A Generalized Structure from Motion Framework for Central Projection Cameras

Christiano Couto Gava and Didier Stricker

German Research Center for Artificial Intelligence, Department Augmented Vision,
Trippstadterstr. 122, 67663 Kaiserslautern, Germany
Kaiserslautern University of Technology, Department of Computer Science
Gottlieb-Daimlerstr., 67663 Kaiserslautern, Germany
{Christiano.Gava,Didier.Stricker}@dfki.de
http://av.dfki.de/en/,http://www.informatik.uni-kl.de/en/

Abstract. This paper presents a novel Structure from Motion (SfM) framework designed for central projection cameras. The goal is to support future large scale multi-view 3D reconstruction algorithms. We believe that these algorithms will be able to benefit from several different sources of visual information. Accordingly, SfM approaches will need to handle this variety of image sources, such as perspective, wide-angle and spherical images. However, this issue has not yet been addressed. Current state of the art techniques are not able to handle heterogeneous images simultaneously. Therefore, we introduce SPHERA, a generalized SfM framework designed for central projection cameras. By adopting the unit sphere as underlying model it is possible to treat single effective viewpoint cameras in a unified way. We validate our framework on synthetic and real datasets. Results show that SPHERA is a powerful framework to support upcoming algorithms and applications on large scale 3D reconstruction.

Keywords: Structure from motion, spherical images, multi-view 3D reconstruction, large scale

1 Introduction

The popularity of full panoramic images has significantly increased during the past few years. This is confirmed by the growing variety of spherical image acquisition hardware and software packages available nowadays [1–5]. Mobile devices such as smartphones and tablets feature easy-to-use Apps that allow the user to capture panoramas within seconds. Additionally, panoramic images offer the possibility to create immersive environments where the user experiences a first-person view, such as Google Street View [6]. Immersive visualization systems find appliance in a number of applications, e.g. documentation, education, preservation of cultural heritage, gaming, city planing, etc. Clearly, these applications can further benefit from 3D information. This makes full spherical images specially attractive for immersive visualization as well as 3D reconstruction.

There are several ways to classify multi-view 3D reconstruction algorithms. One of them concerns the distance between the images relative to the scene, the so called baseline. Recent *narrow* baseline approaches are capable of simultaneously recovering camera poses and 3D geometry from a video sequence [7, 8]. However, these approaches are normally restricted to indoor, office-like, environments. *Wide* baseline techniques, on the other hand, are better suited for large scale reconstruction, but assume camera poses have been previously determined [9, 10]. In other words, they implicitly demand Structure from Motion (SfM) to recover the camera poses before the 3D model can be computed.

To perform SfM, spherical images are more suitable than standard perspective images. Due to their wide field of view, scene features are observed in more images, thus increasing the number of constraints on camera poses. Consequently, methods have been derived to perform SfM on wide field of view cameras. More specifically, [11–13] address SfM on omnidirectional images, while [14–16] deal with full spherical images. Not surprisingly, perspective SfM has been extensively studied e.g. by [17–23]. Although these approaches have shown to work well for the specific image type they were designed for, up to the authors knowledge they are unable to handle images of any other type.

Another relevant aspect of SfM algorithms is whether the camera poses are estimated globaly or incrementaly. Usually, global methods split the camera pose estimation into two parts. The first part aims at recovering the rotation matrices of all cameras. The second part uses the global rotations obtained in the first part to determine the translation of all cameras. The later may be performed independently of the scene structure [20] or along with it [21]. The main reason for this splitting is that the estimation of relative translations is inaccurate in case of narrow baseline, whereas relative rotations can be precisely recovered regardless of the baseline, provided enough point correspondences. Global methods have the advantages of evenly distributing errors among all cameras and being independent of an initial pair of cameras. They traditionally solve a linear system of equations (which minimize an *algebraic* error), combined or followed by bundle adjustment (BA) [24] to refine camera poses. However, if a new camera is added afterwards, the entire pipeline has to be executed again.

Incremenal methods are initialized by computing the poses of a selected camera pair. Then, point correspondences are triangulated and the resulting 3D points are used to select the next camera. Once the pose of the new camera is determined, new 3D points are created and the procedure is repeated. In other words, the poses of all cameras along with a sparse representation of the scene structure are recovered by alternating between triangulation and resectioning. An advantage of these methods is the possibility to obtain the optimal pose every time a new camera is added. This happens because each pose is estimated using BA which minimizes the reprojection error, carrying along a meaningful *geometric* interpretation, instead of an algebraic error. Another advantage of incremental methods is the ability to later include new cameras without necessarily rerunning the entire pipeline. Incremenal pipelines may suffer from drift caused by accumulated errors. Consequently, loop closure may become an issue.

Nevertheless, it has been shown in [22] that re-triangulation of existing point correspondences is able to redistribute accumulated errors as well as deal with loop closures. Moreover, one of the most successful SfM algorithms is Bundler [17], which implements an incremental pipeline.

Given the current effort to reconstruct ever growing environments [10, 25, 26], every source of visual information shall be taken into account, regardless of the shape of image surface. This is an issue that has not yet been addressed. Apart from performance and accuracy, another highly desirable feature of 3D reconstruction algorithms is to update and improve the scene model whenever new images are available. Here again, the ability to deal with different camera types is essential. Therefore, we present SPHERA, a novel Structure from Motion framework to bridge the gaps between current SfM methods for central projection cameras. We build on the model proposed in [27] and adopt the unit sphere to represent images and to treat heterogeneous camera types in a unified way. Our approach dynamically selects the best information available to recover camera poses and scene structure, allowing new images to be integrated efficiently. Experiments on synthetic and real image sequences validate our framework as a valuable contribution to support large scale 3D reconstruction algorithms.

1.1 Related Work

The work presented in [11] uses epipolar geometry to compute scene structure from an omnidirectional vision system mounted on a robot platform. However, the camera pose problem is not addressed. In [12], Micusik and Pajdla focus on omnidirectional images with a field of view larger than 180^o and devise a camera model specific for that type of image. Although scene structure can be recovered, the technique is limited to the two-view geometry problem. Consequently, the proposed camera model can hardly be used in a more generic SfM approach. Bagnato et al. present in [13] a variational approach to achieve egomotion estimation and 3D reconstruction from omnidirectional image sequences. Nonetheless, the environment must be densely sampled so that the relationship between image derivatives and 3D motion parameters is still valid. Thus, this approach can not be used in a more general, sparse SfM pipeline.

A method to recover camera poses from a set of spherical images on a sparsely sampled environment is presented in [14]. However, SfM is performed based on panoramic cubes computed for each spherical image. The camera poses are recovered by casting the spherical problem back to the standard perspective problem. In [16], spherical images are used to estimate the relative camera poses and to build a map of the environment. To simplify the problem, Aly and Bouguet assume planar motion, i.e. all camera frames must lie on the same plane. This assumption strongly limits the applicability of the proposed technique. Our approach is closely related to [15], as both exploit full spherical images to deliver a sparse representation of the scene along with recovered camera poses. Nevertheless, the method presented by Pagani and Stricker was designed exclusively for spherical cameras, whereas our framework naturally handles any kind of central projection camera. Additionally, SPHERA allows to dynamically select a subset of the cameras to optimize and speed up BA with little to no loss of accuracy, as detailed in Section 3.

Not surprisingly, our pipeline has similarities with some SfM methods derived exclusively for perspective images. For instance, Wu proposes an incremental SfM where loop closure does not need to be explicitly detected [22]. His algorithm tracks *under-reconstructed* camera pairs, i.e. pairs with low ratio between their common 3D points and number of point correspondences. Then, based on a geometric sequence, re-triangulation is performed for all under-reconstructed camera pairs. Wu shows that this re-triangulation is able to reduce drift errors without explicitly detecting loops even for long image sequences. Our framework incorporates this idea. However, as we aim at high accuracy, re-triangulation is performed during every step of BA, instead of follwoing a predefined sequence. Another incremental method has been proposed in [23]. The authors introduce an *algebraic* cost function formulated on pairwise epipolar constraints as a more efficient alternative to the traditional reprojection error. Their algorithm eliminates structure from BA aiming at speeding up convergence. Nevertheless, their final solution lacks the accuracy of geometric based cost functions. Therefore, their pipeline requires two or three additional iterations of the classical BA (which takes the structure back into account) at the end to improve precision. As described in Section 3, we also consider only the camera parameters for BA to reduce the dimension of the parameter search space. Contrary to [23], we implicitly model scene structure through the re-triangulation mentioned above. Moreover, instead of an algebraic error, SPHERA minimizes a reprojection error defined directly on the surface of the unit sphere.

2 Background

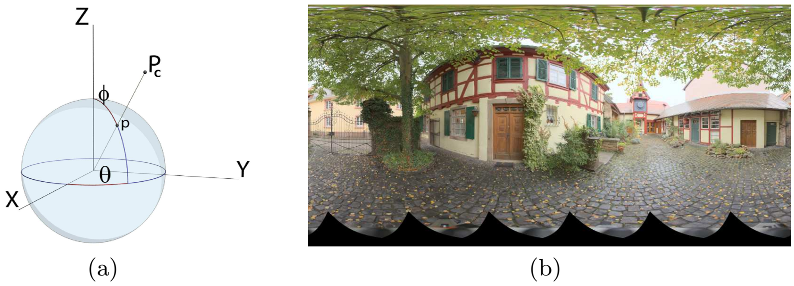
2.1 Spherical Images

A spherical image is a $180^o \times 360^o$ environment mapping that allows an entire scene to be captured from a single point in space. Consequently, every visible 3D point P_W given in world coordinate system can be mapped onto the image surface. This is done by a two-step process. First, analogue to the perspective case, P_W is represented in the camera coordinate system as $P_C = RP_W + t$, with R and t representing the camera rotation matrix and translation vector. Second, and different from the perspective projection, P_C is projected onto the image surface by scaling its coordinates, as shown in Fig. 1-(a). Without loss of generality, we assume a unit sphere. Thus, the scaling becomes a normalization and $p = P_C / \|P_C\|$.

Spherical images are stored as a 2D pixel-map as depicted in Fig. 1-(b). This map is obtained using a latitude-longitude transformation, with $0 \leq \phi \leq \pi$ and $0 \leq \theta \leq 2\pi$.

2.2 Sphere as Unifying Model

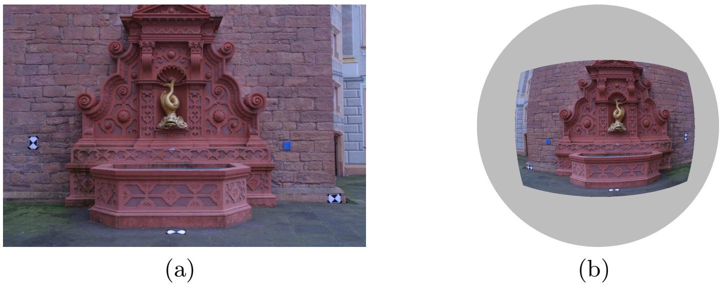
Our approach is grounded on the seminal work developed in [27], where the authors proposed a unifying model for the projective geometry of vision sys-



(a) (b)

Fig. 1. (a) Spherical coordinates and illustration of the spherical projection. (b) Pixel-map of a spherical image.

tems having a single effective viewpoint. These vision systems are commonly referred to as central projection cameras and include catadioptric sensors featuring conic mirrors of different shapes, such as parabolic, hyperbolic or elliptic. Geyer and Daniilidis showed that any central catadioptric projection is equivalent to a two-step mapping via the sphere. It is well known from the pinhole model that standard perspective imaging characterizes a single viewpoint system. Nonetheless, perspective images are also central catadioptric systems with a virtual planar mirror and are, therefore, covered by the aforementioned model. In practice, that means it is possible to treat these central projection systems as spherical cameras, provided the mapping from the original image surface to the sphere is known. This mapping may be seen as a warping transformation from the original image to the unit sphere. As an example, Fig. 2 shows the result of warping a perspective image onto the sphere.



(a) (b)

Fig. 2. Example of (a) an original perspective image [28] and (b) its warped version. The warped image appears mirrored due to the viewpoint ("outside" the unit sphere).

2.3 Spherical Camera Pose Estimation

Epipolar Geometry The epipolar geometry for full spherical cameras has already been presented in [29]. Thus, here we provide a short overview. Consider a pair of spherical cameras C_0 and C_1. Let R and t be the associated rotation matrix and translation vector. A point p_0 on the surface of C_0, along with the centers of the cameras, define a plane Π that may be expressed by its normal vector $n_\Pi = Rp_0 \times t = [t]_\times Rp_0$, where $[t]_\times$ is the skew-symmetric matrix representing the cross-product. For any point p_1 on C_1 belonging to Π the condition $p_1^T n_\Pi = 0$ holds, which is equivalent to $p_1^T [t]_\times Rp_0 = 0$, where $E = [t]_\times R$ is the essential matrix [18]. The condition $p_1^T E p_0 = 0$ is known as the epipolar constraint and is the same result obtained in the perspective case. This shows that the epipolar constraint is independent of the shape of the image surface.

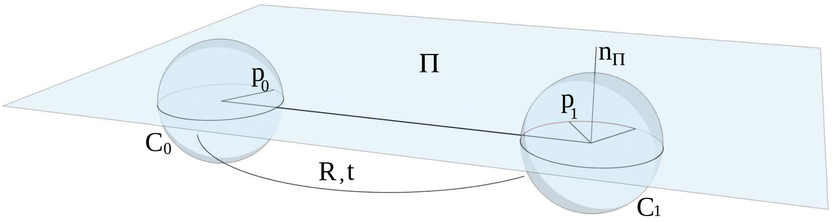


Fig. 3. Epipolar geometry for two spherical images.

Pose Estimation There are mainly two techniques for computing camera poses. The first is useful for *relative* pairwise pose estimation, typically when only 2D image correspondences (2D-2D correspondences) are available. Without loss of generality, one of the cameras is assumed as reference and R and t represent the pose of the second camera. In this case, R and t may be determined with e.g. the 5-point algorithm [30]. The second technique is normally used when a number of 3D scene points and their respective projections onto an image are known, i.e. a set of 2D-3D correspondences is available. This configures a *Perspective-n-point* (PnP) problem, which can be solved with a minimum of 6 correspondences [31].

3 The Proposed Approach

Given a set of images of a scene, our goal is to accurately estimate the pose of all cameras as well as to recover a sparse 3D point cloud of the underlying scene representing its geometry. The set of central projection cameras is defined as

$$\mathcal{C} = \left\{ C_j = \left[\hat{R}_j | \hat{t}_j \right] \mid \hat{R}_j \in SO\left(3\right), \ \hat{t}_j \in \mathbb{R}^3 \right\}, \tag{1}$$

where $j = 0, .., M-1$, M is the total number of cameras and $\hat{R}_j$ and $\hat{t}_j$ are the rotation matrix and translation vector representing the estimated pose of camera C_j. To aid the non-linear optmization, we adopt an axis-angle parameterization for the rotation matrix and C_j is then parameterized by a vector $\rho_j \in \mathbb{R}^6$. All together, the cameras are parameterized by a vector $\rho \in \mathbb{R}^m$, with $m = 6M$.

Likewise, we denote the set of sparse 3D points reconstructed along with the camera poses as

$$\mathcal{P} = \left\{ \hat{P}_i \in \mathbb{R}^3 \right\}, \tag{2}$$

where $i = 0, .., N-1$, N is the number of points and $\hat{P}_i$ holds the estimated coordinates of a scene point P_i.

We then formulate the problem of recovering all camera poses along with a sparse point representation of the scene as a non-linear optimization problem. The parameter vector ρ is optimized in order to minimize

$$\min_{\rho} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} e_{ij}\left(\rho\right), \tag{3}$$

where $e_{ij}\left(\rho\right)$ is a cost function for each point $\hat{P}_i$ and camera C_j. The parameters ρ^+ that minimize Eq. 3 are the sought camera poses. Note that this optimization depends exclusively on the camera parameters ρ. In a classical BA scenario, for N 3D points and M cameras, a total of $3N + 6M$ parameters have to be optimized. Different from the classical BA, we reduce the complexity of the problem by dropping the structure and considering only the camera parameters. This leads to an important advantage: the dimension of the parameter search space is at most $6M$, a significant reduction compared to $3N + 6M$, what is particularly convenient in case of large scale scenes. Nonetheless, structure is jointly estimated. Inspired by [22], point correspondences are re-triangulated at every step of our BA, updating the structure with the most recent camera parameters and reducing drift due to accumulated errors.

3.1 Reprojection Error and Visibility Map Models

The cost function $e_{ij}\left(\rho\right)$ in Eq. 3 represents the reprojection error of a point $\hat{P}_i$ on camera C_j and is defined as

$$e_{ij}\left(\rho\right) = cos^{-1}\left(p_{ij}\hat{p}_{ij}\right), \tag{4}$$

where $p_{ij}\hat{p}_{ij}$ is the scalar product between the expected projection p_{ij} and the measured projection $\hat{p}_{ij}$ obtained with $\hat{P}_i$, $\hat{R}_j$ and $\hat{t}_j$. The expected projection p_{ij} is determined by the keypoint location corresponding to P_i. Note that as $-1 \leq p_{ij}\hat{p}_{ij} \leq 1$, we have $0 \leq e_{ij}\left(\rho\right) \leq \pi$ and it is not necessary to take the absolute value in Eq. 4. Furthermore, we do not use any approximation of the reprojection error as in [15]. As we aim at high accuracy, the error defined in Eq. 4 is the exact geodesic distance, i.e. the exact angular deviation, between p_{ij} and $\hat{p}_{ij}$. Additionally, to each point P_i we associate a visibility map

$$\mathcal{V}_i = \left\{ \left(C_j, p_{ij}\right) \mid C_j \in \mathcal{C},\ p_{ij} \in \mathcal{S}^2 \right\}, \tag{5}$$

where $\mathcal{S}^2$ represents the unit sphere. We denote the pair $\left(C_j, p_{ij}\right)$ as the *observation* of a scene point P_i on camera C_j.

3.2 Sub-set Constraints

SPHERA implements an incremental pipeline, that is, starting from an initial pair, cameras are sequentially added until all poses have been estimated. More specifically, we draw from $\mathcal{C}$ an initial pair of cameras and once their poses are determined they are used to initialize a set $\mathcal{C}^{'}$ representing the current set of calibrated cameras. Then, one by one, cameras are added to $\mathcal{C}^{'}$ until $|\mathcal{C}^{'}| = |\mathcal{C}|$. After adding a camera to $\mathcal{C}^{'}$, BA is performed to refine the poses of all calibrated cameras. However, this is not always necessary. As calibration progresses, previously added cameras become more stable, i.e. their poses no longer change significantly. After some time, refining their poses brings no improvement. This is often true for large image datasets. The exceptions to this are loop closures and later addition of new images to the dataset.

To address this issue, we introduce a sub-set $\mathcal{C}^{*} \subset \mathcal{C}^{'}$ to hold the cameras for which pose refinement is unnecessary. Cameras belonging to $\mathcal{C}^{*}$ will be regarded as fixed and their poses will not be updated during BA. A camera $C_j \in \mathcal{C}^{'}$ is added to $\mathcal{C}^{*}$ when the update on its pose is no longer significant. This is achieved with the introduction of two measurements in the following way. After BA, we measure the update on its rotation matrix δ_{R_j} and translation vector δ_{t_j} computed as

$$\delta_{R_j} = \| log \left(R_j^{k-1} \left(R_j^k \right)^T \right) \|, \tag{6}$$

$$\delta_{t_j} = \frac{\| t_j^k - t_j^{k-1} \|}{\| t_j^{k-1} \|}, \tag{7}$$

where k stands for calibration step, i.e. it is incremented after each BA. The right-hand side of Eq. 6 is a metric in $SO\,(3)$ and can be efficiently computed with quaternions [32]. Then, if $\delta_{R_j} < \tau_r$ and $\delta_{t_j} < \tau_t$, with $\tau_r \geqslant 0$ and $\tau_t \geqslant 0$, C_j is added to $\mathcal{C}^{*}$. Clearly, once C_j is added to $\mathcal{C}^{*}$, $\delta_{R_j} = 0$ and $\delta_{t_j} = 0$ in the subsequent calibration steps. Therefore, to correctly handle loop closures and to locally update camera poses whenever new images are included in the dataset, a third measurement is required. This measurement allows to *remove* cameras from $\mathcal{C}^{*}$ so that they may be optimized once again. It is based on the visibility of scene points and works as follows. Assume $C_j \in \mathcal{C}^{*}$ and C_j observes N_j 3D points. The visibility measurement δ_{v_j} of a camera C_j is then defined as

$$\delta_{v_j} = \| \nu_j^k - \nu_j^{k-1} \|, \; with \tag{8}$$

$$\nu_j = \sum_{n=0}^{N_j - 1} \eta_n, \; \eta_n = \begin{cases} 1, \; if \; \mathcal{V}_n \; increased \\ 0, \; otherwise \end{cases}. \tag{9}$$

Note that δ_{v_j} is independent of the camera pose. Also, it does not measure how many new 3D points C_j observes. Instead, it measures, among all 3D points visible in C_j, how many had their visibility maps updated, i.e. are now visible in at least one new camera. Then, if $\delta_{v_j} > \tau_v$, C_j is removed from $\mathcal{C}^{*}$ and will be taken into account in the next calibration step. In our implementation, we also use the visibility measurement along with the first two to decide whether a camera should be added to $\mathcal{C}^{*}$. Together, τ_r, τ_t and τ_v form the sub-set constraints.

Remark 1. The re-triangulation of point correspondences is beneficial as it reduces drift due to accumulated errors. However, it increases the overall computational cost. The sub-set constraints prevent the re-triangulation of points $\hat{P}_i$ that are seen exclusively by cameras in $\mathcal{C}^*$, thus further improving performance.

3.3 Minimizing the Reprojection Error

As discussed above, recovering the camera poses and scene structure can be achieved by solving a bundle adjustment problem [24]. SPHERA minimizes a reprojection error formulated directly on the surface of the unit sphere (see Eqs. 3 and 4). This is interpreted as finding the camera poses that maximize the alignment between the rays defined by all predicted and measured projections. This is true for any central projection camera.

After the introduction of the visibility map in Section 3.1, we may now rewrite Eq. 3 in the form shown in Eq. 10. We adopted the framework available in [33] as the core non-linear solver upon which SPHERA is built.

$$\min_{\rho} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \gamma_{ij} e_{ij}\left(\rho\right), \ \gamma_{ij} = \begin{cases} 1, \ if \ C_j \in \mathcal{V}_i \\ 0, \ otherwise \end{cases} \tag{10}$$

In practice, we solve a modified version of Eq. 10, where only cameras $C_j \in \mathcal{C}^{'} \setminus \mathcal{C}^*$ and the most reliable points are used. These points are defined as

$$\mathcal{P}^* = \left\{ \hat{P}_i \in \mathcal{P} \mid e_{ij}\left(\rho\right) < \tau_e, \ \forall \left(C_j, p_{ij}\right) \in \mathcal{V}_i \right\}, \tag{11}$$

where τ_e is a threshold imposed to all individual reprojection errors $e_{ij}\left(\rho\right)$.

4 Evaluation

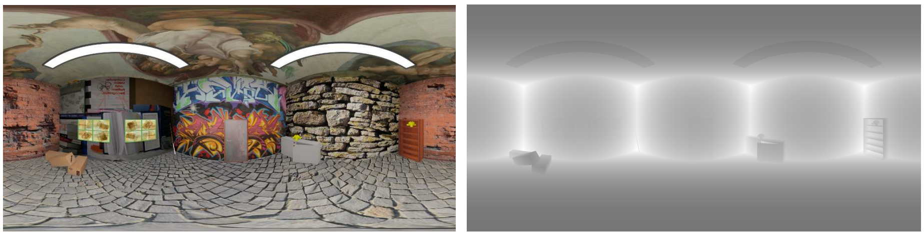
4.1 Preliminaries

Keypoints are detected and matched using the method proposed in [34], where a multi-scale keypoint detector and matcher was developed for high resolution spherical images. Nonetheless, it is worth mentioning that SPHERA is completely independent of how keypoints are detected, described and matched. Consequently, any other keypoint detector and matcher may be adopted (see Section 4.3).

We validate our framework using synthetic spherical as well as real perspective and spherical images. The resolution of all spherical images presented below is 14142×7071 (100 Mega-pixels). Experiments are divided into four categories: The first category consists of a set of synthetic spherical cameras where the goal is to validate our framework on spherical images using groundtruth. The second is composed exclusively of real perspective images. Here, the idea is to show that our framework is suitable for standard SfM, i.e. it may be used even when no spherical image is available. The third category consists of spherical images only, where we compared SPHERA to the work presented in [15] in two different real world scenarios. The fourth and last category is a hybrid dataset where real perspective and spherical images are used simultaneously. The aim is to demonstrate SPHERA's ability to improve scene geometry estimation whenever more images are available, independent of their types[1]. Whenever available, groundtruth data is used for evaluation. Otherwise, we rely on the global mean reprojection error computed taking all images and all reconstructed points into account.

4.2 Synthetic Dataset

An artificial room with dimensions 6x6x3 meters was created using [35] and 72 spherical images were rendered (see Fig. 4-(a)). The poses of these artificially generated cameras were used as groundtruth. Additionally, the depth map shown in Fig. 4-(b) was stored and serves to measure the accuracy of the recovered scene geometry.



(a) (b)

Fig. 4. (a) Sample image of the synthetic dataset. (b) Groundtruth depth map used to evaluate the accuracy of scene geometry estimation (contrast enhanced to improve visualization).

After detecting and matching keypoints with Gava's approach, camera poses and scene structure were recovered with SPHERA. Residual errors were computed in the following way. The position error is the Euclidean distance between the groundtruth and estimated camera positions. To measure the orientation error, we chose again the function presented in [32], which in this context may be written as $\|log\left(R\hat{R}^T\right)\|$, with R the desired rotation and $\hat{R}$ the estimated rotation matrix. For details we refer to [32]. The residual error of a reconstructed point $\hat{P}_i$ is computed as $\|\hat{P}_i - P_i\|$, where the coordinates of P_i are obtained as follows. A virtual spherical camera is located at the origin of the global coordinate system. The projection of $\hat{P}_i$ onto this virtual camera delivers $p_i^{'}$. Then

[1] Assuming central projection cameras.

$P_i = I_{dm}\left(p_i^{'}\right) p_i^{'}$, where $I_{dm}\left(p_i^{'}\right)$ is the groundtruth depth retrieved from the stored depth map.

We first ran our pipeline ignoring the sub-set constraints and with τ_e equivalent to 5 pixels. Although τ_e is an angular deviation, for convenience we converted and presented it in pixels. Table 1 presents the resulting errors in camera poses and scene reconstruction.

	orient. error [degree]	pos. error [mm]	recon. error [mm]
μ	0.009	0.68	0.482
σ	0.03	2.8	0.9

Table 1. Errors in camera poses and sparse scene reconstruction for the synthetic dataset. Mean and standard deviation are identified by μ and σ, respectively.

τ_v [# points]	100		1000	
τ_t [%]	1	5	1	5
orient. error [degree]	0.0091	0.0091	0.0094	0.0095
pos. error [mm]	0.76	0.78	0.76	0.79
recon. error [mm]	0.487	0.494	0.581	0.604
time [%]	49.7	48.2	15.4	14.9

Table 2. Average errors in camera poses and sparse scene reconstruction for the synthetic dataset with sub-set constraints. The last line shows the running time relative to the total time needed when no sub-set constraints are used.

Figure 5 shows the reconstructed point cloud, with approximately $156K$ points. The rendered spheres and their corresponding coordinate frames reflect the recovered camera poses. We adopted the Odysseus Studio [36] to visualize and present our results.

A second experiment aimed at evaluating the impact of the sub-set constraints on camera pose estimation, the sparse reconstruction of the scene and the overall performance gain. We ran our pipeline varying the sub-set constraints within the ranges $\tau_r = [0.25°, 2°]$ in steps of 0.25°, $\tau_t = [0.01, 0.05]$ and $\tau_v = [100, 1000]$. We noticed that, for this experiment, varying τ_r had little impact on the final results. Thus, Table 2 summarizes the average values. Time values are relative to the total time required when no sub-set constraints are used. The standard deviations for the rotation, position and reconstruction errors were below 0.04°, 3.25 mm and 1.2 mm, respectively. On the other hand, the standard deviations for the performance gain were approximately 10% for $\tau_v = 100$ and 7% for $\tau_v = 1000$. This is probably due to different gradient descent paths chosen by the non-linear optimizer [33].

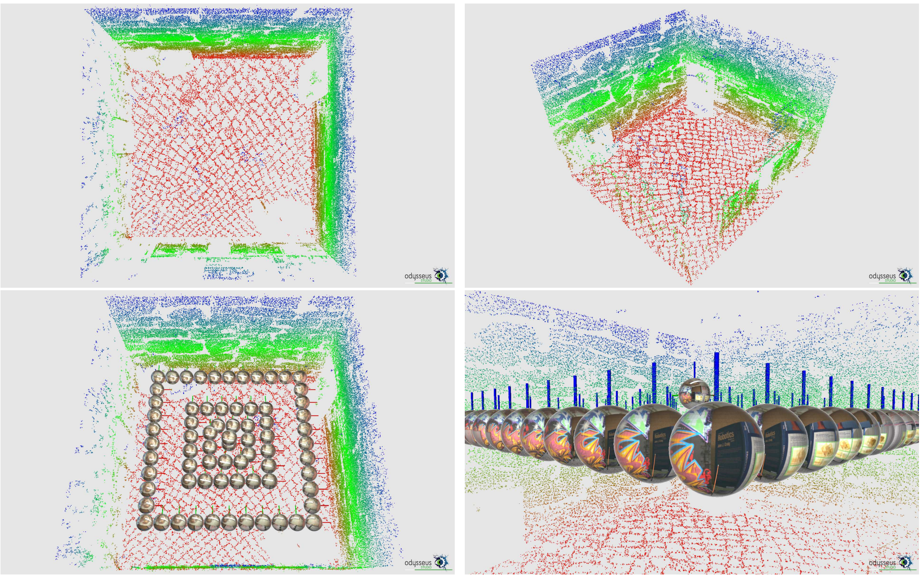


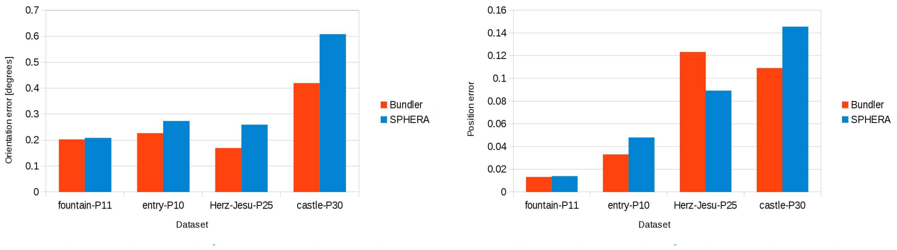
Fig. 5. Reconstructed point cloud and recovered camera poses obtained with SPHERA. Details on the floor and walls can be easily seen.

4.3 Perspective Datasets

To validate our approach on perspective images, we compared it to Bundler [17], a popular software developed for SfM on standard perspective images. Bundler is the camera calibration tool currently used in [10, 26, 37], and is publicly available.

The experiments presented in this section were carried out on the datasets published in [28]. For each dataset, we ran Bundler on the original images and SPHERA on the corresponding warped images as shown in Fig. 2. To ensure a fair comparison, we ran our pipeline using the same keypoints detected by Bundler [38] after warping their coordinates to the unit sphere. This eliminates the influence of image feature location on the evaluation. Moreover, it shows SPHERA's independence of keypoint detectors as pointed out in Section 4.1. Results on camera pose estimation are summarized in Fig. 6. Orientation errors were obtained as in the previous section. Position errors, however, were computed after preprocessing the estimated camera positions. To account for the differences in scale, the baseline between the closest camera pair was normalized and the remaining camera positions were scaled accordingly. After that, the Euclidean distance was measured as in Section 4.2.

As can be seen, Bundler performs slightly better and the reason is as follows. Bundler works exclusively on perspective images and optimizes the camera poses along with their individual intrinsic parameters such as focal length and lens distortion. In contrast, SPHERA has been designed to operate on any kind of central projection camera, but the optimization of intrinsic parameters has not been integrated yet. Therefore, for the experiments presented in this



(a) (b)

Fig. 6. (a) Orientation error and (b) position error on perspective image datasets obtained with Bundler and SPHERA. See text for details.

section, we used a constant focal length in our pipeline and a variable focal length for Bundler. In fact, the differences observed in Fig. 6 are proportional to the variance of the focal length within each dataset, see Table 3. The exception is Herz-Jesu-P25, where Bundler delivers smaller orientation error whereas SPHERA provides better camera positions.

dataset	σ_f [pixel]	**range** [pixel]
fountain-P11	8.49	23.02
entry-P10	10.97	28.41
Herz-Jesu-P25	4.01	16.15
castle-P30	20.44	118.86

Table 3. Variation of focal lengths estimated with Bundler. The second column shows the standard deviation and the third column the difference between maximum and minimum values. Note that, except for the Herz-Jesu-P25 dataset, the differences in Fig. 6 are proportional to the variation of the focal length.

4.4 Spherical Datasets

In this section we compare SPHERA and the approach presented in [15]. We ran both pipelines on two datasets. The first dataset consists of 9 spherical images captured inside one of the Mogao Caves, in China. The second dataset contains 35 spherical images taken at the Saint Martin Square in Kaiserslautern, Germany, and represents outdoors, more challenging, environments. Due to the lack of groundtruth data for these datasets, we based our evaluation on the global mean reprojection error. The assumption is that the correlation observed in Section 4.2 can be used to infer the relative accuracy of the estimated scene geometry.

As can be seen in Fig. 7, SPHERA improves the reprojection error on both datasets, specially on the St. Martin Square. In the case of the Mogao Cave, due to its simple geometry and rich texture (Fig. 8-(a)), only few points are discarded based on Eq. 11, what explains the small difference in the reprojection error for this dataset. The St. Martin Square dataset is more challenging (Fig. 8-(b)). It contains many low textured regions, depth discontinuites, occlusions as well as repetitive patterns. Therefore, several points are inconsistent and discarding them from the camera pose estimation leads to the difference observed in Fig. 7. These results suggest that SPHERA delivers more accurate scene structures. Figure 8 displays the sparse point clouds yielded by our framework, where details of the surroundings are accurately reconstructed.

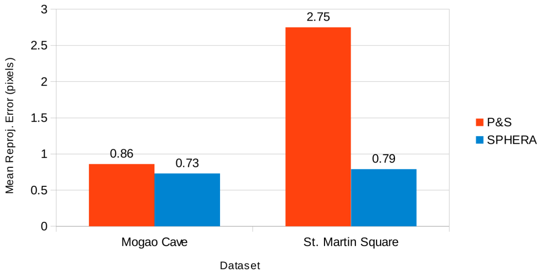


Fig. 7. Global mean reprojection error on spherical image datasets obtained with [15] and SPHERA. See text for details.

4.5 Hybrid Dataset

In this section we evaluate the SPHERA framework on a hybrid dataset composed of perspective and spherical images. The idea is to show that our framework naturally handles different central projection cameras simultaneously. This dataset is composed of the same 35 spherical images used in the previous experiment and additional 11 perspective images of resolution 3888×2592 pixels. As shown in Fig. 9, the reprojection error obtained with spherical images (same as previous experiment) is better than the error for perspective images.

The main reason spherical camera pose estimation is better than its perspective counterpart is due to their wide field of view. As can be seen in Fig. 10, matches between spherical images cover the entire scene and thus impose more constraints on cameras' poses. As expected, the reprojection error decreases when perspective and spherical images are used simultaneously.

5 Conclusions

This paper presents SPHERA, a novel unifying Structure from Motion framework designed for central projection cameras. The goal is to cover the gaps between pipelines developed for perspective, spherical and catadioptric images

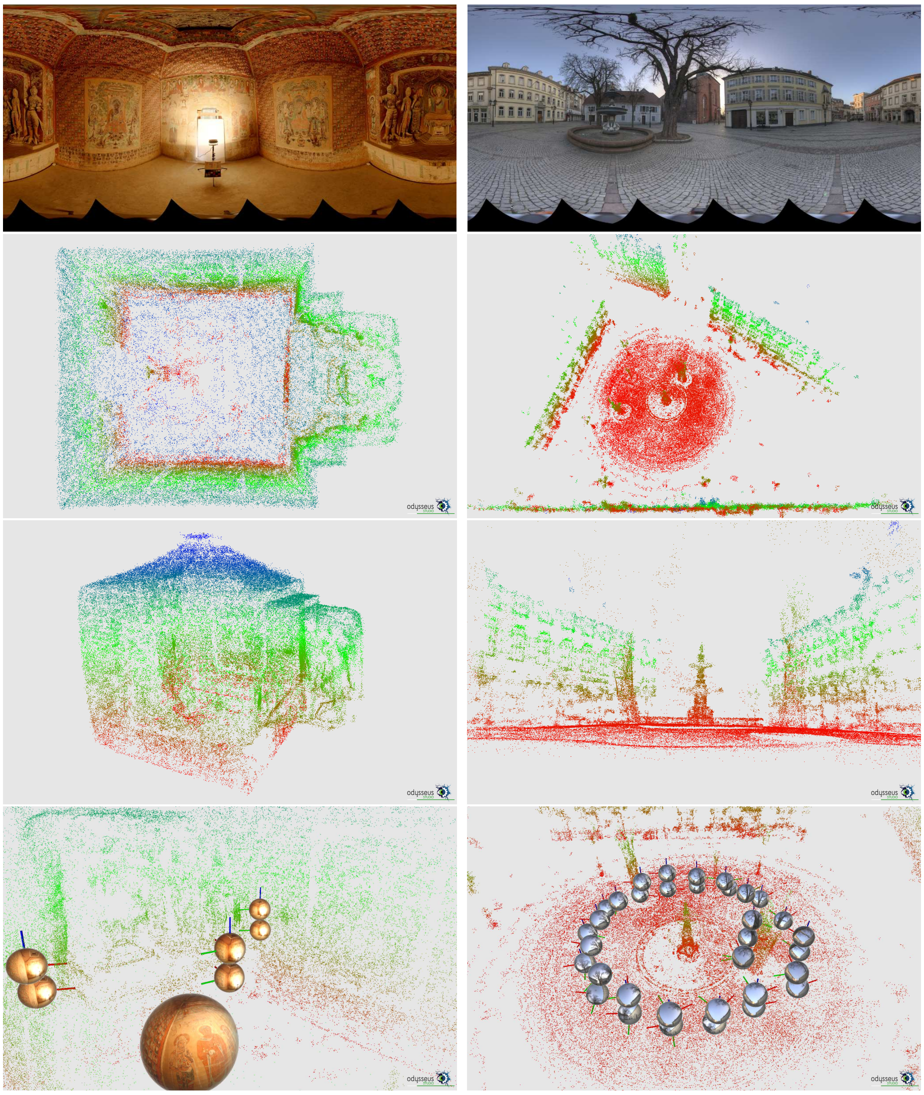


Fig. 8. First row: Sample images of the Mogao Cave and St. Martin Square datasets. Second to fourth rows: reconstructed point clouds delivered by SPHERA, containing approximately 106K and 197K 3D points for the Mogao Cave and St. Martin Square, respectively.

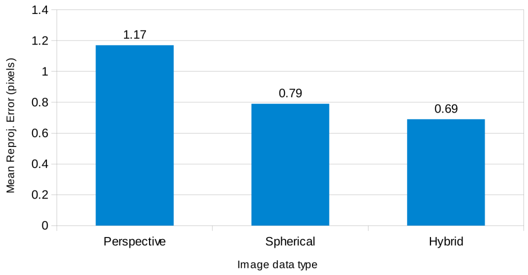
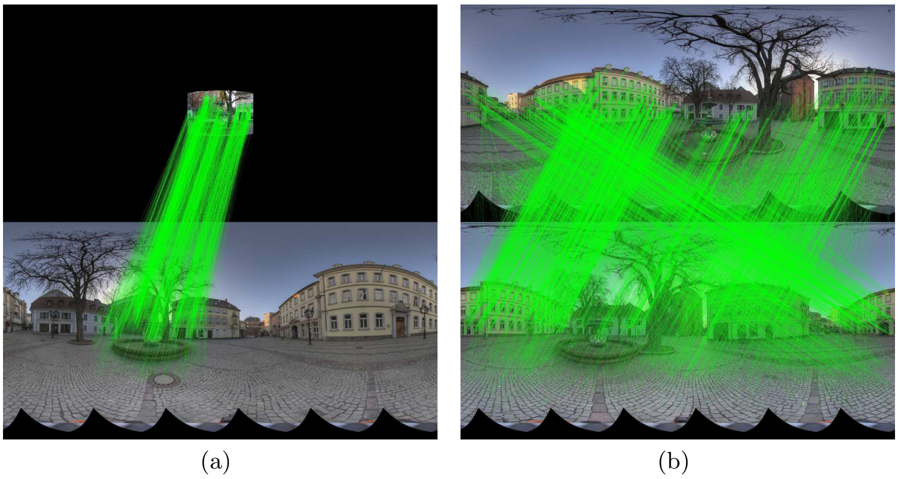


Fig. 9. Global mean reprojection error for the hybrid St. Martin Square experiment. Note how it decreases when perspective and spherical images are used together.

and to support future large scale 3D reconstruction algorithms. Through extensive quantitative evaluation on synthetic and real image sequences, we showed that our approach delivers high quality camera pose as well as scene geometry estimations when compared to state of the art approaches optimized for specific camera types.

Future work aims at integrating the optimization of intrinsic parameters to increase the accuracy of pose estimation of perspective cameras. Additionally, we plan to validate our framework on larger, hybrid image datasets, supported by groundtruth data. Finally, SPHERA will be the underlying SfM mechanism in our upcoming dense multi-view reconstruction approach.


(a)

(b)

Fig. 10. (a) Symmetric matches between a warped perspective image and a spherical image. (b) Symmetric matches between two full spherical images.

Acknowledgments. The authors would like to thank Richard Schulz for the creation of the synthetic dataset. This work was funded by the project DENSITY (01IW12001).

References

1. Civetta 360^o Digital Imaging, http://www.weiss-ag.org/solutions/civetta/
2. LizardQ, http://www.lizardq.com
3. Seitz Roundshot, http://www.roundshot.ch
4. THETA, https://theta360.com/en/
5. New House Internet Services BV, http://www.ptgui.com
6. Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., Weaver, J.: Google Street View: Capturing the World at Street Level. Computer 43 (2010)
7. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, Washington DC (2007)
8. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: Dense Tracking and Mapping in Real-time. In: International Conference on Computer Vision, pp. 2320–2327. IEEE Computer Society, Washington DC (2011)
9. Hiep, V.H., Keriven, R., Labatut P., Pons, J.-P.: Towards high-resolution large-scale multi-view stereo. In: Computer Vision and Pattern Recognition, pp 1430–1437. IEEE Computer Society (2009)
10. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Towards Internet-scale Multi-view Stereo. In: Computer Vision and Pattern Recognition. (2010)
11. Chang, P., Hebert, M.: Omni-Directional Structure from Motion. In: IEEE Workshop on Omnidirectional Vision, pp. 127–133. IEEE Computer Society, Washington (2000)
12. Micusik, B., Pajdla, T.: Structure from Motion with Wide Circular Field of View Cameras. Trans. Pattern Anal. Mach. Intell. 28, 1135–1149 (2006)
13. Bagnato, L., Frossard, P., Vandergheynst, P.: A Variational Framework for Structure from Motion in Omnidirectional Image Sequences. J. of Mathematical Imaging and Vision 41, 182–193 (2011)
14. Kangni, F. Laganiere, R.: Orientation and Pose recovery from Spherical Panoramas. In: IEEE International Conference on Computer Vision, pp. 1–8. IEEE Computer Society, Los Alamitos (2007)
15. Pagani, A., Stricker D.: Structure from Motion using full spherical panoramic cameras. In: OMNIVIS (2011)
16. Aly, M., Bouguet, J.-Y.: Street view goes indoors: Automatic pose estimation from uncalibrated unordered spherical panoramas. In: Winter Conference on Applications of Computer Vision (2012)
17. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: SIGGRAPH, pp. 835–846. ACM, New York (2006)
18. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)
19. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: An invitation to 3D vision, from images to models. Springer Verlag (2003)
20. Arie-Nachimson, M., Kovalsky, S.Z., Kemelmacher-Shlizerman, I., Singer, A., Basri, R.: Global Motion Estimation from Point Matches. In: 3DIMPVT (2012)

21. Olsson, C., Enqvist, O.: Stable Structure form Motion for Unordered Image Collections. In: SCIA, LNCS 6688 (2011)
22. Wu, C.: Towards Linear-time Incremental Structure from Motion. In: 3DV (2013)
23. Rodríguez, A.L., López-de-Teruel, P.E., Ruiz, A.: Reduced Epipolar Cost for Accelerated Incremental SfM. In: Computer Vision and Pattern Recognition. (2011)
24. Triggs, B., Mclauchlan, P. F., Hartley, R. I., Fitzgibbon, A. W.: Bundle Adjustment: A Modern Synthesis. In: International Workshop on Vision Algorithms: Theory and Practice, pp. 298-372. (1999)
25. Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building Rome on a Cloudless Day. In: European Conference on Computer Vision, pp. 368–381. Springer-Verlag, Berlin, Heidelberg (2010)
26. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a day. In: International Conference on Computer Vision, pp. 72–79. Kyoto (2009)
27. Geyer, C., Daniilidis, K.: Catadioptric projective geometry. Int. Journal of Computer Vision. 43, 223-243 (2001)
28. Strecha, C., Hansen, W., Van-Gool, L., Fua, P., Thoennessen, U.: On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In: Computer Vision and Pattern Recognition. (2008)
29. Torii, A., Imiya, A., Ohnishi, N.: Two- and Three- View Geometry for Spherical Cameras. In: OMNIVIS. (2005)
30. H. Stewénius, H., Engels, C., Nistér, D.: Recent Developments on Direct Relative Orientation. Journal of Photogrammetry and Remote Sensing 60, 284–294 (2006)
31. Quan, L., Lan, Z.: Linear N-Point Camera Pose Determination. Trans. Pattern Anal. Mach. Intell. 21, 774–780 (1999)
32. Huynh, D.Q.: Metrics for 3D Rotations: Comparison and Analysis. J. of Mathematical Imaging and Vision 35, 155-164 (2009)
33. Agarwal, S., Mierle, K., Others: Ceres Solver, `https://code.google.com/p/ceres-solver/`
34. Gava, C.C., Hengen, J.M., Tätz, B., Stricker, D.: Keypoint Detection and Matching on High Resolution Spherical Images. In: International Symposium on Visual Computing, pp. 363-372. Rethymnon (2013)
35. Blender, `http://www.blender.org/`
36. Odysseus Studio, `http://av.dfki.de/odysseus-studio`
37. Furukawa, Y., Ponce, J.: Accurate, Dense, and Robust Multi-View Stereopsis. Trans. Pattern Anal. Mach. Intell. (2008)
38. Lowe, D.G., Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision 60, 91–110 (2004)