

Structure from Motion using full spherical panoramic cameras

Alain Pagani and Didier Stricker
German Research Center for Artificial Intelligence
Kaiserslautern - Germany
alain.pagani@dfki.de

Abstract

In this paper, we investigate the use of full spherical panoramic images for Structure from Motion (SFM) algorithms. We introduce several error models for solving the pose problem and the relative pose problem for central projection cameras, and study the effect of these models on the convergence in the algorithm’s optimization steps. We formulate our method using a spherical imaging model, that covers all central projection cameras, including catadioptric and dioptric systems, as well as the standard projective pinhole camera. We use this error formulation in a Structure from Motion pipeline with full spherical panoramic cameras.

1. Introduction

Acquisition of panoramic or full spherical images has become a very common practice for documentation and visualisation of scenes. Applications like Google Street View or Google Art Project [3] deliver millions of spherical views, and (full-)panoramic imaging is not anymore prerogative of professional photographers. In contrast to standard omnidirectional cameras using fisheye lenses or catadioptric system using curved mirrors, spherical imaging is not limited to a hemispherical field of view, but captures a complete scene in a $360^\circ \times 180^\circ$ panorama from a single point in space. There are different ways to acquire such an image. This can be done using a standard camera and specialized hardware like motorized spherical panorama heads (e.g. [18, 16, 19, 23, 1]) and dedicated automatic software (e.g. [22, 4]), or more conveniently by using complete hardware/software packages that automatize all the steps, like Weiss AG’s Civetta Camera [29] or SpheronVR’s SpheroCam [24]. Note that these systems necessitate a precise calibration, but we do not focus on this calibration in this paper. Instead, we consider *calibrated* cameras, and can therefore assume that the image surface is a unit sphere. Interestingly, this representation is valid for all types of cameras where all reflected rays intersect at a single point, known as *cen-*

tral panoramic cameras [27]. Thus our model is valid for a wide class of omnidirectional camera, once the calibration parameters are known.

In this paper, we focus on full-spherical panoramic images in the context of Structure From Motion (SFM), *i.e.* we use spherical cameras to obtain a sparse reconstruction of a 3D scene, together with the pose of the cameras. While SFM has been thoroughly studied for perspective cameras (see *e.g.* [11]) and omnidirectional hemispherical cameras ([15, 26]), little attention has been given to fully spherical cameras. In [28], Torii *et al.* describe geometric aspects of two- and three- view geometry for spherical cameras. Our work extends this study by describing further algorithms required for SFM in spherical cameras. In addition, we present a novel error measure for spherical cameras that is suited for the complete class of calibrated central cameras.

1.1. Related work

The bases of epipolar geometry for different kind of cameras have been studied for example in [30]. In [21], the authors show that image data normalization to unit vectors is necessary for omnidirectional epipolar geometry. Similarly, Kanatani suggests using unit N -vectors as pixels in [12]. Evaluation of error functions for epipolar geometry has been addressed in [5] for *perspective* cameras. For spherical images, [8, 9] presents three types of reprojection errors for epipolar geometry, but all of them are based on rectification of the images to bring the epipoles to a canonic position, operation that can be quite costly.

Structure from Motion has been studied before for hemispherical cameras by different authors [6, 26] but usually the pose estimation problem is not addressed. Epipolar geometry is solved using the algebraic error. In [14], Micusik proposed a calibration procedure for omnidirectional cameras that uses the epipolar geometry. The case of fully panoramic spherical cameras has only been partially addressed by [28], where the basic equations for complete spheres are presented. This work has been extended in [?, ?], where a complete SFM pipeline for dense image sets is presented. [13] proposes to compute Structure from Mo-

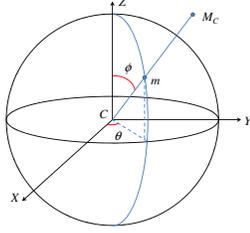


Figure 1. Local camera coordinate system

tion for spherical panoramas by projecting the spherical image onto a panoramic cube, thus casting the spherical problem to a set of perspective cameras. In contrast, we propose Structure from Motion algorithms that use directly spherical images and spherical coordinates, and derive adapted error functions.

In this paper, we focus on the study of different errors approximations for the computation of the epipolar geometry and the pose estimation of calibrated spherical cameras.

The remainder of this paper is organized as follows: section 2 presents our model for a spherical camera, derives the main algorithms used in Structure from Motion for spherical images and presents the different types of errors we propose to study. Section 3 analyzes the proposed error functions in synthetic experiments and section 4 presents results of our algorithm using real images.

2. Spherical camera definition and algorithms

2.1. Definition

In this section, we define the spherical camera model that we use in this paper. We consider the central projection of points in space onto a spherical imaging surface as showed in figure 1. The camera is defined by its center C and by a local Euclidean coordinates system. A 3D point M defined by its Cartesian coordinates \mathbf{M}_C in the coordinate system of the camera is projected onto the sphere's surface by the central projection centered on C to the point $\mathbf{m} = \mathbf{M}_C / \|\mathbf{M}_C\|$. Thus the image points on the sphere are unit 3-dimensional vectors, in the spirit of the N -vectors in [12]. Note that there is no ambiguity about antipodal points: although the ray passing through the center C and the point M crosses the sphere at two antipodal points, the image is uniquely defined by the fact that \mathbf{m} and \mathbf{M}_C must have the same direction, *i.e.* $\mathbf{m}^T \mathbf{M}_C$ is positive. In addition to its Cartesian coordinate system, the spherical camera has a spherical coordinates system. The spherical coordinates (r, θ, ϕ) are related to the Cartesian coordinates (X, Y, Z) by

$$r = \sqrt{X^2 + Y^2 + Z^2}, \theta = \tan^{-1}\left(\frac{Y}{X}\right), \phi = \cos^{-1} \frac{Z}{r}$$

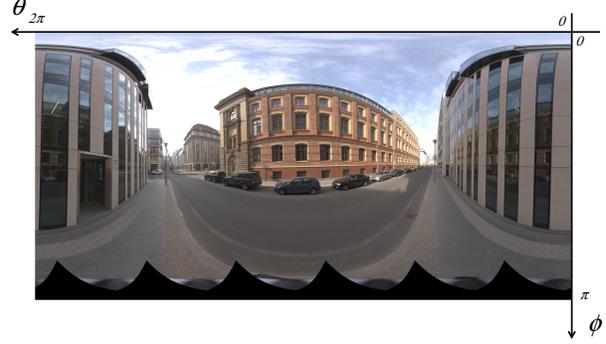


Figure 2. Example of full-panoramic spherical image.

Because the camera is a full panorama, we should ideally have a color information for any value of (θ, ϕ) in $(0, 2\pi) \times (0, \pi)$. In practice, this information is stored in a high resolution *environment map image*, using the latitude-longitude projection, and we use pixel interpolation to get color values for floating (θ, ϕ) values. Figure 2 shows an example of a full panoramic spherical image.

We assume the existence of a global coordinate system (the *world* coordinate system), in which the camera pose (position and orientation) can be defined. If a point M has the Cartesian coordinates \mathbf{M}_W in the world coordinate system and \mathbf{M}_C in the camera coordinate system, then the pose of the camera is defined by the rotation \mathbf{R} and the translation \mathbf{t} such that:

$$\mathbf{M}_C = \mathbf{R}\mathbf{M}_W + \mathbf{t} \quad (1)$$

2.2. SFM algorithms for a full panorama

In this section, we derive the basis algorithms used in the Structure from Motion pipeline for spherical cameras. The main idea is to compute the relative pose between two first cameras using point correspondence and epipolar geometry, then triangulate all matched points to get 3D coordinates (first structure). After this initialization, for each remaining spherical camera, we successively perform following steps: (1) add one new camera in the set, (2) get 2D-3D correspondences from the matches with already triangulated points, (3) compute the pose of the new camera using 2D-3D correspondences and (4) triangulate all possible points from matches between calibrated cameras. After each new camera is added, we use a bundle adjustment step to refine the pose and the structure. For all optimizations in the pipeline, we use a novel error measure that we will detail in the next section.

2.2.1 Point matching for spherical cameras

The algorithms used in SFM require point correspondences between two views as an input. In order to find robust matches despite non-linear transformation between two views, we adopt a matching-by-synthesis approach in the

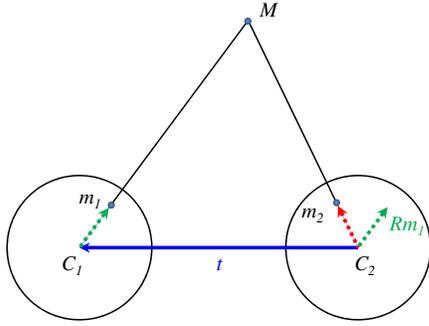


Figure 3. Epipolar geometry for spherical cameras.

spirit of the ASIFT method [17]. Note that because this method uses a local projection model, we can assume a gaussian noise distribution in the angular description of the point.

2.2.2 Relative pose problem

The epipolar geometry of two spherical camera is described in [28]. For completeness, we recall it briefly here. Let C_1 and C_2 be the center of two spherical cameras. Without loss of generality, we can assume that the coordinate system of the first camera is the world coordinate system, *i.e.* $C_1 = \mathbf{0}$, and \mathbf{R} and \mathbf{t} define the position and orientation of the second camera. A point M is projected in \mathbf{m}_1 and \mathbf{m}_2 in the first and second camera respectively (see Figure 3). Ideally, the points M , \mathbf{m}_1 , C_1 , \mathbf{m}_2 and C_2 lie on the same plane — the *epipolar plane* — and this can be expressed by stating that the vectors \mathbf{m}_2 , $\mathbf{R}\mathbf{m}_1$ and \mathbf{t} are coplanar. This is expressed using the epipolar constraint

$$\mathbf{m}_2^T \mathbf{E} \mathbf{m}_1 = 0, \quad \mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (2)$$

where $[\cdot]_{\times}$ is a skew-symmetric matrix expressing the cross-product operation. It is well known that the matrix \mathbf{E} can be computed linearly using eight pairs of correspondences. Because equation 2 uses coplanarity constraints only without considering the orientation of the rays, the factorization of \mathbf{E} into \mathbf{R} and \mathbf{t} generally has 4 solutions [11]. In perspective cameras, the correct solution is found by stating that the depths (z -value) of the 3D points must be positive. For spherical cameras, the same constraint applies, but the positivity of the depth must be verified using the sign of the dot product $\mathbf{m}^T \mathbf{M}_C$.

Geometrical interpretation. If we look at the geometrical aspects of equation 2, we can consider that the vector $\mathbf{n} = [\mathbf{t}]_{\times} \mathbf{R}\mathbf{m}_1$ is the normal of the plane defined by the vectors \mathbf{t} and $\mathbf{R}\mathbf{m}_1$ (see Figure 4). Equation 2 can be reformulated

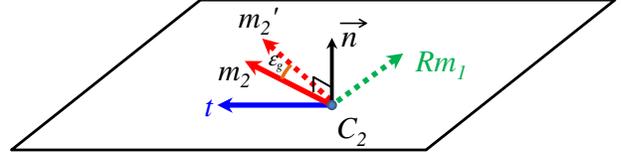


Figure 4. Geometrical interpretation of the epipolar error.

ulated $\mathbf{m}_2^T \mathbf{n} = 0$, which expresses the fact that \mathbf{m}_2 should be perpendicular to \mathbf{n} . In presence of noise however, this is not exactly the case. If we call \mathbf{m}_2' the noisy measurement of \mathbf{m}_2 , then $\mathbf{m}_2'^T \mathbf{n}$ is the cosine of the angle between \mathbf{m}_2' and \mathbf{n} , or equivalently the sine of the angular error ϵ_g between the plane and the image point. We therefore use the angular error for the epipolar constraint [?]:

$$\epsilon_g = \sin^{-1}(\mathbf{m}_2'^T \mathbf{E} \mathbf{m}_1) \quad (3)$$

ϵ_g is the error we will consider for SFM, in particular for non linear optimization algorithms. ϵ stands for epipolar error, and g for *geodesic*, as the angular error is equivalent to the geodesic distance between the plane and the point on the surface of the sphere. Note that the formulation is valid if \mathbf{m}_1 , \mathbf{m}_2 and \mathbf{n} are unit vectors, *i.e.* if \mathbf{E} is parametrized using a unit length for the vector \mathbf{t} . The advantage of this error is that it is defined on the sphere surface (or equivalently in terms of angular error) instead of on an image plane. Angular errors are more generic than epipolar distance, because it can be applied for any central camera independently of the projection geometry, and avoids the computation of epipolar curves (line for perspective cameras, great circles for spherical images) in the images.

Note on normalization. In the perspective case, it is recommended to normalize the point coordinates to improve stability. This normalization is known as *conditioning* [11], where the points are transformed such that the mean is at origin and their average norm is $\sqrt{2}$. For spherical images however, this step is not necessary, as all the points have unit length, and are usually spread around the center of the camera.

Symmetry in error measures. When measuring the error in image space, the residuals are usually made symmetric by adding the distances in both images. For example, the symmetric squared geometric error in perspective images is defined by:

$$\frac{(\mathbf{m}_2^T \mathbf{E} \mathbf{m}_1)^2}{[\mathbf{E} \mathbf{m}_1]_1^2 + [\mathbf{E} \mathbf{m}_1]_2^2} + \frac{(\mathbf{m}_1^T \mathbf{E} \mathbf{m}_2)^2}{[\mathbf{E}^T \mathbf{m}_2]_1^2 + [\mathbf{E}^T \mathbf{m}_2]_2^2}, \quad (4)$$

where $[\cdot]_i$ denotes the i th element of a vector. In our case, the error ϵ_g has been defined using image 2, but the computed value would be the same in image 1. Indeed, the value computed in equation 2 does not change if a global

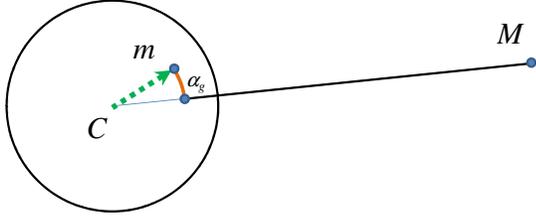


Figure 5. Geometrical interpretation of the reprojection error.

rotation of all vectors by \mathbf{R}^{-1} is applied. By virtue of the triple product, we then have:

$$\begin{aligned} \mathbf{m}_2^T [\mathbf{t}]_{\times} \mathbf{R} \mathbf{m}_1 &= (\mathbf{R}^{-1} \mathbf{m}_2)^T [\mathbf{R}^{-1} \mathbf{t}]_{\times} \mathbf{m}_1 \\ &= \mathbf{m}_1^T [-\mathbf{R}^{-1} \mathbf{t}]_{\times} \mathbf{R}^{-1} \mathbf{m}_2, \end{aligned}$$

which is the expression of the error in the first image.

2.2.3 Pose problem

The pose problem — or *Perspective-n-point* (PnP) problem — can be defined as the problem of finding a camera's position and orientation from 2D-3D correspondences. We assume that a number of points with known (world) 3D coordinates \mathbf{M}_W can be identified in an image as 2D point \mathbf{m} , and the task of the PnP problem is to find the transformation parameters \mathbf{R} and \mathbf{t} that minimize the reprojection error. Different solutions to this problem have been proposed [20], and for the spherical camera, we use the Direct Linear Transform (DLT) [2] as first estimate of the pose, followed by non-linear refinement with a novel error measure. DLT states that the image point \mathbf{m} is collinear to the 3D point in the camera coordinate system, so that the cross product between \mathbf{m} and $\mathbf{R} \mathbf{M}_W + \mathbf{t}$ is zero. Writing this cross product delivers three equations in the elements of \mathbf{R} and \mathbf{t} , among which two only are linearly independent. Six identified points are therefore necessary to solve for the pose. After the pose has been computed, the orthogonality of \mathbf{R} can be enforced using SVD.

Geometrical interpretation of the reprojection error.

The linear method described so far delivers only a first estimate of the pose of the camera. This step should be followed by a non-linear least-squares optimization of the pose. Here again, we define the geodesic reprojection error as the angular error between the point \mathbf{m} and the reprojection of the point \mathbf{M} on the sphere (see Figure 5). The geodesic error is defined by [?]:

$$\alpha_g = \cos^{-1} \left(\frac{\mathbf{m}^T \mathbf{M}}{\|\mathbf{M}\|} \right) \quad (5)$$

α_g is suited for spherical images and for calibrated central cameras and can be used in non linear optimization as well as for distance computation in thresholded outlier rejection schemes.

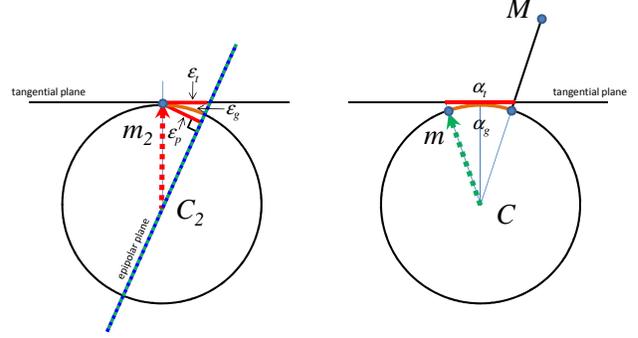


Figure 6. Tangential approximations of the errors.

2.3. Error approximations

So far we have presented the basic building blocks needed in Structure from Motion in full panoramic spherical images. We have derived two exact geodesic error formulations for the relative pose problem and the perspective-n-point problem. These errors require however the computation of inverse trigonometric function for each correspondence. In the case of high-resolution spherical images, we usually get around 7.000 correspondences for an image pair, and the computational cost can be critical. We therefore introduce different approximations of these errors (see Figure 6).

Distance of ray to epipolar plane The sine of the geodesic distance is the perpendicular distance from the epipolar plane to the image point \mathbf{m}_2 . So taking the absolute value of the algebraic distance defined in equation 2 amounts to approximating the geodesic distance by the perpendicular projection distance. This is valid if image points are unit vectors, so we define the *projected distance* ϵ_p as:

$$\epsilon_p = \frac{|\mathbf{m}_2^T \mathbf{E} \mathbf{m}_1|}{\|\mathbf{m}_2\| \|\mathbf{E} \mathbf{m}_1\|} \quad (6)$$

The left part of Figure 6 shows the distance ϵ_p for a quite big angle ϵ_g for visualization purposes, but in practice the angle is much smaller, so that the local projection is a good approximation of the error.

Tangential error for epipolar distance. For the computation of the error in the epipolar distance, we project the geodesic error ϵ_g *locally* on a plane tangential to the sphere on the point \mathbf{m}_2 . The resulting error is $\epsilon_t = \tan(\epsilon_g)$, which turns out to have the mathematical form:

$$\epsilon_t = \frac{\epsilon_p}{\sqrt{1 - \epsilon_p^2}} \quad (7)$$

Because we usually compute the square of that error, the overall computational cost amounts to one squaring, one subtraction and one division.

Tangential error for reprojection distance. When reprojecting a 3D point on the sphere surface, we end up with two points on the sphere. In this case, we consider the tangential plane tangent on a point that lies exactly between the image point and the reprojection of the 3D point (see Figure 6, right). The resulting error is $\alpha_t = 2 \tan(\alpha_g/2)$, that is:

$$\alpha_t = 2 \sqrt{\frac{1 - \frac{\mathbf{m}^T \mathbf{M}}{\|\mathbf{M}\|}}{1 + \frac{\mathbf{m}^T \mathbf{M}}{\|\mathbf{M}\|}}} \quad (8)$$

Here again, the computational load is kept small, as only one addition, one subtraction and one division is required to compute the square of the error.

In the next section, we evaluate the errors defined above in synthetic experiments.

3. Minimization using error approximations

So far we have derived the errors for full panoramic spherical images, and proposed new approximations for these errors. We now evaluate the validity of these approximations in different experiments.

3.1. Choice of the epipolar distance

In a first experiment, we tested several epipolar distances in a synthetic scenario. Given two spherical images with known positions, we randomly generated n 3D points around the two cameras. The 3D points are then projected in the images to get the coordinates of the 2D points. We assume a perfect matching of points, so each 3D point leads to a pair of corresponding points in the images. We then add a given amount of noise in the spherical coordinates by adding a small value τ to ϕ and θ sampled from a uniform distribution in $-\sigma, +\sigma$. We then compute the epipolar matrix \mathbf{E} using the 8 point algorithm. This first estimate is then optimized by a Levenberg–Marquardt error minimization. We compare the results obtained for different error measures in the optimization step: the geodesic distance ϵ_g of equation 3, the perpendicular projection ϵ_p of equation 6, the tangential distance ϵ_t of equation 7, and the Sampson distance usually recommended [11] for optimizing the epipolar geometry in perspective images:

$$\epsilon_s = \frac{(\mathbf{m}_2^T \mathbf{E} \mathbf{m}_1)^2}{[\mathbf{E} \mathbf{m}_1]_1^2 + [\mathbf{E} \mathbf{m}_1]_2^2 + [\mathbf{E}^T \mathbf{m}_2]_1^2 + [\mathbf{E}^T \mathbf{m}_2]_2^2}, \quad (9)$$

with $[\cdot]_i$ defined as in equation 4. After convergence, we measure the geodesic distance ϵ_g between the corresponding points. We repeat this experiment T times and plot the mean error for different values of σ . Figure 7 shows the results for a number of points $n=200$, $T=1000$, and different values of σ varying between 0.1 and 2 degrees.

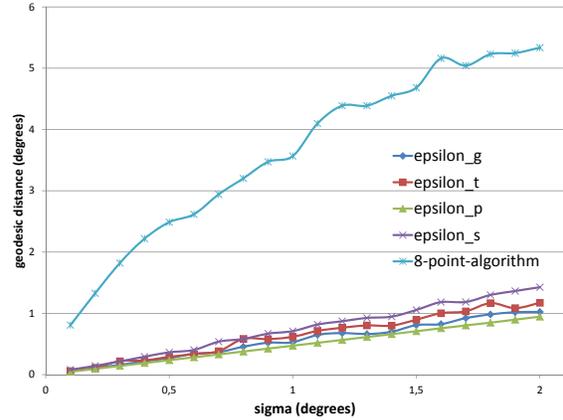


Figure 7. Residual error after optimization of the epipolar matrix

The result show that the 8 point algorithm alone is not sufficient to compute a reliable epipolar matrix. When minimizing the reprojection error, our experiment show that the Sampson distance is not adequate for the spherical geometry, as it is constantly the worse result. A possible reason is that the Sampson distance is a first order approximation of the geometric distance in the case of perspective cameras, *i.e.* the error that is actually minimized is measured in the image plane. The results also show that the tangential distance ϵ_t is less good than the geodesic distance ϵ_g . Surprisingly, the perpendicular distance to the epipolar plane ϵ_p seems to provide better results than the minimization using the geodesic distance. This is probably due to better convergence properties with this distance. We therefore recommend using ϵ_p as distance to minimize for spherical cameras.

3.2. Choice of the point to point distance

We now conduct a similar experiment for comparing distances in the computation of the pose from 2D-3D correspondences. Similarly, we randomly generate n 3D points around one spherical image with known position. The 3D points are then projected into the image to get the coordinates of the 2D points, to which we and add a given amount of noise as in the previous experiment. We then compute the pose of the camera with the DLT algorithm, and optimize this result by a Levenberg–Marquardt error minimization. The error measures we compare are the following: the geodesic distance α_g of equation 5, the tangential distance α_t of equation 8, and the Euclidean distance between the 2D point \mathbf{m} and the projection of the 3D point \mathbf{M} :

$$\alpha_e = \left\| \mathbf{m} - \frac{\mathbf{M}}{\|\mathbf{M}\|} \right\| \quad (10)$$

After convergence, we measure the geodesic distance

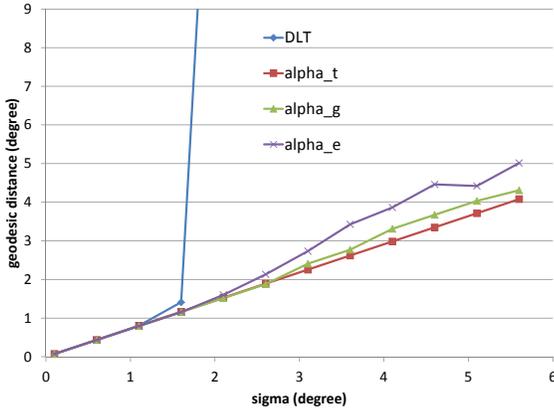


Figure 8. Residual error after optimization of camera pose

α_g between the 2D point and the reprojection of the corresponding 3D point. We repeat this experiment T times and plot the mean error for different values of σ . Figure 8 shows the results for a number of points $n=200$, $T=1000$, and different values of σ varying between 0.1 and 6 degrees.

The result show that the DLT algorithm is stable until a value of $\sigma = 1.5$ degrees, then rapidly diverges and is not reliable anymore. However, the different minimization strategies recover a correct pose from the erroneous DLT result. If we compare the result of optimization using different distances, we see that using the euclidean distance between points α_e provides poor results for large values of σ . Using the geodesic distance α_g produces better results, but the tangential distance α_t constantly provides the best convergence in the optimization. Here again, a possible explanation is that this distance is more suited for finding the global optimum. We therefore recommend using α_t as distance to minimize when solving the PnP problem in spherical cameras.

4. 3D Reconstruction with spherical cameras

In order to illustrate the use of our structure from motion algorithms for full spherical panoramic cameras, we now present results obtained with images of an urban environment.

4.1. Structure from motion pipeline

We follow a classical approach for the SFM pipeline: we start by detecting salient points in all the images using the spherical variant of ASIFT described in section 2.2.1. For the matching process, we can automatically compute the matches for *all* pairs of cameras, or for a subset of pairs using user-provided prior information on neighborhood between cameras. We then choose two cameras as initial cameras, and solve the epipolar geometry for this pair. This is

done in two steps: first the epipolar matrix \mathbf{E} is computed linearly using the eight-point algorithm that minimizes the algebraic error of equation 2. Outliers are rejected using RANSAC [7] and the perpendicular error ϵ_p as defined in equation 6. We then factorize the matrix \mathbf{E} in \mathbf{R} and \mathbf{t} and set a unit length for the translation vector. This allows for a parametrization of \mathbf{E} using 5 parameters. Using this parametrization, we optimize the epipolar error using ϵ_p in the Levenberg–Marquardt optimization algorithm. When the optimal pose is found, we can triangulate the correspondences to find the 3D position of the points.

After this initialization, we successively perform following steps: (1) Among the non-calibrated cameras, select the one that shares the most matches with already calibrated cameras. (2) get 2D-3D correspondences from the matches with already triangulated points. (3) compute a first estimate of the pose of the new camera linearly using the DLT algorithm, using α_t as defined in equation 8 for RANSAC–based outlier rejection. (4) optimize this pose using α_t in the Levenberg–Marquardt optimization algorithm. (4) triangulate all possible points from matches between calibrated cameras. (5) Apply a sparse bundle adjustment technique that globally minimizes the reprojection error for all already calibrated cameras and all 3D points.

The novelty of our approach resides in the usage of full panoramic spherical cameras together with SFM techniques, and in the usage of novel error formulations for the optimization steps.

The advantage of using a full panoramic spherical camera is that they have the ability to see extremely large parts of the surroundings, as noted by [15] and [25].

4.2. Results

We tested our methods in a real scenario by acquiring 35 spherical images of a urban environment. The acquisition time for 35 images is approximately one hour. We find approximately 50.000 affine SIFT points on each image. After pairwise matching, we find up to 7000 matches for each image pair. Figure 9 top (left and right) shows the distribution of the 38.287 3D points found after the SFM step, as well as the position of the cameras on the floor.

Once all cameras are calibrated and a coarse structure has been found, we compute a dense reconstruction based on a Multiple View Stereo (MVS) method of Furukawa and Ponce [10]. This method was originally formulated to generate dense point clouds for calibrated perspective images, and we adapted it for spherical views. The resulting point cloud contains over 50 millions points. Figure 9 bottom shows several views of the reconstructed point cloud. Note that the output of our algorithm is a dense point cloud, not a surface. However, in many regions, the density of the points is such that the point cloud appears as a continuous surface.

The quality that we obtain for the dense point cloud as

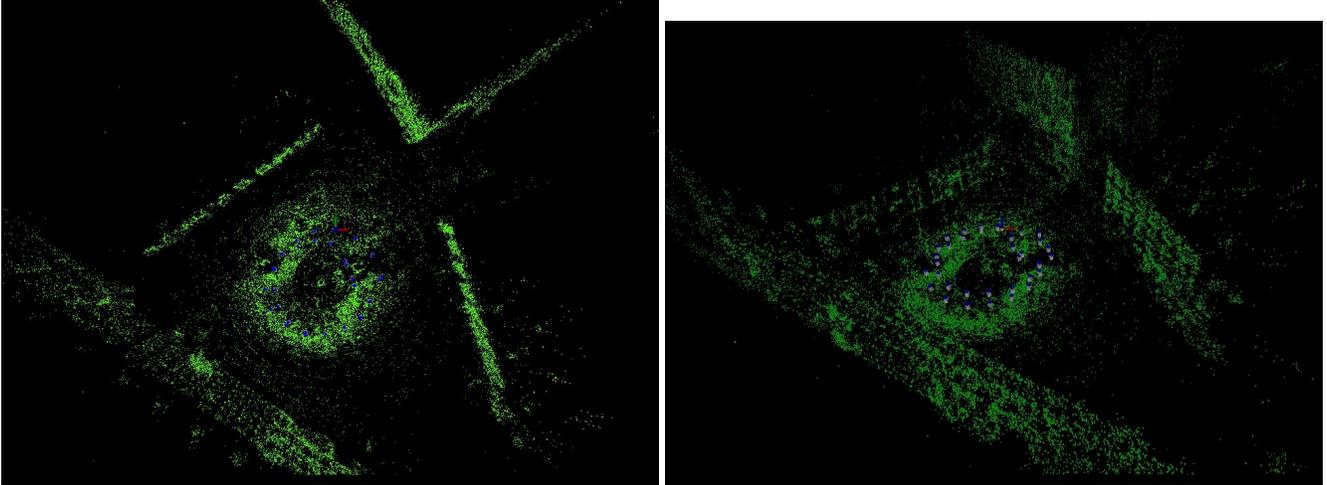


Figure 9. 3D reconstruction of a city place. Top: results of the structure from motion method. Middle and bottom: Dense point cloud generated using multiple view stereo with the calibrated cameras as input.

assesses the precision of the calibration step we propose in Structure from Motion. The resulting reconstruction can be

visualized in a video provided as supplementary material.

5. Conclusion

In this paper, we presented different algorithms for Structure from Motion estimation from full panoramic spherical images. After the derivation of the solutions to the pose problem and the relative pose problem, we showed what are the exact geodesic errors to consider in these solutions, and proposed several approximations of these errors. We showed that the perpendicular distance to the epipolar plane is the best error for epipolar geometry and that our novel tangential error for the point-to-point distance is the best one for solving the PnP problem (pose from 2D-3D correspondences). We validated these results in a practical scenarios where we reconstructed a large scene in an urban environment using spherical images.

Acknowledgment This work has been partially funded by the project CAPTURE (01IW09001).

References

- [1] 360 Precision. www.360precision.com. 1
- [2] Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proc. Symposium on Close-Range Photogrammetry*, pages 1–18, 1971. 4
- [3] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, June 2010. 1
- [4] AutoPano. autopano.kolor.com. 1
- [5] M. Brckner, F. Bajramovic, and J. Denzler. Experimental evaluation of relative pose estimation algorithms. In *VIS-APP*, 2008. 1
- [6] P. Chang and M. Hebert. Omni-directional structure from motion. In *Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS)*, 2000. 1
- [7] M. A. Fishler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comp. Mach.*, 24(6):381–395, 1981. 6
- [8] J. Fujiki. Three types of reprojection error on spherical epipolar geometry. In *Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS)*, 2008. 1
- [9] J. Fujiki, A. Torii, and S. Akaho. Epipolar geometry via rectification of spherical images. In *MIRAGE*, 2007. 1
- [10] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *PAMI*, 1:1–14, 2008. 6
- [11] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000. 1, 3, 5
- [12] K. Kanatani. *Geometric computation for machine vision*. Oxford University Press, Inc., New York, NY, USA, 1993. 1, 2
- [13] F. Kangni and R. Laganire. Orientation and pose recovery from spherical panoramas. In *OMNIVIS*, 2007. 1
- [14] B. Micusik and T. Pajdla. Estimation of omnidirectional camera model from epipolar geometry. In *CVPR*, 2003. 1
- [15] B. Micusik and T. Pajdla. Structure from motion with wide circular field of view cameras. *PAMI*, 28:1135 – 1149, 2006. 1, 6
- [16] MK Panorama Systeme. www.vr-head.com. 1
- [17] J. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2:438–469, 2009. 3
- [18] Nodal Ninja Panoramic Tripod Heads. www.nodalninja.com. 1
- [19] Novoflex. www.novoflex.com. 1
- [20] A. Pagani and D. Stricker. A comparative study of seven solutions to the p-n-p problem. In *Proc. ORASIS*, 2003. 4
- [21] T. Pajdla, T. Svoboda, and V. Hlavac. Epipolar geometry of central panoramic cameras. *Panoramic Vision: Sensors, Theory, and Applications*, 2001. 1
- [22] PTgui Pro. www.ptgui.com. 1
- [23] Seitz Roundshot. www.roundshot.ch. 1
- [24] Spheron VR SpheroCam HDR. www.spheron.com/en/intruvision/solutions/spheroCam-hdr.html. 1
- [25] B. Streckel and R. Koch. Lens model selection for visual tracking. In *Annual Symposium of the German Association for Pattern Recognition (DAGM)*, 2005. 6
- [26] T. Svoboda and T. Pajdla. Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, 49:23–37, 2002. 1
- [27] T. Svoboda, T. Pajdla, and V. Hlavac. Epipolar geometry for panoramic cameras. In *Proc. of ECCV*, 1998. 1
- [28] A. Torii, A. Imiya, and N. Ohnishi. Two- and three- view geometry for spherical cameras. In *OMNIVIS*, 2005. 1, 3
- [29] Weiss AG Civetta camera. www.weiss-ag.org/business-divisions/360-technology. 1
- [30] Z. Zhang. Epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27:161–198, 1998. 1