Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding 103 (2006) 218–228
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

A generic structure-from-motion framework

Srikumar Ramalingam [a,b,*], Suresh K. Lodha [b], Peter Sturm [a]

[a] INRIA Rhône-Alpes, Montbonnot/Grenoble, France
[b] Department of Computer Science, University of California, Santa Cruz, USA

Received 2 February 2005; accepted 7 June 2006

Communicated by Seth Teller

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 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. The proposed methods are evaluated via experiments on two cross-camera scenarios—a pinhole used together with an omni-directional camera and a stereo system used with an omni-directional camera.

Keywords: Calibration; Structure from motion; Motion estimation; Pose estimation; 3D reconstruction; Non-central camera

1. Introduction and motivation

Many different types of cameras including pinhole, stereo, catadioptric, omni-directional and non-central cameras have been used in computer vision. Some of these, especially the omni-directional class, provide more stable ego-motion estimation and larger fields of view than pinhole cameras [2,25,21]. Naturally, a larger field of view allows to reconstruct 3D scenes using fewer images, although the spatial resolution is lower, i.e. pinhole cameras can provide more useful texture maps. Non-central cameras, a review of which is given 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 omni-directional camera (which detects moving objects) and several narrow-field-of-view pan-tilt-zoom cameras that can be used to take close-up pictures of objects. Also while reconstructing complete environments, it is helpful to have a combination of omni-directional and traditional images: the traditional ones (narrow field-of-view, i.e. high spatial resolution) give good accuracy locally, whereas the omni-directional images would be good for registering images scattered throughout the environment to a single reference frame. Despite these advantages, a general, unified, structure-from-motion approach for handling different camera systems, does not exist yet.

* Corresponding author.
E-mail addresses: Srikumar.Ramalingam@inrialpes.fr (S. Ramalingam), lodha@soe.ucsc.edu (S.K. Lodha), Peter.Sturm@inrialpes.fr (P. Sturm).
URLs: http://www.soe.ucsc.edu/~srikumar (S. Ramalingam), http://www.soe.ucsc.edu/~lodha (S.K. Lodha), http://perception.inrialpes.fr/member.php3?id_auteur=24 (P. Sturm).

1077-3142/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.06.006

This statement holds also for camera calibration: most existing calibration methods are parametric and camera dependent [15,9]. For example, in the pinhole camera we use a 3×3 matrix, called calibration matrix, to store the internal parameters of a camera. This matrix along with the camera pose provides the necessary calibration information. Similarly, calibration of optical distortions and of central or non-central catadioptric systems or other omni-directional 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 [12] (cf. also [5,11]): 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 [30,31,28]. 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 3.2–3.4.

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 [17], are generalized in Section 3.2, following [26]. As for triangulation, various algorithms have been proposed for pinhole cameras in [14]. 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 [15,34], 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 [14,34]. 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 use for the general camera model. The second cost function is, as usually desired, based on the re-projection 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 [14], and the fact that it leads to a least-squares type cost function, for which efficient optimization methods exist, such as Gauss–Newton or Levenberg–Marquardt. There is a major challenge in applying this cost function to the general imaging model used here, due to the fact that we have no analytical projection equation, 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 of 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 a stereo system, $k = n$ for oblique cameras [24], 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 rays in every cluster are intersected by a plane to synthesize a perspective image. This allows us to formulate an analytical 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: we have applied the structure-from-motion algorithm to two cross-camera scenarios—a pinhole camera used together with an omni-directional camera, and a stereo system (interpreted as a single non-central camera) used together with an omni-directional camera. We compare the performances with ground truth where available, and 3D reconstruction from pinhole images, obtained using classical techniques.

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 [15]. In the last decade or so, omni-directional cameras and non-central cameras have also been used for 3D reconstruction [3,2,7,16]. Recently, Mičušik et al. extended multi-view metric 3D reconstruction to central fish-eye cameras [20]. 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 [10]. Omni-directional images, with known camera motion, obtained from a GPS or a robot, have also been used in 3D reconstruction [19,4]. 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 apply equally well to all types of cameras—pinhole, stereo, omni-directional, 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 multiview relations for different cameras. Geyer and Daniilidis [8] 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 [29]. Our work is complementary to these efforts in enhancing and promoting the use of cross-camera scenarios for practical applications. This paper is an extended version of [27].

3. Generic structure-from-motion

Fig. 1 describes the pipeline for the proposed generic structure-from-motion approach.

3.1. Generic camera calibration

We use the generic calibration approach developed in [30,31] 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 [12], 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 [30,31], 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 [30,31]).

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 [17]. We briefly summarize how the classical problem is usually solved [15]. 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 eight correspondences or more. In the minimal case of five correspondences, an efficient non-linear minimal algorithm, which gives exactly the theoretical maximum of 10 feasible solutions, was only recently introduced [22]. Once the essential matrix is estimated, the motion parameters R and T can be extracted relatively straightforwardly [22].

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 [15], i.e. 6-vectors X and X'. The epipolar constraint extends naturally to rays, and manifests itself by a 6×6 essential matrix, defined as:

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

The epipolar constraint then writes: $X'^{\mathsf{T}}\mathscr{E}X = 0$ [26]. Linear estimation of $\mathscr{E}$ requires 17 correspondences. Once $\mathscr{E}$ is estimated, motion can again be extracted straightforwardly: R can simply be read off $\mathscr{E}$, as the upper left or lower right 3×3 sub-matrix, or the average of both. The obtained R will usually not obey the orthonormality constraints of a rotation matrix. We correct this by computing the orthonormal matrices that are closest to the original matrices (in the sense of the Frobenius norm). This can be done in the following way. Let the SVD of the estimated R be given by $R = U\Sigma V^{\mathsf{T}}$. An orthonormal estimate for the rotation matrix R is then given by UV^{T}, plus possibly a multiplication of the whole matrix by -1, to make its determinant equal to $+1$ (otherwise, the recovered matrix represents a reflection and not a rotation). This approximation is also reasonable because we anyway refine the rotation matrix using bundle adjustment.

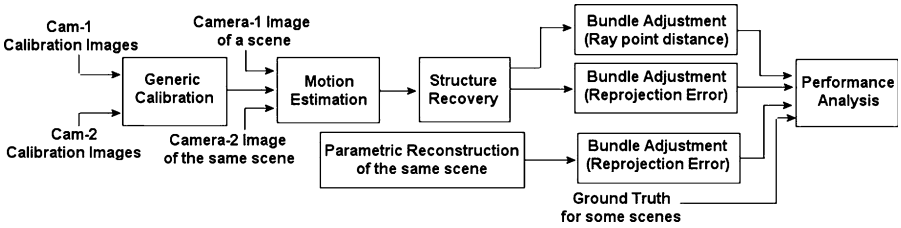

Fig. 1. The overall pipeline of the generic structure-from-motion approach.

The next step is the computation of the translation component T. Note that 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). Later in Section 5 we will observe a scenario with stereo camera and a central omni-directional camera. Since the stereo camera (by considering it as a single general camera) models a non-central camera we automatically extract the scale information during the motion estimation. However in experiments involving only central systems, we need to use some knowledge about the scene to obtain the scale information. In any case the evaluation methods are independent of the absolute scale of the scene.

Estimation of T can be done as follows: $\mathscr{E}$ is usually estimated up to scale, and we first eliminate this ambiguity. Let A and B be the upper left and lower right 3×3 submatrices of $\mathscr{E}$. We estimate a scale factor λ, that minimizes the sum of the squared Frobenius norms of $\lambda A - R$ and $\lambda B - R$. This is a simple linear least squares problem. Then, multiply $\mathscr{E}$ with λ and let C be the upper right 3×3 submatrix of the product. We compute T as the vector that minimizes the Frobenius norm of $C + [T]_\times R$. This is again a linear least squares problem.

Other algorithms for computing R and T from $\mathscr{E}$ are possible of course, but in any case, the computation may be followed by a non-linear optimization of R and T (by carrying out the associated sub-part of a bundle adjustment). Also note that the theoretical minimum number of required correspondences for motion estimation is 6 instead of 5 (due to the absence of the scale ambiguity), and that it might be possible, though very involved, to derive a minimal 6-point method along the lines of [22]. More details on motion estimation are available in [32].

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 ith ray using a starting point A_i and the direction, represented by a unit vector B_i. We apply the mid-point method [14,26], i.e. determine P that is closest in average to the n rays. Let us represent generic points on rays using position parameters λ_i. Then, P is determined by minimizing the following expression over X, Y, Z and the λ_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:

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

where $\mathtt{I}_3$ is the identity matrix of size 3×3. 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 for example [34]. Here, we even get a closed-form solution, based on:

$$M^{-1} = \begin{pmatrix} \frac{1}{n}\{\mathtt{I}_3 + BB^\mathsf{T}C^{-1}\} & C^{-1}B \\ B^\mathsf{T}C^{-1} & \mathtt{I}_n + B^\mathsf{T}C^{-1}B \end{pmatrix},$$

where $B = (B_1 \cdots B_n)_{3\times n}$ and $C = n\mathtt{I}_3 - BB^\mathsf{T}$.

The closed-form solution for P (C^{-1} can be computed in closed-form) is then:

$$P = \frac{1}{n}\{\mathtt{I}_3 + BB^\mathsf{T}C^{-1}\}\sum_{i=1}^{n} A_i - C^{-1}\sum_{i=1}^{n} B_i B_i^\mathsf{T} A_i.$$

To summarize, 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 3×3 matrix.

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. A literature review on algorithms for pinhole cameras is given in [13]. 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 via scalars λ_i, like in the previous section: $A_i + \lambda_i B_i$.

We know the structure of the observed object, i.e. we know the mutual distances d_{ij} between the 3D points. We can thus write equations on the unknowns λ_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 three equations that are quadratic in three 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 eight theoretical solutions. For pinhole cameras, at least four of them can be eliminated because they would correspond to points lying behind the camera [13], a concept that is not applicable (at least in a direct way) to non-central cameras. In any case, a unique solution can be obtained using one or two

additional points [13]. More details on pose estimation for non-central cameras are given in [6,23].

4. Bundle adjustment

4.1. Ray-point bundle adjustment

This technique minimizes the distance between projection rays and 3D points, over camera motion and 3D structure. We briefly describe our cost function. Let $C_j = (X_j, Y_j, Z_j)^{\mathsf{T}}$ be the 3D coordinates of the jth point. Consider the ith image and assume that the projection ray corresponding to C_j is the kth ray of the camera. Let this ray be represented like above by a base point A_k and a direction B_k (B_k is chosen to have unit norm). Note that here, we assume these are known, since we consider calibrated cameras. Let R_i and T_i be the pose of the camera for the ith image. Then, points on the considered projection ray are represented by a scalar λ:

$$A_k + T_i + \lambda R_i B_k.$$

We now seek to compute the (squared) distance between this ray and the point C_j. It is given by:

$$e_{ijk} = \min_{\lambda_{ijk}} \|A_k + T_i + \lambda_{ijk} R_i B_k - C_j\|^2.$$

It can easily be computed in closed-form; the λ_{ijk} minimizing the above expression is:

$$\lambda_{ijk} = B_k^{\mathsf{T}} R_i^{\mathsf{T}} (C_j - A_k - T_i).$$

Bundle adjustment consists then in minimizing the sum of all squared distances e_{ijk} (for all available matches between points and pixels/rays), over the 3D point positions and the camera motions. This is a non-linear least squares problem, and appropriate optimization methods such as Gauss–Newton or Levenberg–Marquardt may be used for its solution.

Note that this bundle adjustment is completely generic: due to working with projection rays, it may be applied to any calibrated camera, be it central or non-central. One might include the calibration in the optimization and minimize the cost function also over projection ray coordinates (in that case, the representation using a base point and a direction may not necessarily be the best choice). This is easy to write down and implement, but one needs sufficient data to get meaningful estimates: in a fully non-central model for example, each estimated ray needs at least two associated 3D points, i.e. the pixel associated with that ray, has to correspond to actual interest points in at least two images. This can only be achieved for sufficiently many rays if a reliable *dense* matching is possible.

4.2. Re-projection-based bundle adjustment

We now describe some challenges in using re-projection-based bundle adjustment for the generic imaging model and our approaches to overcome these. In the generic imaging model, there is no analytical projection equation, since calibration more or less corresponds to a lookup table that gives projection ray coordinates for individual pixels (or, image points). Thus, to project a 3D point, search and interpolation are required: one searches for a certain number (could be equal to 1) among the camera's projection rays that are closest to the 3D point. The coordinates of the image point can then be computed by interpolating the coordinates of the pixels associated with these rays. Efficient optimization for re-projection-based bundle adjustment would require the computation of derivatives of this projection function; although numerical differentiation is possible and rather straightforward, it is time-consuming.

We solve this problem by considering a camera as a cluster of central cameras: given a set of rays belonging to a non-central camera, we partition them into k clusters of rays, each having its own optical center. For example, $k = 2$ for a stereo system. In addition we also impose the condition that each ray should be contained by only one cluster. In the following, we describe a simple clustering method and then, how we perform bundle adjustment over these ray clusters.

The clustering is obtained using a 3D Hough transform (mapping rays in 3D to 3D points), which we explain briefly. First we transform the "ray space," consisting of rays in space, to a discretized "point space," where we use a counter (initialized to zero) for every 3D point. Then every ray updates the counters (increase by 1) of the 3D points lying on it. Next we identify the 3D point having the largest count. This point becomes the center of the first cluster and the rays that contributed to its count, are grouped to form the cluster. The contribution of these rays to other points' count is then deleted, and the process repeated to determine other clusters. With a reasonably good resolution for the point space in 3D Hough transform, we can obtain the correct number of clusters in simple configurations such as stereo camera and multi-camera network, where the centers are distinct. However in catadioptric systems having complex caustics, the resolution of 3D point space in Hough transform determines the number of discrete clusters we can obtain. Each such cluster is in the following interpreted as a central camera. We synthesize a *perspective* image for each one of them, that will be used in the parameterization for the bundle adjustment. A perspective image for a cluster of rays, can be easily computed by intersecting the rays with some properly chosen plane, henceforth denoted as image plane (cf. [30]). We thus generate k perspective images, one per cluster of rays. Each of them is parameterized by the position of its optical center (the center point of the cluster), the directions of the projection rays and the position of the image plane. We have thus created a parameterization for an analytical projection equation from 3D points to 2D coordinates (instead of only a lookup table between rays and pixels). It is used in bundle adjustment to compute and minimize the re-projection error simultaneously on all these synthesized images.

We now briefly describe how to choose an image plane for a cluster of rays. To do so, we propose to minimize the "uncertainty" in the intersection points of image plane and rays: ideally the rays should be perpendicular to the plane, and therefore we find the plane's orientation which minimizes the sum of all acute angles between the plane and rays:

$$\min_{m_1,m_2,m_3} \sum_{i=1}^{n} (m_1 l_1^i + m_2 l_2^i + m_3 l_3^i)^2,$$

where (l_1^i, l_2^i, l_3^i) refers to the direction of the ith ray (unit vector) and (m_1, m_2, m_3) is the normal of the image plane. The normal is given as the unit null-vector of the matrix:

$$\begin{pmatrix} \sum (l_1^i)^2 & \sum l_1^i l_2^i & \sum l_1^i l_3^i \\ \sum l_1^i l_2^i & \sum (l_2^i)^2 & \sum l_2^i l_3^i \\ \sum l_1^i l_3^i & \sum l_2^i l_3^i & \sum (l_3^i)^2 \end{pmatrix}.$$

The distance between the image plane and the center of the cluster does not matter as long as we keep it the same for all clusters. Thus we place the image planes at the same distance for all the individual clusters.

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. For example in a completely non-central camera, which is referred to as an oblique camera [24], where each ray belongs to a separate central cluster. In that case we consider a plane perpendicular to that ray and the center will be kept at infinity. Our re-projection-based algorithm will be exactly the same as a ray-point approach.

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. Finally 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. In addition to allowing the use of a re-projection based approach, our clustering technique makes a compromise between fully central (stability) and fully non-central (generality).

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.

In general non-central omni-directional cameras are constructed using mirrors and lenses. These catadioptric configurations, constructed using spherical, parabolic and hyperbolic mirrors, are either central or approximately central. The second scenario can either be approximated to a central camera or accurately modeled using a large number of clusters. On following the second option we observe the following. First, it is very difficult to cluster in the presence of noise. Second, the bundle adjustment is more or less the same as the ray-point one. Thus it was not necessary for us to demonstrate the clustering for non-central omni-directional cameras. More precisely the re-projection based approach is meaningful only to non-central configurations with distinct clusters such as stereo and multi-camera scenarios.

5. Results and analysis

We consider three indoor scenarios:

- A house scene captured by an omni-directional camera and a stereo system (cf. Fig. 4(b)).
- A house scene captured by an omni-directional and a pinhole camera (same scene as in Fig. 4(b)).
- An objects scene, which consists of a set of objects placed in random positions as shown in Fig. 4(a), captured by an omni-directional and a pinhole camera.

The following cameras were used: Nikon Coolpix 5400 as pinhole camera, the "Bumblebee stereo camera," and the Nikon Coolpix 5400 with an "FC-E8" fisheye converter to give omni-directional images with a field of view of $360° \times 183°$.

We first briefly describe how the cameras used were calibrated, and then present experiments and results with the algorithms described in this paper.

5.1. Calibration

We calibrate three types of cameras in this work. They are pinhole, stereo, and omni-directional systems. Sample calibration images for these are shown in Fig. 2 and some visual calibration information is given in Fig. 3.

5.1.1. Pinhole camera

Fig. 3(a) shows the calibration of a regular digital camera using the single center assumption [31].

5.1.2. Stereo system

Here we calibrate the left and right cameras separately as two individual central cameras. In the second step we capture images of a 3D scene and compute the motion between the two cameras using the technique described in Section 3.2. Finally, using the computed motion we obtain the rays of the two cameras in the same coordinate system, which thus constitutes the calibration information for this non-central system.

5.1.3. Omni-directional camera

We assume the camera to be central. Fig. 3(c) shows that we have used more than three calibration grids to calibrate the camera, which is due to the fact that the minimum required number of three images is seldom sufficient to completely calibrate the whole field of view.

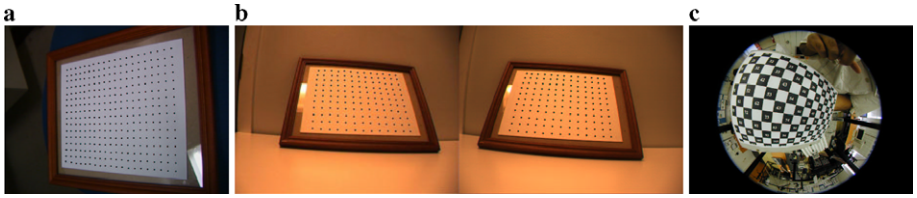

Fig. 2. Sample calibration images (not necessarily the ones used for the calibration, as shown in Fig. 3). For (a) pinhole and (b) stereo, circular calibration targets are used. For (c) omni-directional, checkerboard grids are used.

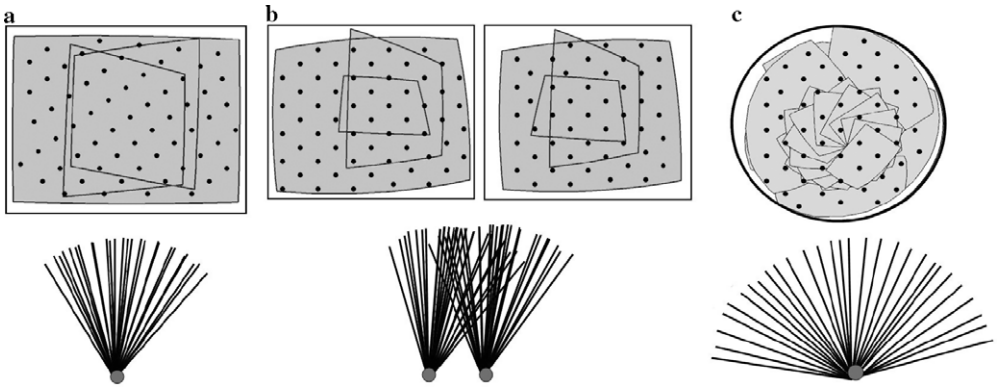

Fig. 3. Calibration information. (a) Pinhole. (b) Stereo. (c) Omni-directional. The shading shows the calibrated regions, i.e. the regions of pixels for which projection rays were determined. The 3D rays shown on the bottom correspond to the image pixels marked in black. We also show the outlines of the calibration grids (enclosing the image pixels).

Thus we placed a checkerboard grid, shown in Fig. 2(c), on a turntable and captured a sequence of images to cover the entire field of view. Then we used a few overlapping images to obtain a partial initial calibration [31]. This provides, in a single coordinate system: the pose of the calibration grid for the images used, the position of the camera's optical center and the direction of projection rays for the pixels in the overlap region. Then, for each calibration grid whose pose has been determined, we can compute projection rays for all pixels covered by that grid's image: the rays are simply given by joining the camera's optical center and the point on the grid corresponding to the pixels under consideration. Furthermore, the pose of further calibration grids can be computed, as soon as they cover sufficiently many pixels with already known projection rays (the pose estimation method of Section 3.4 is used). This process of alternating between grid pose estimation and projection ray computation, is repeated until all grid poses have been determined. Finally, all poses are refined using the ray-point bundle adjustment algorithm explained in Section 4.1. The calibrated image region shown in Fig. 3(c) was obtained using 23 images.

5.2. Motion and structure recovery

Two scenarios are considered here: combining an omni-directional camera with either a pinhole camera or a stereo system.

5.2.1. Pinhole and omni-directional

Since the omni-directional camera has a very large field of view and consequently lower resolution compared to the pinhole camera, the images taken from close viewpoints from these two cameras have different resolutions as shown in Fig. 4(a). This poses a problem in finding correspondences between images. Operators like SIFT [18], are scale invariant, but not fully 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. From the matched points, we triangulated the 3D structure. The result suggests that the algorithms used here (calibration, motion estimation, and triangulation) are correct and work in practice.

5.2.2. Stereo system and omni-directional

Here, we treat the stereo system as a single, non-central camera; the same procedure as for the above case are applied: manual matching, motion estimation, triangulation. The only difference is that the same scene point may appear twice in the stereo camera, but this makes no difference for our algorithms. Although a simple 3D structure is used here, the result again suggests that the algorithms are correct. This experiment underlines the fact that they are generic, i.e. may be used for any camera and combination of cameras that are modeled by the generic imaging model.

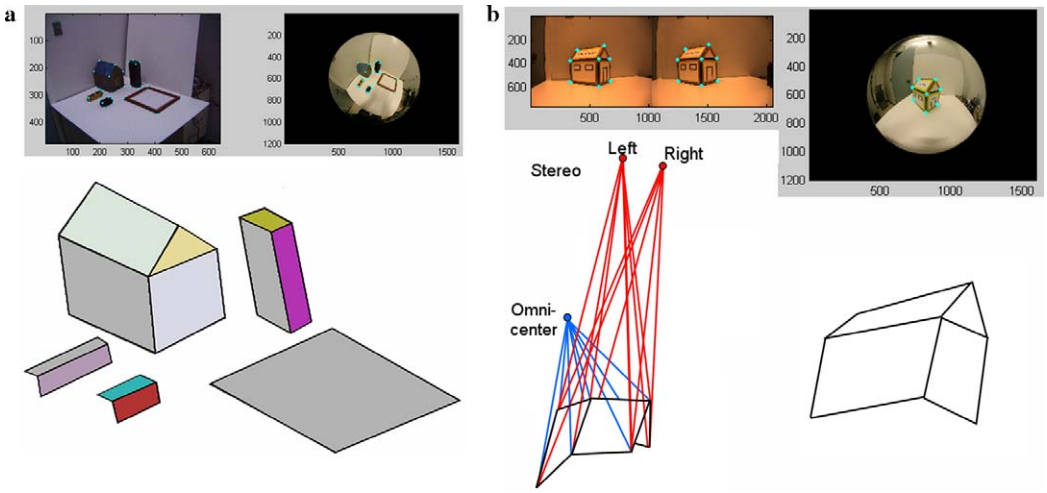

Fig. 4. Results of motion estimation and 3D reconstruction for cross-camera scenarios. (a) Pinhole and omni-directional. (b) Stereo and omni-directional. Shown are the reconstructed 3D points, the optical centers (computed by motion estimation) and the projection rays used for triangulation.

5.3. Pose estimation

We conducted a simple experiment to test the accuracy of the pose estimation algorithm, described in Section 3.4. A calibration grid was placed on a turntable in near to vertical position. We captured omni-directional images of the grid at 14 different rotation angles; a sample image is shown in Fig. 2(c). The grid's pose at each of the 14 positions was computed (the camera was previously calibrated, as explained in Section 5.1). This is shown in Figs. 5(a) and (b). Fig. 5(c) shows the extension of a line in the grid's coordinate system, for the different poses. Due to the turntable motion, these should envelope a quadric close to a cone, which indeed is the case. A complete quantitative analysis is difficult, but we evaluated how close the trajectories of individual grid points are to being circular (as they should be, due to the turntable motion). The least-squares circle fit for one of the grid points, from its 14 recovered positions, is shown in Fig. 5(d). The least-squares fit error was found to be as low as 0.64% with respect to the overall scene size (largest distance between two grid points in this scene).

5.4. 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. Accuracy is measured as follows: the reconstructed 3D points are first scaled such that the sum of squared distances from their centroid equals 1. This way, the accuracy measurements become relative to scene size. Then, we compute all possible pairwise distances between reconstructed 3D points. These are then compared to ground truth values if available. We also compare them to the analogous information obtained from 3D reconstruction using pinhole images only and classical structure-from-motion methods: motion estimation, triangulation and re-projection-based bundle adjustment for perspective cameras [15].

5.4.1. House scene

For the house scene (cf. Fig. 4(b)), ground truth is available (manual measurement of distances). We compute the relative error between reconstructed distances d_{ij} and ground truth distances $\bar{d}_{ij}$ between all pairs (i,j) of 3D points:

$$\frac{|d_{ij} - \bar{d}_{ij}|}{\bar{d}_{ij}}.$$

Table 1 shows the mean of these relative errors, given in percent. Values are shown for three camera setups: omni-directional image combined with a pinhole or a stereo system, and two pinhole images. Three methods are evaluated: classical (perspective) algorithms (called "Parametric"

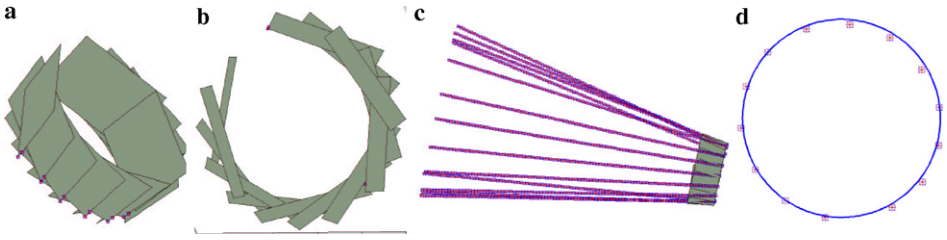

Fig. 5. Experiment on pose estimation. (a and b) Estimated poses of calibration grid in 14 positions. (c) Extensions of a line on the calibration grid, in all 14 positions. (d) Least squares circle fit to the estimated positions of one grid point.

Table 1
Statistics for the house scene

Scene	Points	Camera 1	Camera 2	Parametric (it, error)	Ray-point (it, error)	Re-projection (it, error)
House	8	Stereo	Omni	—	(26, 2.33)	(7, 1.54)
House	8	Pinhole	Omni	—	(18, 3.05)	(5, 4.13)
House	8	Pinhole	Pinhole	(8, 2.88)	—	—

it refers to the number of iterations of bundle adjustment and *error* refers to the *mean relative error* on distances between 3D points, expressed in percent.

here), and generic algorithms, with the two different bundle adjustment methods (“Ray-Point” and “Re-projection”). Histograms giving the distribution of relative distance errors are also shown, in Fig. 6.

As for the two generic bundle adjustment methods, we observe that the re-projection method converges faster than the ray-point method. Both bundle adjustments reduce the error in the initial 3D estimate (after motion estimation and triangulation) significantly. As for the accuracy, each one of the two bundle adjustments is better than the other one in one scenario.

We also observe that the generic approaches perform better than the classical parametric one in the case they use an omni-directional camera and a stereo system; this is not surprising since one more image is used than the two pinhole images of the classical approach. Another possible reason might be the use of more number of parameters as compared to classical approaches. Thus they will have a good local minima. Nevertheless, this again confirms the correctness and applicability of our generic approaches. It is no surprise either that performance is worse for the combination of a pinhole and an omni-directional image, since the spatial resolution of the omni-directional image is much lower than those of the pinhole images.

5.4.2. Objects scene

For this scene (cf. Fig. 4(a)), no complete ground truth is available. We thus computed the differences between point distances obtained in reconstructions with the three methods. Concretely, for some methods X and Y, we compute, for all point pairs (i,j):

$$\frac{|d_{ij}^X - d_{ij}^Y|}{d_{ij}^Y},$$

where d_{ij}^X respectively d_{ij}^Y are pairwise distances obtained by using methods X and Y, respectively. Fig. 7 shows the histograms for this measure and Table 2 gives some details on this scene and the number of iterations for the different methods. In this scenario as well, re-projection method converges faster than the ray-point method.

The mean values of the above measure are as follows:

X = Ray-point Y = Parametric $\rightarrow$ 4.96

X = Re-projection Y = Parametric $\rightarrow$ 5.44

X = Re-projection Y = Ray-point $\rightarrow$ 0.69

We observe that the refinements produced by both bundle adjustments seem to be comparable to each other.

5.4.3. Outdoor scene

Fig. 8 shows results for a 3D reconstruction of an outdoor scene from two images, one omni-directional and the other pinhole. The reconstruction has 121 3D points. Fig. 8c–e allow a qualitative evaluation of the reconstruction, e.g. reasonable recovery of right angles (between window edges or between walls). We analyzed the reconstruction quantitatively, by measuring the deviations

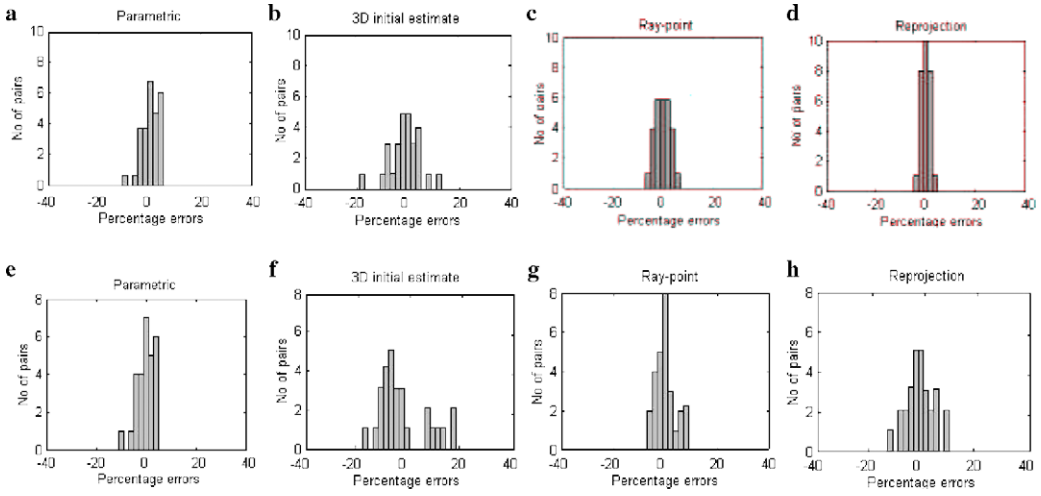

Fig. 6. Histograms for the house scene. Top: results for the combination of stereo and an omni-directional image (besides for the left column, where two pinhole images are used). Bottom: combination of a pinhole and an omni-directional image. Please note that the different graphs are scaled differently along the y-axis.

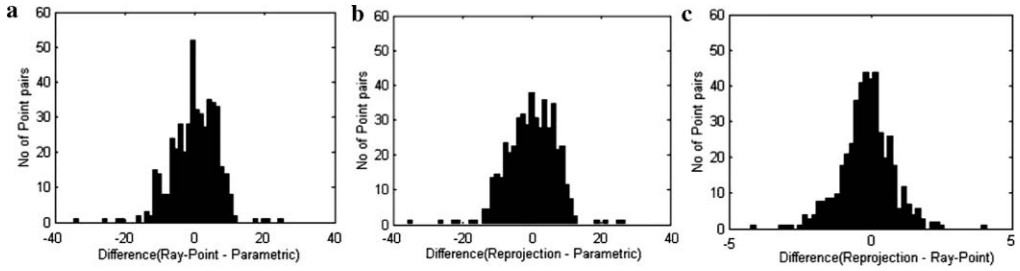

Fig. 7. Histograms for the relative distance errors for the objects scene. Please note that the histograms are scaled differently along both axes.

Table 2
Details on the objects scene

Scene	Points	Camera 1	Camera 2	Parametric	Ray-point	Re-projection
Objects	31	Pinhole	Omni	7	25	5

The last three columns give the number of iterations of bundle adjustment for the three methods used.

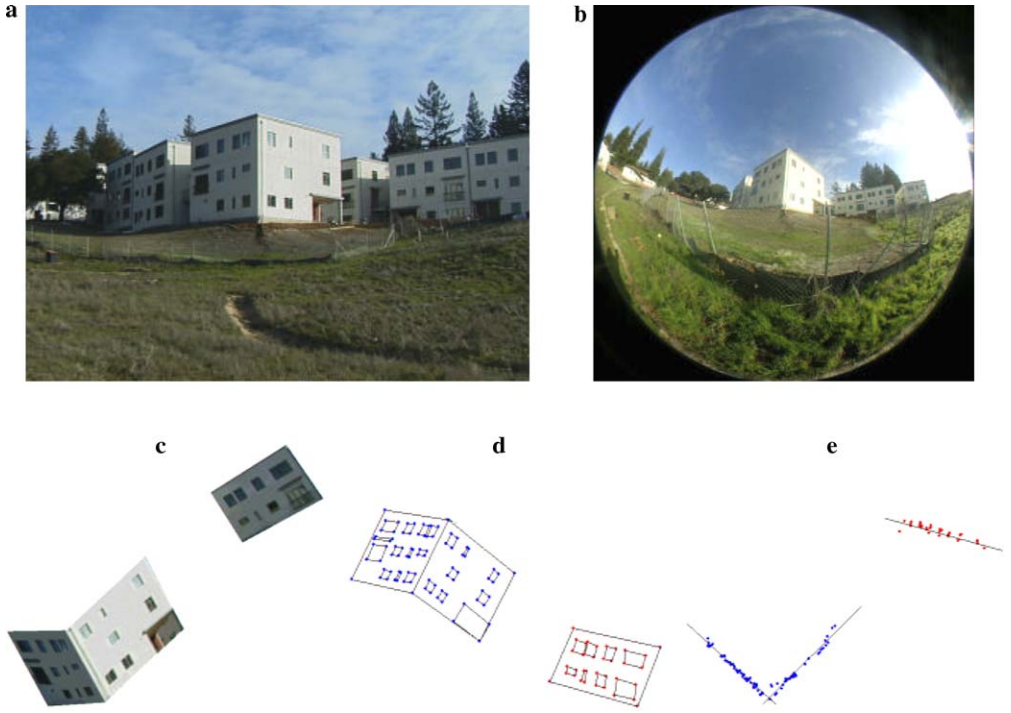

Fig. 8. Outdoor scene. (a) Pinhole image. (b) Omni-directional image. (c) Texture-mapped model. (d) Mesh representation. (e) Top view of the points. We reconstructed 121 3D points, which lie on three walls shown in the images.

from right angles and from coplanarity for appropriate sets of points. To do so, we computed a least-squares plane for coplanar points and measured the residual distances. We then compute the mean distance, and express it relative to the overall size of the scene (largest distance between two points in the scene).

We also measure the angle between planes that ideally should be orthogonal, and consider the deviation from 90°. The errors are found to be low (cf. Table 3), considering that the images are certainly not ideal for the reconstruction task. Table 3 also contains these error measures for the house and objects scenes used above.

Table 3
ΔPlanarity and ΔOrthogonality refer to the mean residual for the least squares plane fit (relative to scene size and expressed in percent) and to the mean errors of deviations from right angles (see text for more details)

Scene	Camera 1	Camera 2	ΔPlanarity (Ray-point, Re-projection)	ΔOrthogonality (Ray-point, Re-projection)
House	Stereo	Omni	(0.37, 0.27)	(5.1, 3.5)
Objects	Pinhole	Omni	(0.38, 0.42)	(3.8, 4.14)
Outdoor	Pinhole	Omni	(0.59, 0.63)	(4.2, 5.4)

6. Conclusions

We have designed and developed a generic approach for structure-from-motion, that works for any camera or mixture of cameras that fall into the generic imaging model used. Our approach includes methods for motion and pose estimation, 3D point triangulation and bundle adjustment. Promising results have been obtained for different image sets, obtained with three different cameras: pinhole, omni-directional (fisheye) and a stereo system. 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.

Acknowledgments

This work was partially supported by the following grants: NSF grant ACI-0222900, Multidisciplinary Research Initiative (MURI) grant by Army Research Office under contract DAA19-00-1-0352 and grant from the European Community under the EST Marie-Curie project Visitor. We are very thankful to Tomáš Pajdla and Branislav Mičušik for the data. We thank the anonymous reviewers for valuable feedback.

References

[1] S. Baker, S. Nayar. A theory of catadioptric image formation, in: Internat. Conf. on Computer Vision, Bombay, India, 1998, pp. 35–42.
[2] H. Bakstein, Non-central cameras for 3D reconstruction. Technical Report CTU-CMP-2001-21, Czech Technical University, Prague, 2001.
[3] H. Bakstein, T. Pajdla, An overview of non-central cameras, in: Computer Vision Winter Workshop, Ljubljana, Slovenia, 2001.
[4] R. Bunschoten, B. Krose, Robust scene reconstruction from an omnidirectional vision system, IEEE Trans. Robotics Autom. 19 (2) (2003) 351–357.
[5] G. Champleboux, S. Lavallée, P. Sautot, P. Cinquin, Accurate calibration of cameras and range imaging sensors: the NPBS Method, in: Internat. Conf. on Robotics and Automation, Nice, France, 1992, pp. 1552–1558.
[6] C.S. Chen, W.Y. Chang, On pose recovery for generalized visual sensors, IEEE Trans. Pattern Anal. Mach. Intell. 26 (7) (2004) 848–861.
[7] P. Doubek, T. Svoboda, Reliable 3D reconstruction from a few catadioptric images, in: Workshop on Omni-directional Vision, Copenhagen, Denmark, 2002, pp. 71–78.
[8] C. Geyer, K. Daniilidis, A unifying theory of central panoramic systems and practical implications, in: Eur. Conf. on Computer Vision, Dublin, Ireland, 2000, pp. 445–461.
[9] C. Geyer, K. Daniilidis, Paracatadioptric camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 687–695.
[10] C. Geyer, K. Daniilidis, Structure and motion from uncalibrated catadioptric views, in: Conf. on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, 2001, pp. 279–286.
[11] K.D. Gremban, C.E. Thorpe, T. Kanade. Geometric camera calibration using systems of linear equations, in: Internat. Conf. on Robotics and Automation, Philadelphia, Pennsylvania, USA, 1988, pp. 562–567.
[12] M.D. Grossberg, S.K. Nayar, A general imaging model and a method for finding its parameters, in: Internat. Conf. on Computer Vision, Vancouver, Canada, 2001, pp. 108–115.
[13] R.M. Haralick, C.N. Lee, K. Ottenberg, M. Nolle, Review and analysis of solutions of the three point perspective pose estimation problem, Internat. J. Comput. Vision 13 (3) (1994) 331–356.
[14] R.I. Hartley, P. Sturm, Triangulation, Comput. Vision Image Understand. 68 (2) (1997) 146–157.
[15] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, 2000.
[16] S.B. Kang, R. Szeliski, 3-D scene data recovery using omnidirectional multibaseline stereo, Internat. J. Comput. Vision 25 (2) (1997) 167–183.
[17] H.C. Longuet-Higgins, A computer program for reconstructing a scene from two projections, Nature 293 (1981) 133–135.
[18] D.G. Lowe, Object recognition from local scale-invariant features, in: Internat. Conf. on Computer Vision, Kerkyra, Greece, 1999, pp. 1150–1157.
[19] J. Mellor, Geometry and texture from thousands of images, Internat. J. Comput. Vision 51 (1) (2003) 5–35.
[20] B. Mičušik, D. Martinec, T. Pajdla, 3D metric reconstruction from uncalibrated omni-directional images, in: Asian Conf. on Computer Vision, Jeju Island, Korea, 2004.
[21] J. Neumann, C. Fermüller, Y. Aloimonos, Polydioptric camera design and 3D motion estimation, in: Conf. Computer Vision and Pattern Recognition, Madison, Wisconsin, USA, 2003, pp. 294–301.
[22] D. Nistér, An efficient solution to the five-point relative pose problem, Conf. on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA, 2003, pp. 195–202.
[23] D. Nistér, A minimal solution to the generalized 3-point pose problem, in: Conf. on Computer Vision and Pattern Recognition, Washington, USA, 2004, pp. 560–567.
[24] T. Pajdla, Stereo with oblique cameras, Internat. J. Comput. Vision 47 (1) (2002).
[25] S. Peleg, Y.Pritch, M. Ben-Ezra, Cameras for stereo panoramic imaging, in: Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA, 2000, pp. 208–214.
[26] R. Pless, Using many cameras as one, in: Conf. on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA, 2003, pp. 587–593.
[27] S. Ramalingam, S.K. Lodha, P. Sturm, A generic structure-from-motion algorithm for cross-camera scenarios, in: Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras, Prague, Czech Republic, 2004, pp. 175–186.
[28] S. Ramalingam, P. Sturm, S.K. Lodha, Towards complete generic camera calibration. Conference on Computer Vision and Pattern Recognition, San Diego, USA, 2005, pp. 1093–1098.
[29] P. Sturm, Mixing catadioptric and perspective cameras, in: Workshop on Omni-directional Vision, Copenhagen, Denmark, 2002, pp. 60–67.
[30] P. Sturm, S. Ramalingam, A generic calibration concept-theory and algorithms, Research Report 5058, INRIA, 2003.
[31] P. Sturm, S. Ramalingam, A generic concept for camera calibration, Eur. Conf. on Computer Vision, Prague, Czech Republic, 2004, pp. 1–13.
[32] P. Sturm, Multiview geometry for general camera models, Conf. on Computer Vision and Pattern Recognition, San Diego, USA, 2005, pp. 206–212.
[33] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle adjustment, A modern synthesis, in: B. Triggs, A. Zisserman, R. Szeliski (Eds), Workshop on Vision Algorithms: Theory and Practice, Springer Verlag, 2000, pp. 298–375.