Orientation and Pose estimation of Panoramic Imagery

Robert Laganière and Florian Kangni
VIVA Research Lab
School of Information Technology and Engineering
University of Ottawa
800 King Edward
Ottawa, Ontario
K1N 6N5
Canada
laganier,fkangni@site.uottawa.ca
1-613-562-5800
Fax: 1-613-562-5664

January 5, 2010

Orientation and Pose estimation of Panoramic Imagery

Abstract

In a database of geo-referenced images, determining the exact position of each panorama is an important step in order to ensure the consistency of the visual information. This paper addresses the problem of camera pose recovery from spherical (360^o) panoramas. The $3D$ information is extracted from a database of panoramic images sparsely distributed over a scene of interest. We present a two-stage algorithm to recover the position of the omni-directional cameras 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 *panorama alignment*. Full camera positions are then computed using bundle adjustment. During this step, the previously computed panorama orientations, used to feed the global optimization process, can be further refined. Results are shown for indoor and outdoor panorama sets.
Keywords: omni-directional panorama pose estimation bundle adjustmentstructure from motion

1 Introduction

When a collection of images showing a given environment has been captured, one important step consists in the camera pose recovery. This problem consists essentially in using visual information extracted from the set of images in order 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 take advantage of the motion parameters, in a virtual navigation application, to adequately position the images with respect to the environment (or a map representation of it). The fact that the panoramas are properly aligned also ensures that the information displayed when hopping from one panorama to another is visually consistent. Virtual objects can be inserted into the pictured scene and the motion parameters can be used for interpolating intermediate viewpoint generation.

The problem of pose recovery from large set of images has been the object of several studies in the literature, particularly in the context of city navigation. Images or video sequences are taken from a moving vehicle that follows some path inside the environment. Incremental or piece-wise reconstruction is then achieved using bundle adjustment in order to position all these images with respect to the environment. With the introduction of more sophisticated digital cameras, panoramic images are now often used to build such image-based models of large environments. Positional information is often obtained using GPS, but while this is appropriate for virtual navigation systems (e.g. Google Street View), this solution can be inadequate in applications requiring more accurate positioning.

In this paper, we present an approach for orientation and pose recovery in the context of a database of spherical panoramas sparsely captured in an

environment. We show that it is possible to perform the reconstruction using a two-stage procedure where rotations are first estimated followed by a full pose recovery process. It must be noted that the approach described here could also applies 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 breaks the set into smaller groups sharing common field-of-views.

In addition, splitting the pose recovery problem makes it applicable to various contexts where only part of the positional information is available. In some cases, the position of each panorama is known with an acceptable accuracy from a GPS device. Only rotation recovery is thus required in order to align the panoramas. This alignment is essential in virtual navigation in order to avoid jittering when hopping from one panorama to another. In other cases, the orientations are already known (e.g through the use of an inertial device), pose recovery can be applied directly. When no initial estimates of the camera positions are available, splitting the pose recovery process into two steps makes the estimation more reliable. The two-stage procedure benefits from the accuracy of the initial estimate of the structure orientation-wise which results in higher chances of convergence toward the global optimal solution.

2 Related works

Zhang [1] 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 [2]. The focus is however on how efficient motion extraction from spherical cameras allows the distinction between a pure translation and a pure rotation. In [3] and [4], global solutions to the pose recovery problem are given respectively for one generalized camera and for a pair of generalized cameras. [4] 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 $SO(3)$ [5]. 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 [6] 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 used to estimate the rotations is correlation-based. Optimization over the $SO(3)$ group is also discussed in [7].

Finally, in [8] 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.

3 Two-stage orientation and pose recovery

Determining the exact position of the panoramas in a database of geo-referenced images allows ensuring the consistency of the visual information. When available, the GPS information attached to each image constitutes a first estimate that, in most application, needs to be validated and refined. When the GPS coordinates cannot be obtained, such as in indoor scenes, algorithmic pose estimation becomes the only way to position the panoramas in the environment.

In order to solve the pose estimation problem applied to spherical images, we have designed an algorithm that is divided into two stages:

1. The first stage is the estimation of the rotation of each panorama with respect to a global world frame. A bundle adjustment approach in conjunction with pair-wise essential matrices is used to recover each rotation in a global solution. Once these rotations are known, one can then obtain a configuration in which all camera frames are aligned with respect to each other. This configuration achieves what will be designated as *panorama alignment*. This procedure is described in Section 4.

2. The second and final stage of the pose recovery is to start from the aligned set of panoramas to estimate their relative positions with respect to one another. Since negligible rotation now exists between each camera frame, the problem becomes a pure translation estimation. The final solution, up to a scale, is obtained using an error minimization approach. Section 5 describes this second procedure.

The accurate positioning information thus obtained can then be used to control the navigation inside the image database. Intersection point can be determined; redundant information (panorama taken from practically the same point of view) can be eliminated. Also, the fact that the images are properly aligned make sure that these ones are displayed in a visually consistent way when hopping from one panorama to another. It relates each image with respect to the other, therefore visual information (e.g. textual annotation) can be propagated from images to others.

Before we give the details of our pose recovery approach, we briefly review in the following subsection the basic projective geometry concept related to panoramic imaging.

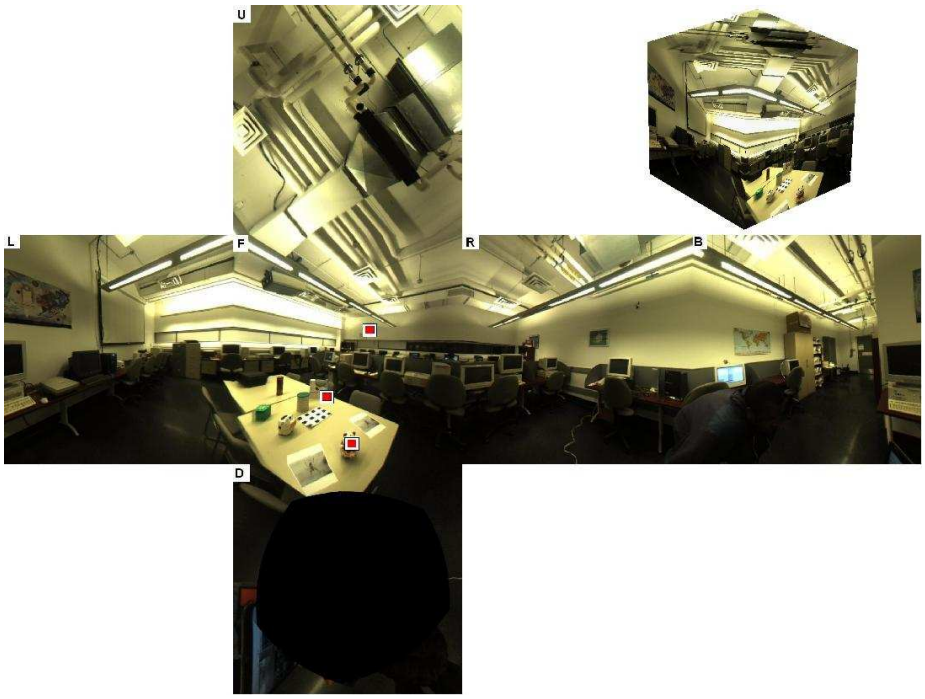


Figure 1: Cube in perspective view and laid out in cross pattern

3.1 Panoramas projective geometry

Spherical panoramas are produced by a central projection camera that collects all light rays coming from all directions. In our case, they are composed from the images captured by the Point Grey Ladybug camera. It is essentially a camera composed of 6 sensors (1024×768 pixels each), 5 laterals and 1 pointing upwards that capture a view of the world at 360 degrees around the azimuth completed by a top view.

Since the camera's sensors have overlapping fields of view, they can be accurately calibrated. It is therefore possible to fuse the six images to form an almost complete spherical panorama. The resulting plenoptic function can be re-projected on any type of surface : sphere, cylinder, cube, etc. We use here a cubic representation made of six identical faces, each of them acting as a standard perspective projection camera with 90^o field of view. The calibration matrix K associated with each sensor for each face of the cube is then given by:

$$K = \begin{pmatrix} -\frac{L}{2} & 0 & \frac{L}{2} \\ 0 & \frac{L}{2} & \frac{L}{2} \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

where L is the size of a cube in pixels, as set by the user at the generation stage.

This representation has been shown to be very convenient to handle and can be stored and rendered very efficiently on standard graphic hardware [9][10]. It must however be stressed out that this choice of representation is arbitrary and the results presented applies to any other representation.

When several spherical panoramas are analyzed, they are linked by the usual projective relations. By definition the essential matrix is :

$$E = [t]_{\times} R \tag{2}$$

Up to a scale, E characterizes completely the geometry between two panoramas, just as in the case of conventional stereo images. It follows the following constraints for two matching points p and p':

$$p'^{T} E p = 0 \tag{3}$$

Calibration information being known, p and p' are expressed as 3D coordinates of the image points on the 3D projection surface (a cube in our case). Estimating E then becomes a matter of solving the classical problem of the epipolar geometry estimation using for example the 8-point algorithm [11].

4 Panorama alignment

This section describes the *panorama alignment* procedure which constitutes the first step in our pose estimation process. Panorama alignment consists essentially in removing the rotational component that separates the different panorama-centric 3D frames. Since N panoramas have to be aligned all at once, we have chosen to base our solution on a bundle adjustment algorithm. The choice of such an approach is most of all guided by the fact that it is a *global* solution that is seek, i.e. where each variable (rotation in this case) has its influence on the final result. Bundle adjustment has proven its effectiveness in many applications where camera poses were computed in order to provide a $3D$ representation of a scene [12, 13, 14, 15]. Obviously, a a naive approach where the motions extracted from the pairwise essential matrices are chained together would have lead to an unacceptable accumulation of errors.

The core of the concept relies on the error or cost function that is to be minimized so that the parameters are estimated properly. In the usual case of camera pose recovery for example, the cost function is the sum of all re-projection errors for all observed points of the scene. In the case of panoramas alignment, the cost function expresses the re-projection error by evaluating the epipolar constraint for all available pairs of panoramas.

4.1 Epipolar constraint in spherical panoramas

Let f be an interest point in a scene (a 3D point) visible in cubic panorama c and $\bar{c}$ with respective cube coordinates $p(c, f)$ and $p(\bar{c}, f)$. If $R(c)$ and $R(\bar{c})$ are the respective "aligning" rotations of each of these cubes with respect with

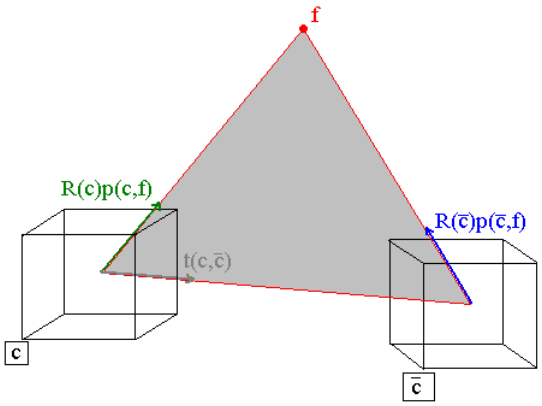


Figure 2: The epipolar constraint.

the world coordinate frame system, and $t(c, \bar{c})$ is the unit vector in c pointing in the direction of the projection center of $\bar{c}$, then epipolar constraint suggests that the vectors $R(c)p(c, f)$, $R(\bar{c})p(\bar{c}, f)$ and $R(c)t(c, \bar{c})$ be coplanar (see Fig.2). This is expressed by the following triple scalar product and *residual*:

$$r(c, \bar{c}, f) = (R(c)p(c, f) \times R(\bar{c})p(\bar{c}, f)) \bullet R(c)t(c, \bar{c}) \tag{4}$$

Ideally, the coplanarity implies that :

$$r(c, \bar{c}, f) = 0 \tag{5}$$

This gives rise a natural choice for the objective function that is to be minimized. Since the coplanarity constraint has to be verified for all matches between all possible pairs of panoramas, the total error is given by:

$$e(\Theta) = \sum_{c \in \mathcal{C}} \sum_{\bar{c} \in \mathcal{M}(c,\bar{c})} \sum_{c \in \mathcal{F}(c,\bar{c})} r(c, \bar{c}, f)^2 \tag{6}$$

The set of all panoramas that are of interest in the bundle adjustment algorithm is noted $\mathcal{C}$. For a given panorama c, the set of panoramas that share some matches with the latter is noted $\mathcal{M}(c)$. This error function is similar to the one used by [12], but here adapted to the case of panoramas. The minimization of this cost function results in estimates of the different rotations $R_1, \cdots, R_N$ represented by their Rodrigues vector all stated under the single parameter $\Theta = (\omega_{1_x}, \omega_{1_y}, \omega_{1_z}, \cdots, \omega_{N_x}, \omega_{N_y}, \omega_{N_z})$.

4.2 Solving for all rotations

This section describes the implementation of the algorithm summarized by Equation (6). The steps mentioned here follow the framework described in [12] to solve the problem of bundle adjustment for N images. Let us, beforehand, note $\mathcal{B}$ the set of panoramas for which rotations are being optimized.

The algorithm requires that for all available pairs of panoramas $(c, \bar{c})$ that share matches, it is necessary to first compute the associated essential matrix $E(c, \bar{c})$ and then extract the translation unit vector $t(c, \bar{c})$ and the rotation $\rho(c, \bar{c})$ between panoramas c and $\bar{c}$.

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

- Add a panorama c in $\mathcal{C}$ that is not in $\mathcal{B}$ and that has the highest total number of matches with the panoramas in $\mathcal{B}$ (or $\mathcal{C}$ if it is the first iteration)
- Identify the panorama $\bar{c}$ of $\mathcal{B}$ that best matches c and initialize the rotation $R(c)$ associated with the current panorama as follows :

$$R(c) = \begin{cases} R(\bar{c})\rho(\bar{c}, c) & \text{, if } \mathcal{B} \text{ contains at least one panorama} \\ I_3 & \text{, otherwise} \end{cases}$$

- Minimize the error function given in (6) except the fact that the first sum is over the adjuster $\mathcal{B}$ instead of the whole set of panoramas $\mathcal{C}$:

$$e(\Theta) = \sum_{c \in \mathcal{B}} \sum_{\bar{c} \in \mathcal{M}(c,\bar{c})} \sum_{c \in \mathcal{F}(c,\bar{c})} r(c, \bar{c}, f)^2 = 0$$

- Process the next panorama

} $\rho(c, \bar{c})$ is the rotation extracted from the essential matrix linking c and $\bar{c}$ and I_3 the 3 by 3 identity matrix. The initialization described above can be thought of as bringing back the panorama c to the world frame configuration first then rotating it into the best matching cube position. This causes both involved panoramas to be aligned but does not guarantee alignment with all the other panoramas of $\mathcal{B}$.

For the error minimization, we use a non linear solution available in Matlab under **lsqnonlin** with zero as a goal and the Jacobian of the residual defined in the next section.

4.3 Jacobian evaluation for minimization

To be able to specify the variation of the objective function that is minimized in the previous section, it is necessary to evaluate the variation of the residuals that are squared and summed up in the error function. We recall that Θ stood

for all possible parameters involved in the process describing all rotations that are being optimized. Considering that the goal here is to compute $\frac{\partial r}{\partial \Theta}$, equation (5) gives us the residual involving 2 panoramas c and $\bar{c}$ and a given feature f:

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

Dropping the indices for sake of simplicity, the residual attached to a particular feature f becomes:

$$r = (Rp \times \bar{R}\bar{p}) \bullet Rt \tag{7}$$

This triple scalar product (7) can be rewritten using the permutation tensor ε_{ijk}:

$$\begin{aligned} r &= \varepsilon_{ijk} R_{im} p_m \bar{R}_{jn} \bar{p}_n R_{kr} t_r \\ &= \varepsilon_{ijk} R_{im} \bar{R}_{jn} R_{kr} p_m \bar{p}_n t_r \end{aligned} \tag{8}$$

Let us consider the tensor D that depends only on the components of the vectors p, $\bar{p}$ and t :

$$D_{mnr} = p_m \bar{p}_n t_r \tag{9}$$

(9) in (8) gives rise to the following tensorial expression of the residual:

$$r = \varepsilon_{ijk} R_{im} \bar{R}_{jn} R_{kr} D_{mnr} \tag{10}$$

[7] gives us an expression of the derivative of a given rotation R with respect to the associated Rodrigues vector ω:

$$\frac{\partial R_{\mu\nu}}{\partial \delta\omega_\alpha} = -\varepsilon_{\alpha\mu\rho} R_{\rho\nu} \tag{11}$$

This expression is the one used in [7] in the author's approach of the minimization problem over the space of rotations using a gradient method with rotations matrices represented by their exponential form. Equations (10) and (11) thus allow us to write :

$$\frac{\partial r}{\partial \delta\omega_\alpha} = \varepsilon_{ijk} \bar{R}_{jn} D_{mnr} \left[\frac{\partial R_{im}}{\partial \delta\omega_\alpha} R_{kr} + R_{im} \frac{\partial R_{kr}}{\partial \delta\omega_\alpha} \right] \tag{12}$$

$$= -\varepsilon_{ijk} \bar{R}_{jn} D_{mnr} \left[\varepsilon_{\alpha i\rho} R_{\rho m} R_{kr} + \varepsilon_{\alpha k\beta} R_{im} R_{\beta r} \right] \tag{13}$$

(13) gives us the expression of partial derivatives of r with respect to the components of the Rodrigues vector associated with the rotation R. For the components $\bar{\omega}_\alpha$ related to rotation $\bar{R}$, we have a similar expression :

$$\frac{\partial r}{\partial \delta\bar{\omega}_\alpha} = -\varepsilon_{ijk} \varepsilon_{\alpha j\rho} \bar{R}_{\rho n} R_{im} R_{kr} D_{mnr} \tag{14}$$

Theoretically, residuals and Jacobians are evaluated in a closed form. Practically, the different sums involved are implemented as loops meaning the latter

entities are computed incrementally. Note also that the speed of the algorithm can be increased by using the *sparse matrix* approach mentioned in [16].

Note that although a large amount of outliers in the match set can be eliminated during the essential matrices E estimation step, some may survive and therefore affect the panorama alignment step. In order to eliminate these outliers, one could apply the $x84$ rejection rule [17]. This statistical approach proceeds by computing the Median Absolute Deviation of all residuals. All matches with residuals over $5.2MAD$ would be eliminated from the set and a new solution would then be recomputed.

5 Pose estimation and structure recovery: evaluating translations

The panorama alignment step created a new set of N cubic panoramas with their respective frames aligned. Therefore to complete the pose recovery, one has to estimate the position of each of the new camera frames with respect to a world frame.

This stage relies mainly on the principle used in "classic" planar image pose recovery [1, 14] where the $3D$ points corresponding to matches are recovered at the same time as the motion parameters of the camera, both up to a scale and through a single global optimization procedure.

In usual SfM algorithms, the criterion that is minimized is the re-projection error enforcing the fact that for all computed $3D$ points, their projection in each camera should ideally correspond to the image points of the original matches that were used. The situation dealt with in this paper is one for which the rotations are already found and the calibration (i.e the intrinsic camera parameters) are already known. The criterion is therefore simplified since only translations or relative positions are involved. Figure 3 illustrates the choice of the minimization criterion. Very similar to what is done with regular $2D$ images, this criterion should insure that the projecting rays of all $3D$ points corresponding to matches in all panoramas should be aligned with their corresponding image points. In other terms, if we insure that, with respect to each reference frame, the ray through the center of the cube of concern and the $3D$ point (i.e ray_1 in Figure 3) and the ray through the corresponding match in the same reference frame (i.e ray_2) are collinear and of same direction, then we guarantee a minimal re-projection error. This can be expressed by a simple dot product on normalized direction vectors, v_1 and v_2; their colinearity being verified by:

$$v_1 \bullet v_2 = 1 \text{ or } 1 - v_1 \bullet v_2 = 0 \tag{15}$$

To apply (15) to the studied case, let us consider X_i the i^{th} $3D$ point estimated, t_j the relative position of panorama 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.

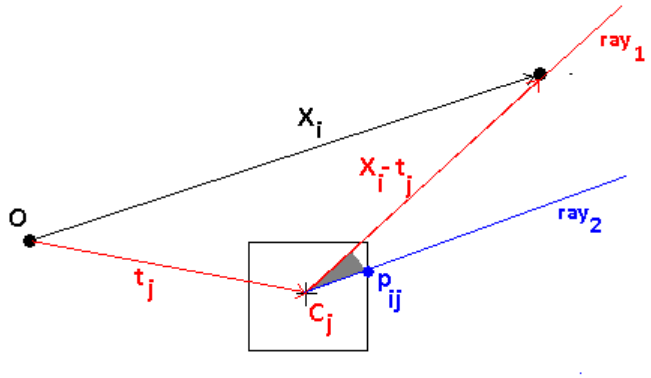


Figure 3: Minimization criterion illustration.

The residual for this particular point is given by :

$$r_{ij} = 1 - \tilde{p}_{ij} \bullet \frac{(X_i - t_j)}{N_{ij}} \tag{16}$$

with :

$$\tilde{p}_{ij} = \frac{p_{ij}}{\|p_{ij}\|} \quad \text{and} \quad N_{ij} = \|X_i - t_j\| \tag{17}$$

Using the tensor notation, Equation (16) can be rewritten as:

$$r_{ij} = 1 - \tilde{p}_{ij}^{\alpha} \frac{(X_i^{\alpha} - t_j^{\alpha})}{N_{ij}} \tag{18}$$

Overall, if $i \in [1, \cdots, n_X]$ and $j \in [1, \cdots, n_C]$, with n_X and n_C respectively the number of $3D$ points reconstructed and the number of panoramas involved, the error e that is minimized is given by :

$$e = \sum_{i=1}^{n_X} \sum_{j=1}^{n_C} r_{ij}^2 \tag{19}$$

Minimizing this global error results in the best approximate $3D$ reconstruction of the scene of interest.

5.1 Initialization

Since a global optimization solution was chosen, a first estimate of the $3D$ structure is necessary and will serve as a seed for the algorithm. This means

that one must obtain approximate positions of the cubic panoramas and of the generated $3D$ points from the available input data (matches across all panoramas). Triangulation will be used here for computing the first approximate of all the unknowns. It is a simple and fast way to obtain a rough estimate of the structure.

First a reference panorama C_r has to be chosen and will obviously have a position t_r equal to 0.

Next, the *scale* of the reconstruction needs to be fixed by setting the length between the reference panorama and a given panorama C_k. This is done by computing the essential matrix E_{rk} from the available matches. The choice can be random if all pairs of panoramas have the same number of matches or can be based on the best matching panorama. Once E_{rk} is found, the unit direction vector t_{rk} is extracted. The position of panorama C_k, that is not a priori known since only the direction is known, is then fixed to be $t_k = t_{rk}$. The distance between both panoramas is set to 1 and represents the unit of measure for all subsequent estimations even for $3D$ points.

With the previous two initializations in place, it is then possible to obtain all other panorama positions estimates from the available matches. The procedure involves triplets of panoramas for example (C_k, C_l, C_m). As shown in Figure 4, epipoles e_{km} and e_{lm} are to be recovered implying the computation of essential matrices E_{km} and E_{lm}. This leads us to the following equation:

$$\alpha e_{km} - \beta e_{lm} + \gamma(e_{km} \times e_{lm}) = t_{kl} \tag{20}$$

α and β are solved for and allow us to recover the position t_{km} of C_m wit respect to C_k as follows:

$$t_{km} = \frac{\alpha e_{km} + (t_{kl} + \beta e_{lm})}{2} \tag{21}$$

And overall, the position of panorama C_m with respect to the reference panorama C_r is given by:

$$t_m = t_k + t_{km} \tag{22}$$

Choosing the reference panorama and the scale of the reconstruction allows us to recover all estimates of panorama positions by using the suitable triplets. If many positions are found for one panorama as a result of many triplets involving the same panorama, then the average is used as the first estimate.

The estimated position of each $3D$ point with respect to the reference frame is also obtained using the same approach.

5.2 Jacobian estimation : residual derivation

The variables of the optimization algorithm minimizing the error given in (19) are the $3D$ points of the scene as well the positions of each one of the panoramas. The initialization procedure explained above described a simple way to get a rough approximation of all variables grouped under $\Theta = \{X_1, \cdots, X_{n_X}, t_1, \cdots, t_{n_C}\}$. To express the progression of the minimization through the variation of the error function, the Jacobian associated to all residuals has to be computed. As

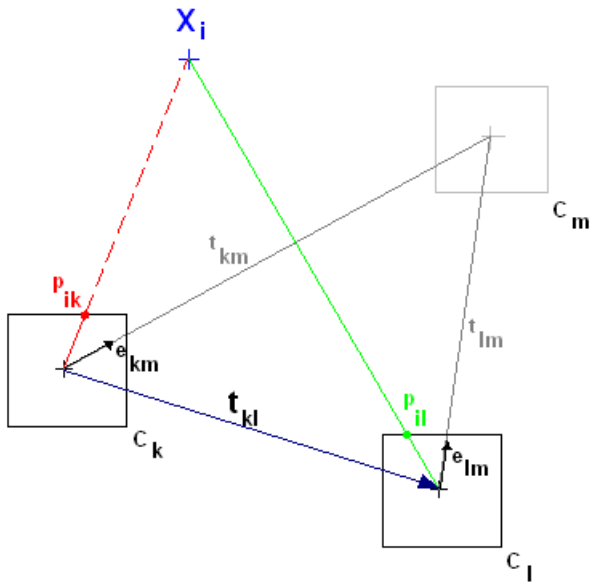


Figure 4: Triangulation for $3D$ points and for cubic panoramas.

a consequence, it is necessary to establish the derivative of any residual r_{ij} as given in (18) with respect to each one of the elements of Θ, the variable vector. We therefore have the following expressions for the residual derivatives:

$$\begin{cases} \text{If } \Theta_m \in \{X_1, \cdots, X_{n_X}\}, & \frac{\partial r_{ij}}{\partial \Theta_m^\beta} = \delta_{mi}(-\tilde{p}_{ij}\frac{\delta_{\alpha\beta}}{N_{ij}} + \frac{1-r_{ij}}{N_{ij}^2}(X_i^\beta - t_j^\beta)) \\ \text{If } \Theta_m \in \{t_1, \cdots, t_{n_C}\}, & \frac{\partial r_{ij}}{\partial \Theta_m^\beta} = \delta_{mj}(\tilde{p}_{ij}\frac{\delta_{\alpha\beta}}{N_{ij}} - \frac{1-r_{ij}}{N_{ij}^2}(X_i^\beta - t_j^\beta)) \end{cases} \tag{23}$$

A total of $n_X n_C$ residuals r_{ij} are computed at each iteration and as a result the Jacobian is a $n_X n_C$ by $3(n_X + n_C)$ sparse matrix. As an implementation note, the optimization algorithm was performed in Matlab using the function *lsqnonlin* with activated Jacobian estimation.

6 Results

This first part of this section presents the results obtained for two different sets of panoramas. The first set depicts an indoor laboratory scene and the second one, an outdoor scene. The second part of this section shows results obtained in situations where the GPS data recorded during the capture of the panoramas give erroneous information about the pose of panoramas. Using the proposed algorithm, we recover the spatial configuration of the panorama sets up to a

Figure 5: Indoor cubic panorama set before alignment (top and bottom faces truncated)

scale. This is done for a linear sequence and a cross shaped set.

6.1 Indoor Scene

Our first experiment was conducted on four randomly captured panoramas in our laboratory. Figure 5 and Figure 6 illustrate the alignment step. The "truncated" versions of the cubic panoramas that are shown in these images (top and bottom faces were removed) are vertically concatenated to ease the observation of the effect of alignment. Note how the same scene elements appear in each aligned image for a the same viewing direction (for instance the windows in the background, or the computer monitors in the center of the aligned cameras). It can also be observed that the difference in tilt of the different cameras have also been eliminated.

Figure 6: Indoor cubic panorama set after alignment (top and bottom faces truncated)

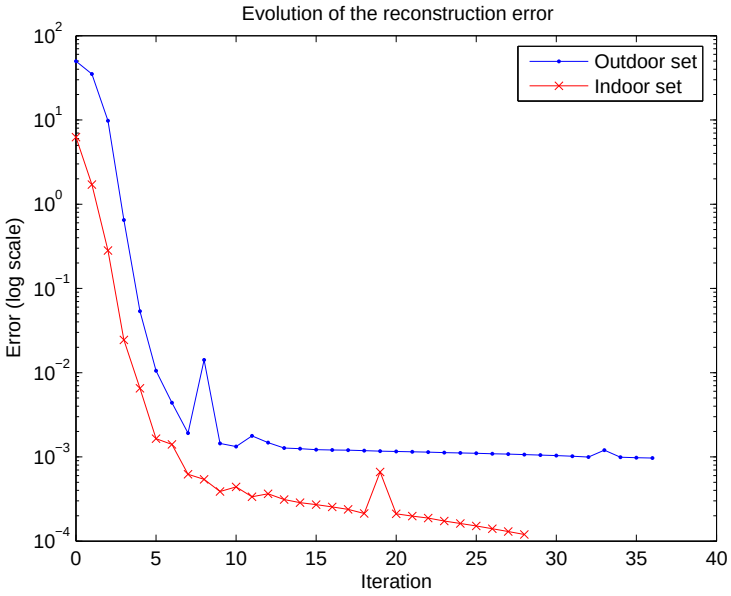


Figure 7: Evolution of the reconstruction error for both sequences during the bundle adjustment of stage 2.

Figure 8: Scooter used in the capture of the outdoor set

6.2 Outdoor Scene

The set used in this particular experiment was captured using a scooter designed to capture a large database of panoramic images in an outdoor environment. Figure 8 shows the PointGrey Ladybug camera mounted on a scooter with an on-board computer for capture and storage purposes. This outdoor set contains 6 panoramas. Figure 9 and figure 10 show the panoramas respectively before and after alignment.

Following the alignment step, the complete pose recovery was performed and the results are displayed as a perspective and top view (Figure 11) of the scene with some elements of the closest building. In this example, we observed that the higher values of residuals correspond to matches that include feature points that were inaccurately localized across the panoramas. This is in part due to the considerable change in viewpoints of the cameras. Figure 12 shows the complete pose recovery from a chosen point of view, with panoramas in their estimated configuration. The evolution of the overall reconstruction error during the second stage is shown in the semi-log graph of Figure 7. Values less than 10^{-3} are reached after about 20 iterations.

6.3 Linear Sequence

The set of panoramas used in this experiment belongs to a linear sequence and can be seen in Figure 13. For the chosen panoramas - eight to be more precise - the GPS data remained constant (the device did not refreshed during the capture of these images) throughout the capture process preventing any pose estimation from the latter data. Pose estimation was applied to the 8 panoramas of the sequence and the $3D$ reconstruction up to a scale can be seen in Figure 14 (left: top view of the set, right: perspective view). The scooter was driven in a straight line direction during the capture and as expected, a linear sequence of panoramas resulted from the pose estimation process.

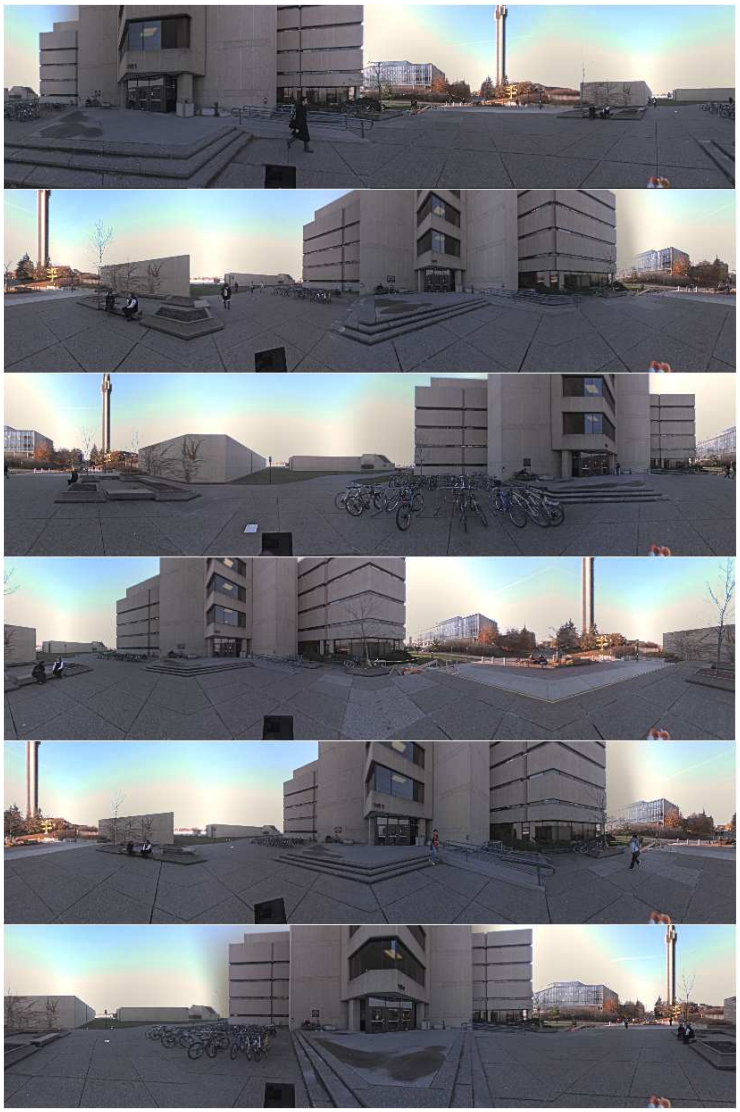


Figure 9: Outdoor cubic panorama set before alignment (top and bottom faces truncated)

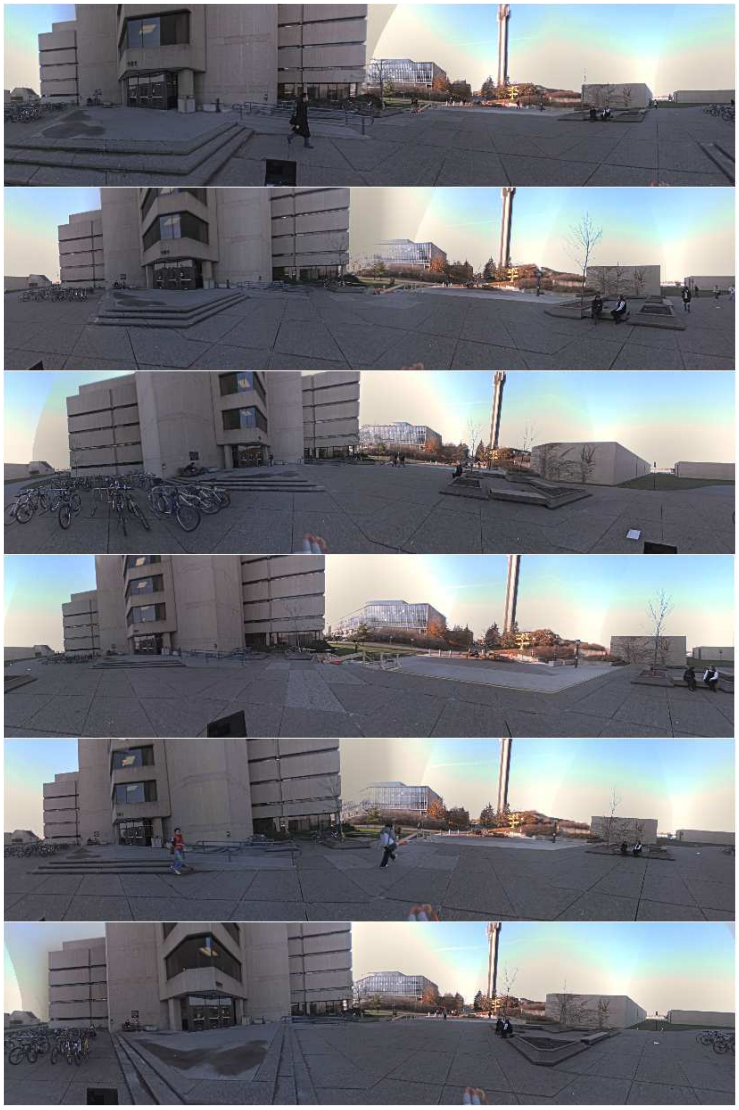


Figure 10: Outdoor cubic panorama set after alignment (top and bottom faces truncated)

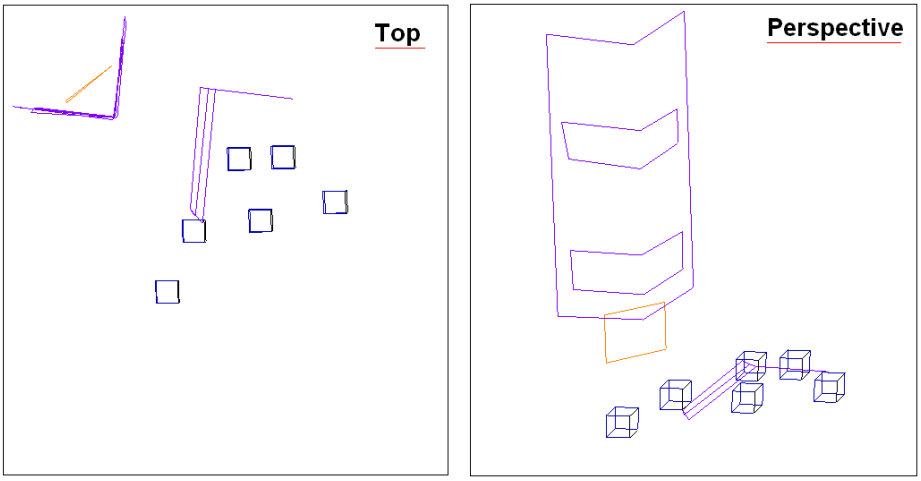


Figure 11: Outdoor scene with a few objects and the 6 aligned panoramas (cubes) at their computed locations

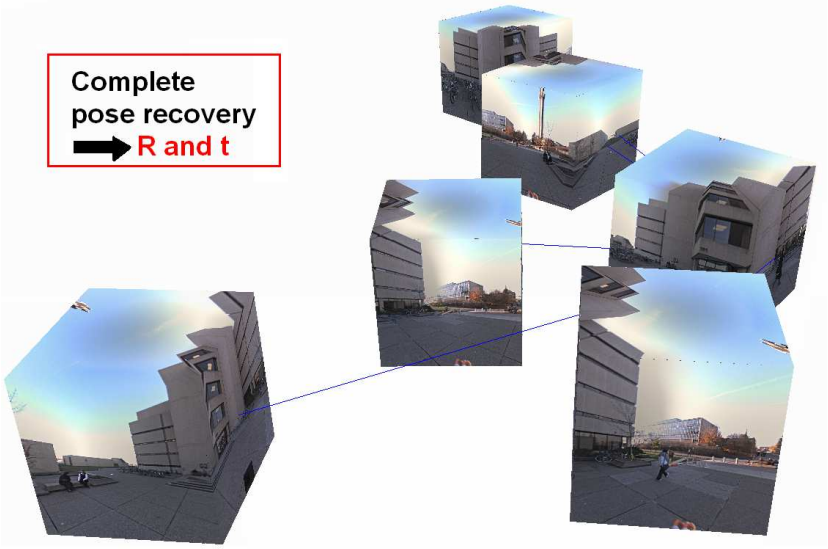


Figure 12: Pose recovery for the outdoor set of panoramas

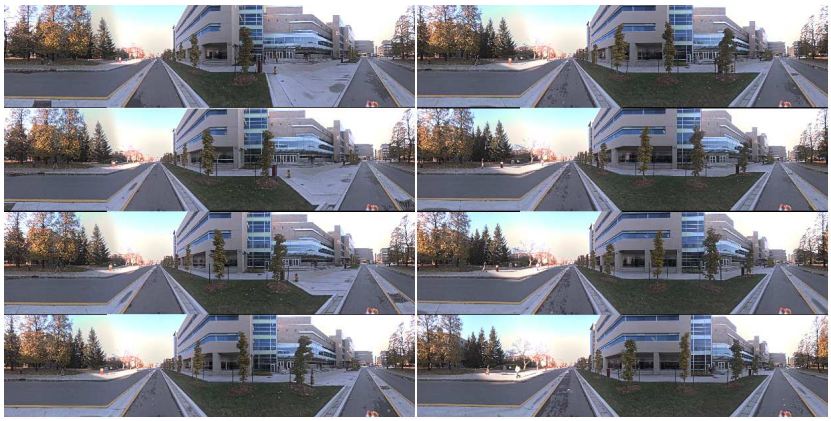


Figure 13: Panoramas of the linear set

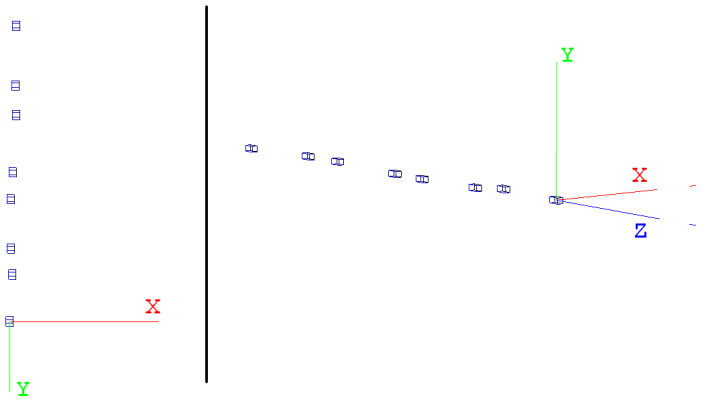


Figure 14: Reconstruction from the linear set: top view (left) and perspective view (right)

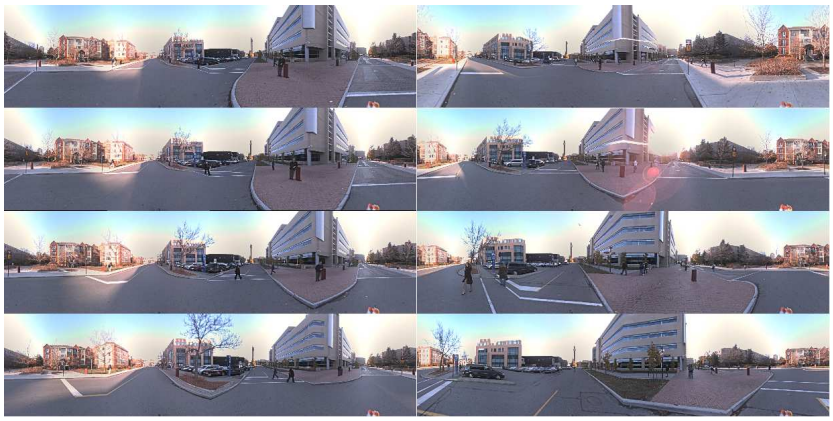


Figure 15: Cross shaped panorama set. Left, from top to bottom: panoramas 50 to 53. Right, from top to bottom: cubes 77 to 80

6.4 Cross shaped set

For this experiment, the panoramas were captured following two almost perpendicular paths. 8 were used here: panoramas with numbers 50 to 53 captured in one direction and panoramas with numbers 77 to 80 in the other direction. The sequence in shown in Figure 15. In this case again, the GPS locations collected by our GPS devices are partly erroneous. The same coordinate was assigned to panoramas 50 to 53 and the panoramas 77 to 80 do not form a linear path as expected as illustrated in Figure 16(a). Applying the proposed pose estimation on these panoramas allows to recover their relative locations up to a scale as seen on Figure 16. The expected cross shaped with almost perpendicular paths is then successfully obtained.

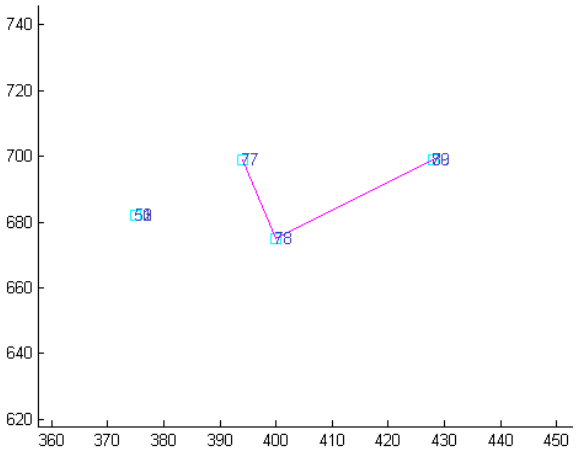
7 Conclusion

A two stage algorithm to achieve pose recovery for a given set of panoramas was presented here.

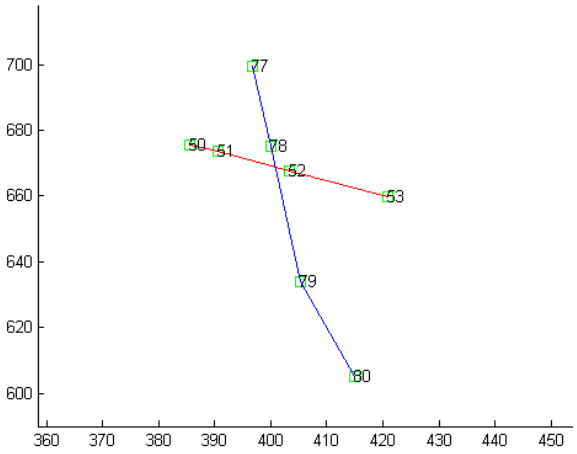
The first step consisted in what was designated as *panorama alignment* i.e rotations recovery. Alignment is a multi-panorama process that resulted in panoramas having parallel reference frames. This configuration was exploited in the second stage of the algorithm.

The second stage used the aligned panoramas to compute the estimated positions of the panoramas up to a scale. Basically a minimization of re-projection error criterion was used to refine an initial structure computed through multiple triangulation. Combining this information with the rotations recovered in the first stage allowed us to complete the pose recovery procedure.

Results were given for an indoor set and an outdoor set for both stages of



(a) From GPS coordinates



(b) From pose estimation, up to a scale

Figure 16: Pose estimation for the cross shaped set

the algorithm. Additional results on two other sets showed how the use of such an algorithm could help panorama localization in cases where the GPS unit is unavailable or produces erroneous coordinates due to different reasons: slow refresh rate, stable satellite connection throughout capture, etc. 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.

References

[1] Z. 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*, vol. 14, pp. 2938–2950, Nov. 1997.

[2] T. Svoboda, T. Pajdla, and V. Hlaváč, "Motion estimation using central panoramic cameras," in *IEEE International Conference on Intelligent Vehicles*, S. Hahn, Ed. Stuttgart, Germany: Causal Productions, October 1998, pp. 335–340.

[3] D. Nistér and H. Stewénius, "A minimal solution to the generalised 3-point pose problem," *J. Math. Imaging Vis.*, vol. 27, no. 1, pp. 67–79, 2007.

[4] H. Stewnius and D. Nistr, "Solutions to minimal generalized relative pose problems," in *In Workshop on Omnidirectional Vision*, 2005.

[5] C. Rodrigo, K. Ankita, and D. Kostas, "Structure from motion with known camera positions," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 477–484.

[6] A. Makadia and K. Daniilidis, "Rotation recovery from spherical images without correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1170–1175, 2006.

[7] F. Kuehnel, "On the minization over so(3) manifolds," Research Institute for Advanced Computer Science (RIACS), USA, Tech. Rep. TR.0312, 2003.

[8] A. Levin and R. Szeliski, "Visual odometry and map correlation." in *CVPR (1)*, 2004, pp. 611–618.

[9] 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*, October 2005, pp. 103 – 108.

[10] 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*, October 2005, pp. 103 – 108.

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

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

[13] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – A modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Verlag, 2000, pp. 298–375. [Online]. Available: citeseer.ist.psu.edu/triggs00bundle.html

[14] S. Ramalingam, S. K. Lodha, and P. Sturm, "A generic structure-from-motion framework," *Comput. Vis. Image Underst.*, vol. 103, no. 3, pp. 218–228, 2006.

[15] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *Int. J. Comput. Vision*, vol. 80, no. 2, pp. 189–210, 2008.

[16] O. Faugeras and Q.-T. Luong, *The geometry of multiple images*, 1st ed. Cambridge University Press, ISBN: 0262062208, 2001.

[17] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto, "Improving feature tracking with robust statistics," *Pattern Analysis and Applications*, vol. 2, pp. 312–320, 1999. [Online]. Available: citeseer.ist.psu.edu/fusiello99improving.html