynamic response. The 3DM-G sensor estimates the orientation at rest by combining the gravity vector read from three accelerometers with the magnetic north measured by three embedded magnetometers. The third vector is obtained by applying the cross-product of the two previous vectors. This provides an absolute and precise orientation within the earth referential over a long period of time. These data are combined with the readings of three gyroscopes by going through a set of internal filters. Rapid motions can then be detected from the gyroscope readings while the orientation is corrected over a long period of time by the accelerometers and the magnetometers. Table 4 shows the standard deviation of the orientation over a period of 20 seconds with the sensor being at rest. A dynamic test was also conducted for verifying the precision of the orientation in motion. To do so, the unit was positioned in space, moved to a random motion and finally returned to the same starting position. The result can be seen in Figure 2 where the pitch, yaw and roll angles are plotted along time.

Figure 1. Experimental setup: camera and IMU (below the platform).

Angle	Std Dev.(°)
Pitch	0.0541
Yaw	0.0261
Roll	0.1085

Table 1. Standard deviation of the orientation of the sensor at rest (test performed over 20 seconds).

The camera intrinsic parameters have also to be estimated using a standard camera calibration approach [18].

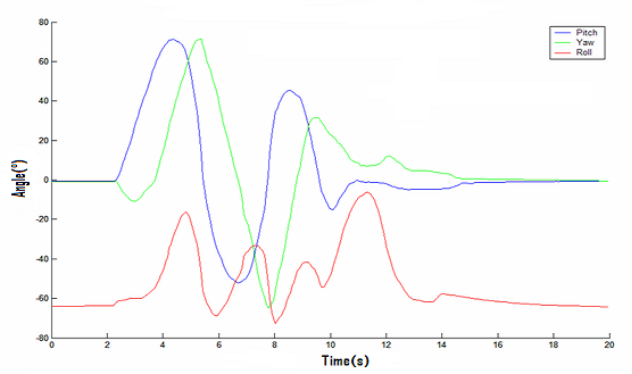

Figure 2. Dynamic test of the IMU. Roll, pitch and yaw evolution curves.

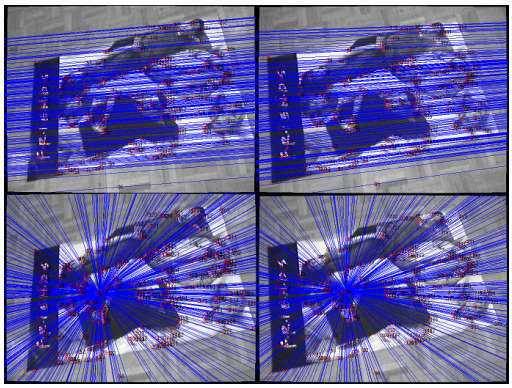

Figure 3. Top: epipolar lines obtained when exploiting the IMU. Bottom: epipolar lines obtained without using the IMU.

The projection model includes two terms for radial distortion. All image feature points are corrected (undistorted) before estimating fundamental matrices. The orientation transformation between the IMU and the camera has also been estimated in order to apply the rotation with respect to a common axis for both the IMU and the camera. For that purpose, we use the estimated orientations of the camera during the camera calibration process. Actually, the calibration software provides the pose of the camera (the extrinsics) with respect to the calibration target at each position. The orientation provided by the IMU, R_{sensi}, is also recorded at each of those i^th positions. The equation linking the orientations of the camera to those of the IMU, is

$$R_{camij} = R_{sc}R_{sensj}R_{sensi}^T R_{sc}^T = R_{sc}R_{sensij}R_{sc}^T \quad (9)$$

where R_{camij} is the rotation matrix linking two poses i and j of the camera and the rotation R_{sc} links the orientations of the IMU and the camera. In the equation, the notation R_{sensij} is used to describe the relative rotation between two poses of the IMU. The estimation of R_{sc} requires a minimum of 2 poses of the camera and at least one 3D point of the calibration target. From Equation 9, a new equation can be built for each 3D point as follows

$$R_{camij}X = (R_{sc}R_{sensi}R_{sensj}^T R_{sc}^T)X + e. \quad (10)$$

Using non-linear optimization, R_{sc} can then be estimated by minimizing the squared error (e^2)

$$e^2 = ((R_{sc}R_{sens1}R_{sens1}^T R_{sc}^T)X - R_{cam12}X)^2. \quad (11)$$

5 Results

5.1 Stabilizing the fundamental matrix estimate

First we demonstrate that the system efficiently stabilizes the estimate of the fundamental matrix even in the case of a planar scene. Actually, in this case the mapping is described by a homography between both images. Following the development in section 2.2, the addition of the inertial sensor automatically constrains the estimated matrix to be of rank 2. It is thus more stable and robust to noise. This avoids discrimating between the two cases using numerical analysis. The result shown in Figure 3 depicts a typical case for a planar scene. While both top images depict the epipolar lines estimated with the orientation from the inertial sensor, the two images at the bottom show the case where only image information is considered. In this latter case, we can see that the epipolar lines are much less coherent with the camera displacement which was a left to right motion. In recent years, more robust methods have been developed for estimating fundamental matrices. The rank of the matrices can be reinforced or assessed numerically. However, using the IMU to help recover F makes the process more efficient. This is particularly interesting when modeling scenes with planar sections such as floor and walls.

5.2 Some examples of 3D reconstruction

In this section, models of object and environment centric are reconstructed. In each case, the entire sequence has been incrementally processed frame by frame. For each sequence, the output model includes the computed camera positions as well as the set of observed 3D points.

5.2.1 Object centric modeling

The modelled object is a textured cylinder box. The drawing on the box makes it possible to easily extract feature points all around the object. The entire sequence is composed of 37 images. Four of them are shown in Figure 4.

Figure 4. Images 1, 16, 28 and 37 of the cylinder box sequence.

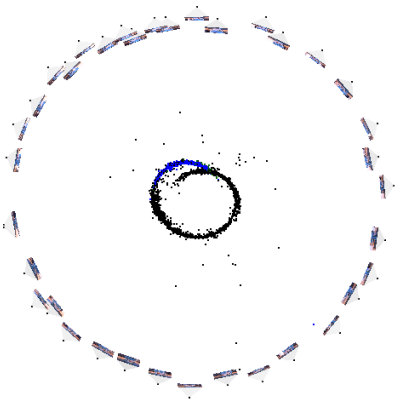

Figure 5. 3D Reconstruction (2937 points) of the cylinder without non-linear optimization (squared pixel error: 3.76).

A top view of the model obtained at the end of the acquisition is seen in Figure 5. It took less than one second per image to build the model on a 2.4 GHz Pentium IV. It can be seen that the cylinder form is perceptible, but that it does not close well. This highlights the effect of the accumulated error on the camera position over the sequence. Moreover, if a position in the sequence has been estimated more or less precisely due to a more important error on the orientation provided by the sensor or due to a low amount of matches, then all following positions will be affected. This is a characteristic of incremental modeling approaches. Another cause of error is the inability of the system to manage physical points previously tracked and then lost during the sequence. To improve the model, it can be optimized by using a bundle adjustment [19]. The projection error is then reduced to 2.03 squared pixel and, as seen in Figure 6, the cylinder box model greatly improves. It is also possible to improve the model by applying the bundle adjustment after each new image. Various strategies can be adopted to improve computation efficiency such as limiting the number of images. We discuss the strategy adopted in section 5.3. The result shown in Figure 10 shows the final model after applying incremental bundle adjustment.

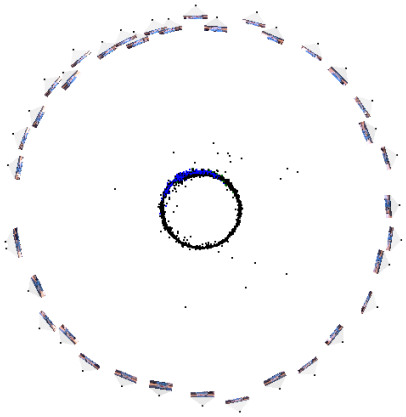

Figure 6. 3D Reconstruction (2937 points) of the cylinder with non-linear optimization (squared pixel error: 2.03).

5.2.2 Environment centric modeling

Figure 7. Images 1, 15, 25 and 38 of the room sequence.

The environment to be reconstructed here is a bedroom where many feature points can be extracted for the purpose of modeling. As opposed to the case of an object, the camera positions are enclosed within the scene and aim towards the exterior. The entire sequence is composed of 38 images. Four images of the sequence are shown in Figure 7. The final model obtained after direct incremental modeling can be seen in Figure 8. The required processing time was comparable to the previous case. It was a little bit higher than one second per frame since for this model, the number of points is larger. The behaviour of the system is similar to the previous case. After applying a bundle adjustment to

the model, the projection error reduces from 3.13 to 1.72 squared pixels. The result can be seen in Figure 9.

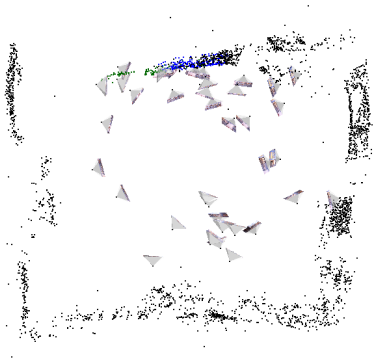

Figure 8. 3D Reconstruction (3261 points) of the room without non-linear optimization (squared pixel error: 3.13).

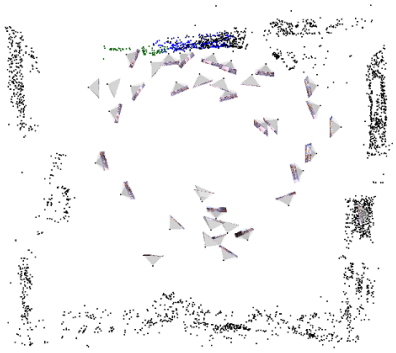

Figure 9. 3D Reconstruction (3261 points) of the room with non-linear optimization (squared pixel error: 1.72).

5.3 Incremental bundle adjustment

An incremental bundle adjustment has also been implemented to help reduce the accumulated error through the sequence. To do so, a bundle adjustment is first applied to the first three images. Then, for every following image, only the pose is optimized. This process was tested with both the cylinder and the room sequences. The results are shown in Figure 10 and Figure 11 respectively. Interestingly, the error is comparable to the error after applying the global bundle adjustment. It converges to 2.17 and 1.72 squared pixels respectively. The total required processing times increased slightly for both sequences (46.03 and 39.03 seconds).

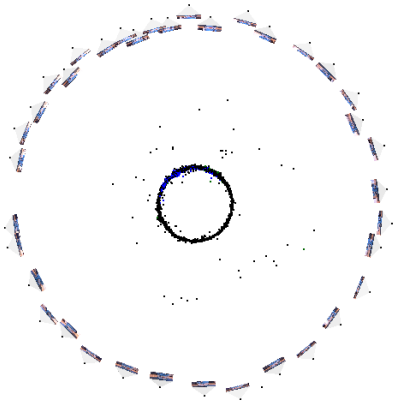

Figure 10. 3D Reconstruction (2856 points) of the cylinder with the incremental bundle adjustement (squared pixel error: 2.17).

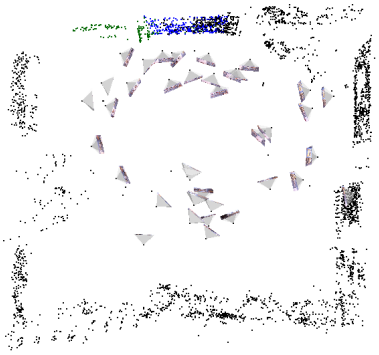

Figure 11. 3D Reconstruction (3621 points) of the room with the incremental bundle adjustement (squared pixel error: 1.72).

6 Conclusion

We have shown how an inertial sensor can improve the efficiency of matching and contribute to camera motion estimation and 3D recovery of feature points. This is accomplished by reducing the minimal number of points to match as well as by making the system to be solved, linear and stable even in the presence of planar sections. In a sequence with wide baseline snapshot images, matching and camera motion were performed incrementally by triplets. However, it is clear that this type of sensor must be complemented by vision - photogrammetry - measurements to improve the model quality over the whole sequence. Actually, a standard deviation of one tenth of a degree leads to significant accumulation of error that was compensated for by using bundle

adjustment techniques, in their classic or incremental versions. Remarkably, inertial sensors have not been widely used in computer vision systems yet. As new technologies become accessible, new 3D modeling methods should be developed.

Acknowledgement

The authors express their gratitude to B. Ricard, researcher at RDDC-DRDC Valcartier, for his valuable input in the course of the project.

References

[1] A.J. Davison, *Real-Time Simultaneous Localisation and Mapping with a Single Camera* iccv, p. 1403, Ninth IEEE International Conference on Computer Vision (ICCV'03) - Volume 2, 2003.

[2] C. Lyness, O.C. Marte, B. Wong, P. Marais, *Low-cost model reconstruction from image sequences* in proceedings of the 1st international conference on Computer graphics, virtual reality and visualisation. 2001. Camps Bay, Cape Town, South Africa: ACM Press.

[3] D. Nister, *Automatic Passive Recovery of 3D from Images and Video* 3dpvt, pp. 438-445, Second International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'04), 2004.

[4] M. Pollefeys and L.V. Gool, *From images to 3D models* Communications of the ACM, 45(7): p. 50-55, 2002.

[5] C. Tomasi and T. Kanade, *Shape and motion from image streams under orthography: A factorization approach* International Journal of Computer Vision, 9(2): p. 137-154, 1992.

[6] L. Chai, W.A. Hoff, and T. Vincent. *3-D Motion and Structure Estimation Using Inertial Sensors and Computer Vision for Augmented Reality* 2002, MIT Press. pp. 474-492.

[7] M. Kanbara, H. Fujii, H. Takemura and N. Yokoya, *A Stereo Vision-Based Augmented Reality System with an Inertial Sensor* in Proceedings of ISAR'2000, pp. 97-100, Munich, Germany, 2000.

[8] J. Lobo and J. Dias, *Vision and Inertial Sensor Cooperation Using Gravity as a Vertical Reference* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 12, pp. 1597-1608, Dec., 2003.

[9] S. You and U. Neumann. *Fusion of Vision and Gyro Tracking for Robust Augmented Reality Registration* in Proceedings of IEEE Conference on Virtual Reality (VR'2001), pp. 71-78, 2001.

[10] T. Okatani and K. Deguchi. *Robust estimation of camera translation between two images using a camera with a 3D orientation sensor* in Proceedings of the 16th International Conference on Pattern Recognition, pp. 275-278, Québec, 2002.

[11] A.W. Fitzgibbon. and A. Zisserman, *Automatic camera recovery for closed or open images sequences* in Proceedings of 5th European Conference on Computer Vision. 1998. Freiburg, Germany: Springer.

[12] Z. Zhang and Y. Shan, *Incremental Motion Estimation through Modified Bundle Adjustment* In Proceedings International Conference on Image Processing (ICIP), Vol.II, pp. 343-346, Barcelona, Spain, 2003.

[13] T. Mukai and N. Ohnishi. *The Recovery of Object Shape and Camera Motion Using a Sensing System with a Video Camera and a Gyro Sensor* in Proceedings of the 7th International Conference on Computer Vision (ICCV'99). 1999. Kerkyra, Greece: IEEE Computer Society Press.

[14] D. G. Lowe, *Distinctive image features from scale-invariant keypoints* International Journal of Computer Vision, Vol. 60, pp 91-110, 2004.

[15] W. Forstner and E. Gulch, *A fast operator for detection and precise location of distinct points, corners and centers of circular features* In Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, pp. 281-305, 1987.

[16] M.A. Fischler and R.C. Bolles, *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography* 24(6): pp. 381-395, 1981.

[17] P. Torr and A. Zisserman, *Robust Parametrization and Computation of the Trifocal Tensor* Image and Vision Computing, 15: pp. 591-605, 1997.

[18] J.-Y. Bouguet, *Camera Calibration Toolbox for Matlab* http://www.vision.caltech.edu/bouguetj/.

[19] B. Triggs and P. McLauchlan, R. Hartley and A. Fitzgibbon, *Bundle Adjustment – A Modern Synthesis, in Vision Algorithms: Theory and Practice* B. Triggs, A. Zisserman, and R. Szeliski, Springer-Verlag. pp. 298-372, 2000.