A linear approach to motion estimation using generalized camera models

Hongdong Li, Richard Hartley, Jae-hak Kim

InfoEng, RSISE, Australian National University
Canberra Labs, National ICT Australia (NICTA)
{firstname.lastname}@anu.edu.au

Abstract

A well-known theoretical result for motion estimation using the generalized camera model *is that 17 corresponding image rays can be used to solve linearly for the motion of a generalized camera. However, this paper shows that for many common configurations of the generalized camera models (e.g., multi-camera rig, catadioptric camera etc.), such a simple 17-point algorithm does not exist, due to some previously overlooked ambiguities.*

We further discover that, despite the above ambiguities, we are still able to solve the motion estimation problem effectively by a new algorithm proposed in this paper. Our algorithm is essentially linear*, easy to implement, and the computational efficiency is very high. Experiments on both real and simulated data show that the new algorithm achieves reasonably high accuracy as well.*

1. Introduction

In the study of "using many cameras as one" for motion estimation [13], Pless has derived the *generalized epipolar constraint (GEC)* expressed in terms of the generalized essential matrix which is a 6×6 matrix capturing the 6 degrees-of-freedom motion of the multi-camera rig system . In his original derivation, the key idea is to use the Generalized Camera Model (GCM) to replace the **image pixels** by a set of unconstrained **image rays**, each described by a Plücker line vector $\mathbf{L}$.

As an analogy to the conventional epipolar equation for a pinhole cameras, the GEC is also written as a homogeneous 6×6 matrix equation, giving

$$\mathbf{L}'^{\top} \begin{bmatrix} \mathtt{E} & \mathtt{R} \\ \mathtt{R} & 0 \end{bmatrix} \mathbf{L} = 0, \qquad (1)$$

where $\mathtt{E} = [\mathbf{t}]_{\times}\mathtt{R}$ is similar to the conventional essential matrix, and $\mathbf{L}$ and $\mathbf{L}'$ are two Plücker line vectors representing two corresponding rays. One substantial difference between this GEC and the conventional epipolar equation is that all 6 degrees of freedom (of camera motion) can be recovered by solving the GEC.

This GEC also suggests a 17-point (or 17-ray) algorithm that could possibly be used to solve the problem. Because there are in total 18 unknowns, one may think to solve for $\mathtt{E}$ and $\mathtt{R}$ linearly by using 17 pair of corresponding image rays.

Specifically, the GEC can be re-written as $\mathtt{A}\mathbf{X} = 0$, where $\mathtt{A}$ is an $N \times 18$ equation matrix and $\mathbf{X}$ is the vector of unknowns made up of the entries of $\mathtt{E}$ and $\mathtt{R}$. The usual way of solving this system is to take the Singular Value Decomposition (SVD) of $\mathtt{A} = \mathtt{UDV}^{\top}$, in which case the solution for $\mathbf{X}$ is the last column of the matrix $\mathtt{V}$, corresponding to the smallest singular value ([6]). We refer to this as the *standard SVD algorithm*. This conclusion has been explicitly claimed by many other authors as well [17, 10, 14, 11].

Somewhat surprisingly, no one has actually given numerical results for such an SVD algorithm in a real imaging situation. The original paper of Pless gives no account of numerical tests on the 17-point algorithm.

It is shown in the present paper that in fact the standard SVD algorithm **does not work at all** in many common scenarios involving the generalized camera model. For example, we have found it is not possible to apply the standard SVD algorithm to *non-overlapping multi-camera rigs* in a straight-forward way. The reason is very simple, for these common camera configurations the rank of the above coefficient matrix $\mathtt{A}$ is less than 17 (under noise-free condition). In other words, there are extra **ambiguities** (degeneracies/singularities) in the linear equation system, which lead to a whole *family* of solutions.

In this paper, we characterize three typical generalized camera configurations—locally-central, axial and stereo—where we prove that the ranks are 16, 14 and 14, respectively.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

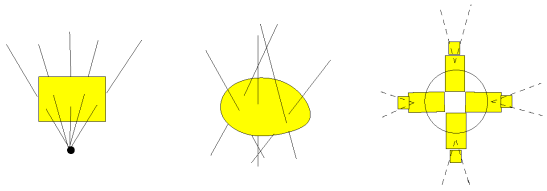

Figure 1. *This paper studies the motion estimation problem using a Generalized Camera. Here illustrates 3 typical instances of the generalized camera. The left one is a traditional pinhole camera. The central one shows a generalized camera model containing a set of image rays or 'raxels' ([5]). The right one is a non-overlapping multi-camera rig.*

Yet, more remarkably, we show that even though the ranks are deficient, it is still possible to solve for the camera motion linearly and uniquely (without suffering from the degeneracy). Specifically, we show the following results.

1. For motion of a general multi-camera rig, where image points are not matched from one camera to another, the equation matrix $\mathtt{A}$ will have rank 16. Consequently, there exists a 2-parameter family of solutions for $(\mathtt{E}, \mathtt{R})$ to the GEC equations.

2. For an axial camera, defined by the property that all imaging rays intersect a single line the equation matrix has rank 16, and hence a 2-parameter family exists for $(\mathtt{E}, \mathtt{R})$.

3. For a multi-camera rig with all the cameras mounted in a straight line, the rank will be 14. Thus, there exists a 4-parameter family of solutions. This includes the popular case of stereo (binocular) head. This setup is a mixture of the two cases enumerated above.

4. Remarkably, we show that: despite the rank deficiency of the equation matrix, it is still possible to solve the $\mathtt{E}$ part **uniquely** by our new **linear algorithm**. From this $\mathtt{E}$ we may further estimate the rotation $\mathtt{R}$ and translation $\mathtt{t}$, as well as the scale of the motion.

The above results may seem paradoxical given the existence of multi-dimensional families for $(\mathtt{E}, \mathtt{R})$, but it will be made clear in the rest of this paper. The key observation is that all ambiguity lies only in the estimate of $\mathtt{R}$ but not in the estimate of $\mathtt{E}$, provided simple conditions are met.

Alternation. Although our linear algorithm for computing the motion gives generally favorable performance, the result may be further improved.

For this purpose, we propose an iterative scheme via alternation between solving for the rotation $\mathtt{R}$ and the translation $\mathtt{t}$. Each iteration is a simple solution of linear equations of low dimension. This scheme seems straightforward and easy to implement, yet we point out critical pitfalls, and suggest remedy accordingly.

2. Previous work

Using non-traditional non-central cameras for motion estimation has recently attracted much attention from both the research community and practitioners. The most common case is that of a set of many cameras, often with non-overlapping views, attached to a vehicle. This is particularly relevant to recent application to urban mapping ([2]).

Pless predicted that it is possible to solve the equation system linearly using 17 corresponding rays (hereafter we will refer to this algorithm as the 17-point algorithm). In his paper, he however hinted that the generalized epipolar equations may accept multiple non-unique solutions. In other words, there might be ambiguities for certain cases. This important issue remains however unexplored in that paper.

Later on, Sturm unified the theory of multi-view geometry for generalized camera models [17][18]. He also mentioned that in order to solve the non-central generalized epipolar equations 17 points are necessary. Again, no experiment was provided.

This is somewhat surprising, because, by contrast, many more *nonlinear algorithms* (and results) have been reported for this problem. Molana and Geyer used a nonlinear manifold optimization approach to solve the generalized motion estimation problem [10]. Lhuillier proposed a method based on iterative bundle adjustment minimizing a new angular error [7]. Papers [16] and [1] used Groebner basis techniques to derive a polynomial solver for the minimal case of the problem. A recent paper [11] confirmed the existence of ambiguity, and provided a non-linear incremental bundle adjustment solution to it. Schweighofer and Pinz suggested a globally-convergent nonlinear iterative algorithm [14]. They derived an object-space cost function and used 3D triangulation as the intermediate goal of minimization. It is well known that those non-linear algorithms often require proper initialization. This made us even curious, as none of the above nonlinear algorithms actually used the linear 17-point algorithm for initialization.

There are other related research efforts on motion estimation using non-traditional cameras. Frahm, Koser and Koch [4] proposed a motion estimation algorithm using multi-camera rigs. Dellaert and Tariq [19] designed an embedded (miniature) multi-camera rig for helping people suffering from visual impairment. Neumann and Fermuller et al. solved motion estimation from optical flow using polydioptric camera [12]. Vidal et al used multiple panoramic images for estimating egomotion [15]. Pajdla et al derived the epipolar geometry for the crossed-slit camera [3]. Clipp and Pollefey et al in [2] suggested an *ad hoc* method that combines the five-point algorithm [9] with a scale estimator.

In the following sections, we solve the problem effectively and linearly. The estimates we obtain are remarkably accurate.

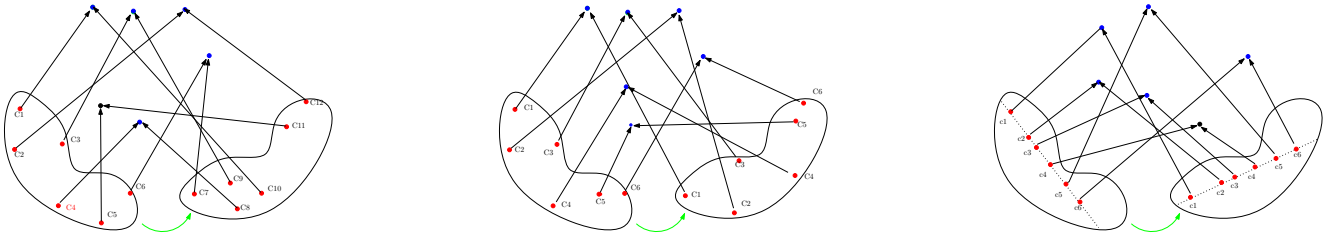

Figure 2. *From left to right, the figure illustrates two-view motion estimation from a generalized camera of 3 different types (cases). Left: the most general case, where all the image rays are entirely unconstrained; Middle: the locally-central case, where for every 3D scene point the projection center is unique and fixed locally in camera frame; Right: the axial case, where all image rays must intersect a common line called as axis. Note that in the latter two cases the relative positions of all the camera centers are fixed in the local frame. In this paper we show that the ranks of generalized epipolar equations corresponding to the above three cases are respectively* $r = 17, r = 16$ *and* $r = 14$*. The standard SVD algorithm works only for the first case, while our new algorithm applies to all 3 cases.*

3. Analysis of Degeneracies

The derivation of the generalized epipolar equation can be easily obtained from the Plücker coordinate representation of 3D lines (see [13]). An image ray passing through a point $\mathbf{v}$ (e.g., camera center) with unit direction $\mathbf{x}$ can be represented by a Plücker 6-vector $\mathbf{L} = (\mathbf{x}^\top, (\mathbf{v} \times \mathbf{x})^\top)^\top$.

Using this representation we then re-state the GEC as follows:

$$\mathbf{x}_i{}^\top \mathtt{E} \mathbf{x}'_i + \mathbf{x}_i{}^\top \mathtt{R} (\mathbf{v}'_i \times \mathbf{x}'_i) + (\mathbf{v}_i \times \mathbf{x}_i)^\top \mathtt{R}\, \mathbf{x}'_i = 0\,. \quad (2)$$

In the ground-truth solution to these equations, 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.

3.1. Degeneracies

We identify several degeneracies for the set of equations arising from (2), which cause the set of equations to have smaller than expected rank. Suppose we wish to show that the rank of the system is $r < 17$. 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. In this paper, we do not explicitly exhibit examples where the rank attains the claimed value, but in all cases this has been verified by example.

The most general case. In the most general case (see fig-2 left), 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 a degenerate case of a "generalized camera" consisting of a set of locally central projection rays (see fig-2 middle). The commonly used (non-overlapping) multi-camera rigs are examples of this case. When the camera rig moves from an initial to a final position, points are tracked. We assume no point from one component camera to another is used, so that all point correspondences are for points seen in the same camera. 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 (2) are now

$$\mathbf{x}_i{}^\top \mathtt{E} \mathbf{x}'_i + \mathbf{x}_i{}^\top \mathtt{R} (\mathbf{v}_i \times \mathbf{x}'_i) + (\mathbf{v}_i \times \mathbf{x}_i)^\top \mathtt{R}\, \mathbf{x}'_i = 0\,. \quad (3)$$

Now let $(\mathtt{E}, \mathtt{R})$ be a 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 $(\mathtt{0}, \mathtt{I})$ in (3) results in $\big(\mathbf{x}_i{}^\top (\mathbf{v}_i \times \mathbf{x}'_i) + (\mathbf{v}_i \times \mathbf{x}_i)^\top \mathbf{x}'_i\big)$, 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 (2) is found: the ambiguity is contained entirely in the estimation of $\mathtt{R}$, while the essential matrix $\mathtt{E}$ is still able to be determined uniquely up to scale.

Axial cameras. Our second example of degenerate configuration is what we will call an axial camera (see fig-2 right). 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 non-central catadioptric or fisheye cameras mounted with collinear axes.

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 will assume that the origin of the world coordinate system lies on the axis, and examine the solution set of equations (2) for this case.

In this 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 (2) then takes the form

$$\mathbf{x}_i{}^\top \mathtt{E} \mathbf{x}'_i + \alpha_i (\mathbf{w} \times \mathbf{x}_i)^\top \mathtt{R}\, \mathbf{x}'_i + \alpha'_i \mathbf{x}_i{}^\top \mathtt{R} (\mathbf{w} \times \mathbf{x}'_i) = 0\,. \quad (4)$$

Suppose that $(\mathtt{E}, \mathtt{R})$ is the true solution to these equations. Another solution is given by $(0, \mathbf{w}\mathbf{w}^\top)$, It satisfies the equation (4) 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 (4) 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. We have seen already that for locally central projections, $(0, \mathtt{R})$ 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 (4). We may now identify a further solution $(0, [\mathbf{w}]_\times)$, since

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

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

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

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 4-dimensional family of solutions.

4. Algorithm

Next, we shall give a new algorithm based on (2) 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 (2) 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 $\mathtt{A}(\mathrm{vec}(\mathtt{E})^\top, \mathrm{vec}(\mathtt{R})^\top)^\top = \mathbf{0}$. 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 are $\mathtt{R} = \mathtt{I}$ 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 non-linear, 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 fam-

ily of solutions (of dimension 2 to 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 9-dimension 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

$$\mathtt{A}(\mathrm{vec}(\mathtt{E})^\top, \mathrm{vec}(\mathtt{R})^\top)^\top = \mathbf{0}$$

we find the solution that minimizes

$$\|\mathtt{A}\,(\mathrm{vec}(\mathtt{E})^\top, \mathrm{vec}(\mathtt{R})^\top)^\top\| \text{ subject to } \|\mathtt{E}\| = 1\ ,$$

instead of $\|(\mathrm{vec}(\mathtt{E})^\top, \mathrm{vec}(\mathtt{R})^\top)\| = 1$ 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 [6] (section 9.6, page 257) in a more general form. Here we summarize the method. Write the equations as $\mathtt{A}_\mathtt{E}\mathrm{vec}(\mathtt{E}) + \mathtt{A}_\mathtt{R}\mathrm{vec}(\mathtt{R}) = \mathbf{0}$, where $\mathtt{A}_\mathtt{E}$ and $\mathtt{A}_\mathtt{R}$ are submatrices of $\mathtt{A}$ consisting of the first and last 9 columns. Finding the solution that satisfies $\|\mathrm{vec}(\mathtt{E})\| = 1$ is equivalent to solving

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

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

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 GEC equations in terms of a world coordinate system where the origin lies on the axis. This is an essential (non-optional) 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 mid point between the two cameras. In addition, we scale the rays such that each of the two cameras lies at unit distance from the origin, in opposite directions.

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.

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 GEC.

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 (2) and compute $\mathbf{t}$ directly from the relationship

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

The only unknown in this set of equations is the translation vector $\mathbf{t}$ which may then be computed using linear-least squares. The computed $\mathbf{t}$ will not have scale ambiguity.

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

1. **Given:** a set of correspondences $(\mathbf{x}_i, \mathbf{v}_i) \leftrightarrow (\mathbf{x}'_i, \mathbf{v}'_i)$. 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.

2. **Normalization:** Center the cameras by the transformations $\mathbf{v}_i \leftarrow \mathbf{v}_i - \bar{\mathbf{v}}$, $\mathbf{v}'_i \leftarrow \mathbf{v}'_i - \bar{\mathbf{v}}$ for some $\bar{\mathbf{v}}$, normally the centroid of the different camera centers. For axial cameras, it is essential that $\bar{\mathbf{v}}$ be a point on the axis of the cameras. Next, scale so that the cameras are approximately unit distance from the origin.

3. Form the set of linear equations $\mathtt{A}_\mathtt{E}\mathrm{vec}(\mathtt{E}) + \mathtt{A}_\mathtt{R}\mathrm{vec}(\mathtt{R}) = \mathbf{0}$ using (2).

4. Compute the pseudo-inverse $\mathtt{A}_\mathtt{R}^+$ of $\mathtt{A}_\mathtt{R}$ and write the equation for $\mathrm{vec}(\mathtt{E})$ as $\mathtt{B}\,\mathrm{vec}(\mathtt{E}) = 0$, where $\mathtt{B} = (\mathtt{A}_\mathtt{R}\mathtt{A}_\mathtt{R}^+ - \mathtt{I})\mathtt{A}_\mathtt{E}$. Solve this equation using the standard SVD algorithm to find $\mathtt{E}$.

5. Decompose $\mathtt{E}$ to get the twisted pair of rotation matrices $\mathtt{R}$ and $\mathtt{R}'$.

6. Knowing possible rotations, solve equations (2) to compute $\mathbf{t}$ linearly. The equations are non-homogeneous 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. 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

$$\sum_i \|\mathbf{x}_i{}^\top [\mathbf{t}]_\times \mathtt{R}\, \mathbf{x}'_i + \mathbf{x}_i{}^\top \mathtt{R}(\mathbf{v}'_i \times \mathbf{x}'_i) + (\mathbf{v}_i \times \mathbf{x}_i)^\top \mathtt{R}\, \mathbf{x}'_i\|^2. \quad (5)$$

The matrix $\mathtt{R}$ so found may not be orthogonal, but it may be corrected at the end. Note that the cost (5) being minimized decreases at each step of the iteration. 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 can not enforce this constraint.

The way to solve this is by modifying the equations (2) 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 (5) by β results in a cost function

$$\sum_i \left| \mathbf{x}_i{}^\top [\hat{\mathbf{t}}]_\times \mathtt{R}\, \mathbf{x}'_i + \beta \left(\mathbf{x}_i{}^\top \mathtt{R}(\mathbf{v}'_i \times \mathbf{x}'_i) + (\mathbf{v}_i \times \mathbf{x}_i)^\top \mathtt{R}\, \mathbf{x}'_i \right) \right|^2 \quad (6)$$

in which $\hat{\mathbf{t}}$ is a unit vector, and 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, that 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, since otherwise will result in increasing cost.

By a sequence of such alternating steps, the algebraic cost function associated with (6) is minimized subject to $\|\hat{\mathbf{t}}\| = 1$ and $\|\mathtt{R}\| = 3$, the cost diminishing at every step. In this way, we can not 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.

6. Experiments

To demonstrate the proposed algorithm works well in practice, we conduct extensive experiments on both synthetic data and real image data using multi-camera rig configuration.

For both real and simulated experiments, the algorithm's performance is characterized by the following criteria.

- Error in $\mathtt{E}$: $\epsilon_E = \frac{\|\mathtt{E}-\hat{\mathtt{E}}\|}{\|\mathtt{E}\|}$;
- Error in $\mathtt{R}$: $\epsilon_R = \|\mathtt{R} - \hat{\mathtt{R}}\|$;
- Angle difference in $\mathtt{R}$: $\delta_\theta = \cos^{-1}\left(\frac{\mathrm{Tr}(\mathtt{R}\hat{\mathtt{R}}^\top)-1}{2}\right)$;
- Direction difference in $\mathbf{t}$: $\delta_\mathbf{t} = \cos^{-1}\left(\frac{\mathbf{t}^\top \cdot \hat{\mathbf{t}}}{\|\mathbf{t}\|\|\hat{\mathbf{t}}\|}\right)$;

The symbols used above are defined as following. We denote $\mathtt{E}, \mathtt{R}, \mathbf{t}$ as the ground-truth information used in simulations, and $\hat{\mathtt{E}}, \hat{\mathtt{R}}, \hat{\mathbf{t}}$ the corresponding estimated versions. Note that all these data (matrix or vector) are defined with absolute scales (rather than defined up to a scale). Further denote the $\hat{\mathtt{R}}$ and $\hat{\mathbf{t}}$ as the final results obtained after the *alternation*, while $\hat{\mathtt{E}}$ is computed by $\hat{\mathtt{E}} = [\hat{\mathbf{t}}]_\times \hat{\mathtt{R}}$. All the norms are Frobenius norms.

6.1. Tests on simulated multi-camera rigs

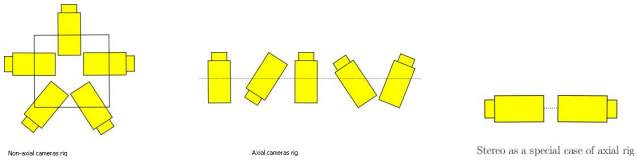

Figure 3. *We simulate three different configurations of multi camera rigs. Left: a general non-axial camera rig; middle: an axial camera rig; right: a non-overlap stereo head. The standard SVD algorithm is not applicable to any of these cases.*

We simulate three cases of multi-camera rigs (see fig-3). The first two cases—one is non-axial and one is axial—each consists of five pinhole cameras, and the last case is a non-overlapping stereo (binocular) system . The synthetic image size is about 1000×1000 pixels. We do not use any cross camera matches, therefore they both satisfy the locally-central projection model.

We build up GEC equations for each of these three cases. When there is no noise, the observed rank of each of the equations is 16, 14 and 14. This has confirmed our theoretical prediction.

We add Gaussian noise of 0.05 degrees in std to the direction part of the Plücker vectors. This is a reasonable level of noise, as in the image plane it roughly corresponds to one pixel std error in our experiments.

We test our linear algorithm+alternation algorithm. Experiments confirm that the alternation procedure always decreases the residual and thus improves the estimation accuracy. An average convergence curve is illustrated in fig-4.

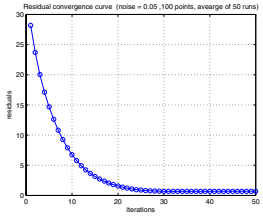

Figure 4. *An average convergence curve of the alternation procedure, i.e., residual error v.s. number of iterations. The curve was generated by averaging 50 tests with 0.05 degrees std noise.*

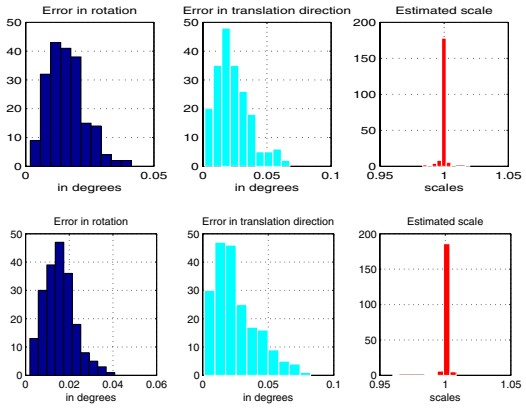

Figure 5. *Histograms of estimation accuracy based on* 1000 *randomly simulated tests. Top row: results for a non-axial multi-camera rig; Bottom row: results for an axial camera rig. In all these tests, we introduce angular noise at level std=0.05 degrees. The number of image rays are 100.*

Figure-5 gives the histograms of estimation accuracy for the first two cases. Figure-6 shows the estimation accuracy as a function of noise levels for both cases. The number of random trials is 1000.

Figure-7 shows results for the two-camera case, i.e.,a stereo system. For comparison we also gives the estimation accuracy obtained by deliberately using only one camera (i.e., the monocular case). To be fair we use the same set (same number) of matching points. It is clearly that our algorithm gives much better results.

6.2. Tests on real Ladybug camera

For real data experimentation, we use PointGrey's Ladybug omnidirectional camera, which is a multi-camera rig with 6 pinhole camera units. During experiments, we move the Ladybug in a controlled way along a prescribed trajectory on a plotting paper (with a coordinates grid) placing on a planar table. In this way we can obtain measured ground-truth motion (see fig-8). At each step of the movements of the camera we take one image from each camera, and call them a 'frame'. In total we have captured 101 frames for an

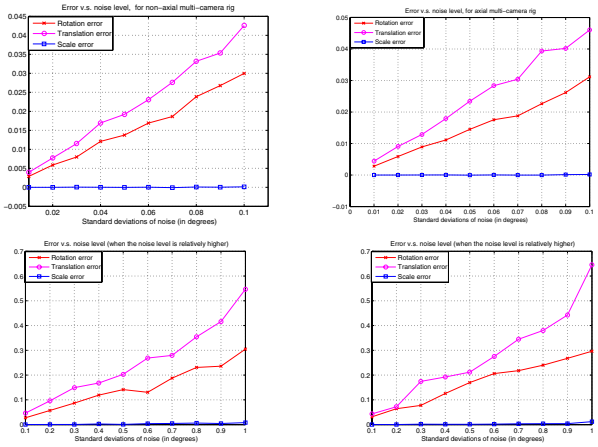

Figure 6. *This figure shows estimation accuracy (in rotation, translation, scale) as a function of noise level. The error in scale estimate is defined as* $|1 - \frac{\|t\|}{\|\hat{t}\|}|$. *Top row left: results for simulated non-axial camera rigs; Top row right: results for simulated axial camera rigs. Bottom row shows the same results as in the top row, but with much higher levels of noise.*

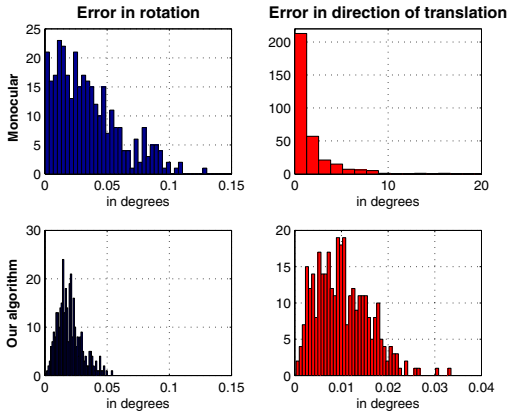

Figure 7. *Experiment results for a 2-camera stereo system: Top row: estimation errors in rotation and translation direction by using one camera only (i.e., monocular); Bottom row: estimation errors obtained by the proposed method.*

entire trajectory. Some sample captured images are shown in fig-9. After removing radial distortions of each individual image, we use the KLT tracker to find feature matches for every pair of images. These matches are then subjected to manual checking and modification to ensure that they are accurate and outlier-free. The measured ground-truth motion between every neighboring frames is about ± 30-degree rotation and around 1-cm translation.

After all these careful preparations we apply the new linear+alternation algorithm to matches, and obtain the following trajectories, shown in fig-10.

Remember that only two images are used to compute

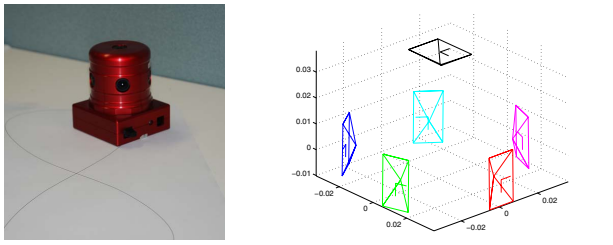

Figure 8. *Left: our experiment setup. A ladybug camera is moving along a plotted planar curve; Right: spatial configurations of its six camera units.*

Figure 9. *Some sample images used in the real date experiment.*

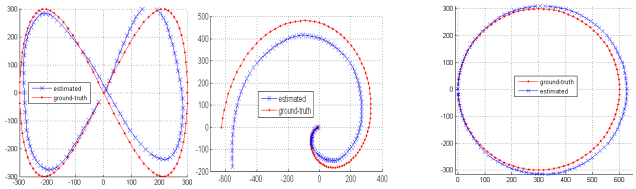

Figure 10. *Measured ground-truth trajectories vs. recovered trajectories. From left to right we test three different type of curves, an ∞-shaped curve, a spiral and a circle. Each sequence has 101 frames.*

the incremental motion between them. Even with errors accumulate over 100 frames, the recovered trajectories are remarkably good, not show obvious drift. This has validated the effectiveness of our algorithm. During the tests, we manually checked the feature matches to rule out mismatches. Otherwise, the recovered trajectory would be skewed, indicating that the algorithm is sensitive to outliers. To address the outlier issue, we are currently investigating a new algorithm based on Linear Programming ([8]).

7. Summary

Although the standard 17-point algorithm does not work for many of the common Generalized Camera configurations (such as axial camera arrays, non-overlapping multiple-camera rigs, non-central catadioptric cameras or non-overlapping binocular stereo), a new linear algorithm (based on as few as 14 or 16 points) is proposed which allows us to solve linearly for the generalized essential matrix and the 6-dof motion of the generalized camera.

This linear algorithm, combined with an alternation scheme, solves the generic motion estimation problem efficiently and accurately, as long as care is taken to find good matches and to avoid the trivial *null* solution. The accuracy of the proposed method is very good, in particular for a linear approach. Using this method for initializing a nonlinear algorithm (e.g., bundle adjustment algorithm based on geometric error metric) would no doubt give even better results.

Acknowledgement. This research was partly supported by NICTA as represented by the Department of Broadband, Communications and the Digital Economy of Australian Government. The authors wish to thank anonymous reviewers for their invaluable suggestions.

References

[1] M. Byröd, K. Josephson, and K. Åström. Improving numerical accuracy of grbner basis polynomial equation solver. In *ICCV*, 2007.

[2] B. Clipp, J. Kim, J. Frahm, M. Pollefeys, and R. Hartley. Robust 6dof motion estimation for non-overlapping, multi-camera systems. Technical Report TR07-006, UNC, ftp://ftp.cs.unc.edu/pub, 2007.

[3] D. Feldman, T. Pajdla, and D. Weinshall. On the epipolar geometry of the crossed-slits projection. In *ICCV*, page 988, Washington, DC, USA, 2003. IEEE Computer Society.

[4] J.-M. Frahm, K. Köser, and R. Koch. Pose estimation for Multi-Camera Systems. In *DAGM*, 2004.

[5] M. D. Grossberg and S. K. Nayar. A general imaging model and a method for finding its parameters. In *ICCV*, pages 108–115, 2001.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[7] M. Lhuillier. Effective and generic structure from motion using angular error. In *ICPR*, pages 67–70. IEEE Computer Society, 2006.

[8] H. Li. A practical algorithm for l-infinity triangulation with outliers. In *CVPR*, 2007.

[9] H. Li and R. Hartley. Five-point motion estimation made easy. In *ICPR '06*, pages 630–633, 2006.

[10] R. Molana and C. Geyer. Motion estimation with essential and generalized essential matrice. In *Imaging Beyond the Pinhole Camera (Daniilidis and Klette ed.)*. Springer-Verlag, 2007.

[11] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real time structure from motion. In *Proc. BMVC*, 2007.

[12] J. Neumann, C. Fermüller, and Y. Aloimonos. Polydioptric cameras: New eyes for structure from motion. In *DAGM*, pages 618–625, London, UK, 2002. Springer-Verlag.

[13] R. Pless. Using many cameras as one. In *CVPR03*, pages II: 587–593, 2003.

[14] G. Schweighofer and A. Pinz. Fast and globally convergent structure and motion estimation for general camera models. In *Proc. BMVC*, volume 1, pages 206–212, 2006.

[15] O. Shakernia, R. Vidal, and S. Sastry. Infinitesimal motion estimation from multiple central panoramic views. In *Proc. MOTION'02*. IEEE Computer Society, 2002.

[16] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *ICCV Workshop on Omnidirectional Vision*, Beijing China, Oct. 2005.

[17] P. Sturm. Multi-view geometry for general camera models. In *CVPR*, volume 1, pages 206–212, jun 2005.

[18] P. Sturm, S. Ramalingam, and S. Lodha. On calibration, structure from motion and multi-view geometry for generic camera models. In *Imaging Beyond the Pinhole Camera (Daniilidis and Klette ed.)*. Springer-Verlag New York, Inc., 2006.

[19] S. Tariq and F. Dellaert. A Multi-Camera 6-DOF Pose Tracker. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2004.