# A generic structure-from-motion algorithm for cross-camera scenarios

Srikumar Ramalingam, Suresh K. Lodha, Peter Sturm

# A Generic Structure-from-Motion Algorithm for Cross-Camera Scenarios

Srikumar Ramalingam[1] Suresh K. Lodha[1] Peter Sturm[2]

[1] Dept. of Computer Science, University of California, Santa Cruz, USA,
{srikumar,lodha}@soe.ucsc.edu; http://www.soe.ucsc.edu/~{srikumar,lodha}
[2] INRIA Rhône-Alpes, 38330 Montbonnot, France
Peter.Sturm@inrialpes.fr; http://www.inrialpes.fr/movi/people/Sturm/

**Abstract.** We introduce a generic structure-from-motion approach based on a previously introduced, highly general imaging model, where cameras are modeled as possibly unconstrained sets of projection rays. This allows to describe most existing camera types (at least for those operating in the visible domain), including pinhole cameras, sensors with radial or more general distortions, catadioptric cameras (central or non-central), etc. We introduce a structure-from-motion approach for this general imaging model, that allows to reconstruct scenes from calibrated images, possibly taken by cameras of different types (cross-camera scenarios). Structure-from-motion is naturally handled via camera independent ray intersection problems, solved via linear or simple polynomial equations. We also propose two approaches for obtaining optimal solutions using bundle adjustment, where camera motion, calibration and 3D point coordinates are refined simultaneously. One is relatively straightforward, minimizing distances between 3D points and projection rays. The other minimizes reprojection error; the general imaging model does not provide analytical expressions for the reprojection error and its derivatives, which are desirable for efficient optimization. To achieve this, we propose to approximate the set of projection rays of a general non-central camera by several clusters of central rays, allowing us to formulate an analytical cost function. We present results for two cross-camera scenarios – a pinhole used together with an omnidirectional camera and a stereo system used with an omnidirectional camera. Using ground-truth and 3D reconstruction results from classical techniques, we show that our generic algorithm is simple, general and accurate for extensions to various cross-camera and multi-camera scenarios.

## 1 Introduction and Motivation

Many different types of cameras including pinhole, stereo, catadioptric, omnidirectional and non-central cameras have been used in computer vision. Some of these cameras, especially the omnidirectional class, provide more stable egomotion estimation and larger fields of view [2, 21, 18]. Naturally, larger fields of view allow to reconstruct 3D scenes using fewer images, although the spatial resolution is lower, e.g. pinhole cameras can provide more useful texture maps. Non-central cameras, a review of which is present in [3], eliminate the scale ambiguity in motion estimation and thereby we do not need ground control points

for scale computation. Thus using a variety of cameras will facilitate and enhance the 3D reconstruction in both geometry and texture. For example, we can build a surveillance system with one static omnidirectional camera (which detects moving objects) and several narrow-field-of-view pan-tilt-zoom cameras that can be used to take closeup pictures of objects. Also while reconstructing complete environments, it is helpful to have a combination of omnidirectional and traditional images: the traditional ones (narrow field of view, i.e. high spatial resolution) give good accuracy locally, whereas the omnidirectional images would be good for registering images scattered throughout the environment to a single reference frame. Inspite of these advantages, a general, unified, structure-from-motion approach for handling different camera systems, does not exist yet.

This statement holds also for camera calibration: most existing calibration methods are parametric and camera dependent [12, 7]. For example, in the pinhole camera we use a $3 \times 3$ matrix, called intrinsic matrix, to store the internal parameters of a camera. This matrix along with the camera pose provide the necessary calibration information. Similarly, calibration of optical distortions and of central or non-central catadioptric systems or other omnidirectional cameras, has been done using various specific parametric camera models. A non-parametric model and approach to camera calibration, referred to as the generic imaging model, was recently introduced by Grossberg and Nayar [9]: camera calibration is formulated as computing a 3D projection ray for every image pixel. Their method requires several images of calibration objects, with known relative motions between image acquisitions. We have recently introduced a more general calibration approach, that does not need a specific experimental setup; it only requires taking images of calibration objects, from completely unknown viewpoints [24, 25]. This technique is used for calibrating the systems used in our experiments. Section 3.1 provides a brief overview of the calibration algorithm.

Besides proposing algorithms, we want to stress, in this paper, that most basic structure-from-motion problems can be formulated in a unified, camera independent manner, typically as ray intersection type problems. This is shown for pose and motion estimation and triangulation, in Sections 3.2 to 3.4.

The main contribution of this paper is the description of an approach for 3D scene reconstruction from images acquired by any camera or system of cameras following the general imaging model. Its building blocks are motion estimation, triangulation and bundle adjustment algorithms, which are all basically formulated as ray intersection problems. Classical motion estimation (for pinhole cameras) and its algebraic centerpiece, the essential matrix [14], are generalized in Section 3.2, following [22]. As for triangulation, various algorithms have been proposed for pinhole cameras in [11]. In this work, we use the *mid-point* approach because of its simplicity, see Section 3.3. Initial estimates of motion and structure estimates, obtained using these algorithms, are refined using bundle adjustment [12, 27], i.e. (non-linear in general) optimization of all unknowns. This is described in Section 4. Bundle adjustment needs a good initial solution, and also depending on the cost functions the convergence rate and the optimality of the final solutions vary [11, 27]. In this work we utilize two different cost functions to design and implement two different bundle adjustment algorithms. The first cost function is based on minimizing the distance between 3D points

and associated projection rays, which we refer to as the *ray-point* method. The main reason for using this cost function is that it was straightforward to extend to non-parametric cross-camera scenarios. The second cost function is, as usually desired, based on the reprojection error, i.e. the distance between re-projected 3D points and originally measured image points (possibly weighted using uncertainty measures on extracted image point coordinates). The main reason for using this cost function is its statistical foundation [11].

There is a major challenge in extending the existing parametric bundle-adjustment algorithms (reprojection based) to a non-parametric scenario, i.e. the general imaging model. This is due to the fact that no analytical projection equation exists, and thus no analytical expression for the re-projection error based cost function and its derivatives. In order to address this challenge, we approximate the $n$ rays from a given camera, central or non-central, by $k$ clusters of central rays, i.e. rays that intersect in a single point. For example we have $k = 1$ for central cameras (e.g. pinhole), $k = 2$ for stereo cameras, $k = n$ for oblique cameras [20], etc. Each such cluster of rays, therefore, corresponds to a single central camera. Given any 3D point we find the corresponding cluster of rays to which it belongs. The bunch of rays in every central cluster is intersected by a plane to synthesize a perspective image. This allows us to formulate a parametric function that maps the 3D point to a 2D pixel on the synthesized image, and thus to drive bundle adjustment. Details are discussed in Section 4.2.

Experimental results with two cross-camera scenarios are given in Section 5.

## 2 Previous Work and Background

We briefly explain previous efforts in 3D reconstruction using various cameras. Pinhole cameras have a long history of being employed for 3D reconstruction [12]. In the last decade or so, omnidirectional cameras and non-central cameras have also been used for 3D reconstruction [3, 2, 5, 13]. Recently, Micusik et al. extended the multiple view metric 3D reconstruction to central fish-eye cameras [17]. Central catadioptric cameras such as para-catadioptric systems (orthographic camera facing a parabolic mirror) were calibrated and utilized in 3D reconstruction by Geyer and Daniilidis [8]. Omni-directional images, with known camera motion, obtained from a GPS or a robot, have also been used in 3D reconstruction [16, 4]. All these efforts have utilized parametric calibration techniques and camera dependent structure-from-motion algorithms. In contrast, in this work we utilize a generic camera calibration technique and a generic structure-from-motion algorithm that applies equally well to all types of cameras – pinhole, stereo, omnidirectional etc.

The importance of using cross-camera networks for 3D reconstruction and video surveillance has been observed by few researchers as yet. One of the first steps in this direction is the process of proposing unifying models and multi-view relations for different cameras. Geyer and Daniilidis [6] developed a unified theory that encompasses all central catadioptric systems, observed by Baker and Nayar in [1]. Sturm developed multi-view relations for any mixture of para-catadioptric, perspective or affine cameras [23]. Our work is complementary to these efforts in enhancing and promoting the use of cross-camera scenarios for practical applications.

## 3 Generic Structure-from-Motion

Figure 1 describes the pipeline for the generic structure-from-motion algorithm.

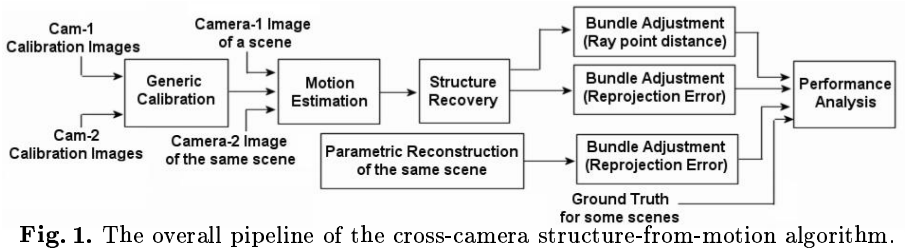



**Fig. 1.** The overall pipeline of the cross-camera structure-from-motion algorithm.

### 3.1 Generic Camera Calibration

We use a generic calibration approach developed in [25], an extension of [9], to calibrate the different camera systems. For the sake of completeness, we briefly explain the algorithm. Calibration consists in determining, for every pixel, the 3D projection ray associated with it. In [9], this is done as follows: two images of a calibration object with known structure are taken. We suppose that for every pixel, we can determine the point on the calibration object, that is seen by that pixel. For each image and each pixel, we thus obtain two 3D points. Their coordinates are usually only known in a coordinate frame attached to the calibration object; however, if one knows the motion between the two object positions, one can align the coordinate frames. Then, every pixel's projection ray can be computed by simply joining the two observed 3D points.

In [25], we propose a more general approach, that does not require knowledge of the calibration object's displacement. In that case, three images need to be taken at least. The fact that all 3D points observed by a pixel in different views, are on a line in 3D, gives a constraint that allows to recover both the motion and the camera's calibration. The constraint is formulated via a set of trifocal tensors, that can be estimated linearly, and from which motion, and then calibration, can be extracted (details are given in [24]).

### 3.2 Motion Estimation

We describe how to estimate ego-motion, or, more generally, relative position and orientation of two calibrated general cameras. This is done via a generalization of the classical motion estimation problem for pinhole cameras and its associated centerpiece, the essential matrix [14]. We briefly summarize how the classical problem is usually solved [12]. Let $R$ be the rotation matrix and $T$ the translation vector describing the motion. The essential matrix is defined as $E = [T]_\times R$. It can be estimated using point correspondences $(x, x')$ across two views, using the epipolar constraint ${x'}^\mathsf{T} E x = 0$. This can be done linearly using 8 correspondences or more. In the minimal case of 5 correspondences, an efficient non-linear minimal algorithm, which gives exactly the theoretical maximum of 10 feasible solutions, was only recently introduced [19]. Once the essential matrix is estimated, the motion parameters $R$ and $T$ can be extracted relatively straightforwardly [19].

In the case of our general imaging model, motion estimation is performed similarly, using pixel correspondences $(x, x')$. Using the calibration information,

the associated projection rays can be computed. Let them be represented by their Plücker coordinates, i.e. 6-vectors $X$ and $X'$. The epipolar constraint extends naturally to rays, and manifests itself by a $6 \times 6$ essential matrix, defined as:

$$\mathcal{E} = \begin{pmatrix} R & -E \\ 0 & R \end{pmatrix}$$

The epipolar constraint then writes: ${X'}^{\mathsf{T}} \mathcal{E} X = 0$ [22]. Once $\mathcal{E}$ is estimated, motion can again be extracted straightforwardly (e.g., $R$ can simply be read off $\mathcal{E}$). Linear estimation of $\mathcal{E}$ requires 17 correspondences.

There is an important difference between motion estimation for central and non-central cameras: with central cameras, the translation component can only be recovered up to scale. Non-central cameras however, allow to determine even the translation's scale. This is because a single calibrated non-central camera already carries scale information (via the distance between mutually skew projection rays). One consequence is that the theoretical minimum number of required correspondences is 6 instead of 5. It might be possible, though very involved, to derive a minimal 6-point method along the lines of [19]. More details on motion estimation are omitted due to lack of space.

### 3.3 Structure Recovery/Triangulation

We now describe an algorithm for 3D reconstruction from two or more calibrated images with known relative position. Let $P = (X, Y, Z)^{\mathsf{T}}$ be a 3D point that is to be reconstructed, based on its projections in $n$ images. Using calibration information, we can compute the $n$ associated projection rays. Here, we represent the $i$th ray using a starting point $A_i$ and the direction, represented by a unit vector $B_i$. We apply the mid-point method [11, 22], i.e. determine $P$ that is closest in average to the $n$ rays. Let us represent generic points on rays using position parameters $\lambda_i$. Then, $P$ is determined by minimizing the following expression over $X, Y, Z$ and the $\lambda_i$: $\sum_{i=1}^{n} \|A_i + \lambda_i B_i - P\|^2$.

This is a linear least squares problem, which can be solved e.g. via the Pseudo-Inverse, leading to the following explicit equation (derivations omitted):

$$\begin{pmatrix} P \\ \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix}_{3+n} = \underbrace{\begin{pmatrix} n\mathtt{I}_3 & -B_1 & \cdots & -B_n \\ -B_1^{\mathsf{T}} & 1 & & \\ \vdots & & \ddots & \\ -B_n^{\mathsf{T}} & & & 1 \end{pmatrix}^{-1}}_{M_{(3+n)\times(3+n)}} \begin{pmatrix} \mathtt{I}_3 & \cdots & \mathtt{I}_3 \\ -B_1^{\mathsf{T}} & & \\ & \ddots & \\ & & -B_n^{\mathsf{T}} \end{pmatrix}_{(3+n)\times(3n)} \begin{pmatrix} A_1 \\ \vdots \\ A_n \end{pmatrix}_{3n}$$

where $\mathtt{I}_3$ is the identity matrix of size $3 \times 3$. Due to its sparse structure, the inversion of the matrix $M$ in this equation, can be performed very efficiently, as typically done in bundle adjustment [27]. Overall, the triangulation of a 3D point using $n$ rays, can by carried out very efficiently, using only matrix multiplications and the inversion of a symmetric $3 \times 3$ matrix (details omitted).

### 3.4 Pose Estimation

Pose estimation is the problem of computing the relative position and orientation between an object of *known* structure, and a calibrated camera. We don't use it

in this work, but describe it for the sake of completeness. A literature review on algorithms for pinhole cameras is given in [10]. Here, we briefly show how the minimal case can be solved for general cameras. For pinhole cameras, pose can be estimated, up to a finite number of solutions, from 3 point correspondences (3D-2D) already. The same holds for general cameras. Consider 3 image points and the associated projection rays, computed using the calibration information. We parameterize generic points on the rays as in the previous section: $A_i + \lambda_i B_i$.

We know the structure of the observed object, meaning that we know the mutual distances $d_{ij}$ between the 3D points. We can thus write equations on the unknowns $\lambda_i$, that parameterize the object's pose:

$$\|A_i + \lambda_i B_i - A_j - \lambda_j B_j\|^2 = d_{ij}^2 \quad \text{for } (i,j) = (1,2), (1,3), (2,3)$$

This gives a total of 3 equations that are quadratic in 3 unknowns. Many methods exist for solving this problem, e.g. symbolic computation packages such as MAPLE allow to compute a resultant polynomial of degree 8 in a single unknown, that can be numerically solved using any root finding method.

Like for pinhole cameras, there are up to 8 theoretical solutions. For pinhole cameras, at least 4 of them can be eliminated because they would correspond to points lying behind the camera [10]. As for general cameras, determining the maximum number of feasible solutions requires further investigation. In any case, a unique solution can be obtained using one or two additional points [10].

## 4 Bundle Adjustment

### 4.1 Ray-point bundle adjustment

This technique minimizes the distance between projection rays and 3D points. We briefly describe our cost function. As in the previous section, $A_i + \lambda_j B_i$ is used to represent generic points on a ray. The distance between this ray, in a coordinate system represented with $(R, T)$, and a 3D point $(X_j, Y_j, Z_j)$ is given below:

$$e_i = \|A_i + T + \lambda_j R B_i - \begin{pmatrix} X_j \\ Y_j \\ Z_j \end{pmatrix} \|$$

The parameter $\lambda_j$ is computed in closed form using least squares. In a general scenario having $n$ 3D points, each point may be associated with two or more rays coming from different camera views. We add all the ray-point distances (for every point with all the rays linked to it) to compute the overall cost function, which will be used in bundle adjustment to simultaneously refine the camera motion, possibly calibration, and 3D structure. The most important feature of this approach is that it is directly applicable to non-central and generically calibrated cameras.

### 4.2 Re-projection based bundle adjustment

We now describe the challenges in using re-projection based bundle adjustment for generically calibrated cameras and our approaches to overcome these challenges. In generic calibration, there is no mathematical function to relate the

pixel and a ray. The calibration result is a lookup, which maps every pixel to a ray. Non-linear optimization functions like Levenberg-Marquardt algorithms, which are used in bundle adjustment, use the derivatives of error functions. Computation of these derivatives poses the main problem in our scenario. For computing derivatives we need the ray associated to sub-pixel image coordinates. While this is easily possible in parametrically calibrated cameras, it is not straightforward in non-central cameras. One could try to use some kind of interpolation followed by numerical approaches for differentiation. However in noncentral and other non-perspective scenarios it is difficult to use interpolation.

We consider any camera as a cluster of central cameras. Therefore, given a set of rays belonging to a non-central camera, one can cluster the rays into $k$ buckets. For example $k = 2$ for a stereo camera. An interesting future area of research is to find the number of clusters and to design clustering algorithm. Once the rays corresponding to each central cluster is determined we synthesize a perspective image for each one of them. Perspective image for a cluster of rays belonging to a central camera can be easily computed by intersecting the rays with some plane. If the non-central camera has $k$ central cameras, we generate $k$ perspective images. Each of these perspective images can be parameterized based on the central rays and the intersecting plane, that allows us to create a parametric association from 3D points to 2D sub-pixels. In bundle adjustment we compute the reprojection error on all these synthesized images.

An important question is what criterion to use in choosing the plane of intersection while generating synthesized perspective images. We propose to minimize the "uncertainty" in the intersection points when the plane intersects the bunch of central rays: ideally the rays have to be perpendicular to the plane. Therefore we find an orientation which minimizes the sum $M$ of all acute angles between the plane and intersecting rays. That cost function $M$, and the required normal to the plane $(m_1, m_2, m_3)$ are given as:

$$M = \sum_{i=1}^{n} (m_1 l_1^i + m_2 l_2^i + m_3 l_3^i), \quad m_j = \sqrt{\frac{\sum l_j^i}{\sum ((l_1^i)^2 + (l_2^i)^2 + (l_3^i)^2)}}, \ \ j = 1, 2, 3$$

where $(l_1^i, l_2^i, l_3^i)$ refers to the direction of the $i_{th}$ ray. It is useful to discuss what happens to our algorithm in extreme cases. The first case is when we have only one ray in a cluster. In that case we consider a plane perpendicular to that ray and the center will be kept at infinity. The next interesting case is that of a highly non-central camera, where the number of clusters is very large. We will have to generate many perspective images and if we use the above optimization criterion for computing the normal for the intersecting plane, then this algorithm tends to become a ray-point distance based bundle adjustment. For example in a completely non-central camera, which is referred to as an oblique camera [20], where each ray belongs to a separate central cluster, our reprojection based algorithm will be exactly the same as a ray-point approach. On the contrary if the camera has just one cluster it becomes the conventional re-projection based algorithm, if the image coordinates in the synthesized perspective image match with that of the original image.

A possible improvement to the above approach is to identify a plane and generate a perspective view where the image coordinates are close to the original

image coordinates, which would better preserve the noise model in the image. Preliminary results with this approach are promising.

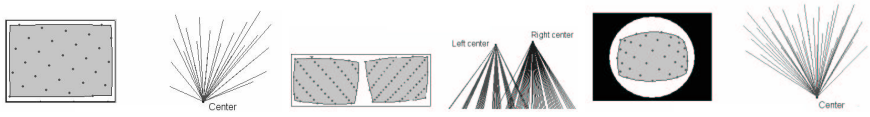
## 5 Results and Analysis

We consider three indoor scenarios:

- A house scene captured by omnidirectional and stereo system.
- A house scene captured by omnidirectional and pinhole camera.
- An objects scene, which consists of a set of objects placed in random positions as shown in Figure 3b, captured by an omnidirectional and a pinhole camera.

### 5.1 Calibration

We calibrate three types of cameras in this work. They are pinhole, stereo, and omni-directional systems. *Pinhole Camera:* Figure 2a shows the calibration of a pinhole camera using the single center assumption [25]. *Stereo camera:* Here we calibrate the left and right cameras separately as two individual pinhole cameras. In the second step we capture an image of a same scene from left and right cameras and compute the motion between them using the technique described in section 3.2. Finally using the computed motion we obtain both the rays of left camera and the right camera in the same coordinate system, which essentially provides the required calibration information. *Omni-directional camera:* Our omni-directional camera is a Nikon Coolpix-5400 camera with E-8 Fish-Eye lens. Its field of view is $360 \times 183$. In theory, this is just another pinhole camera with large distortions. The calibration results are shown in Figure 2. Note that we have calibrated only part of the image because three images are insufficient to capture the whole image in an omnidirectional camera. By using more than three boards it is possible to cover the whole image.

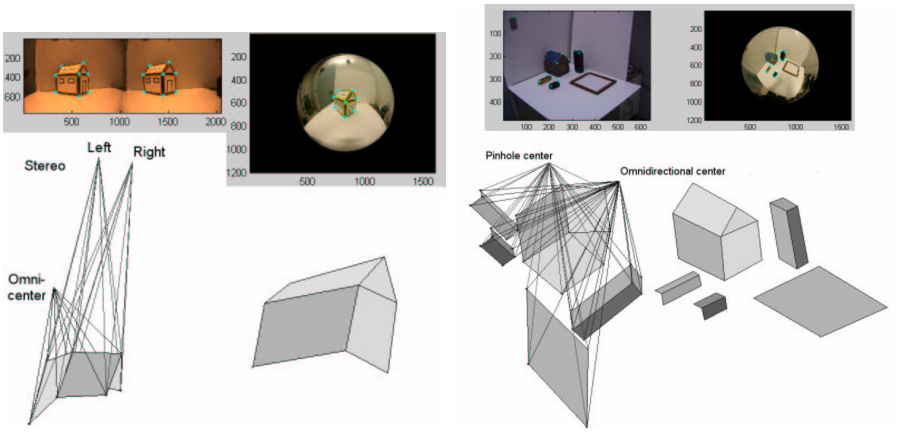



(a) (b) (c)

**Fig. 2.** a) Pinhole b) Stereo c) Omni-directional. The shading shows the calibrated region and the 3D rays on the right correspond to marked image pixels.

### 5.2 Motion and Structure Recovery

*Pinhole and Omni-directional:* Pinhole and omni-directional cameras are both central. Since the omni-directional camera has a very large field of view and consequently lower resolution compared to pinhole camera, the images taken from close viewpoints from these two cameras have different resolutions as shown in Figure 3. This poses a problem in finding correspondences between keypoints. Operators like SIFT [15], which are scale invariant, are not camera invariant. Direct application of SIFT failed to provide good results in our scenario. Thus we had to manually give the correspondences. One interesting research direction would be to work on the automatic matching of feature points in these images.




(a) (b)

**Fig. 3.** a) stereo and omni-directional b) pinhole and omni-directional. We intersect the rays corresponding to the matching pixels in the image to compute the 3D points.

*Stereo system and Omni-directional:* A stereo system can be considered as a non-central camera with two centers. The image of a stereo system is a concatenated version of left and right camera images. Therefore the same scene point appears more than once in the image. While finding image correspondences one keypoint in omni-directional image may correspond to 2 keypoints in stereo image as shown in Figure 3a. Therefore in the ray-intersection we intersect three rays to find one 3D point.

### 5.3 Bundle Adjustment Statistics

We discuss the convergence rate, error criteria and performance of the two bundle adjustment algorithms. Convergence rate is measured by the number of iterations. We consider all possible pairwise distances between reconstructed 3D points. To this purpose we normalize the 3D data to make the sum of squared distances from the centroid equal to unity. We have used ground-truth and parametric 3D reconstruction using classical techniques, to compare the performance of bundle adjustment. Parametric 3D reconstruction is obtained by parametric calibration of a pinhole camera, capturing of pinhole images, reconstruction by least-squares method and finally refinement of 3D scenes using parametric reprojection based bundle adjustment [12].

For the house scene, ground-truth is available. We compute the percentage error of a distance pair joining $i_{th}$ and $j_{th}$ points using method A as follows.

$$PE(i,j) = \frac{d(i,j)_A - d(i,j)_{gt}}{d(i,j)_{gt}}$$

where $d(i,j)_A$ and $d(i,j)_{gt}$ refer to the pairwise distance between the $i_{th}$ and $j_{th}$ points obtained using method A and ground-truth respectively. We then use the

absolute percentage errors of all the pairs to compute *mean absolute percentage error*. The statistics are presented in Table 1. We show the histograms of the pairs binned according to the percentage errors in Figures 4 and 5. We observe that the reprojection method converges faster than the ray-point method. The two bundle adjustments reduces the error of the initial 3D estimate. For the house scene constructed using a stereo and an omni-directional camera, the mean absolute percentage error of the reprojection method is smaller than that of the ray-point. On the contrary in the case of house scene constructed using a pinhole and an omnidirectional camera, ray-point method is slightly better than the reprojection method. We also observe that both the ray-point and the reprojection methods outperform the parametric approach in the house scene using stereo and omnidirectional cameras. However the parametric approach does better when using a pinhole and an omni-directional camera.

| Scene | Cam-1 | Cam-2 | No of Pts | Parametric (Iters, Error) | Ray-Point (Iters, Error) | Reprojection (Iters, Error) |
|---|---|---|---|---|---|---|
| House | stereo | omni | 8 | (8,0.0288) | (26,0.0233) | (7,0.0154) |
| House | pinhole | omni | 8 | (8,0.0288) | (18,0.0305) | (5,0.0413) |

**Table 1.** Statistics for the house scene which has the ground-truth. *Iters* refer to the number of iterations and *Error* refers to the *mean absolute percentage error*.

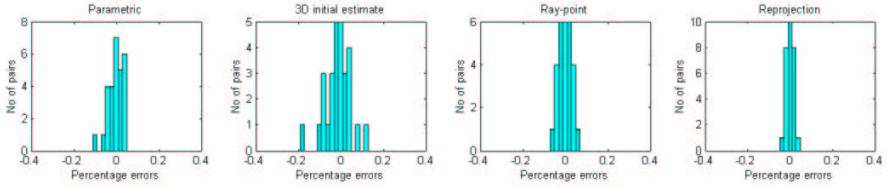



**Fig. 4.** Histogram of the house scene constructed using a stereo and an omnidirectional image. Please note that the number of pairs along y-axis across different diagrams are not same.

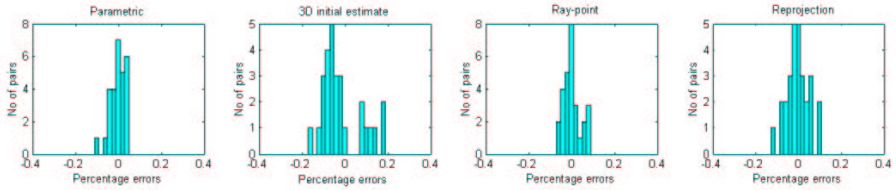



**Fig. 5.** Histogram for the house scene constructed using a pinhole and omnidirectional camera. Please note that the number of pairs along y-axis across different diagrams are not same.

For the objects scene, we also reconstruct 3D scenes by using a pinhole camera and parametric calibration. However as observed earlier, there is no guarantee that parametric reconstruction will be more accurate than ray-point and reprojection methods. In the absence of ground-truth we use *difference* measures between different methods to analyze the performance. Difference measures between method A and method B can be obtained for a pair containing $i_{th}$ and $j_{th}$ points as given below:

$$DM(i,j) = \frac{d(i,j)_A - d(i,j)_B}{d(i,j)_B}$$

where $d(i,j)_A$ and $d(i,j)_B$ are pairwise distances obtained by using methods A and B respectively. Figure 6 shows the histogram binning the pairs according to the difference measures. The statistics are presented in Table 2. In this case, as well, reprojection method converges faster than the ray-point method. We observe that the refinements produced by both the bundle adjustments are comparable to each other. We also show the cross-camera 3D reconstruction of an outdoor scene having 121 3D points in Figure 7.

| Scene | Cam-1 | Cam-2 | No of Pts | Param Iters | Ray-Pt Iters | Reproj Iters | mean absolute difference measure |
|---|---|---|---|---|---|---|---|
| Objects | pinhole | omni | 31 | 7 | 25 | 5 | \|Ray-Pt - Param\|=0.0496<br>\|Reproj - Param\|=0.0544<br>\|Reproj -Ray-Pt\|=0.0069 |

**Table 2.** Statistics for the objects scene which do not have the ground-truth. Here *Iters* refer to the number of iterations.

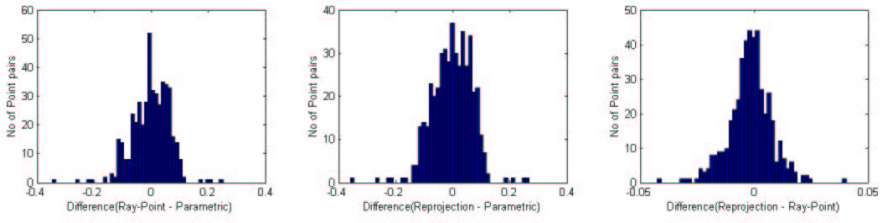
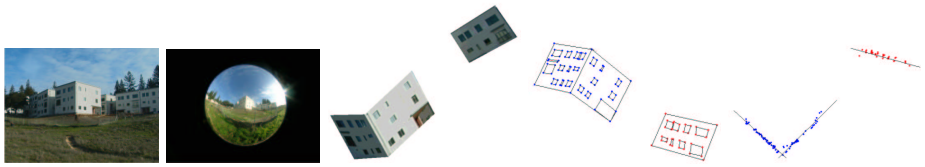



**Fig. 6.** Histograms for the difference errors in the *objects* scene. Please note that the bin sizes along x-axis and the number of pairs along y-axis across different diagrams are not same.



(a) (b) (c) (d) (e)

**Fig. 7.** a) pinhole image b) omnidirectional image c) Texture-mapped model d) Mesh representation e) Top view of the points. We reconstruct 121 3D points, which lie on three walls shown in the images.

## 6 Conclusions and Future Work

We have designed and developed a generic algorithm to mix different cameras in the SfM problem, including the extension of bundle adjustment for generic scenarios. We have obtained promising results for two cross-camera scenarios – pinhole with omnidirecitonal and stereo with omnidirectional. Using simulations and real data, we are interested in investigating our approach and the clustering issues in more exotic catadioptric cameras and multi-camera configurations.

**Acknowledgements.** This work was partially supported by the NSF grant ACI-0222900 and by the Multidisciplinary Research Initiative (MURI) grant by Army Research Office under contract DAA19-00-1-0352. We also like to thank the anonymous reviewers for their valuable suggestions for future work.

## References


1. S. Baker and S. Nayar. A theory of catadioptric image formation. *ICCV*, 1998.
2. H. Bakstein. Non-central cameras for 3D reconstruction. Research Report CTU-CMP-2001-21, Czech Technical University, Prague, 2001.
3. H. Bakstein and T. Pajdla. An overview of non-central cameras. *Computer Vision Winter Workshop*, Ljubljana, Slovenia, 2001.
4. R. Bunschoten and B. Krose. Robust scene reconstruction from an omnidirectional vision system. *IEEE Transactions on Robotics and Automation*. 2002.
5. P. Doubek and T. Svoboda. Reliable 3D reconstruction from a few catadioptric images. *OMNIVIS*, 2002.
6. C. Geyer and K. Daniilidis. A unifying theory of central panoramic systems and practical implications. *ECCV*, 2000.
7. C. Geyer and K. Daniilidis. Paracatadioptric camera calibration. *PAMI*, 2002.
8. C. Geyer and K. Daniilidis. Structure and motion from uncalibrated catadioptric views. *CVPR*, 2001.
9. M.D. Grossberg and S.K. Nayar. A general imaging model and a method for finding its parameters. *ICCV*, 2001.
10. R.M. Haralick, C.N. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *IJCV*, 1994.
11. R.I. Hartley and P. Sturm. Triangulation. *CVIU*, 1997.
12. R.I. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.
13. S.B. Kang and R. Szeliski. 3-D scene data recovery using omnidirectional multibaseline stereo. *IJCV*, 1997.
14. H.C. Longuet-Higgins. A Computer Program for Reconstructing a Scene from Two Projections. *Nature*, 1981.
15. D.G. Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999.
16. J. Mellor. Geometry and texture from thousands of images. *IJCV*, 2003.
17. B. Micusik and T. Pajdla. 3D metric reconstruction from uncalibrated omnidirectional images. *ACCV*, 2004.
18. J. Neumann, C. Fermuller, and Y. Aloimonos. Polydioptric Camera Design and 3D Motion Estimation. *CVPR*, 2003.
19. D. Nister. An Efficient Solution to the Five-Point Relative Pose Problem. *CVPR*, 2003.
20. T. Pajdla. Stereo with oblique cameras. *IJCV*, 2002.
21. S. Peleg, Y.Pritch, and M. Ben-Ezra. Cameras for stereo panoramic imaging. *CVPR*, 2000.
22. R. Pless. Using Many Cameras as One. In *CVPR*, 2003.
23. P. Sturm. Mixing catadioptric and perspective cameras. *OMNIVIS*, 2002.
24. P. Sturm and S. Ramalingam. A generic calibration concept-theory and algorithms. Research Report 5058, INRIA, 2003.
25. P. Sturm and S. Ramalingam. A generic concept for camera calibration. *ECCV*, 2004.
26. R. Swaminathan, M.D. Grossberg, and S.K. Nayar. A perspective on distortions. *CVPR*, 2003.
27. B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle Adjustment, A modern synthesis. *Workshop on Vision Algorithms: Theory and Practice*, 2000.