Motion Estimation for Nonoverlapping Multicamera Rigs: Linear Algebraic and L_∞ Geometric Solutions

Jae-Hak Kim, *Member*, *IEEE*, Hongdong Li, *Member*, *IEEE*, and
Richard Hartley, *Fellow*, *IEEE*

Abstract—We investigate the problem of estimating the ego-motion of a multicamera rig from two positions of the rig. We describe and compare two new algorithms for finding the 6 degrees of freedom (3 for rotation and 3 for translation) of the motion. One algorithm gives a linear solution and the other is a geometric algorithm that minimizes the maximum measurement error—the optimal L_∞ solution. They are described in the context of the General Camera Model (GCM), and we pay particular attention to multicamera systems in which the cameras have nonoverlapping or minimally overlapping field of view. Many nonlinear algorithms have been developed to solve the multicamera motion estimation problem. However, no linear solution or guaranteed optimal geometric solution has previously been proposed. We made two contributions: 1) a fast linear algebraic method using the GCM and 2) a guaranteed globally optimal algorithm based on the L_∞ geometric error using the branch-and-bound technique. In deriving the linear method using the GCM, we give a detailed analysis of degeneracy of camera configurations. In finding the globally optimal solution, we apply a rotation space search technique recently proposed by Hartley and Kahl. Our experiments conducted on both synthetic and real data have shown excellent results.

Index Terms—Multicamera rigs, generalized camera, motion estimation, epipolar equation, branch and bound, linear programming.

1 Introduction

Motion estimation using a monocular or stereo camera has been studied for decades in computer vision. Most research have focused on the central projection camera model (or pinhole camera model). Recently, interest in exploiting the Generalized Camera Model (or generic imaging model) has increased substantially, partly because many new nonconventional camera models (for example, catadioptric camera) have been developed. An important example of the Generalized Camera Model is a rigidly coupled multiple camera system attached to a moving vehicle or platform. This type of camera setup is receiving more and more attention because of its use in applications such as urban mapping and model building. In Fig. 1a, for instance, a set of cameras mounted on a vehicle can be used to find the position of the vehicle by using information obtained from images captured by the mounted cameras; Fig. 1b shows another commercial six-camera system—PTGrey's Ladybug camera. The overlap between the fields of view of the six cameras is minimal.

Nonoverlapping multicamera rigs are of particular interest in practice. Because the component cameras have little or no overlap in their fields of views, the effective overall field of view is wider, leading to efficient data acquisition. A further benefit of the wide field of view is that the ego-motion may be determined with greater accuracy. It is important and interesting to note that most conventional binocular stereo heads are not instances of such nonoverlapping multicamera rigs because the two cameras usually share a large portion of the field of view. In this paper, we are interested in a system consisting of multiple rigidly configured pinhole cameras with possibly nonoverlapping fields of view, such as the two examples shown in Fig. 1.

In the most general setting, a multicamera rig can be treated as a special case of the General Camera Model (GCM). The GCM is an abstract imaging model, which generalizes many nonconventional nonpinhole cameras, including multicamera rig systems. Specifically, it replaces the conventional concept of image pixels by a set of unconstrained image rays. Grossberg and Nayar in [1] first gave a systematic study of this imaging model. A hierarchical analysis of general camera models in terms of multiview geometry is given by Sturm in [2]. Pless [3] has studied how to use this GCM to derive motion estimation algorithm using the multiple camera rigs, which is the subject of this paper. He proposed two linear algorithms, one for the discrete motion case and the other for the continuous motion case. However, the particular case of nonoverlapping cameras was not fully explored. As will be made clear in this paper, his algorithms do not apply directly to this important type of GCM because of a degeneracy in the set of linear equations obtained using his method. We propose a modification to his method that applies to the nonoverlapping camera configuration.

- *J.-H. Kim is with the Department of Computer Science, Queen Mary, University of London, Mile End Road, London E1 4NS, United Kingdom. E-mail: jaehak@dcs.qmul.ac.uk.*
- *H. Li and R. Hartley are with the Department of Information Engineering, Research School of Information Sciences and Engineering, Australian National University, Canberra, ACT 0200, Australia, and the National ICT Australia (NICTA). Email: {Hongdong.Li, Richard.Hartley}@anu.edu.au.*

Manuscript received 9 Aug. 2008; revised 5 Mar. 2009; accepted 31 Mar. 2009; published online 9 Apr. 2009.
Recommended for acceptance by F. Kahl.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-08-0479.
Digital Object Identifier no. 10.1109/TPAMI.2009.82.

0162-8828/10/$26.00 © 2010 IEEE Published by the IEEE Computer Society

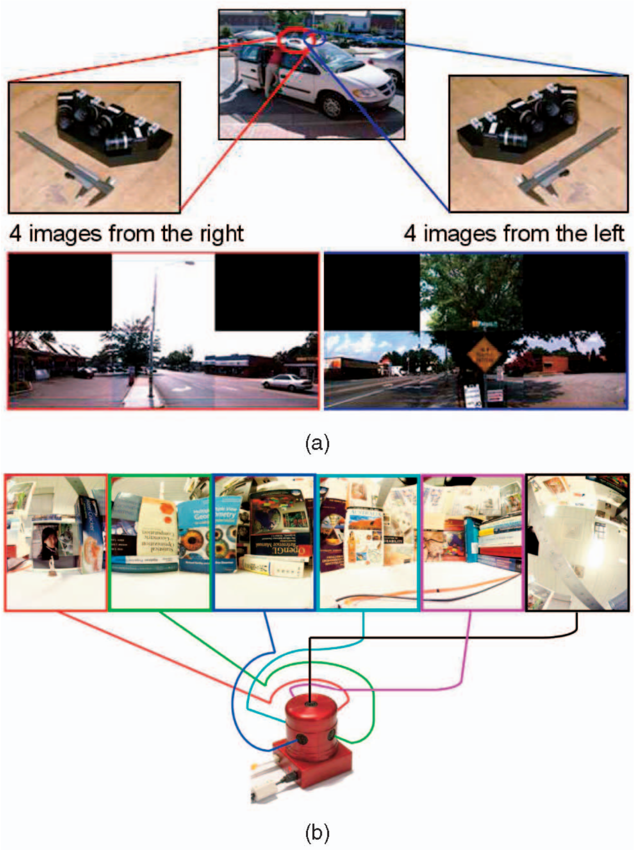


Fig. 1. An example of nonoverlapping multicamera rigs: (a) 8-camera system of 3D Urban modeling project (courtesy of UNC-Chapel Hill) and (b) 6-camera system of Ladybug2 omnidirectional camera.

Although many iterative *nonlinear* algorithms have been proposed ([4], [5]) to solve this relative motion problem, it is surprising that no reliable linear algorithm has been described. This is curious because local nonlinear algorithms usually need a good initial estimate for refinement. However, to our knowledge, little research has been done to give a linear algorithm for the nonoverlapping camera case. In this paper, we will address this issue and give a simple linear solution which can be used to initialize any nonlinear algorithm.

An obvious problem with local nonlinear methods is that they may result in a local minimum. In other words, no global optimality is guaranteed. Recently, quasiconvex optimization to obtain truly globally optimal solutions under the L_∞ norm was introduced and has become a fruitful direction of research [6], [7], [8], [9], [10], [11], [12]. Our nonlinear algorithm in this paper is based on such L_∞ global optimization. In particular, we use the branch-and-bound technique to achieve guaranteed optimality. By using some geometric heuristics, we have made the branch-and-bound computation fast enough to produce accurate estimates within a reasonable time.

In this paper, we describe two algorithms for the relative motion of a GCM. First, we briefly introduce the generalized camera model and categorize types of GCMs according to their degeneracy. From this analysis, we derive a linear algorithm, based on Pless's work, for estimating relative motion. This result may be used for initializing our L_∞ globally optimal method, which is described next. This nonlinear method minimizes a meaningful geometric cost function in L_∞ norm by repeated applications of Linear Programming (LP).

Experiments with both synthetic and real data are carried out to assess the linear method and optimal nonlinear method. These experiments show that our proposed method gives a reliable and accurate solution to the motion estimation problem for nonoverlapping multicamera rigs.

2 Previous Work

Grossberg and Nayar described the general imaging model as a mapping of scene rays to pixels, and presented a concept of "raxels" that represent geometric, radiometric, and optical properties [1]. They also provided a calibration method for this general imaging model using structured light patterns. Pless used Plücker line coordinates [13] to represent scene rays and derived an epipolar constraint equation for generalized cameras [3]. He predicted that 17 point correspondences are enough to solve the generalized essential matrix. Although it may appear that his equations give a linear solution to the relative motion problem, many important camera configurations, such as those discussed in the current paper, lead to a set of degenerate equations from which a linear solution is not immediately possible. A hierarchy of generalized camera models and essential matrices for the different camera models are shown by Sturm in [2]. However, none of these researches has shown any experiments with a linear method for estimating an essential matrix for generalized cameras. In our research, we show and extensively verify a linear method for solving the relative motion problem for generalized cameras.

There exist many nonlinear algorithms for solving for the generalized essential matrix for generalized cameras. Lhuillier used bundle adjustment for generalized cameras by using angular error instead of 3D errors [5]. Stewénius et al. used a Gröbner basis to solve for the generalized essential matrix and Byröd et al. improved the numerical accuracy of the Stewénius et al.'s method [14], [15]. These methods based on a Gröbner basis approach solve polynomial equations to compute the generalized essential matrix and apply to the minimal case only.

Mouragnon et al. [4] considered the dimension of the solution set of the generalized epipolar equations in the case of locally central cameras, both axial and nonaxial. The authors confirmed that there are ambiguities in the solution for the generalized epipolar equations, and indeed found the same rank conditions that we report here. They do not appear to consider the case of axial nonlocally central cameras, a class that includes noncentral catadioptric and pushbroom cameras. Their approach to solving this problem involved a nonlinear algorithm. They carried out experiments with axial-type cameras only and used an incremental bundle adjustment method to refine their results. Schweighofer and Pinz gave an iterative algorithm for generalized cameras to estimate their structure and motion by minimizing an object-space error, which is the distance between a point in 3D and the projection of the point onto a scene ray [16]. All of these methods require a good initial estimate for their nonlinear optimization process. However, none of the methods actually used the linear 17-point algorithm for initialization.

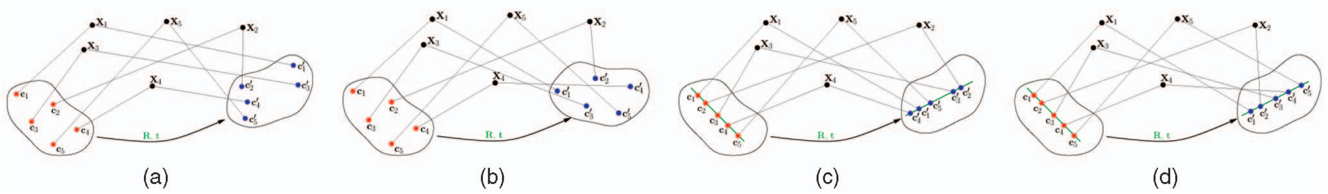


Fig. 2. (a) The most general case: Pixels in an image correspond to general lines in space. (b) The locally central case: Corresponding image rays in two images with a moving camera pass through a common point (their local center) in the camera's coordinate frame. For example, the two centers, $\mathbf{c}_i$ and $\mathbf{c}'_i$, of the rays passing through each point $\mathbf{X}_i$ are the same point relative to the coordinate frame of the camera. Compare this with (a), where this condition does not hold. (c) The axial case: All of the image rays meet a common line, the axis. (d) The locally central, axial case: Image rays meet a common line and corresponding rays pass through the same point on that line. The ranks of the generalized epipolar equations in each of these cases are 17, 16, 16, and 14, respectively.

There is some related work estimating relative motion for nontraditional cameras. Frahm et al. [17] proposed a pose estimation algorithm for multicamera systems by putting a virtual camera into a multicamera system. Also, a similar approach to estimate relative motion for nonoverlapping multicamera systems using Second-Order Cone Programming (SOCP) was proposed by Kim et al. [18]. However, that approach does not provide a unified framework for global estimation. They first estimate the motion of each camera separately, and then as a second stage, combine the individual motions to find a solution. The accuracy of that method is limited by the accuracy of the estimate of the motion of each camera.

Parts of this work have been published in conference papers [19], [20].

3 Generalized Cameras

A generalized camera is a model for an imaging situation in which pixels in the image correspond to specified rays (straight lines) in space, but with no other limitation on how incoming light rays project onto an image. The image value at a pixel records the response (for instance, color) of some point along its associated ray. There can be multiple centers of projection, or indeed no centers of projection at all. This camera model is relatively general and includes cameras such as perspective cameras, fish-eye cameras, central or noncentral catadioptric cameras, linear or nonlinear pushbroom cameras [21], whiskbroom cameras, panoramic cameras [22], as well as multicamera rigs and insect eyes. It is worth noting, however, that it does not cover certain important classes of cameras, such as Synthetic Aperture Radar (SAR) images and the rational cubic camera model [23] used in many surveillance images. This is because the possible 3D points that map to a given image point in such images do not all lie on a straight line.

The generalized camera model, along with certain important special cases, is illustrated in Fig. 2.

Multicamera systems. For us, the most important type of generalized camera is the multicamera system. A multicamera system is a set of cameras placed rigidly on a moving platform or vehicle, possibly having nonoverlapping or minimally overlapping fields of view. We assume that the complete calibration of the camera system is known.

Since the multicamera system moves rigidly, it is convenient to consider it as a single image device, or a single generalized camera. Indeed, in conformity with the definition of generalized camera, each pixel in the set of images taken at one time corresponds to a ray in space. Note that different rays pass through different camera centers. The algorithms developed in this paper for generalized cameras will apply to any such multicamera system.

However, we are mostly interested in cameras with minimal or no overlap in their field of view. In such an instance, we further assume that, during the motion of the rig, points are tracked only within individual images and that point tracks do not pass from one camera to another. Under this assumption, the multicamera system is an example of a special class of generalized camera called a "locally central camera." This means that a 3D point $\mathbf{X}$ is seen by the same camera from two different positions of the multicamera rig, and most particularly, the rays corresponding to the two images of $\mathbf{X}$ pass through the same point in the local camera coordinate frame, the said camera center.

A multicamera system may also be classed as an "axial camera" if the centers of all the cameras lie on a common line, the axis. This will always be the case for two-camera rigs. Axial and locally central camera systems have special degeneracies, which will be explored in this paper.

Derivation of Pless's equations. Let a light ray be described by a point $\mathbf{v}$ lying on the ray and a unit direction vector $\mathbf{x}$ pointing along the ray. The 6-vector $\mathbf{L} = (\mathbf{x}^\top, (\mathbf{v} \times \mathbf{x})^\top)^\top$ is known as the Plücker coordinates of the line. Note that the vector $\mathbf{L}$ is independent of the particular point $\mathbf{v}$ chosen on the line.

The condition for two lines with Plücker coordinates $\mathbf{L}$ and $\mathbf{L}'$ to meet may be simply expressed by the equation

$$\mathbf{L}'^\top \begin{bmatrix} \mathtt{0} & \mathtt{I} \\ \mathtt{I} & \mathtt{0} \end{bmatrix} \mathbf{L} = 0.$$

When a euclidean transformation consisting of rotation $\mathtt{R}$ and translation $\mathbf{t}$ is applied to a line $\mathbf{L}$, it is transformed into a different line according to the transformation rule

$$\mathbf{L} \mapsto \mathbf{L}_{\text{rot}} = \begin{bmatrix} \mathtt{R} & \mathtt{0} \\ \mathtt{E} & \mathtt{R} \end{bmatrix} \mathbf{L},$$

where $\mathtt{E} = [\mathbf{t}]_\times \mathtt{R}$ is the essential matrix corresponding to the motion. Here, $[\mathbf{t}]_\times$ is the skew symmetric matrix representing the vector (cross) product in the sense that $\mathbf{x}^\top [\mathbf{t}]_\times = (\mathbf{x} \times \mathbf{t})^\top$ for any vector $\mathbf{x}$.

Pless's equation expresses the fact that a line $\mathbf{L}$, when transformed via a euclidean motion, will meet another line $\mathbf{L}'$. These lines represent the rays belonging to two cameras meeting in a matched point in space. By combining the two equations above, we obtain

$$\begin{aligned} 0 &= \mathbf{L}'^{\top} \begin{bmatrix} 0 & \mathtt{I} \\ \mathtt{I} & 0 \end{bmatrix} \mathbf{L}_{\text{rot}} \\ &= \mathbf{L}'^{\top} \begin{bmatrix} 0 & \mathtt{I} \\ \mathtt{I} & 0 \end{bmatrix} \begin{bmatrix} \mathtt{R} & 0 \\ \mathtt{E} & \mathtt{R} \end{bmatrix} \mathbf{L} \\ &= \mathbf{L}'^{\top} \begin{bmatrix} \mathtt{E} & \mathtt{R} \\ \mathtt{R} & 0 \end{bmatrix} \mathbf{L}. \end{aligned}$$

This is Pless's Generalized Epipolar Constraint and it must be satisfied by every pair of matching rays from a moving Generalized Camera.

The Generalized Epipolar Constraint may be expanded out to give the following equation:

$$\mathbf{x}'^{\top}\mathtt{E}\mathbf{x} + (\mathbf{v}' \times \mathbf{x}')^{\top}\mathtt{R}\mathbf{x} + \mathbf{x}'^{\top}\mathtt{R}(\mathbf{v} \times \mathbf{x}) = 0. \qquad (1)$$

Equation (1) may be written as a linear equation of the form $\mathbf{a}_i^{\top}\mathbf{y} = 0$, where $\mathbf{y}$ is a vector made up of the unknown entries of matrices $\mathtt{E}$ and $\mathtt{R}$. By putting together the equations derived from N such matched point pairs, we may construct a set of equations

$$\begin{aligned} \mathtt{A}\mathbf{y} &= \begin{bmatrix} \mathbf{a}_1^{\top} \\ \mathbf{a}_2^{\top} \\ \vdots \\ \mathbf{a}_N^{\top} \end{bmatrix} (\mathtt{E}_{11}, \mathtt{E}_{12}, \ldots, \mathtt{E}_{33}, \mathtt{R}_{11}, \ldots, \mathtt{R}_{33})^{\top} \\ &= \mathtt{A} \begin{pmatrix} \text{vec}(\mathtt{E}) \\ \text{vec}(\mathtt{R}) \end{pmatrix} = 0, \end{aligned} \qquad (2)$$

where $\text{vec}(\mathtt{E})$ and $\text{vec}(\mathtt{R})$ are 9-vectors whose elements are taken in column-major order from $\mathtt{E}$ and $\mathtt{R}$, respectively.

From a minimum of $N = 17$ such matches, one seeks to solve these equations to find $\mathtt{E}$ and $\mathtt{R}$ up to a common scale. From $N > 17$ points, a least-squares solution can be sought. In particular, let $\mathtt{A} = \mathtt{UDV}^{\top}$ be the Singular Value Decomposition (SVD) of $\mathtt{A}$, where $\mathtt{D}$ is a diagonal matrix containing the singular values in order of decreasing magnitude. Then the least-squares solution $\binom{\text{vec}(\mathtt{E})}{\text{vec}(\mathtt{R})}$ is the last column of $\mathtt{V}$. This will be referred to as the *standard SVD solution* method.

Unfortunately, as will be seen later, the set of equations constructed in this way does not have rank 17 except in the most general camera case. Hence, it is impossible to solve these equations directly to find the correct solution.

Change of origin. Consider what happens if the origin of coordinates is changed via a translation so that $\mathbf{v}_i$ and $\mathbf{v}'_i$ are changed. The form of the equations (1) is correspondingly changed and the solution also changed. If $(\mathtt{E}, \mathtt{R})$ is the solution to the equations (1), then, under a change of coordinates $\mathbf{v}_i \mapsto \mathbf{v}_i + \mathbf{s}$ and $\mathbf{v}'_i \mapsto \mathbf{v}'_i + \mathbf{s}$, the solution is modified as follows:

$$\begin{aligned} (\mathtt{E},\ \mathtt{R}) &\mapsto (\mathtt{E} - [\mathbf{s}]_{\times}\mathtt{R} + \mathtt{R}[\mathbf{s}]_{\times},\ \mathtt{R}) \\ &= (\mathtt{E} - [\mathbf{s}]_{\times}\mathtt{R} + [\mathtt{R}\mathbf{s}]_{\times}\mathtt{R},\ \mathtt{R}) \\ &= ([\mathbf{t} - \mathbf{s} + \mathtt{R}\mathbf{s}]_{\times}\mathtt{R}, \mathtt{R}). \end{aligned} \qquad (3)$$

In the ground-truth solution to the equations (1), the matrix $\mathtt{E}$ is of the form $[\mathbf{t}]_{\times}\mathtt{R}$, where $\mathtt{R}$ is the same matrix occurring in the second and third terms of this equation. There is no simple way of enforcing this condition in solving the equations, however, so, in a linear approach, we solve the equations ignoring this condition. Our initial goal is to examine the linear structure of the solution set to these equations under various different camera geometries.

4 Analysis of Degeneracies

We identify several degeneracies for the set of equations arising from (1) which cause the set of equations to have smaller than expected rank. These examples are illustrated in Fig. 2.

Genericity argument. Suppose we wish to show that the rank of the system is equal to $r < 17$. We assume that there are at least r equations arising from point correspondences via the equations (2). To show that the rank is r, we will exhibit a linear family of solutions to the equations. If the linear family of solutions has rank $18 - r$, then the equation system must have rank no greater than r.

The reader may object that this argument only places an upper bound on the rank of the equation system. However, to show that the system does not have smaller rank, it is sufficient to exhibit a single example in which the rank has the claimed value. This will mean that generically (that is, for almost all input data) the rank will indeed reach this upper bound.

This claim is justified by the following argument: Consider the set of equations arising from a set of point correspondences $(\mathbf{x}_i, \mathbf{v}_i) \leftrightarrow (\mathbf{x}'_i, \mathbf{v}'_i)$. Equation (2) arising from such a correspondence is easily seen to have coefficients that are polynomials in the variables $\mathbf{x}_i, \mathbf{x}'_i, \mathbf{v}_i, \mathbf{v}'_i$. For the set of equations arising from several correspondences to have rank less than r, it is necessary and sufficient that all $r \times r$ subdeterminants vanish. Each such subdeterminant gives rise to a polynomial equation in the input data variables. The system will have rank less than r only on the input data forming a zero-set for all these polynomials—in other words, on a variety in the euclidean space of all possible input data. However, a variety in $\mathbb{R}^N$ is either dense nowhere or else consists of the whole of $\mathbb{R}^N$. Exhibiting a single point not on the variety (corresponding to a set of input data for which the rank is equal to r) rules out the second possibility. This demonstrates that generically the rank must be r.

An example of how to construct an example of the required type is given in the Appendix. The example shows that the rank is indeed 14 in the locally central axial case described soon. It is easily modified to determine the rank for the other configurations.

The most general case. In the most general case (see Fig. 2a), the camera is simply a set of unconstrained image rays in general position. For this case, the rank of the obtained generalized epipolar equation will be 17. Therefore, a unique solution is readily solvable by the standard SVD method. We do not further consider this case in the paper.

Locally central projection. Next, we consider the case of a "generalized camera" consisting of a set of locally central projection rays (see Fig. 2b). The commonly used (non-overlapping) multicamera rigs are examples of this case. When the camera rig moves from an initial to a final position, points are tracked. We assume that points are not tracked from one camera to another. Thus, a point that is seen at one time instant by one of the cameras constituting the multicamera rig can only be seen to correspond with a point seen by the same camera at a later point in time. We assume further that each component camera is a central projection camera so that all rays go through the same point, i.e., the camera center. We will refer to this as *locally central projection*.

Since rays are represented in a coordinate system attached to the camera rig, the correspondence is between points $(\mathbf{x}_i, \mathbf{v}_i) \leftrightarrow (\mathbf{x}'_i, \mathbf{v}_i)$, where $\mathbf{v}_i$ is the camera center. In particular, note that $\mathbf{v}'_i = \mathbf{v}_i$. The equations (1) are now

$$\mathbf{x}_i^\top \mathtt{E}\mathbf{x}'_i + \mathbf{x}_i^\top \mathtt{R}(\mathbf{v}_i \times \mathbf{x}'_i) + (\mathbf{v}_i \times \mathbf{x}_i)^\top \mathtt{R}\,\mathbf{x}'_i = 0. \qquad (4)$$

Now let $(\mathtt{E}, \mathtt{R})$ be one solution to this set of equations, with $\mathtt{E} \neq \mathtt{0}$. It is easily seen that $(\mathtt{0}, \mathtt{I})$ is also a solution. In fact, substituting $(\mathtt{0}, \mathtt{I})$ into (4) results in $\mathbf{x}_i^\top(\mathbf{v}_i \times \mathbf{x}'_i) + (\mathbf{v}_i \times \mathbf{x}_i)^\top \mathbf{x}'_i$, which is zero because of the antisymmetry of the triple product.

Generically, the rank is not less than 16, so a complete solution to this set of equations is therefore of the form $(\lambda\mathtt{E}, \lambda\mathtt{R} + \mu\mathtt{I})$, a two-dimensional linear family. From this formulation, an interesting property of the set of solutions to (1) is found: The ambiguity is contained entirely in the estimation of $\mathtt{R}$, while the essential matrix $\mathtt{E}$ is still determined uniquely up to scale.

Axial cameras. Our second example of degenerate configuration is what we will call an axial camera (see Fig. 2c). This is defined as a generalized camera in which all the rays intersect in a single line, called the axis. There are several examples of this which may be of practical interest.

1. A pair of rigidly mounted central projection cameras (for instance, ordinary perspective cameras).
2. A set of central projection cameras with collinear centers. We call this a linear camera array.
3. A set of noncentral catadioptric or fish-eye cameras mounted with collinear axes.
4. A linear pushbroom or whiskbroom camera.

The first two cases are also locally central projections, provided that points are not tracked from one camera to others.

To analyze this configuration, we may choose the origin of the world coordinate system to lie on the axis, and examine the solution set of equations (1) for this case.

For a different coordinate origin, the solution will be related to this particular solution according to (3). We see that in different coordinate systems, the value of $\mathtt{E}$ changes. This is why we insist on the requirement that the origin of the camera coordinate system should lie on the axis.

In such a coordinate system, we may write $\mathbf{v}_i = \alpha_i\mathbf{w}$ and $\mathbf{v}'_i = \alpha'_i\mathbf{w}$, where $\mathbf{w}$ is the direction vector of the axis. Equation (1) then takes the form

$$\mathbf{x}_i^\top \mathtt{E}\mathbf{x}'_i + \alpha_i(\mathbf{w} \times \mathbf{x}_i)^\top \mathtt{R}\,\mathbf{x}'_i + \alpha'_i\mathbf{x}_i^\top \mathtt{R}(\mathbf{w} \times \mathbf{x}'_i) = 0. \qquad (5)$$

Suppose that $(\mathtt{E}, \mathtt{R})$ is the true solution to these equations. Another solution is given by $(\mathtt{0}, \mathbf{w}\mathbf{w}^\top)$. It satisfies (5) because $(\mathbf{w} \times \mathbf{x}_i)^\top\mathbf{w} = \mathbf{w}^\top(\mathbf{w} \times \mathbf{x}'_i) = 0$. Generically, the equation system has rank 16, so the general solution to (5) for an axial camera is $(\lambda\mathtt{E}, \lambda\mathtt{R} + \mu\mathbf{w}\mathbf{w}^\top)$.

Note the most important fact that the $\mathtt{E}$ part of the solution is constant and the ambiguity only involves the $\mathtt{R}$ part of the solution. Thus, we may retrieve the matrix $\mathtt{E}$ without ambiguity from the degenerate system of equations. It is important to note that this fact depends on the choice of coordinate system such that the origin lies on the axis. Without this condition, there is still a two-dimensional family of solutions, but the solution for the matrix $\mathtt{E}$ is not invariant.

Locally central-and-axial cameras. If in addition we assume that the projections are locally central, then further degeneracies occur (see Fig. 2d). We have seen already that for locally central projections, $(\mathtt{0}, \mathtt{I})$ is also a solution. However, in the case of an axial camera array, a further degeneracy occurs. The condition of local centrality means that $\alpha_i = \alpha'_i$ in (5). We may now identify a further solution $(\mathtt{0}, [\mathbf{w}]_\times)$ since

$$\begin{aligned}(\mathbf{w} \times \mathbf{x}_i)^\top[\mathbf{w}]_\times\mathbf{x}'_i + \mathbf{x}_i^\top[\mathbf{w}]_\times(\mathbf{w} \times \mathbf{x}'_i)\\ = (\mathbf{w} \times \mathbf{x}_i)^\top(\mathbf{w} \times \mathbf{x}'_i) \;+\; (\mathbf{x}_i \times \mathbf{w})^\top(\mathbf{w} \times \mathbf{x}'_i) = 0.\end{aligned}$$

In summary, in the case of a locally central axial camera, the complete solution set is of the form

$$(\alpha\mathtt{E},\ \alpha\mathtt{R} + \beta\mathtt{I} + \gamma[\mathbf{w}]_\times + \delta\mathbf{w}\mathbf{w}^\top), \qquad (6)$$

under the assumption that the coordinate origin lies on the camera axis. Once more, the $\mathtt{E}$ part of the solution is determined uniquely up to scale, even though there is a four-dimensional family of solutions.

Cross-slits camera. A camera that has been considered in certain graphics applications [24], [25] is the cross-slits camera. This has the property that each ray must meet two separate nonintersecting straight lines. Such a camera is not locally central. Since each straight line gives rise to one degree of ambiguity, it follows that the rank of the equation system for a cross-slits camera is at most 15.

Let the coordinate origin be on one of these two lines, so a point on this line is of the form $\alpha_i\mathbf{v}$. A point on the other line may be expressed as $\mathbf{a} + \beta_i\mathbf{w}$ for some vectors $\mathbf{a}$ and $\mathbf{w}$. We write $\mathbf{v}_i = \alpha_i\mathbf{v}$ and $\mathbf{x}_i = \mathbf{a} + \beta_i\mathbf{w} - \alpha_i\mathbf{v}$.

We also consider a second cross-slits camera and express the analogous quantities with primes. It is not necessary to assume that the two cameras are identical, so $\mathbf{a}$, $\mathbf{v}$, and $\mathbf{w}$ may be different from $\mathbf{a}'$, $\mathbf{v}'$, and $\mathbf{w}'$. With this notation, it may be verified that Pless's equation (1) has special solutions $(\mathtt{E}, \mathtt{R}) = (\mathtt{0}, \mathbf{v}'\mathbf{v}^\top)$ and $(\mathtt{E}, \mathtt{R}) = ([\mathbf{a}']_\times\mathbf{w}'\mathbf{w}^\top - \mathbf{w}'\mathbf{w}[\mathbf{a}]_\times, \mathbf{w}'\mathbf{w}^\top)$.

It follows that if $(\mathtt{E}, \mathtt{R})$ represents the true motion, then there are at least the following solutions to the epipolar equations:

$$(\mathtt{E} + \mu\big([\mathbf{a}']_\times\mathbf{w}'\mathbf{w}^\top - \mathbf{w}'\mathbf{w}[\mathbf{a}]_\times\big), \mathtt{R} + \lambda\mathbf{v}'\mathbf{v}^\top + \mu\mathbf{w}'\mathbf{w}^\top).$$

On the other hand, experiments suggest that, generically, the rank is 13, which implies the existence of two other solutions. At present, we cannot identify these solutions, and the question will not be further considered in this paper.

4.1 Summary of Degeneracy Results

The results we have obtained for solutions to (1) are as follows:

1. For the most general case, the rank of the generalized epipolar equation is exactly 17 and the equation

gives a unique solution. The scale can be estimated by properly normalizing the rotation matrix.

2. For a locally central projection, the general solution is of the form $(\alpha\mathtt{E}, \alpha\mathtt{R} + \beta\mathtt{I})$. Despite the degeneracy, we may retrieve $\mathtt{E}$ (up to scale) from the solution family.
3. For axial cameras, the general solution is of the form $(\alpha\mathtt{E}, \alpha\mathtt{R} + \beta\mathbf{w}\mathbf{w}^\top)$, as long as the equations are written in a coordinate system with the axis passing through the origin. In this case, we may retrieve $\mathtt{E}$ (up to scale) from the solution family.
4. For axial cameras with locally central projection (for instance, a pair of perspective cameras or several perspective cameras mounted linearly), the general solution is of the form $(\alpha\mathtt{E},\ \alpha\mathtt{R} + \beta\mathtt{I} + \gamma\mathbf{w}\mathbf{w}^\top + \delta[\mathbf{w}]_\times)$, provided the origin is chosen to lie on the axis. Once more, we may retrieve $\mathtt{E}$ up to scale.
5. For the cross-slits camera, there is a three-dimensional set of solutions (including scale).

5 Linear Algorithm

Next, we shall give a new algorithm based on (1) for retrieving the motion of a generalized camera. The algorithm applies to the situations involving locally central and/or axial cameras, where the equation set is rank deficient, resulting in different dimensional families of solutions. Despite the degeneracy, we show how to obtain a unique linear solution.

Linear algorithm. The condition (1) gives one equation for each point correspondence. Given sufficiently many point correspondences, we may solve for the entries of matrices $\mathtt{E}$ and $\mathtt{R}$ linearly from the set of equations $\mathtt{A}\binom{\mathrm{vec}(\mathtt{E})}{\mathrm{vec}(\mathtt{R})} = \mathbf{0}$. However, we have seen that the standard SVD solution to this set of equations gives a whole family of solutions. If one ignores the rank deficiency of the equations, totally spurious solutions may be found.

It was observed that for locally central projections, one trivial solution to the linear system is $\mathtt{E} = 0$ and $\mathtt{R} = \mathtt{I}$. In practice, this solution is often found using the standard SVD algorithm. The corresponding motion is given by $\mathtt{R} = \mathtt{I}$ and $\mathbf{t} = \mathbf{0}$, since $\mathtt{E} = [\mathbf{t}]_\times\mathtt{R}$. This means that the camera rig neither rotates nor translates—a *null* motion. However, for a moving camera, this solution is not compatible with the observation. This shows a very curious property of the algebraic solution that the equation set may be satisfied exactly with zero error even though the solution found is totally wrong geometrically.

Various possibilities for finding a single solution from among a family of solutions may be proposed, enforcing necessary conditions on the essential matrix $\mathtt{E}$ and the rotation $\mathtt{R}$. Such methods will be nonlinear and not easy to implement (e.g., involving many parameter tunings). In addition, observe that the solution $\mathtt{E} = 0$, $\mathtt{R} = \mathtt{I}$ with $\mathbf{t} = \mathbf{0}$ satisfies all compatibility conditions between a rotation $\mathtt{R}$ and $\mathtt{E} = [\mathbf{t}]_\times\mathtt{R}$, and yet is wrong geometrically. We prefer a linear solution avoiding this problem, which will be described next.

The key idea. To avoid the problem of multiple solutions, we observe the crucial fact that although there exists a family of solutions (of dimension 2-4 depending on the case), all the ambiguity lies in the determination of the $\mathtt{R}$ part of the solution. The $\mathtt{E}$ part of the solution is unchanged by the ambiguity. In other words, the family of solutions, when projected down to the nine-dimensional subspace formed by the $\mathtt{E}$ part only, will be well constrained. This suggests using the set of equations to solve only for $\mathtt{E}$, and forget about trying to solve for the $\mathtt{R}$ part, which provides redundant information anyway.

Thus, given a set of equations $\mathtt{A}\binom{\mathrm{vec}(\mathtt{E})}{\mathrm{vec}(\mathtt{R})} = \mathbf{0}$, we find the solution that minimizes $\|\mathtt{A}\binom{\mathrm{vec}(\mathtt{E})}{\mathrm{vec}(\mathtt{R})}\|$ subject to $\|\mathtt{E}\| = 1$ instead of $\|\binom{\mathrm{vec}(\mathtt{E})}{\mathrm{vec}(\mathtt{R})}\| = 1$ as in the standard SVD algorithm. This seemingly small change to the algorithm avoids all the difficulties associated with the standard SVD algorithm.

Solving a problem of this form is discussed in [26, Algorithm 5.4.2, page 595] in a more general form. Here, we summarize the method. Write the equations as $\mathtt{A}_\mathrm{E}\mathrm{vec}(\mathtt{E}) + \mathtt{A}_\mathrm{R}\mathrm{vec}(\mathtt{R}) = \mathbf{0}$, where $\mathtt{A}_\mathrm{E}$ and $\mathtt{A}_\mathrm{R}$ are submatrices of $\mathtt{A}$ consisting of the first and last nine columns. Finding the solution that satisfies $\|\mathrm{vec}(\mathtt{E})\| = 1$ is equivalent to solving

$$(\mathtt{A}_\mathrm{R}\mathtt{A}_\mathrm{R}^+ - \mathtt{I})\mathtt{A}_\mathrm{E}\mathrm{vec}(\mathtt{E}) = \mathbf{0}, \tag{7}$$

where $\mathtt{A}_\mathrm{R}^+$ is the pseudo-inverse of $\mathtt{A}_\mathrm{R}$. This equation is then solved using the standard SVD method, and it gives a unique solution for $\mathtt{E}$.

Details. Note that, typically, $\mathtt{A}_\mathrm{R}$ is rank deficient (that is, it has rank less than 9, its column-dimension). One cannot, therefore, use the formula $\mathtt{A}_\mathrm{R}^+ = (\mathtt{A}_\mathrm{R}^\top\mathtt{A}_\mathrm{R})^{-1}\mathtt{A}_\mathrm{R}^\top$ to compute $\mathtt{A}_\mathrm{R}^+$. Instead, we use the SVD and write $\mathtt{A}_\mathrm{R} = \mathtt{UDV}^\top$. The pseudo-inverse is then $\mathtt{A}_\mathrm{R}^+ = \mathtt{VD}^+\mathtt{U}^\top$, where $\mathtt{D}^+$ is the diagonal matrix defined by $\mathtt{D}_{ii}^+ = \mathtt{D}_{ii}^{-1}$ for $i = 1, \dots, r$ and $\mathtt{D}_{ii}^+ = 0$ otherwise. Here, r is the rank of $\mathtt{A}_\mathrm{R}$. In this case, we observe that $(\mathtt{A}_\mathrm{R}\mathtt{A}_\mathrm{R}^+ - \mathtt{I}) = \mathtt{U}_r\mathtt{U}_r^\top - \mathtt{I}$, where $\mathtt{U}_r$ is the matrix consisting of the first r columns of $\mathtt{U}$.

In each of the degenerate cases we have discussed, the matrix $\mathtt{A}_\mathrm{R}$ is rank deficient. Suppose that the matrix $\mathtt{E}$ is known. Then the set of equations (2) may be written as a set of nonhomogeneous equations

$$\mathtt{A}_\mathrm{R}\mathrm{vec}(\mathtt{R}) = -\mathtt{A}_\mathrm{E}\mathrm{vec}(\mathtt{E})$$

in the unknowns $\mathrm{vec}(\mathtt{R})$. Since one scale factor is set by the choice of scale for $\mathtt{E}$, this set of equations has a set of solutions of dimension 1, 1, or 3 in $\mathbb{R}^9$ for the locally central, axial, or locally central-and-axial cases, respectively. Hence, the matrix $\mathtt{A}_\mathrm{R}$ has rank 8, 8, or 6 for these cases. This known rank should be taken into account when computing the pseudoinverse $\mathtt{A}_\mathrm{R}^+$.

Minimum number of points. Since we solve for $\mathtt{E}$ directly from the set of equations (7), it might be thought that eight points will be sufficient to give a unique solution. However, this is not so. The matrix $(\mathtt{A}_\mathrm{R}\mathtt{A}_\mathrm{R}^+ - \mathtt{I}) = \mathtt{U}_r\mathtt{U}_r^\top - \mathtt{I}$ has a rank deficiency equal to r, where r is the rank of $\mathtt{A}_\mathrm{R}$, and this decreases the rank of the equation matrix. In fact, if $\mathtt{A} = [\mathtt{A}_\mathrm{E}|\mathtt{A}_\mathrm{R}]$ has n rows, stemming from n point correspondences, then the matrix $(\mathtt{A}_\mathrm{R}\mathtt{A}_\mathrm{R}^+ - \mathtt{I})\mathtt{A}_\mathrm{E}$ will have rank at most $n - r$, where r is the rank of $\mathtt{A}_\mathrm{R}$. It follows that we need at least $n = r + 8$ point correspondences to solve for $\mathtt{E}$ up to scale—that is, 16, 16, or 14 points for the three cases, respectively.

Handling axial cameras. The method for solving for axial cameras, and particularly, for the case of two cameras (i.e., stereo head) is just the same, except that we must take care to write the Generalized Epipolar Constraint equations in terms of a world coordinate system, where the origin lies on the axis. This is an essential (nonoptional) step to allow us to compute the matrix $\mathtt{E}$ correctly. In the case of two cameras, it makes sense that the origin should be the midpoint between the two cameras.

Extracting the rotation and translation. The $\mathtt{E}$ part, once found, may be decomposed as $\mathtt{E} = [\mathbf{t}]_\times \mathtt{R}$ to obtain both the rotation and translation up to scale. This problem is a little different from the corresponding method of decomposing the essential matrix $\mathtt{E}$ for a standard pair of pinhole images. There are two differences as follows:

1. The decomposition of $\mathtt{E} = [\mathbf{t}]_\times \mathtt{R}$ gives two possible values for the rotation, differing by the so-called twisted pair ambiguity. This ambiguity may be resolved by cheirality considerations. However, in the GCM case, only one of the two possibilities is compatible with the Generalized Epipolar Constraint.
2. From the standard essential matrix, the translation $\mathbf{t}$ may be computed only up to scale. For a generalized camera, however, the scale of $\mathbf{t}$ may be computed unambiguously. There is only one translation $\mathbf{t}$ that is compatible with the correctly scaled rotation matrix $\mathtt{R}$.

The recommended method of computing the translation $\mathbf{t}$ once $\mathtt{R}$ is known is to revert to the equations (1) and compute $\mathbf{t}$ directly from the relationship

$$\mathbf{x}_i'^\top [\mathbf{t}]_\times (\mathtt{R}\,\mathbf{x}_i) + (\mathbf{v}_i' \times \mathbf{x}_i')^\top \mathtt{R}\,\mathbf{x}_i + \mathbf{x}_i'^\top \mathtt{R}(\mathbf{v}_i \times \mathbf{x}_i) = 0.$$

The only unknown in this set of equations is the translation vector $\mathbf{t}$, which may then be computed using linear least squares. Since this is a nonhomogeneous set of equations, unless $\mathtt{R} = \mathtt{I}$, the computed $\mathbf{t}$ will not have scale ambiguity.

Algorithm description. We now summarize the linear algorithm for solving the Generalized Epipolar Constraints in the case of locally central or axial cameras.

Linear Motion Estimation for Multicamera Systems

Given

A set of correspondences $(\mathbf{x}_i, \mathbf{v}_i) \leftrightarrow (\mathbf{x}_i', \mathbf{v}_i')$ derived from a pair of images from a locally central of axial camera. For the case of locally central projection, $\mathbf{v}_i = \mathbf{v}_i'$ for all i. For the case of an axial camera, all $\mathbf{v}_i$ and $\mathbf{v}_i'$ lie on a single line, which contains the coordinate origin of the camera.

Objective

Estimate the rotation and translation (with scale) of the camera.

Algorithm

1. Form the set of linear equations $\mathtt{A}_{\mathrm{E}}\mathrm{vec}(\mathtt{E}) + \mathtt{A}_{\mathrm{R}}\mathrm{vec}(\mathtt{R}) = \mathbf{0}$ using (1).
2. Compute the pseudoinverse $\mathtt{A}_{\mathrm{R}}^+$ of $\mathtt{A}_{\mathrm{R}}$ and write the equation for $\mathrm{vec}(\mathtt{E})$ as $\mathtt{B}\,\mathrm{vec}(\mathtt{E}) = 0$, where $\mathtt{B} = (\mathtt{A}_{\mathrm{R}}\mathtt{A}_{\mathrm{R}}^+ - \mathtt{I})\mathtt{A}_{\mathrm{E}}$. Solve this equation using the standard SVD algorithm to find $\mathtt{E}$.
3. Decompose $\mathtt{E}$ to get the twisted pair of rotation matrices $\mathtt{R}$ and $\mathtt{R}'$.
4. Knowing possible rotations, solve equations (1) to compute $\mathbf{t}$ linearly. The equations are nonhomogeneous in $\mathbf{t}$, so $\mathbf{t}$ is computed with the correct scale. Keep either $\mathtt{R}$ or $\mathtt{R}'$ and the corresponding $\mathbf{t}$, whichever gives the best residual.

5.1 Alternation

It was indicated that once $\mathtt{R}$ is known, $\mathbf{t}$ may be computed linearly. Similarly, if $\mathbf{t}$ is known, then $\mathtt{R}$ may be computed linearly from the same equations. We solve linearly for $\mathtt{R}$ subject to the condition $\|\mathtt{R}\| = 3$ so as to approximate a rotation matrix.

This suggests an alternating approach in which one solves alternately for $\mathtt{R}$ and $\mathbf{t}$. Since the cost decreases at each step, this alternation will converge to a local minimum of the algebraic cost function

$$\sum_i \left\| \mathbf{x}_i^\top [\mathbf{t}]_\times \mathtt{R}\,\mathbf{x}_i' + \mathbf{x}_i^\top \mathtt{R}(\mathbf{v}_i' \times \mathbf{x}_i') + (\mathbf{v}_i \times \mathbf{x}_i)^\top \mathtt{R}\,\mathbf{x}_i' \right\|^2. \qquad (8)$$

The matrix $\mathtt{R}$ so found may not be orthogonal, but it may be corrected at the end. Unfortunately, the alternation algorithm just given has a problem that manifests itself occasionally. Namely, it returns to the spurious minimum $\mathtt{R} = \mathtt{I}$ and $\mathbf{t} = \mathbf{0}$, i.e., the null motion, even if it may not be geometrically meaningful. Note that we avoided this spurious solution in the linear algorithm by enforcing a constraint that $\|\mathtt{E}\| = 1$. However, since we are computing the value of $\mathbf{t}$ exactly (without scale ambiguity) in this alternation method, we cannot enforce this constraint.

The way to solve this is by modifying the equations (1) as will be explained now. We rewrite the equations as follows: Let $\hat{\mathbf{t}} = \beta\mathbf{t}$ be a unit vector in the direction of $\mathbf{t}$, with β being chosen accordingly. Then, multiplying each term of (8) by β results in a cost function

$$\sum_i \left(\mathbf{x}_i^\top [\hat{\mathbf{t}}]_\times \mathtt{R}\,\mathbf{x}_i' + \beta\left(\mathbf{x}_i^\top \mathtt{R}(\mathbf{v}_i' \times \mathbf{x}_i') + (\mathbf{v}_i \times \mathbf{x}_i)^\top \mathtt{R}\,\mathbf{x}_i' \right) \right)^2 \qquad (9)$$

in which β is an additional unknown. We wish to minimize this cost over all values of β, $\hat{\mathbf{t}}$, and $\mathtt{R}$ subject to the conditions $\|\hat{\mathbf{t}}\| = 1$ and $\|\mathtt{R}\| = 3$. Given $\hat{\mathbf{t}}$ and β, it is easy to solve for $\mathtt{R}$ linearly as described before. Similarly, if $\mathtt{R}$ is known, the problem is a linear least-squares problem in $\hat{\mathbf{t}}$ and β, which we may solve subject to $\|\hat{\mathbf{t}}\| = 1$ in the same way as we solved for $\mathtt{E}$ above.

Note that the trick of multiplying the cost by β, which is the reciprocal of the magnitude of $\mathbf{t}$, prevents $\mathbf{t}$ from converging to zero.

By a sequence of such alternating steps, the algebraic cost function associated with (9) is minimized subject to $\|\hat{\mathbf{t}}\| = 1$ and $\|\mathtt{R}\| = 3$, the cost diminishing at every step. In this way, we cannot fall into the same spurious minimum of the cost function as before. Our alternation serves as a simple add-on to the linear algorithm, which improves accuracy at very low cost. If further accuracy is required, it may be used to initialize a nonlinear bundle-adjustment algorithm.

5.2 Nonlinear Extensions

We have seen above that it is possible to compute the essential matrix linearly from as few as 14 point matches for the locally central-and-axial camera. This is analogous to the 8-point algorithm for computing the essential matrix from a single moving camera. It is well known that it is possible to use its inherent constraints to compute the essential matrix from a smaller number of points. One can use the constraint $\det(\mathtt{E}) = 0$ to solve for $\mathtt{E}$ from seven points [27] or the constraint $2\mathtt{E}\mathtt{E}^\top\mathtt{E} - \mathrm{trace}(\mathtt{E}\mathtt{E}^\top)\mathtt{E} = 0$ to solve from five points. The simplest algorithm in this case is given in [28]. Since the starting point for both of these algorithms is a linear family of solutions, we can apply them to compute the essential matrix for the degenerate multicamera systems that we have considered here. Thus, for the locally central-and-axial camera configuration, it is possible to compute $\mathtt{E}$ from only 11 point correspondences by applying the algorithm of [28]. For locally central or axial cameras, 13 points will be required. We have verified that this method works well for exact data, but is quite sensitive to even low levels of noise.

Note, however, that these are not true minimal solutions for this problem since it is possible to solve for motion from only six point correspondences [14] in the generalized camera case. However, that method does not work for all the degenerate cases considered here. A minimal algorithm for the two-camera case is given in [29].

6 Nonlinear Geometric Solution

The limitations of methods for estimating motion using purely algebraic methods are well known. In broad terms, the quantity minimized in such methods has no relationship to the measurement errors. It is well known that methods that minimize some meaningful geometric error are preferable. We now turn to a different method for computing the motion of a multicamera rig that minimizes a geometric error metric. In this section, we introduce an L_∞ method, which finds a geometrically optimal estimate of the motion in L_∞ error-norm, using a branch-and-bound algorithm.

For multicamera systems, an algorithm was proposed in [18] to estimate the motion of the multicamera systems using SOCP. In that paper, it was shown that the motion problem is the same as a triangulation problem, once the rotation is known. SOCP was applied to obtain a solution for translation of the multiple camera system. However, the method described there used an unstable initial estimate of rotation extracted from an essential matrix from a single camera. Although the method tries to obtain good initial estimates by averaging the selected rotations, the initial estimates come from each camera not from all cameras. Therefore, the rotation estimated from a single camera is still not an optimal solution for the whole system in terms of global optimization. Surely it can be improved if we could estimate the initial rotation from all cameras.

In this paper, we introduce a way of using all cameras to estimate the motion—rotation and translation—from the optimal essential matrix for the multicamera system. The method involves a branch-and-bound algorithm over the space of all rotations.

6.1 Branch-and-Bound Algorithm

In [10], [11], Hartley and Kahl describe a branch-and-bound algorithm for estimating the essential matrix from the motion of a single camera. The algorithm finds the optimal rotation by dividing the space of all rotations into several blocks and testing them one by one to find which one gives the best solution. Rotation space is represented as a three-dimensional space using the angle-axis representation of a rotation. As the algorithm progresses, the blocks may need to be subdivided into smaller blocks in order to get a more accurate answer. Ultimately, after a finite number of steps, one can find the optimal rotation, and hence, translation within any required degree of accuracy.

The key to the branch-and-bound technique is a method of bounding the cost associated with the rotations within a block. Let $\hat{\mathtt{R}}_0$ be the rotation represented by the center of a block in rotation space and r represent the maximum radius of the block (measured in radians). Since the translational part of the motion may be computed optimally (in L_∞ norm) once the rotation is known, we might find this optimal solution assuming the rotation $\hat{\mathtt{R}}_0$, and compute the best residual δ (namely the maximum angular measurement error, or residual, also measured in radians) over all possible choices of translation. Now the key point is that, for all other rotations $\mathtt{R}$ in the rotation block of radius r, the best residual is bounded below by $\delta + r$ (see [10], [11]).

Now, suppose that $\delta_{\min}$ is the best residual found so far in the search, we ask the following question: Is it possible to find a solution with rotation assumed equal to $\hat{\mathtt{R}}_0$ that has residual less than $\delta_{\min} + r$? If the answer is no, it means that no rotation inside the current rotation block can beat the best residual $\delta_{\min}$. In this case, we do not need to consider the current block any further. If, on the other hand, the answer is yes or possibly, then the result is inconclusive. In this case, we subdivide the rotation block by dividing into eight subblocks and keep them for future consideration. This method is guaranteed to find the optimal rotation, and hence, translation within any desired bound within a finite number of steps.

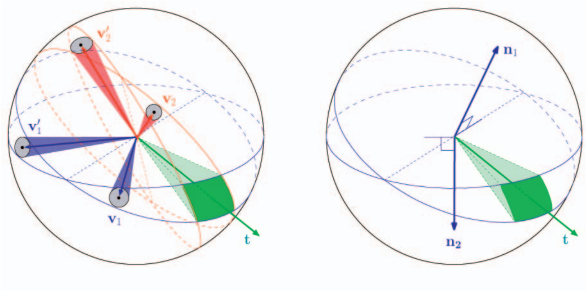
The main computation in the method just described is, for each block, we need to answer a feasibility question: Is it possible with rotation $\hat{\mathtt{R}}_0$ to find a solution with residual less than $\epsilon = \delta_{\min} + r$? We will see that this feasibility problem can be answered very efficiently using LP.

This LP problem arises in the following way: It will be shown that each point correspondence (before and after the motion) must constrain the translation vector of the motion to lie in a wedge of space bounded by a pair of planes. The placement and angle of this wedge depends on the value of ϵ just defined. The feasibility problem has a positive answer if the set of all these wedges (one wedge for every point correspondence) has a common intersection. This is a standard LP feasibility problem, and may be solved quickly and efficiently.

6.2 Theory for Multicamera Systems

We now give more details of the method given above. We assume that a rotation $\hat{\mathtt{R}}$ is given, and our task is to find whether there exists a solution to the motion problem with residual less than a given value ϵ.

Single camera constraints. Let $\mathbf{x} \leftrightarrow \mathbf{x}'$ be a pair of matched points observed in one of the cameras. These represent direction vectors expressed in a coordinate frame attached to the camera rig. Knowing (or rather hypothesizing) the rotation, we may transform one of the vectors so that they are both in the same coordinate system. Therefore, define $\mathbf{v} = \hat{\mathtt{R}}\mathbf{x}$ and $\mathbf{v}' = \mathbf{x}'$. These two vectors and the translation vector must now satisfy the coplanarity condition $\mathbf{t}^\top(\mathbf{v} \times \mathbf{v}') = 0$, which specifies that the three vectors



(a) (b)

Fig. 3. (a) Translation direction $\mathbf{t}$ exists in a region of intersections (shaded as green) of half-spaces bounded by planes that are tangent to two cones having axes $\mathbf{v}_i$ and $\mathbf{v}'_i$. Two matched pairs of points, $\mathbf{v}_1 \leftrightarrow \mathbf{v}'_1$ and $\mathbf{v}_2 \leftrightarrow \mathbf{v}'_2$, give the two intersections of two wedges. The intersection of the two wedges is a polyhedron containing the translation direction $\mathbf{t}$. (b) The two normals of the two half-spaces.

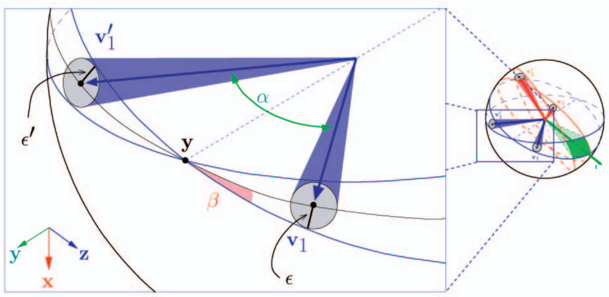


Fig. 4. The angle β, between the planes bitangent to two cones and the plane containing the axes $\mathbf{v}_1$ and $\mathbf{v}'_1$ of the two cones, is determined by the angles α, ϵ, and ϵ', where α is the angle between $\mathbf{v}_1$ and $\mathbf{v}'_1$, and both ϵ and ϵ' are the angle errors at measured image point coordinates of the matched points. The vectors $\mathbf{x}$ and $\mathbf{z}$ are given by $\mathbf{v}_i \times \mathbf{v}'_i$ and $\mathbf{y} \times \mathbf{x}$, respectively, and the vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ construct a basis of a coordinate system.

involved are coplanar. This obviously places a constraint on the vector $\mathbf{t}$.

However, we do not expect this constraint to be satisfied exactly for all point correspondences. Rather, we wish to know if it may be satisfied within a given error bound ϵ. A technical detail discussed in [10], [11] allows us to specify different bounds ϵ and ϵ' on the two points. This is not necessary to follow the argument further, but we will assume that $\mathbf{v}$ and $\mathbf{v}'$ are allowed different error bounds ϵ and ϵ'. If we allow $\mathbf{v}$ and $\mathbf{v}'$ to be perturbed in this way, then this means that they must lie inside cones of radius ϵ and ϵ', respectively, as shown in Fig. 3a.

The translation direction $\mathbf{t}$ must lie inside a wedge bounded by planes tangent to the two cones. The two normals of these planes are shown in Fig. 3b. For several matched points, the translation direction must lie inside all such wedges.

To solve the feasibility problem, we need to express the normals to the planes in terms of $(\mathbf{v}, \epsilon)$ and $(\mathbf{v}', \epsilon')$. Then, answering the feasibility problem is equivalent to solving the LP problem. We give the formulas for the normals below, without full details.

As shown in Fig. 4, let us assume that angles α, β, and ϵ are the angle between two axes of cones, the angle between bitangent planes and the cones, and radius error of matched points, respectively. Let $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ be vectors given by two cones $\mathbf{v}$ and $\mathbf{v}'$, as shown in Fig. 4.

Details of the following derivation are given in [11]. The vectors $\mathbf{x}$ and $\mathbf{z}$ are determined by the axes of two cones $\mathbf{v}$ and $\mathbf{v}'$ and by the vector $\mathbf{y}$, where two great circles meet as shown in Fig. 4. The vector $\mathbf{y}$ is expressed as follows:

$$\mathbf{y} = \frac{\sin(\epsilon)\mathbf{v}' + \sin(\epsilon')\mathbf{v}}{\sin(\beta)\sin(\alpha)}, \tag{10}$$

where β is the angle between the planes bitangent to two cones and the plane containing the axes of the two cones as illustrated in Fig. 4. This angle β is given by

$$\sin^2\beta = \frac{\sin^2(\epsilon) + 2\sin(\epsilon)\sin(\epsilon')\cos(\alpha) + \sin^2(\epsilon')}{\sin^2(\alpha)}, \tag{11}$$

where α, ϵ, and ϵ' are shown in Fig. 4.

The vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ form a basis for a coordinate system and serve to build equations of normals for the two half-spaces. From the work of the authors of [10], [11], given a pair of matched cones on $\mathbf{v}_i \leftrightarrow \mathbf{v}'_i$, we derive the two normals $\mathbf{n}_1$ and $\mathbf{n}_2$ of half-spaces as follows:

$$\mathbf{n}_1 = \sin(\beta)\mathbf{z} + \cos(\beta)\mathbf{x}, \tag{12}$$

$$\mathbf{n}_2 = \sin(\beta)\mathbf{z} - \cos(\beta)\mathbf{x}. \tag{13}$$

These equations provide two normals $\mathbf{n}_1$ and $\mathbf{n}_2$ for planes from a pair of matched points $\mathbf{x} \leftrightarrow \mathbf{x}'$, and eventually, will be used to get an intersection of all half-spaces from all matched pair of points. This is an intersection from only one camera, and the existence of the intersection tells us whether a problem is feasible for the optimal essential matrix in one camera. In this paper, we would like to deal with multiple cameras instead of single camera to find the optimal rotation and translation.

Multiple cameras. We represent each camera by a sphere centered at the camera center. Therefore, we have m spheres for an m-camera system. Associated with each sphere, as in Fig. 3, there is a polyhedral cone with apex positioned at the center of each camera, formed as the intersection of wedges defined by the point correspondences for that camera. These cones represent the direction of motion of each of the cameras. A correspondence of points in the kth camera generates two constraints of the form

$$\mathbf{n}^\top(\mathbf{c}'_k - \mathbf{c}_k) \geq 0, \tag{14}$$

where $\mathbf{c}_k$ is the center of kth camera and $\mathbf{c}'_k$ is the center of the kth camera after the motion. However, the constraints from different cameras involve different variables. To get a set of consistent constraints, we need to transform these cones so that they constrain the final position of a specific chosen one of the cameras, let us say the final position $\mathbf{c}'_1$ of the first camera.

This problem is the same as the triangulation problem considered in [18]. We will see how the cones given by the linear constraints are transformed by the assumed rotation of the camera. This is illustrated in Fig. 5.

To express (14) in terms of $\mathbf{c}'_1$ instead of $\mathbf{c}'_k$, we use the following relationship, which may easily be read from Fig. 5:

$$\mathbf{c}'_1 = \mathbf{c}_k + \hat{\mathtt{R}}(\mathbf{c}_1 - \mathbf{c}_k) + (\mathbf{c}'_k - \mathbf{c}_k)$$

Another way of justifying this equation is to observe that it is equivalent to $\mathbf{c}'_1 - \mathbf{c}'_k = \hat{\mathtt{R}}(\mathbf{c}_1 - \mathbf{c}_k)$, which simply expresses the effect of a rotation on the vector $\mathbf{c}_1 - \mathbf{c}_k$.

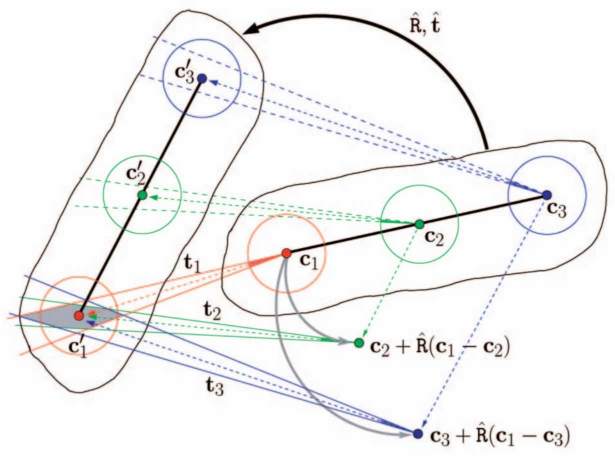

$\hat{\mathsf{R}}, \hat{\mathbf{t}}$
$\mathbf{c}'_3$
$\mathbf{c}'_2$
$\mathbf{c}_3$
$\mathbf{c}_2$
$\mathbf{c}_1$
$\mathbf{t}_1$
$\mathbf{t}_2$
$\mathbf{c}'_1$
$\mathbf{c}_2 + \hat{\mathsf{R}}(\mathbf{c}_1 - \mathbf{c}_2)$
$\mathbf{t}_3$
$\mathbf{c}_3 + \hat{\mathsf{R}}(\mathbf{c}_1 - \mathbf{c}_3)$

Fig. 5. The shaded region is the intersection of three polyhedra located on where each camera sees, $\mathbf{c}'_1$, the center of the first camera after a rigid motion. The shaded region is a feasible solution of the translation of this multicamera system.

By substituting for $(\mathbf{c}'_k - \mathbf{c}_k)$ into (14), we obtain the inequality for multiple camera systems as follows:

$$\begin{aligned} 0 &\leq \mathbf{n}^\top(\mathbf{c}'_k - \mathbf{c}_k) \\ &= \mathbf{n}^\top(\mathbf{c}'_1 - \mathbf{c}_k - \hat{\mathsf{R}}(\mathbf{c}_1 - \mathbf{c}_k)) \\ &= \mathbf{n}^\top\mathbf{c}'_1 - \mathbf{n}^\top(\mathbf{c}_k + \hat{\mathsf{R}}(\mathbf{c}_1 - \mathbf{c}_k)). \end{aligned}$$

This is the specific inequality involving $\mathbf{c}'_1$ after the transformation. Finding a solution satisfying all these inequalities is the same as finding an intersection of all the half-spaces.

We find the center of the first camera after the motion by an intersection of all wedges defined by all pairs of matched points. In other words, we find a solution to a set of linear constraints by Linear Programming. More precisely, this feasibility problem is described as follows:

$$\begin{array}{cl} \text{Does there exist} & \mathbf{c}'_1 \\ \text{satisfying} & \mathbf{n}_{i1}^\top\mathbf{c}'_1 - \mathbf{n}_{i1}^\top(\mathbf{c}_{k_i} + \hat{\mathsf{R}}(\mathbf{c}_1 - \mathbf{c}_{k_i})) \geq 0 \\ & \mathbf{n}_{i2}^\top\mathbf{c}'_1 - \mathbf{n}_{i2}^\top(\mathbf{c}_{k_i} + \hat{\mathsf{R}}(\mathbf{c}_1 - \mathbf{c}_{k_i})) \geq 0 \\ \text{for} & i = 1 \dots N, \end{array}$$

where $\mathbf{n}_{i1}$ and $\mathbf{n}_{i2}$ are the two normals derived from matched point i and k_i is the appropriate index of the camera generating the matched point $i; i = 1, \dots, N$.

The feasible region is the region of space satisfying all these inequalities. Solving this feasibility problem tells us the position of the center of the first camera after the motion, and finally, it gives us the optimal solution of the translation direction vector and its scale value.

Feasibility test. All half-spaces from matched pairs serve as inequalities in this LP problem. Given a total of N matched points in m cameras, the number of inequalities is $2N$. Generally, for five cameras with 100 points, the LP problem is to find the intersection of 1,000 half-spaces. If we use only LP to solve this problem, it will take too much computation time.

We suggest a way to reduce the LP computation time in this particular problem by testing the feasibility at an earlier stage before solving a full LP problem. The feasibility for a multicamera system depends on the feasibility of a single camera. If the feasibility test applied to a single camera fails, then we do not need to look at feasibility for the other cameras. This observation gives a method to reduce the computation time greatly.

The feasibility test for a single camera is done by reducing the number of variables for the translation direction vector to two variables as shown in [10], [11]. This feasibility test for a single camera can be adopted for greater speed of LP in multicamera systems.

The order of matched points also affects the speed of the feasibility test. A larger angle α between two matched points leads to a narrower wedge in which the translation direction must lie, and gives more chance to finish the feasibility test earlier. Thus, these points should be tested first. In our experiments, using a preemptive feasibility test makes the algorithm 90 times faster than an algorithm without this feasibility test.

Degeneracy. It is important to note that if the motion from one frame to the next has no rotation, then the scale of the translation cannot be computed. Because of the independence of the different cameras, there is an overall scale ambiguity, despite having known distances between the cameras. If the rotation is close to zero, the translation will be less reliable. Nevertheless, the rotation computation by the branch-and-bound algorithm will succeed.

7 Branch-and-Bound Algorithm

Given m calibrated cameras with a total of N matched points in each image, we can transform the matched points into vectors on the surface of a sphere by multiplying by the inverse of the calibration matrix and the inverse of the rotation matrix of each camera. With these simplified image vectors, the problem becomes easier to describe. The algorithm to find the optimal solution for the motion of a multicamera system is written as follows:

Optimal L_∞ Motion Estimation for Multicamera Systems

Given

m calibrated cameras with N matched points, $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$

Objective

Estimated optimal rotation and translation with scale.

Algorithm

1. Obtain an initial estimate for the motion by any means (a random guess if necessary) and compute an initial estimate δ_{min} for the minimal residual.
2. Create a queue and populate it with rotation blocks covering the whole of rotation space. (A reasonable choice is to divide the $3D$ rotation space into $10 \times 10 \times 10$ blocks.)
3. Carry out a branch-and-bound algorithm over rotation space, with the following steps (4 to 11).
4. Select a rotation block from the queue and consider its center as an initial estimate of rotation $\hat{\mathsf{R}}$ in rotation space.
5. Multiply $\hat{\mathsf{R}}$ by $\mathbf{x}$ to get axes of two cones $\mathbf{v} = \hat{\mathsf{R}}\mathbf{x}$ and $\mathbf{v}' = \mathbf{x}'$.
6. Let $\epsilon = \delta_{\text{min}} + r$, where r is the radius of the rotation block. Next determine whether there is a solution with rotation $\hat{\mathsf{R}}$ and residual less than ϵ by the following steps (7 to 10).
7. From the two cones about $\mathbf{v}$ and $\mathbf{v}'$ with half vertex-angle errors ϵ, compute two normals $\mathbf{n}_1$ and $\mathbf{n}_2$ from (13). Do this for all correspondences $\mathbf{v} \leftrightarrow \mathbf{v}'$.

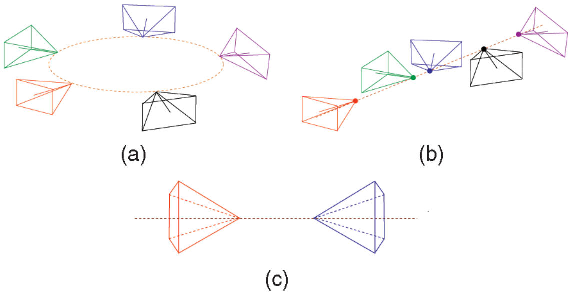


Fig. 6. Three types of generalized cameras used in the experiments with the synthetic data: (a) a general nonaxial camera rig of five cameras ("the locally central case"), (b) an axial camera rig of five cameras ("the locally central-and-axial case"), and (c) a nonoverlapping stereo head of two cameras ("the locally central-and-axial case").

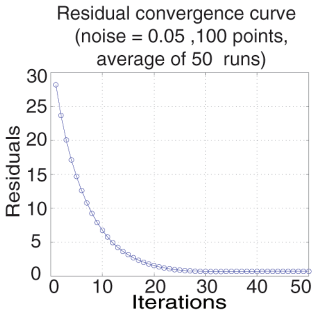


Fig. 7. An average convergence curve of the alternation procedure, i.e., residual error versus number of iterations. The curve was generated by averaging 50 runs with 0.05 degree standard deviation noise.

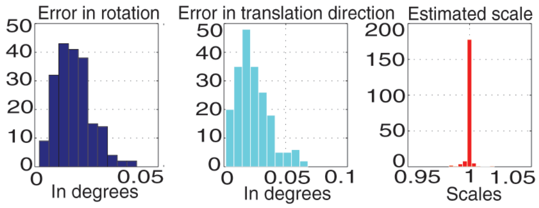


Fig. 8. Histograms of estimation accuracy based on 1,000 randomly simulated tests for a nonaxial multicamera rig. In all of these tests, we introduce angular noise at level of standard deviation 0.05 degree. The number of rays is 100.

8. Transform the two half-spaces to obtain inequality equations $\mathbf{n}_i^\top \mathbf{c}'_1 - \mathbf{n}_i^\top(\mathbf{c}_{k_i} + \hat{\mathtt{R}}(\mathbf{c}_1 - \mathbf{c}_{k_i})) \geq 0$.
9. Solve Linear Programming with the constraints.
10. If it is not a feasible problem, then discard the rotation block and continue. Otherwise divide the selected rotation block into subblocks, and queue these subblocks for further processing. In addition, since the LP problem provides an estimate for the translation $\mathbf{t}$, test the motion pair $(\hat{\mathtt{R}}, \mathbf{t})$ to see if this gives a better solution than the current best.
11. Repeat from step 4 until we achieve a desired error, then return the estimated rotation and translation.

8 Experiments

8.1 Synthetic Experiments for the Linear Method

We carry out three experiments with synthetic data. The synthetic data simulate three commonly used generalized cameras, which are 1) a general nonaxial camera rig, 2) an axial camera rig, and 3) a nonoverlapping stereo head. These three types of generalized cameras are shown in Fig. 6. The image size for each camera is about $1{,}000 \times 1{,}000$ pixels. The three cases have ranks 16, 14, and 14, respectively, from the analysis of the generalized epipolar equations in the previous sections. Gaussian noise with standard deviation 0.05 degree is added to the direction vector in Plücker line coordinates.

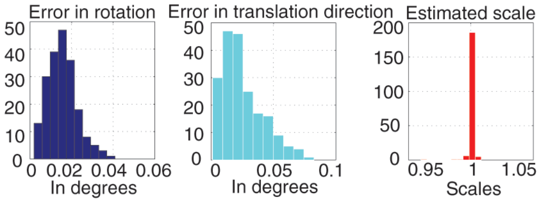


Fig. 9. Histograms of estimation accuracy based on 1,000 randomly simulated tests for an axial camera rig. In all of these tests, we introduce angular noise at a level of standard deviation 0.05 degree. The number of rays is 100.

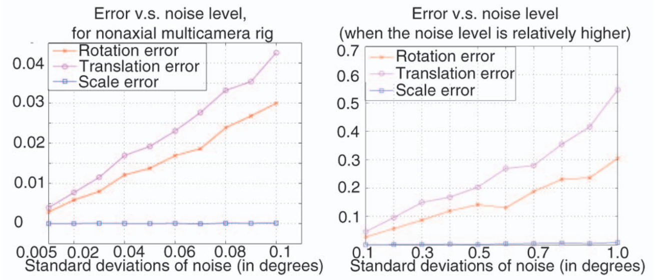


Fig. 10. Estimation accuracy (in rotation, translation, scale) as a function of noise level. The error in scale estimate is defined as $\|1 - \frac{\|\mathbf{t}\|}{\|\hat{\mathbf{t}}\|}\|$. Results are for simulated nonaxial camera rigs.

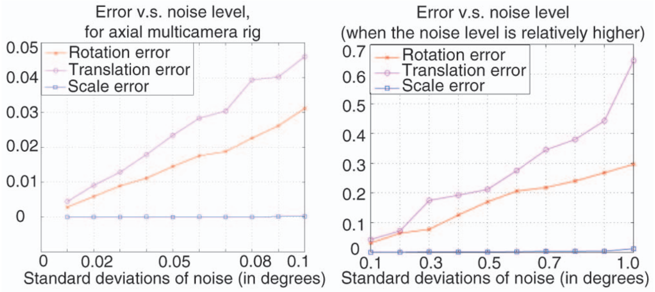


Fig. 11. Estimation accuracy (in rotation, translation, scale) as a function of noise level. The error in scale estimate is defined as $\|1 - \frac{\|\mathbf{t}\|}{\|\hat{\mathbf{t}}\|}\|$. Results are for simulated axial camera rigs.

In Fig. 7, an average convergence curve for 50 runs of the alternation method is plotted. As shown in Fig. 7, the residual error for the alternation method decreases rapidly in less than 20 iterations. For the first two cases in Fig. 6, 1,000 random runs are carried out and histograms of estimation errors are shown in Figs. 8 and 9. Graphs of the errors of the estimated rotation and the estimated translation from 1,000 trials are shown for the first two cases in Figs. 10 and 11. For the nonoverlapping stereo head, errors of the estimated rotation and the estimated translation are shown in Fig. 12. To see how much our method improves the estimations, another experiment with a monocular camera is carried out and the comparison between them is shown in Fig. 12. As seen in Fig. 12, our method gives better estimations than the monocular camera system.

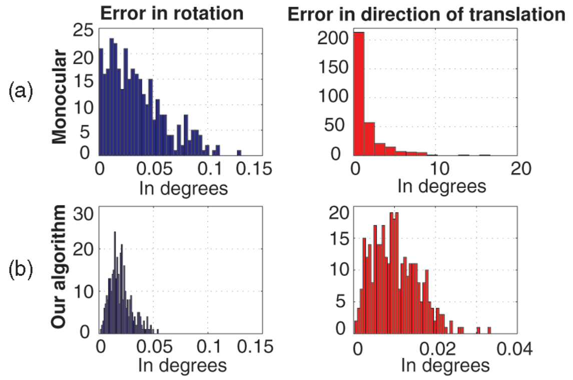


Fig. 12. Experiment results for a 2-camera stereo system. (a) Estimation errors in rotation and translation directions using one camera only (i.e., monocular). (b) Estimation errors obtained by the proposed method.

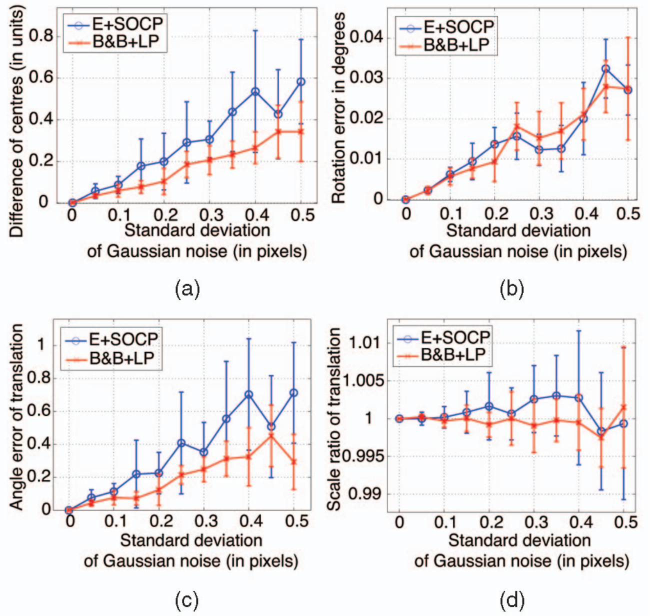


Fig. 13. Comparison of two methods: the SOCP based on the single essential matrix method by Kim et al. [18] (indicated as blue lines, "E+SOCP") and our proposed method based on a branch-and-bound algorithm with LP (indicated as red lines, "B&B+LP"). (a) The difference between the true position of camera and the estimated position of the camera at the final position. (b) Angle error of estimated rotation. (c) Angle error of estimated translation direction. (d) Scale error of estimated translation. The "B&B+LP" method gives more accurate position of camera though it has underestimation of rotation and translation directions compared with the "E+SOCP" method. The difference of the errors is less than 1 degree, so it is minimal. The less scale error of translation in the "B&B+LP" method shows why it estimates a better position of cameras at the final position.

8.2 Synthetic Experiments for the L_∞ Method

Experiments are conducted on synthetic data to show robustness.

A synthetic data set was generated for four cameras with 50 image points randomly located in space. A total of 200 points were projected onto four image planes, and the system of four cameras was moved by a rigid motion of rotation and translation. The 200 points were also projected onto another four image planes of cameras at the final position. When we process this synthetic data to estimate the motion by using our method, the CPU computation time is about 3.5 seconds in a standard Intel Core 2 CPU PC based on 32-bit instructions and a single process. The implementation is written in C++ and uses the GNU Linear Programming Kit (GLPK) [30]. As shown in Fig. 13, several experiments are conducted 10 times on the same synthetic data with increasing noise in pixels, and the distance error of centers is compared with the ground truth and its mean values are shown.

We have compared the performance with another method [18], which we call "E+SOCP" in this paper. This method uses a single essential matrix and SOCP to estimate the motion of multicamera systems. As seen in Fig. 13, our proposed method gives a better estimation for rotation and translation than E+SOCP.

8.3 Real Experiments for the Linear Method

An experiment with real data is carried out. The real data are obtained from a spherical imaging device, the Ladybug2 camera system [31]. The Ladybug2 camera system consists of six cameras in the head unit. There are five cameras along the ring of the head unit and one camera on top of the head unit as shown in Fig. 14. The average distance between cameras on the horizontal ring is about 3.71 cm and the average distance between the five horizontal cameras and the top camera is about 4.57 cm. Although this camera system is mainly used for taking images for spherical or omnidirectional vision, the six cameras are considered here as a multicamera system since the six camera centers are not at the same point. Accordingly, the Ladybug2 camera is a real example of the "locally central" case of generalized cameras.

To acquire the ground truth, a trajectory is generated from a computer-aided drawing tool, as shown in Fig. 15. This trajectory is an ∞-shape and it has marked positions for aligning the camera at every frame. As seen in Fig. 14, the bottom of the camera is flat. So, one of the edges on the bottom of the head unit can be aligned with the marked positions in the experiment. For the alignment, a target point on the edge is marked with a label. Then, the trajectory is printed on a piece of A2-size paper and the printed trajectory is attached under a piece of half-transparent paper with a 1 mm grid. All of the marked positions can be measured in millimeters in 2D coordinates,

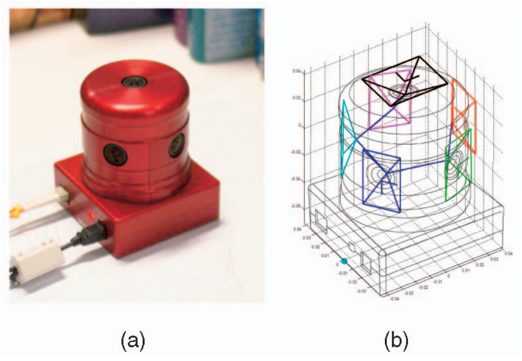


Fig. 14. (a) The Ladybug2 camera system consisting of five cameras on the side and one camera on the top of the head unit. A label is attached on the left-side edge of the bottom of the head unit, which is just under the red LED light. The label is used to align the camera with a trajectory printed on a piece of paper. (b) Positions of the six cameras in the Ladybug2 camera. The positions are retrieved from calibration information provided by Point Gray, Inc. The order of cameras is indicated by colors red, green, blue, cyan, magenta, and black. The label for the alignment is indicated by a cyan dot at the bottom of the head unit.

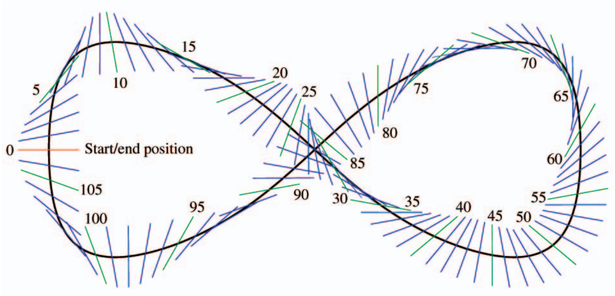


Fig. 15. An ∞-shape trajectory produced by a drawing tool. The trajectory is printed on a piece of paper and is for the path of the Ladybug2 camera in the experiment. The trajectory is a closed-loop and has 108 positions. A starting position and end position are shown by a red line segment, and the frame numbers are shown.

and they provide us the ground truth for the motion of the camera in this experiment.

For features to track in this real experiment, static objects such as books and some boxes are placed around the camera as shown in Fig. 16. The average distance between the camera and the 3D points on the books is about 30 cm. Then, the camera is manually moved and aligned with the marked positions at every frame.

A set of six images is captured by the Ladybug2 camera at each marked position. The number of marked positions is 108, so a total of 648 images are captured in this experiment. The size of each captured image is $1{,}024 \times 768$ pixels. All calibration information is provided by Point Gray, Inc., and the Ladybug SDK library is used to remove radial distortion in images [31].

A sample of six images captured by the Ladybug2 camera in the experiment is shown in Fig. 1b.

Features in the images are detected and tracked through six image sequences using 2D3 software [32]. Because of the wide-angle lenses of the Ladybug2 camera (2.5 mm focal length high-quality microlenses), there is a large amount of radial distortion in the captured images. So, radial distortion correction is applied to the coordinates of the features. After the radial distortion correction, a RANSAC algorithm is used to get rid of outliers from the features [33].

Given all inliers at every frame and camera calibration information, Plücker line coordinates for the inliers are represented in a local coordinate system. One of the six cameras in the Ladybug2 camera system is selected and aligned with the origin of the local coordinate system. With all of these real data, the estimated motion of the camera and its comparison with the ground truth are shown in Fig. 17. A 3D view of the estimated motion and positions of all six cameras of the Ladybug2 camera system are shown in Fig. 18. Note that the trajectory is a closed loop and the estimated positions of the cameras accumulate errors at every frame. Therefore, examining how well the trajectory is closed at the last frame can be one way of verifying the result. In this experiment, the estimation seems fine throughout all frames. However, there is a large displacement in the estimation at the moment of frame number 93. It tells us that the linear method is fairly applicable and gives good results, but in terms of robustness, we need a better way of minimizing residual errors in motion estimation.

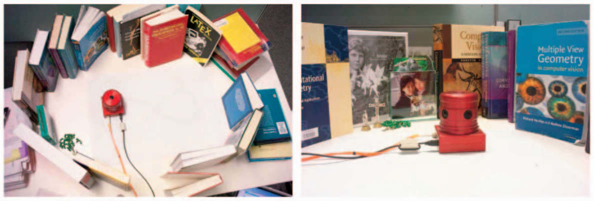


Fig. 16. Experimental setup with the Ladybug2 camera and books surrounding the camera. The camera is placed on a piece of A2-size paper on which the trajectory of 108 positions of the camera is printed.

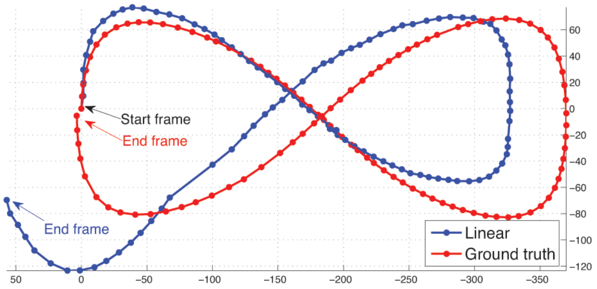
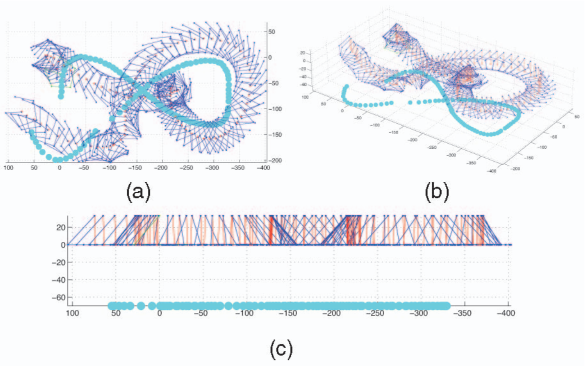


Fig. 17. Estimated motion of the Ladybug2 camera in the real experiment using our proposed "linear method," which is indicated by blue dots and lines. The ground truth of the motion is superimposed by red dots and lines. All of the estimated positions go relatively well until frame number 92 out of 108 frames. At frame number 93, the linear method gives a large amount of displacement error. However, after that frame, the estimation goes well again until the last frame. It tells us that our linear method needs to find some other ways of nonlinear estimation using bundle adjustment to improve the result. The measurement unit in this figure is millimeters.

8.4 Real Experiments for the L_∞ Method

We have used the same real data set that we used for the linear method. For 108 images, the motion of the 6-camera system is estimated and the results are shown and compared with the results of the "E+SOCP" method in Fig. 21. The graph in Fig. 21 shows that the estimated rotation and translation by our proposed method are more accurate than E+SOCP. The estimated trajectory of the cameras is superimposed on the ground truth in Fig. 19. Histograms of translation and rotation errors of the simulated motion are shown in Fig. 20. This analysis shows that the translation



(a)

(b)

(c)

Fig. 18. The estimated motion and position of six cameras of the Ladybug2 camera are plotted. The camera position is indicated by blue dots and lines. The axis of the Ladybug2 camera is shown by red lines. The marked position of the label attached on the head unit, which is aligned with the predefined trajectory, is shown by cyan dots. (a) Top view of the estimated motion and positions of the six cameras; (b) perspective view of the estimated motion and positions of the six cameras; (c) side view of the estimated motion and positions of the six cameras.

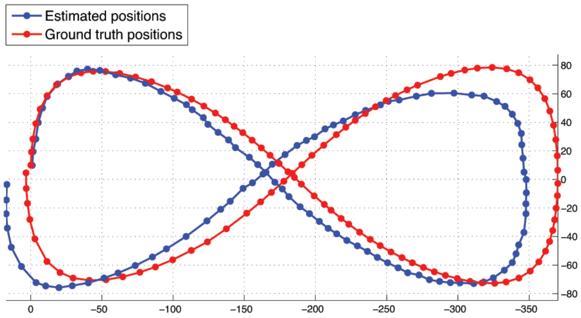


Fig. 19. Top view of the estimated trajectory of the camera and the ground truth from frames 0 to 108. The estimated trajectory is indicated using blue lines and dots to indicate the positions of the camera. The ground truth is illustrated using red lines and dots. The starting position of the cameras is the left middle point, which is (0, 0, 0) in the coordinate system. There is a drift in the estimated motion because of accumulated errors over frames.

direction is sensitive to noise in image coordinates. The estimated trajectories of the Ladybug camera and its six cameras with the marker are shown in Fig. 22.

9 Conclusion

We have discussed two methods of frame-to-frame motion estimation of a multicamera rig, a rapid linear method and a geometric L_∞-optimal algorithm.

Our linear method is fast and easy to implement compared to nonlinear methods. Although the estimate from the linear approach is sensitive to noise, particularly when a motion of the system is close to no-rotation motion, the linear approach is fast enough to be used for real-time applications, and it serves as a good initial guess for nonlinear refinement methods.

The nonlinear approach which gives an optimal solution in L_∞ showed robust estimation result of the motion. It could cure large displacement errors, which may happen in the linear method. However, the estimation of translation direction is more sensitive than that of the rotation in L_∞ as well. Because it uses an exhaustive search over rotation space, the nonlinear method is much slower than the linear method, but its runtime on the order of a few seconds is not exorbitant.

Although we have been concerned chiefly with nonoverlapping cameras in this paper, the methods developed are equally valid for cameras with overlapping fields of view as well.

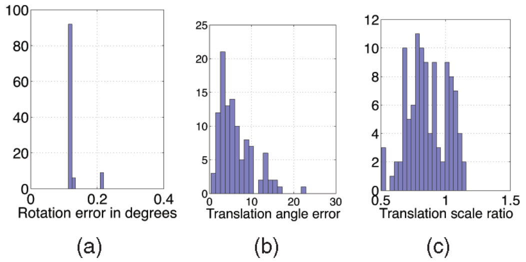


Fig. 20. Histograms of rotation and translation errors on the simulated motion. The simulated motion is generated with 0.1 degree standard deviation noise in the image coordinates. (a) Histogram of rotation errors. (b) Histogram of translation direction errors. (c) Histogram of translation scale errors. These show that the translation direction errors are sensitive to noise.

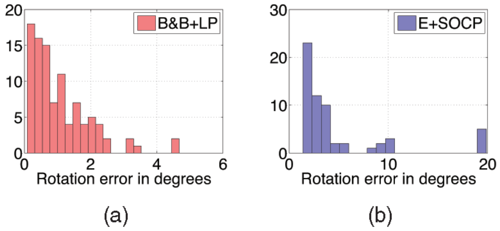


Fig. 21. (a) Histogram of the rotation error by our proposed "B&B+LP" method. It shows 1.08 degrees of the mean and 0.9 degree standard deviation. (b) Histogram of the rotation error by the "E+SOCP" method, which is based on the essential matrix from single camera and SOCP by Kim et al. [18]. It shows 4.73 degrees of the mean and 5.06 degree standard deviation. The proposed "B&B+LP" method estimates the rotation better than the "E+SOCP" method in real data experiments.

Appendix

Rank Conditions

We show how to generate examples that prove the rank conditions stated in this paper. We concentrate on the locally central axial camera. According to (6), the solution set has dimension 4, so the rank of the equation matrix must be at most 14. According to the argument in Section 4, to prove that the rank is generically equal to 14, we need only exhibit one single example where the rank is 14.

We consider a 3D point set consisting of the vertices of two unit cubes. The points (denoted by $\mathbf{X}_i$) are represented by the columns of the following array—15 points in all:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 2 & 2 & 3 & 3 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \end{bmatrix}.$$

This set of points is chosen arbitrarily. Just about any set of points will do. Define a rotation matrix

$$\mathtt{R} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and a translation $\mathbf{t} = (1, 0, 1)^\top$. Then define a second set of points $\mathbf{X}'_i = \mathtt{R}\mathbf{X}_i + \mathbf{t}$. Let $\mathbf{v}_i = (0, 0, 0)$ for $i = 1, \ldots, 7$ and $(0, 0, 1)$ for $i = 8, \ldots, 15$. Now, define points $\mathbf{x}_i = \mathbf{X}_i - \mathbf{v}_i$ and $\mathbf{x}'_i = \mathbf{X}'_i - \mathbf{v}_i$. There is no need to normalize the points $\mathbf{x}_i$ and $\mathbf{x}'_i$ to unit vectors since this will only change the following equation by a constant factor. One may verify that this condition,

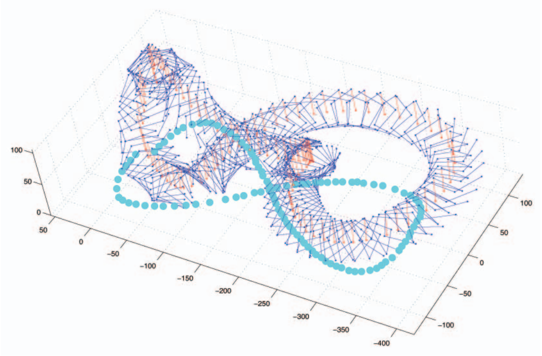


Fig. 22. The top-side view of the path of the six cameras (blue and red lines) and marker (cyan dots).

$$\mathbf{x}_i'^{\top}\mathtt{E}\mathbf{x}_i + (\mathbf{v}_i \times \mathbf{x}_i')^{\top}\mathtt{R}\mathbf{x}_i + \mathbf{x}_i'^{\top}\mathtt{R}(\mathbf{v}_i \times \mathbf{x}_i) = 0,$$

holds with $\mathtt{E} = [\mathbf{t}]_{\times}\mathtt{R}$. Writing this condition as a set of equations of the form (2), one may verify that the matrix $\mathtt{A}$ has rank 14. This may be done by computing the determinant of a 14×14 submatrix. For the sake of proving this rank condition, we have given this example in terms of points with integer coordinates so that one may rigorously show that the rank condition holds without worrying about floating-point numerical precision.

Continuing this example, one may divide matrix $\mathtt{A}$ into two parts $\mathtt{A} = [\mathtt{A}_{\mathrm{E}}\ \mathtt{A}_{\mathrm{R}}]$. Then one verifies that $\mathrm{rank}(\mathtt{A}_{\mathrm{R}}) = 6$ and $\mathrm{rank}((\mathtt{A}_{\mathrm{R}}\mathtt{A}_{\mathrm{R}}^{+} - \mathtt{I})\mathtt{A}_{\mathrm{E}}) = 8$. Finally, solving the equation $(\mathtt{A}_{\mathrm{R}}\mathtt{A}_{\mathrm{R}}^{+} - \mathtt{I})\mathtt{A}_{\mathrm{E}}\mathrm{vec}(\mathtt{E}) = 0$ retrieves the essential matrix $\mathtt{E}$.

Acknowledgments

This research was partly supported by NICTA, a research center funded by the Australian Government as represented by the Department of Broadband, Communications and Digital Economy, and the Australian Research Council through the ICT Centre of Excellence program. Jae-Hak Kim was a PhD student at the Australian National University when this manuscript was first submitted. During the period of the review process, he joined Queen Mary, University of London.

References

[1] M.D. Grossberg and S.K. Nayar, "A General Imaging Model and a Method for Finding Its Parameters," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 108-115, 2001.

[2] P. Sturm, "Multi-View Geometry for General Camera Models," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 206-212, June 2005.

[3] R. Pless, "Using Many Cameras as One," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 587-593, 2003.

[4] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyster, and P. Sayd, "Generic and Real-Time Structure from Motion," *Proc. British Machine Vision Conf.,* 2007.

[5] M. Lhuillier, "Effective and Generic Structure from Motion Using Angular Error," *Proc. Int'l Conf. Pattern Recognition,* pp. 67-70, 2006.

[6] R. Hartley and F. Schaffalitzky, "L_∞ Minimization in Geometric Reconstruction Problems," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 504-509, June 2004, http://dx.doi.org/10.1109/CVPR.2004.140.

[7] F. Kahl, "Multiple View Geometry and the L_∞-Norm," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1002-1009, 2005.

[8] K. Sim and R. Hartley, "Recovering Camera Motion Using L_∞ Minimization," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 1230-1237, 2006, http://dx.doi.org/10.1109/CVPR.2006.247.

[9] K. Sim and R. Hartley, "Removing Outliers Using the L_∞ Norm," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 485-494, 2006, http://dx.doi.org/10.1109/CVPR.2006.253.

[10] R. Hartley and F. Kahl, "Global Optimization through Searching Rotation Space and Optimal Estimation of the Essential Matrix," *Proc. IEEE Int'l Conf. Computer Vision,* Oct. 2007, http://dx.doi.org/10.1109/ICCV.2007.4408896.

[11] R. Hartley and F. Kahl, "Global Optimization through Rotation Space Search," *Int'l J. Computer Vision,* vol. 82, no. 1, pp. 64-79, Apr. 2009,

[12] H. Li, "A Practical Algorithm for L_∞ Triangulation with Outliers," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2007.

[13] J. Stolfi, *Oriented Projective Geometry.* Academic Press Professional, Inc., 1991.

[14] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström, "Solutions to Minimal Generalized Relative Pose Problems," *Proc. Workshop Omnidirectional Vision,* Oct. 2005.

[15] M. Byröd, K. Josephson, and K. Åström, "Improving Numerical Accuracy of Gröbner Basis Polynomial Equation Solvers," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[16] G. Schweighofer and A. Pinz, "Fast and Globally Convergent Structure and Motion Estimation for General Camera Models," *Proc. British Machine Vision Conf.,* 2006.

[17] J.-M. Frahm, K. Köser, and R. Koch, "Pose Estimation for Multi-Camera Systems," *Proc. DAGM,* 2004.

[18] J.-H. Kim, R. Hartley, J.-M. Frahm, and M. Pollefeys, "Visual Odometry for Non-Overlapping Views Using Second-Order Cone Programming," *Proc. Asian Conf. Computer Vision,* vol. 2, pp. 353-362, Nov. 2007, http://dx.doi.org/10.1007/978-3-540-76390-1_35.

[19] J.-H. Kim, H. Li, and R. Hartley, "Motion Estimation for Multi-Camera Systems Using Global Optimization," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2008, http://dx.doi.org/10.1109/CVPR.2008.4587680.

[20] H. Li, R. Hartley, and J.-H. Kim, "Linear Approach to Motion Estimation Using Generalized Camera Models," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2008, http://dx.doi.org/10.1109/CVPR.2008.4587545.

[21] R. Gupta and R.I. Hartley, "Linear Pushbroom Cameras," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 9, pp. 963-975, Sept. 1997, http://dx.doi.org/10.1109/34.615446.

[22] R. Hartley, "Photogrammetric Techniques for Panoramic Cameras," *Proc. SPIE Conf. Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision,* pp. 127-139, Apr. 1993, http://dx.doi.org/10.1117/12.155798.

[23] R.I. Hartley and T. Saxena, "The Cubic Rational Polynomial Camera Model," *Proc. Defense Advanced Research Projects Agency Image Understanding Workshop,* pp. 649-653, 1997.

[24] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall, "Non-Perspective Imaging and Rendering with the Crossed-Slits Projection," technical report, Leibnitz Center, Hebrew Univ. of Jerusalem, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.5676, 2002.

[25] D. Feldman, T. Pajdla, and D. Weinshall, "On the Epipolar Geometry of the Crossed-Slits Projection," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 988-995, 2003.

[26] R.I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision,* second ed. Cambridge Univ. Press, 2004.

[27] R.I. Hartley, "Projective Reconstruction and Invariants from Multiple Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 10, pp. 1036-1041, Oct. 1994, http://dx.doi.org/10.1109/34.329005.

[28] H. Li and R. Hartley, "Five-Point Motion Estimation Made Easy," *Proc. Int'l Conf. Pattern Recognition,* pp. 630-633, Aug. 2006, http://dx.doi.org/10.1109/ICPR.2006.579.

[29] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley, "Robust 6DOF Motion Estimation for Non-Overlapping Multi-Camera Systems," *Proc. Workshop Applications of Computer Vision,* pp. 1-8, Jan. 2008, http://dx.doi.org/10.1109/WACV.2008.4544011.

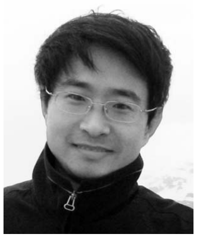
[30] GNU Project, *GNU Linear Programming Kit Version 4.9,* http://www.gnu.org/software/glpk/, 2009.

[31] P.G.R. Inc., "Ladybug2 Camera," http://www.ptgrey.com, 2006.

[32] 2d3 Limited, "2d3 Boujou," http://www.2d3.com, 2005.

[33] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM,* vol. 24, no. 6, pp. 381-395, 1981.

Jae-Hak Kim received the BE degree in computer engineering from Dong-A University, Republic of Korea, in 1997, the MSc degree in computer science and engineering from the Pohang University of Science and Technology (POSTECH), Republic of Korea, in 1999, and the PhD degree in engineering from the Australian National University (ANU) in 2008 with a thesis on multicamera systems. He is currently a postdoctoral research assistant in the Department of Computer Science at Queen Mary, University of London, United Kingdom. He was with the National ICT Australia (NICTA) and also with ANU as a research assistant. His research interests include multiple-view geometry, multicamera systems, feature tracking, global optimization, and nonrigid structure from motion. He is a member of the IEEE and the IEEE Computer Society.



Hongdong Li received the MSc and PhD degrees in information and electronic engineering from Zhejiang University, China, in 1996 and 2000, respectively. He is currently a fellow (senior lecturer) in the Research School of Information Sciences and Engineering (RSISE) at the Australian National University (ANU), Canberra. He is also seconded to the National ICT Australia (NICTA) as a research scientist. After graduation, he was first a lecturer and then an associate professor at Zhejiang University, before joining RSISE ANU and NICTA. His past research projects include handwriting and handwritten Chinese character recognition and autonomous land vehicle navigation (unmanned ground mobile robot). His current research interests include geometric computer vision, structure from motion, image restoration, medical image and bionic eyes, and also discrete combinatorial optimization. He constantly serves as a program committee member and/or reviewer for the top three forums of computer vision research, i.e., ICCV, CVPR, and ECCV. He is a member of the IEEE and the IEEE Computer Society.

Richard Hartley received the degree from the University of Toronto in 1976 with a thesis on knot theory. He is currently with the Computer Vision Group at the Australian National University and also with National ICT Australia, a government-funded research institute. He worked in this area for several years before joining the General Electric Research and Development Center, where he developed a computer-aided electronic design system called the Parsifal Silicon Compiler, described in his book *Digit Serial Computation*. Around 1990, he developed an interest in computer vision, and in 2000, he coauthored (with Andrew Zisserman) a book on multiple-view geometry. He has written papers on knot theory, geometric voting theory, computational geometry, computer-aided design, and computer vision. He holds 32 US patents. In 1991, he was awarded GE's Dushman Award. He is a fellow of the IEEE and a member of the IEEE Computer Society.