Orientation and Pose recovery from Spherical Panoramas

Florian Kangni   Robert Laganière
VIVA Research Lab
School of Information Technology and Engineering
University of Ottawa
fkangni,laganier@site.uottawa.ca

Abstract

This paper addresses the problem of camera pose recovery from spherical images. The $3D$ *information is extracted from a set of panoramas sparsely distributed over a scene of interest. We present an algorithm to recover the position of omni-directional cameras in a scene using pair-wise essential matrices. First, all rotations with respect to the world frame are found using an incremental bundle adjustment procedure, thus achieving what we called cube alignment. The structure of the scene is then computed using a full bundle adjustment. During this step, the previously computed panorama orientations, used to feed the global optimization process, are further refined. Results are shown for indoor and outdoor panorama sets.*

1. Introduction

The problem of camera pose recovery consists essentially in using visual information extracted from a sequence of images to estimate the motion parameters of each of the cameras involved in the capture process. A solution to this problem can be exploited in many ways. One could for example localize objects of the scene from given features or insert a virtual object into the environment at a given position. One could also take advantage of the motion parameters, in a virtual navigation application, to adequately position the images with respect to the environment, to ensure consistent and fluid navigation and to ease the process of image interpolation.

The research project presented in [14] is the framework in which this work is inscribed. Its ultimate goal is to design a virtual navigation system in a real remote environment from which panoramic views are rendered to the user via an image-based procedure. This project uses spherical image sequences to represent the environment. This paper therefore presents a two-stage algorithm to retrieve the camera motion from a set of spherical panoramas.

The first stage is the estimation of the relative rotations with respect to the world frame. An incremental bundle adjustment approach in conjunction with pair-wise essential matrices is used to recover each rotation in a global solution. Once each rotation is known, one can then obtain a configuration in which all camera frames are aligned with respect to each other.

The pose recovery is therefore undertaken using the result of cube alignment as part of the initial structure. The rotations or orientations recovered before hand are refined here as well as features of the scene and panorama location through a bundle adjustment. This procedure benefits from the accuracy of the initial estimate of the structure orientation-wise which results in higher chances of convergence.

It must be noted that the approach described here can also be applied to limited field-of-view (*FOV*) cameras. However, undertaking the rotation estimation prior to the pose recovery is particularly adapted to spherical panoramas as, in this case, it is always possible to undo the rotations and obtain fully aligned panoramas. With limited *FOV* cameras, such strategy would most of the time require breaking the set into smaller groups sharing common field-of-view. In addition, the solution to the panorama alignment problem is by itself of interest in applications where one wishes to work with aligned reference frames. In other cases, where the orientations are already known (e.g through the use of an inertial device), the pose recovery can be applied directly. Conversely, if the position of all cameras is reliably known (using GPS for example), orientation recovery can be performed in order to align all panoramas with respect to each others.

1.1. Related Work

Zhang [21] provides a starting point for any structure from motion procedure based on the essential matrix. His method identifies some constraints used during reconstruction and puts the accent on the refinement of the essential parameters extracted from the essential matrix. Svoboda et al. discuss a method also based on the essential matrix to extract motion information from two spherical cameras [19]. The focus is however on how efficient motion extraction from spherical cameras allows the distinction between a pure translation and a pure rotation. In [17] and [18], global solutions to the pose recovery problem are given respectively for one generalized camera and for a pair of generalized cameras. [18] states that the solution is minimal and requires 3 known rays for the case of one camera and 6 rays for a pair.

Carceroni et al. present a method that relies on some properties of $SO(3)$ [3]. It is a feature-based procedure that uses a GPS to recover camera orientation from known positions of multiple cameras. Constraints on the essential matrices between the images are established resulting in overall rotations estimation. Their work is a special case of the problem discussed in this article in which only the rotation components remain to be estimated. The work presented here applies to spherical panoramas in general unknown positions.

Makadia et al. present in [16] a method to recover the rotation from two spherical images. This method also uses properties of the Special Orthogonal Group $SO(3)$ in addition to "the persistence of image content". It has the advantage of not being based on feature points. One of the images is rotated during the search until it matches the other in a harmonic coefficients space. The matching criterion is correlation-based. Nonetheless, the rotation estimation being the only point of interest of this method no translation information is extracted.

Finally, in [15] is presented a method to compute camera pose from a sequence of spherical images. Many similarities with the method described in the present article can be found from the sensor used for capture to the use of essential matrix for initial pairwise geometry. However, the main difference resides in the use of a rough camera path estimate as additional input of the system to compute camera positions via a map-correlation technique.

1.2. System

The sensor used during our experiment is the PointGrey LadyBug camera. This camera uses 6 sensors, 5 radial and 1 pointing upwards, to capture a 360^o degrees view of a scene. The captured images can then be assembled together and from there re-projected onto any volume. The choice was made to use cubic panoramas, as presented in [6], that have the advantages of "fast rendering with standard graphics hardware and ease of compression and decompression"[1]. Indeed, a cubic panorama or *cube*, is essentially a set of 6 projective cameras, one per face, centered on the same point, that can be considered as a whole or independently. An example of a cubic panorama is displayed in Fig. 1 where the cube is laid out in a cross pattern with its faces in the order from top to bottom and left to right : top, left, front, right, back, down.

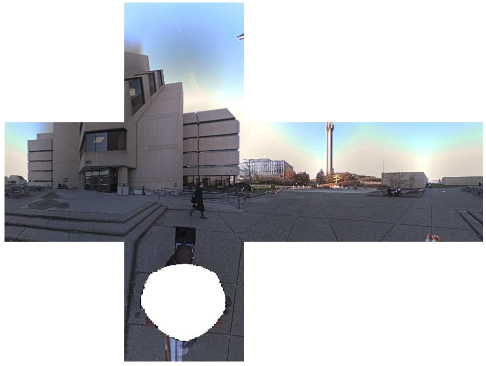

Figure 1. Example of a cubic panorama of the outdoor set laid out in a cross pattern.

In the following sections, some elements on the essential matrix will be presented followed by the presentation of the proposed structure from motion solution in two parts. The first part discusses rotations estimation or cube alignment and the second, the structure computation itself. Some results are presented on two scenes, an outdoor and an indoor ones.

2. Essential Matrix E : the case of cubic panoramas.

The essential matrix introduced by Longuet-Higgins as mentioned in [21] is a powerful tool when studying the relationship between two camera frames. As a matter of fact, [11, 4] among others note that the essential matrix encodes the translation - up to a scale - and the rotation that exist between the frames of interest.

2.1. Computation of E and extraction of the translation t and rotation R

In a classical stereo configuration, matches pairs $(p, \bar{p})$ in normalized camera coordinates are to be provided and they verify the epipolar constraint:

$$\bar{p}^T E p = 0 \quad (1)$$

Using the normalized direct linear transform (*DLT*) algorithm mentioned in [9, 10, 20, 11, 5], one can recover the essential matrix E.

In the case of cubic panoramas, the principle is the same as for limited *FOV* images. The main difference resides in the direct use of normalized coordinates. As a matter of fact, since they are the product of a virtual camera, cubic panoramas are implicitly calibrated; that is user specified cube size

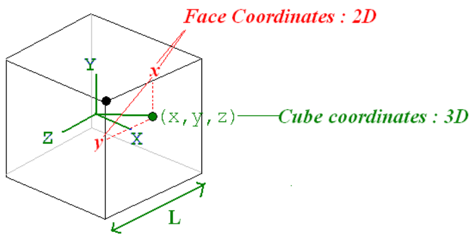

Figure 2. Correspondence $2D$ surface coordinates - $3D$ cube coordinates

and cube resolution when these ones were generated from the re-projection of the Ladybug composite images. The normalized coordinates are therefore directly given by the $3D$ coordinates of a pixel on the cube surface (see Fig.2). In addition, the use of a *RANSAC* scheme when estimating E will ensure to eliminate, at this step, most of the outliers.

Given a pair of cubes C and $\bar{C}$, [11] establishes an expression of the associated essential matrix E as a function of the motion parameters t and R, between C and $\bar{C}$, defined previously:

$$E = [t]_{\times} R \quad (2)$$

Where $[t]_{\times}$ is the anti-symmetric matrix associated with the vector t. Once the essential matrix E computed from matches over the cubes, one can extract t and R using a pair of correspondences $(p, \bar{p})$[5, 4, 11] through the *SVD* decomposition of E (as $E = USV^T$).

3. Computing Rotations : cube alignment

Given a set of N panoramas, captured in a scene, their cubic representations can all be aligned by applying appropriate rotations. That is, we are seeking a configuration in which all cubes are separated by a pure translation. Figure 3 illustrates the problem of cube alignment. In [7], the problem was partly solved by assuming that the translational component could be neglected; here, a general solution is proposed (note that the number N of cubes is assumed to be greater than 2 to eliminate the trivial case solved in section 2.1).

The cube alignment procedure is based on a bundle adjustment approach that is similar to the one described in [2]. Essentially, cubic panoramas are added one after the other to the bundle, forcing a global adjustment of the rotations corresponding to the cubes currently in the bundle. This type of procedure is discussed in detail in [11, 5] and provides a global solution to large estimation problems.

3.1. Notations

The set of all cubes that are of interest in the iterative bundle adjustment algorithm is noted $\mathcal{C}$. The set of all cubes being processed in the current iteration is noted $\mathcal{B}$. For a given cube C, the set of cubes that share some matches with the latter is noted $\mathcal{M}(C)$. Moreover, $\mathcal{F}(C, \bar{C})$ stands for the set of common features for a pair of cubes $(C, \bar{C})$. A rotation R associated to a cube $\bar{C}$ will be noted $R(\bar{C})$, the translation t between cubes C and $\bar{C}$ is noted $t(C, \bar{C})$.

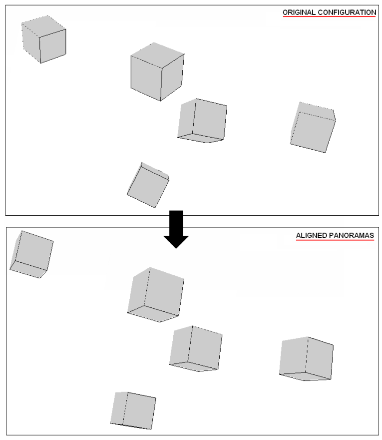

Figure 3. Cube Alignment

3.2. Bundle Adjustment of the Orientations

First, we need to define the objective function to be evaluated at each iteration of the bundle. For a pair of cubes C and $\bar{C}$, considering a feature f appearing in both cubes respectively as $p(C, f)$ and $p(\bar{C}, f)$, the epipolar constraint states the following : the rays sustaining both previously mentioned matches and the translation unit vector $t(C, \bar{C})$ are all coplanar as seen in Figure 4. We therefore have the residual error of feature f associated to pair $(C, \bar{C})$ given by:

$$r(C, \bar{C}, f) = (R(C)p(C, f) \times R(\bar{C})p(\bar{C}, f)) \bullet R(C)t(C, \bar{C}) \quad (3)$$

Overall, the objective function is thus defined by :

$$e(\Theta) = \sum_{C \in \mathcal{C}} \sum_{\bar{C} \in \mathcal{M}(C, \bar{C})} \sum_{f \in \mathcal{F}(C, \bar{C})} r(C, \bar{C}, f)^2 \quad (4)$$

3.3. Algorithm

The bundle adjustment procedure can then take place. For all available pairs of cubes $(C, \bar{C})$ that share matches, it is necessary to first compute the associated essential matrix $E(C, \bar{C})$ then extract the translation unit vector $t(C, \bar{C})$ and the rotation $R(C, \bar{C})$ between cube C and $\bar{C}$ using the method mentioned in section 2.1.

While $\mathcal{B}$ does not contain all cubes of $\mathcal{C}$ {

1. *Add a cube C in $\mathcal{C}$ that is not in $\mathcal{B}$ and that has the highest number of matches with the cubes in $\mathcal{B}$ (or $\mathcal{C}$ if it is the first iteration)*

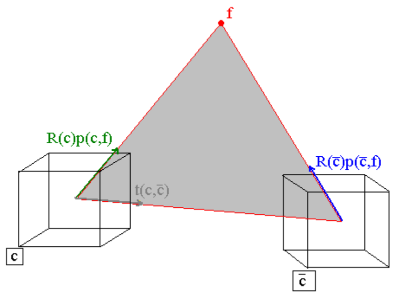

Figure 4. Epipolar constraint: the three shown vectors have to be coplanar.

2. *Identify the cube $\bar{C}$ of $\mathcal{B}$ that best matches C and initialize the rotation $R(C)$ associated with the current cube as follows :*

$$R(C) = \begin{cases} R(\bar{C})R(C,\bar{C}) & \text{, if } \mathcal{B} \text{ contains at least a cube} \\ I_3 & \text{, otherwise} \end{cases}$$

Where $R(C,\bar{C})$ is the rotation extracted from the essential matrix linking C and $\bar{C}$ and I_3 the 3 by 3 identity matrix.

3. *Minimize the error function given in (4) except the fact that the first sum is over the adjuster $\mathcal{B}$ instead of the whole set of cubes $\mathcal{C}$:*

$$e(\Theta) = \sum_{C \in \mathcal{B}} \sum_{\bar{C} \in \mathcal{M}(C,\bar{C})} \sum_{C \in \mathcal{F}(C,\bar{C})} r(C,\bar{C},f)^2$$

We used non linear solution available in Matlab under ***lsqnonlin*** *with zero as a goal and the jacobian of the residual defined in the next paragraph.*

4. *Process the next cube*

}

The initialization in Step 2 can be thought of as bringing back the cube C to the world frame configuration, then rotating it into the best matching cube position. This causes both involved cubes to be aligned but does not guarantee alignment with all the other cubes of $\mathcal{B}$.

In step 3 the variables of the minimization that takes place at each iteration are in fact the rotations represented by their *Rodrigues* vectors. The *Rodrigues* vector is mentioned in more details in [13] as well as elements of the minimization over the $SO(3)$ group. Using the latter in conjunction the *index* notation described - particularly the permutation tensor ε_{ijk} - in [12], one can quantify the variation of the variables all under Θ. A slightly modified version of equation (3) gives us:

$$r = (Rp \times \bar{R}\bar{p}) \bullet Rt \tag{5}$$

We derive r with respect to the elements of R:

$$\frac{\partial r}{\partial \delta\omega_\alpha} = -\varepsilon_{ijk}\bar{R}_{jn}D_{mnr}\left[\varepsilon_{\alpha i\rho}R_{\rho m}R_{kr} + \varepsilon_{\alpha k\beta}R_{im}R_{\beta r}\right] \tag{6}$$

The derivative with respect to the elements of $\bar{R}$:

$$\frac{\partial r}{\partial \delta\bar{\omega}_\alpha} = -\varepsilon_{ijk}\varepsilon_{\alpha j\rho}\bar{R}_{\rho n}R_{im}R_{kr}D_{mnr} \tag{7}$$

In both cases, the tensor D is given by:

$$D_{mnr} = p_m\bar{p}_n t_r \tag{8}$$

Note that the *index* notation in (6) and (7) implies summations over all indices except α. The two rules we followed in our "loose" adaptation of the *index* notation are first, the use of subscript to describe a vector's component, and second the implicit summation property of the classic *index* notation. Overall, in the procedure presented above, an optimization process is undertaken for each cube incrementally added to the set. By proceeding this way, convergence is obtained after a few iterations. Ultimately this method allows us to estimate the optimal overall rotations for each panorama of the sequence. The world reference frame is attached in this case to the first cube selected in the bundle adjustment. Applying properly the rotations to each one of the cubes results in cube alignment.

3.4. Outlier rejection

Although a large amount of outliers in the match set have been eliminated during the essential matrices E estimation step, some may have survived and therefore affect the cube alignment step. In order to eliminate these outliers, we could apply the $x84$ rejection rule [8]. This statistical approach proceeds by computing the *M*edian *A*bsolute *D*eviation of all residuals. All matches with residuals over $5.2MAD$ are eliminated from the set and a new solution is then recomputed.

4. 3D Pose Recovery

In the previous section, the cube alignment allowed accurate camera orientation recovery. This subsequently helps in the computation of the initial structure necessary for the the pose estimation. As a matter of fact, using a classic bundle adjustment approach, an initial scene structure is refined to obtain optimal camera motion parameters as well as $3D$ feature points.

4.1. Full Bundle Adjustment

Let us first consider X_i the i^{th} $3D$ point estimated, t_j and R_j the motion parameters of cube C_j with respect to the world reference frame and finally p_{ij} the original match corresponding to the projection of X_i in C_j. We will note M the number of estimated $3D$ points, N the total number of panoramas and $\mathcal{N}_i$ the set of panoramas in which X_i is visible.

Ideally if X_i is re-projected in each of the cubes C_j of $\mathcal{N}_i$, one should obtain projected points $\bar{p}_{ij}$ equal to the original p_{ij} and this for all points in all cubes : the ray generated by the back projection of p_{ij} into space and the ray going from C_j to X_i should be aligned and of same direction. Similarly as what is done in section 3.2, a geometrical test on a unit dot product of the unit vectors directing those rays is sufficient. This gives rise the minimization criterion materialized by the following test on unit dot product i.e residual re-projection error of X_i in cube C_j:

$$r_{ij} = 1 - \frac{p_{ij}}{\|p_{ij}\|} \bullet \frac{R_j^T(X_i - t_j)}{\|R_j^T(X_i - t_j)\|} \quad (9)$$

with

$$i \in \{1, \cdots, M\} \text{ and } j \in \mathcal{N}_i$$

The overall error to minimize is therefore :

$$e = \sum_{i=1}^{M} \sum_{j \in \mathcal{N}_i} r_{ij}^2 \quad (10)$$

The use of such a minimization procedure implies a prior initialization of the variables to be estimated, in this case, the coordinates of all $3D$ points from matches and the positions of the panoramas.

4.2. Algorithm

4.2.1 Initialization

In any bundle adjustment process, starting with good initial values is of prime importance in order to reduce the probability of being stuck in a local minima. In the preceding section, initial rotation values for the incrementally added cameras were obtained. Here, a global initial configuration for the graph of cameras must be identified.

First, to initialize the positions of the panoramas, two assumptions are made without any loss of generality. The first cube of the sequence C_1 could have its position initialized to $t_1 = (0, 0, 0)$, implying that the latter panorama is the reference. The second assumption fixes the scale of the reconstruction. As a matter of fact, the second cube C_2, has its position forced to $t_2 = t_{12}$ the translation extracted from E_{12}.

Note that it is not necessary to compute all possible E_{jk}. In our case we computed all E_{jk} for $j \in \{1, \cdots, N-1\}$ and $k \in \{j, \cdots, N\}$.

Using the assumptions above and the appropriate essential matrices, therefore appropriate unit direction vectors, one can recover by triangulation an estimate of the positions of the cubes. For example, if the positions of 2 cubes C_j and C_k are known, then the unit translation vectors t_{jl} and t_{kl} can be intersected to obtain the position of cube C_l.

To initialize the $3D$ points X_i, one can also proceed by multiple triangulations over all valid pairs of cubes and then average the results obtained. Once the positions of the cubes and the points are estimated, the minimization procedure is applicable. The results are the positions $t_j, j \in \{1, \cdots, N\}$ of all cubes and the $3D$ points $X_i, i \in \{1, \cdots, M\}$ corresponding to the M N-tuples of matches across all cubes. The fact that the initial rotations R_i were obtained by the previous cube alignment procedure garantees the good quality of the initial estimate.

4.2.2 Iteration update

For simplicity's sake let us drop the index i and j in equation (9) since they only identify the $3D$ point and cubic panorama of interest. A simpler version of (9) is written as :

$$r = 1 - \frac{p}{\|p\|} \bullet \frac{R^T(X - t)}{\|R^T(X - t)\|} \quad (11)$$

This re-projection error r is a function of X, R (represented by its associated $Rodrigues$ vector ω) and t. Grouping all these variables under the symbol $\theta = (\omega, t, X)$, the jacobian of this error function is given by estimating $\frac{\partial r}{\partial \theta}$. Let us consider :

$$\tilde{p} = \frac{p}{\|p\|},\ Q = R^T,\ A = QX,\ B = Qt$$

$$\lambda = \frac{1}{\|R^T(X - t)\|} = \frac{1}{\|A - B\|}$$

Equation (11) becomes :

$$r = 1 - \tilde{p} \bullet \lambda(A - B) \quad (12)$$

For the derivation part, we use, once more, the *index* notation. First, Equation (11) and (12) result in :

$$r = 1 - \tilde{p}_m(\lambda(A_m - B_m)) \quad (13)$$

Note that $\tilde{p}_m = \frac{p_m}{\|p\|}$. The partial derivative of r with respect to parameter vector θ is thus given by :

$$\frac{\partial r}{\partial \theta} = -\tilde{p}_m(\frac{\partial \lambda}{\partial \theta}(A_m - B_m) + \lambda(\frac{\partial A_m}{\partial \theta} - \frac{\partial B_m}{\partial \theta})) \quad (14)$$

The variable θ contains three types of variables each belonging the entities ω (representing R), t and X. There are as many extended versions of Equation (14) as there are types of variables. It results the following expression for $\theta = \omega_\alpha$, any component of the *rotation* vector :

$$\begin{aligned} \frac{\partial r}{\partial \omega_\alpha} &= \tilde{p}((\frac{1}{\lambda^3}(A_i - B_i)(-\epsilon_{\alpha i \beta}(A_\beta - B_\beta)))(A_m - B_m) \\ &\quad - \lambda(\epsilon_{\alpha m \gamma}(A_\gamma - B_\gamma))) \end{aligned} \quad (15)$$

For the *translation* vector components :

$$\frac{\partial r}{\partial t_\alpha} = \tilde{p}((\frac{1}{\lambda^3}(A_i - B_i)(Q_{ik}\delta_{k\alpha}))(A_m - B_m) - \lambda(Q_{mn}\delta_{n_\alpha})) \quad (16)$$

Finally for the $3D$ point components :

$$\frac{\partial r}{\partial X_\alpha} = \tilde{p}((\frac{1}{\lambda^3}(A_i - B_i)(-Q_{ik}\delta_{k\alpha}))(A_m - B_m) + \lambda(Q_{mn}\delta_{n\alpha})) \quad (17)$$

δ is the *Dirac* operator and ϵ is the *permutation tensor*. Equations (15), (16) and (17) can then be used in the optimization algorithm to characterize its variation at each iteration.

5. Results

5.1. Example with an indoor scene

Fig. 5 shows the 4 original panoramas in their cubic representation. Once the rotational components of each panorama have been estimated, one can then regenerate the cubes by undoing the rotations and thus obtained a set of aligned images. Note that the rotation estimation step resulted in a cumulative re-projection error less than to 1×10^{-4} radians (in the order of 10^{-1} to 1 pixel for a cube of side 512 pixels) after an average of 8 iterations per cube added to the bundle. The regenerated aligned cubes are shown in Fig. 6 in which it can be observed that the panoramas are now pointing to the same scene direction. Finally, Fig. 10 shows a virtual perspective view of the reconstructed cube positions given by the described procedure. It illustrates how, for the experiment, the omni-directional camera was randomly oriented during the four scene captures.

5.2. Example with an outdoor scene

One cube of the set of outdoor panoramas is shown in Figure 1. The rotation estimation stage resulted in an error less than 5×10^{-2} radians (in the order of 1 to 10 pixels for a cube of side 512 pixels) after an average of 10 iterations per cube added to the bundle. The panoramas before and after alignment can be seen in Figures 7 and 8 respectively.

Using the pose recovery results, one can also proceed to $3D$ reconstruction of scene elements. Figure 9 displays the top and perspective views of the reconstructed scene in which some scene points were selected in the images to allow a polygonal reconstruction of meaningful scene elements. The main door of the closest building and the windows above it are easily identifiable in the reconstruction; the same goes for the stair steps in front of the main door. These elements are all visible in the view of Figure 1. Figure 11 illustrates the pose estimation results by showing a $3D$ reconstruction of the configuration up to a scale in a coarse representation of the environment. To quantitatively evaluate the accuracy of the reconstructed scene, we selected 20 pairs of segments that are of equal length in the real scene. For each of these pairs, we computed the ratio of the segments length in the reconstruction and obtained an average value of 1.0387 (with a standard deviation of 0.0374).

Figure 5. Indoor scene: the original cubes (1 to 4) laid out (top and bottom faces removed).

Figure 6. Indoor scene: the aligned cubes (1 to 4) laid out (top and bottom faces removed)

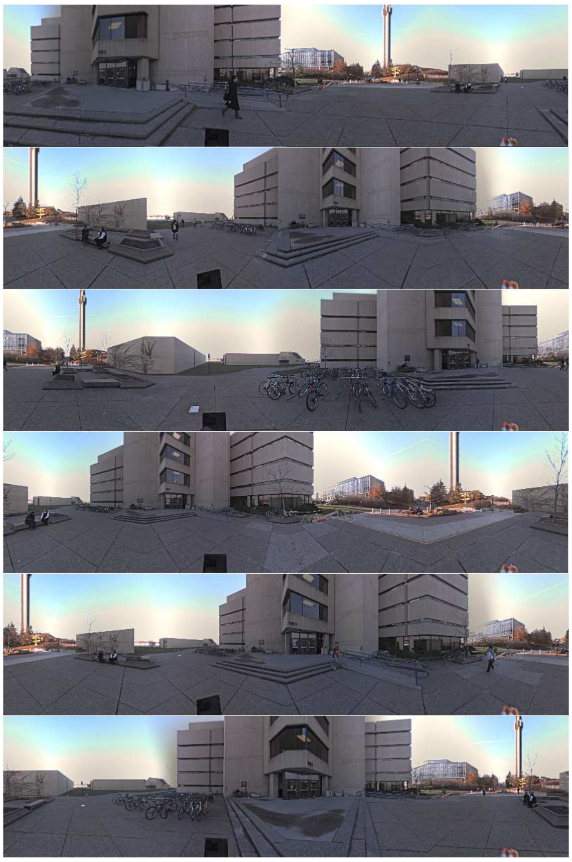

Figure 7. Outdoor scene: the original cubes laid out (top and bottom faces removed).

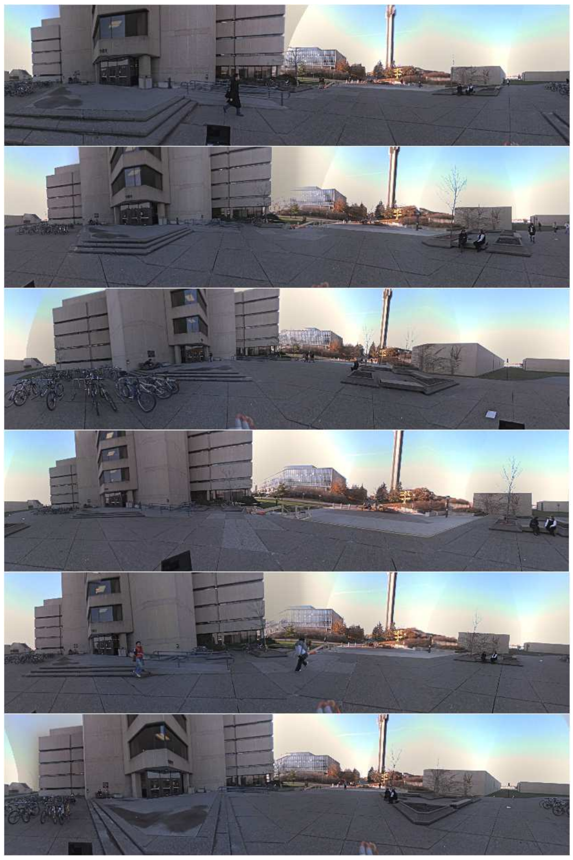

Figure 8. Outdoor scene: the aligned cubes laid out (top and bottom faces removed)

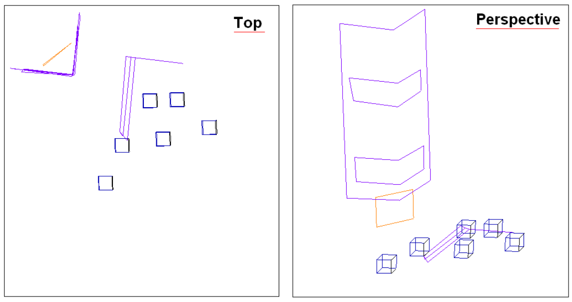

Figure 9. Outdoor scene with a few objects and the 6 aligned panoramas (cubes) at their computed locations.

6. Conclusion

This article presented an algorithm to estimate the motion parameters in the case of a sequence of cubic panoramas based on initial orientation estimation called cube alignment. The essential matrix, extensively used in the completion of such a process, proves to be simple to evaluate in the case of cubes. From that entity, initial motion parameters are found for the main algorithm. First, rotations with respect to a reference cube are evaluated through an incremental bundle adjustment procedure. Then, the pose recovery is achieved by estimating the translations, the $3D$ points from the matches and refining the previously obtained rotations using another optimization phase. To refine the results, it is also possible to iterate a few times over this process. However, from the residual errors obtained, we found the results after one pass to be sufficiently accurate for our application.

Two sets of panoramas were processed and good results were obtained for the reconstructed scenes in both cases. Future improvements to this algorithms will be achieved by improving the matching process as well as integrating the sparse matrix property described in [11] to speed up the optimization time.

References

[1] D. Bradley, A. Brunton, M. Fiala, and G. Roth. Image-based navigation in real environments using panoramas. In *IEEE Int. Workshop on Haptic Audio Visual Environments and their Applications*, pages 103 – 108,

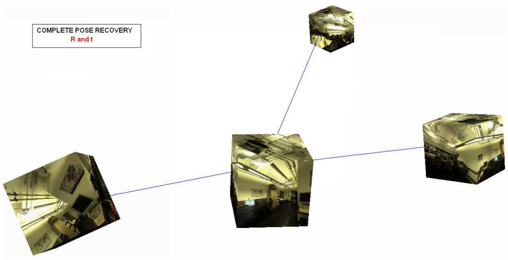

Figure 10. Pose recovery for the indoor set of panoramas.

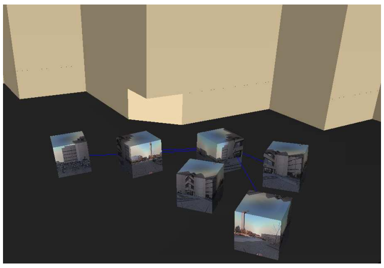

Figure 11. Pose recovery for the outdoor set of panoramas.

October 2005.

[2] M. Brown and D. G. Lowe. Recognising panoramas. In *ICCV '03: Proc. of the 9th IEEE Int. Conf. on Computer Vision*, page 1218, Washington, DC, USA, 2003. IEEE Computer Society.

[3] Rodrigo Carceroni, Ankita Kumar, and Kostas Daniilidis. Structure from motion with known camera positions. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 477–484, Washington, DC, USA, 2006. IEEE Computer Society.

[4] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, ISBN: 0262061589, first edition, 1993.

[5] O. Faugeras and Q-T. Luong. *The geometry of multiple images*. Cambridge University Press, ISBN: 0262062208, first edition, 2001.

[6] M. Fiala. Pano-presence for teleoperation. In *Intelligent Robots and Systems (IROS 2005), 2005 IEEE/RSJ International Conference on*, pages 3798–3802, 2005.

[7] M. Fiala and G. Roth. Automatic alignment and graph map building of panoramas. In *IEEE Int. Workshop on Haptic Audio Visual Environments and their Applications*, pages 103 – 108, October 2005.

[8] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. *Pattern Analysis and Applications*, 2:312–320, 1999.

[9] R. Hartley. In defense of the 8-point algorithm. *IEEE trans. Pattern Analysis and Machine Intelligence*, 19:580–593, 1995.

[10] R. Hartley. Theory and practice of projective rectification. *Int. journal Computer Vision*, 35(2):115–127, 1999.

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[12] J.H Heinbockel. *Introduction to Tensor Calculus and Continuum Mechanics*. Trafford Publishing, ISBN: 1553691334, first edition, 2001.

[13] F.O Kuehnel. On the minization over so(3) manifolds. Technical Report TR.0312, Research Institute for Advanced Computer Science (RIACS), USA, 2003.

[14] VIVA Research Lab. Virtual navigation in image-based representations of real world environments, 2006. www.site.uottawa.ca/research/viva/projects/ibr/.

[15] Anat Levin and Richard Szeliski. Visual odometry and map correlation. In *CVPR (1)*, pages 611–618, 2004.

[16] Ameesh Makadia and Kostas Daniilidis. Rotation recovery from spherical images without correspondences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1170–1175, 2006.

[17] David Nistér and Henrik Stewénius. A minimal solution to the generalised 3-point pose problem. *J. Math. Imaging Vis.*, 27(1):67–79, 2007.

[18] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *OMNIVIS 2005*, 2005.

[19] Tomáš Svoboda, Tomáš Pajdla, and Václav Hlaváč. Motion estimation using central panoramic cameras. In Stefan Hahn, editor, *IEEE International Conference on Intelligent Vehicles*, pages 335–340, Stuttgart, Germany, October 1998. Causal Productions.

[20] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, Sophia-Antipolis Cedex, France, 1996.

[21] Zhengyou Zhang. Motion and structure from two perspective views: from essential parameters to Euclidean motion through the fundamental matrix. *Journal of the Optical Society of America A*, 14:2938–2950, November 1997.