

Estimating Relative Camera Motion from the Antipodal-Epipolar Constraint

John Lim, Nick Barnes, and Hongdong Li

Abstract—This paper introduces a novel antipodal-epipolar constraint on relative camera motion. By using antipodal points, which are available in large Field-of-View cameras, the translational and rotational motions of a camera are geometrically decoupled, allowing them to be separately estimated as two problems in smaller dimensions. We present a new formulation based on discrete camera motions, which works over a larger range of motions compared to previous differential techniques using antipodal points. The use of our constraints is demonstrated with two robust and practical algorithms, one based on RANSAC and the other based on Hough-like voting. As an application of the motion decoupling property, we also present a new structure-from-motion algorithm that does not require explicitly estimating rotation (it uses only the translation found with our methods). Finally, experiments involving simulations and real image sequences will demonstrate that our algorithms perform accurately and robustly, with some advantages over the state-of-the-art.

Index Terms—Multiview geometry, antipodal points, epipolar constraint, structure and motion, Hough, robust estimation.

1 INTRODUCTION

THE problem of estimating relative camera motion (or relative pose) from two views has been extensively studied in the last few decades. The problem involves estimating the rigid body transformation—the translation and rotation—between two cameras using 2D image point correspondences viewed by the cameras (the distances to the 3D world points are unknown, and translation can be found up to an unknown scale only).

In this paper, we introduce new constraints on camera motion by exploiting the geometry of large Field-of-View (FOV) cameras. A useful concept for thinking about large FOV cameras is the *image sphere*, where world points are projected onto a spherical imaging surface (as opposed to the standard image plane) [1], [2], [3]. Note that using image spheres implicitly assumes calibrated cameras.

Central to the proposed constraints is the concept of *antipodal rays*, which are only available in central, large FOV cameras. Two rays are antipodal if they are collinear and point in opposite directions. For example, in Fig. 1, \mathbf{p} and $-\mathbf{p}$ are antipodal in the camera coordinate system centered on C . Antipodal rays will intersect the image sphere at a pair of diametrically opposite points (analogous to the Earth's North and South poles). In practice, antipodal rays may be found in large FOV camera systems such as those shown in Fig. 2.

As we shall see, the use of antipodal rays leads to the novel *antipodal-epipolar* constraint that *decouples* translation and rotation and allows them to be estimated separately, independent of each other. In the conventional epipolar constraint [4], translation and

rotation are entangled together in the essential or fundamental matrix. If differential camera motions are assumed, translation and rotation can be decoupled by an approximate linearization of the motion equations [5], [1]. However, in the more general discrete motion case considered here, this “trick” does not work and decoupling translation and rotation is nontrivial. In fact, this has not been achieved by any other method that we know of.

Consequently, through the antipodal-epipolar formulation, we can express constraints that are *linear in translation* and, independently, constraints that are *linear in rotation*. In contrast, the conventional epipolar constraint is *bilinear* in translation and rotation. In other words, under the special condition of antipodal points, a problem that is nonlinear in general becomes linear (and this is not a mere approximation, as in the case of differential motions).

As a result of the decoupling of motion components, we find ourselves solving *two smaller dimension problems* instead of a single higher dimensional one. This simplifies the problem, leading to important consequences for designing algorithms that are robust to outliers and noise. Basically, it is easier to separate the inlier data from the outliers in a lower dimensional space than in a higher dimensional one. This naturally results in more efficient and robust algorithms, including ones which may be implemented to run in *constant time*, regardless of outlier proportions in the data. (Some existing methods (e.g., [6], [7]) attempt to solve translation and rotation separately; however, the two estimates are not truly decoupled since the methods were based on the conventional epipolar constraint.)

Furthermore, since translation can be estimated independent of rotation, we propose a method by which one may reconstruct scene structure purely from translation estimates and *without* recovering relative camera rotation explicitly (the rotation is implicitly constrained). This translation-only structure-from-motion (SfM) algorithm was previously not a sensible option since past methods recovered rotation and translation simultaneously (in the form of the essential or fundamental matrices). The new SfM algorithm is not only of theoretical interest but also has practical advantages in terms of computation speed, compared to existing techniques.

This paper is organized as follows: Below, we first review existing work. We then derive our geometrical constraints in Section 2. Robust algorithms are presented in Section 3 and scene reconstruction is discussed in Section 4. Section 5 demonstrates the performance of our methods with experiments and we end with some discussion in Section 6.

1.1 Previous Works

A vast body of literature spanning many decades exists in this area of camera motion recovery. Methods can generally be classified into those assuming differential camera motion and those assuming discrete camera motion. The former break down as the size of the motion becomes too large, while the latter may become less accurate if the motion is too small.

Here, our primary concern is with discrete motion methods, which include the work of [8], [9], [10], [11], [12], [13], [14] and many others found in reviews such as [15], [16]. However, bear in mind that many methods assuming differential motion also exist, including [17], [18], [5], [19], [20], [21], [1], [22], [23], and many more—more than can be listed here.

For discrete motion methods, the classical solution relies on the well-known epipolar constraint, which gives linear constraints on camera motion and calibration [4]. Briefly, from Fig. 1, an epipolar plane is given by the two camera centers C , C' , and the world point P . The epipolar constraint then requires the image points \mathbf{p} and \mathbf{p}' to lie on this plane, that is, they satisfy $\mathbf{p}'^T \mathbf{F} \mathbf{p} = 0$, where \mathbf{F} is the fundamental matrix encapsulating the motion and calibration parameters.

• J. Lim is with NICTA and the Department of Engineering, Australian National University, Canberra, ACT 2601, Australia.
E-mail: john.lim@rsise.anu.edu.au.

• N. Barnes and H. Li are with NICTA and the Department of Engineering, Australian National University, Canberra, ACT 2601, Australia, and Bionic Vision Australia.
E-mail: nick.barnes@nicta.com.au, hongdong.li@anu.edu.au.

Manuscript received 13 Aug. 2009; revised 1 Mar. 2010; accepted 20 Apr. 2010; published online 26 May 2010.

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-2009-08-0539.

Digital Object Identifier no. 10.1109/TPAMI.2010.113.

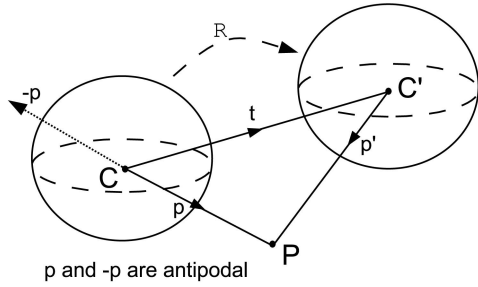


Fig. 1. The world point P is imaged by two cameras centered on C and C' , which are related by a translation t and rotation matrix R . p and p' give a point correspondence. The ray $-p$ is antipodal to p . If there happens to be another world point lying on the ray $-p$ and that point is also visible to camera C' , then we can obtain antipodal-epipolar constraints on t and on R . Also shown are the image spheres centered on cameras C and C' .

This leads to the eight-point algorithm [8], [9] where a minimum of eight points are needed to linearly obtain a unique solution. Alternatively, with an additional singularity constraint, the nonlinear seven-point algorithm may also be used [4].

Meanwhile, for calibrated cameras, the essential matrix, E , satisfying $p^T E p' = 0$ is estimated. The state-of-the-art are the five-point algorithms of Ništer [10], of Kukulova et al. [11], and of Li and Hartley [12], all of which are nonlinear methods requiring five-point correspondences to find the two unknown parameters of translation and three of rotation.

After relative motion between the cameras has been found, it is also possible to recover the structure of the scene using methods such as [24], [25], [26], [27], [28]. These SfM algorithms find the relative distances from the cameras to the world points viewed by those cameras.

The “x-point” motion recovery algorithms described above are typically used as *minimal point solvers* within a robust statistical framework in order to handle point correspondence data that has been contaminated with outliers. These outliers typically come from wrongly matched points, independently moving objects, and such. The RANdom SAMple and Consensus (RANSAC) [29] framework and variants like it (e.g., [30], [31], [32], [33], [34]) are some of the robust algorithms used. These methods perform Monte Carlo sampling to generate hypothesis solutions, which are then scored in order to pick the best solution from among the hypotheses. Basically, they rely on sampling the data enough times such that at least one set of outlier-free observations is picked with high probability.

Antipodal points have previously been proposed for camera motion estimation [35], [36], [37], [38], [39]. However, all such research worked purely under the *differential* camera motion assumption; hence, their usefulness is limited to scenarios where camera motions are small. For example, [35], [38], [39] constrained translation using antipodal optical flow, while [37] worked on the related idea of “matching rays” in cameras with nonoverlapping FOV.

Here, we present a new formulation that considers *discrete* camera motions, leading to constraints and algorithms that work over a wider range of motion sizes. Since we are dealing with the case of calibrated cameras, we benchmark the performance of our motion recovery approach against the state-of-the-art as represented by the five-point algorithm within a robustifying RANSAC framework.

1.2 Notation

We use uppercase, bold letters for 4×1 homogeneous coordinate vectors denoting camera centers and world points (e.g., C , P). Lowercase, bold letters are used for 3×1 homogeneous coordinate vectors denoting image points or rays (e.g., p , q). However, these image points are on the image sphere, rather than the image plane,

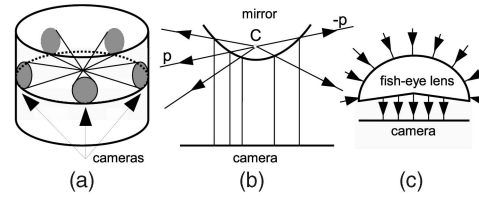


Fig. 2. Antipodal rays are found in any central camera with an FOV larger than 180 degrees. Some examples: (a)–(c) a rig with multiple cameras pointing in different directions, a catadioptric camera, and a fish-eye camera.

so for some ray $p = [p_x \ p_y \ p_z]^T$, p_z is not necessarily unit, but the ray is subject to $p_x^2 + p_y^2 + p_z^2 = 1$. Uppercase nonbold letters (e.g., R , A) refer to matrices.

2 THEORY

Consider the camera center C and world points, P and Q (refer to Fig. 3). The projection of these world points onto C gives the rays p and q . Suppose the world points, and hence the rays, are antipodal relative to C , that is, $p = -q$, where the rays are expressed in the image coordinate frame of C . This is similar to the setup first described in Fig. 1.

The second camera, C' , is related to C by a rigid body transformation—the translation t and rotation R . Projecting the world points onto camera C' gives the rays p' and q' (i.e., the rays are expressed in the coordinate frame of C'). Therefore, p and p' are a pair of point correspondences and q and q' are another pair.

The vector, t , is the translation directed from C to C' , expressed in the coordinate frame of C . Conversely, t' is translation directed from C' to C , expressed in the coordinate frame of C' . Note that there are two unknowns in t (since we cannot recover its scale [4]) and three unknowns in R ; so altogether there are five unknowns to be solved for.

2.1 Constraint on Translation

Everything hinges upon this key fact: All of the points and rays in Fig. 3 lie on a single plane if p and q are antipodal. Vectors p , p' , and t lie on a plane—the all too familiar epipolar plane. Likewise, q , q' , and t also lie on a plane, and if p and q are antipodal, then the two planes are one and the same.

We can express the equation of the epipolar plane purely using the rays p' and q' . The normal vector to the plane is $p' \times q'$, where \times denotes the cross product. Then, we have:

$$t'^T (p' \times q') = 0. \quad (1)$$

This is a linear constraint on the translation, t' , independent of rotation, R . Geometrically, (1) simply states that t' lies on the

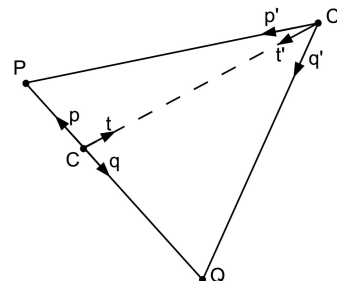


Fig. 3. Cameras C and C' and world points P and Q . Rays p and q are antipodal in the camera C view. Point correspondence pairs arise from p and p' and from q and q' . Vector t is the translation directed from C to C' in the coordinate frame of C and t' is directed from C' to C in the frame of C' . All points and rays shown lie on the same epipolar plane.

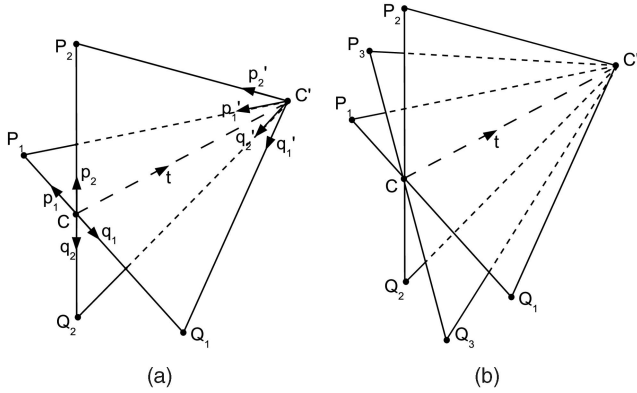


Fig. 4. Pairs of antipodal world points are given by P_i and Q_i , for $i = 1, 2, 3$. (a) Two antipodal pairs give two planes that intersect to give t . (b) Three pairs give three planes constraining the three unknowns of R .

epipolar plane. Let us call this the *antipodal-epipolar constraint on translation*. (We use the “antipodal” qualifier to differentiate this from the usual epipolar constraint, $\mathbf{p}^T E \mathbf{p} = 0$, where E is the essential matrix.)

Note that if the points are antipodal in camera C , then the constraint is on t' ; conversely, if the points are antipodal in camera C' , then the constraint is on t .

It is also important to note that the sign of translation is *unambiguously* recoverable (whereas, in normal essential matrix estimation, it is ambiguous whether the camera translation is t or $-t$ [4]). From Fig. 3, it is clear that t' must lie between \mathbf{p}' and \mathbf{q}' , that is, $(t')^T \mathbf{p}' > 0$ and $(t')^T \mathbf{q}' > 0$. Hence, it cannot lie in the opposite direction.

Ambiguity arises only when both world points \mathbf{P} and \mathbf{Q} are infinitely far away; then \mathbf{p}' and \mathbf{q}' will also be antipodal and it is not possible to determine the sign of t' . Obviously, infinitely distant points can tell us nothing about translation, and this degenerate case is easily detectable as \mathbf{p}, \mathbf{q} will be antipodal, and \mathbf{p}', \mathbf{q}' will also be antipodal.

A minimum of two pairs of antipodal points (which is four points) recovers translation up to a positive scale (since the sign of t is known). One pair of antipodal points constrains translation to lie on a plane. Another pair (that does not lie on the first plane) gives another such plane. Intersecting the two distinct planes gives the direction of translation (Fig. 4a).

2.2 Constraint on Rotation

The relative rotation between the two camera coordinate frames may similarly be constrained by the epipolar planes. The ray \mathbf{p} must lie on the plane given by $\mathbf{p}' \times \mathbf{q}'$, so we have:

$$(R\mathbf{p})^T (\mathbf{p}' \times \mathbf{q}') = 0. \quad (2)$$

This gives a linear constraint on rotation, R , independent of translation. Let us call this the *antipodal-epipolar constraint on rotation*. (Note that $(R\mathbf{q})^T (\mathbf{p}' \times \mathbf{q}') = 0$ gives no extra information since $\mathbf{p} = -\mathbf{q}$.)

The 3×3 rotation matrix effectively has three unknowns since $R^T R = I$ gives six quadratic constraints. Hence, a minimum of three antipodal pairs (three planes) is sufficient to recover rotation. This is shown in Fig. 4b. However, this is a nonlinear solution and nine or more pairs are required for a linear solution.

Just as with the conventional epipolar constraint, the case of purely rotational motion is degenerate and may be easily detected as all of the antipodal points in the first camera will still be antipodal in the second camera.

2.3 Linear and Decoupled Constraints

The conventional epipolar equation is $\mathbf{p}^T E \mathbf{p} = 0$, where the essential matrix is the cross product of t and R , i.e., $E = [t]_{\times} R$. This equation is *bilinear* in both t and R . Contrast that with (1), which is linear in t (and independent of R), and with (2), which is linear in R (and independent of t).

The decoupling of translation and rotation has some advantages. For example, errors in the t estimate will not propagate and contribute to errors in R and vice versa. Other advantages include breaking up the problem into two lower dimensional ones which are easier to solve. In the next section, we will see how this naturally leads to improved algorithms.

3 ROBUST ALGORITHMS FROM ANTIPODAL CONSTRAINTS

We discuss two possible robust algorithms—a RANSAC-type method and a Hough-like voting approach. Using one of the two, we first robustly recover translation. This stage also segments the data into inliers and outliers. The inliers are then used in the next stage to recover rotation linearly.¹

These algorithms aim to demonstrate the viability of the geometrical constraints introduced, and as such, play an illustrative role rather than a definitive one. Many other algorithms utilizing the antipodal constraints are possible, and the reader may replace them with their favorite robust estimation method.

3.1 Robustly Finding Translation

3.1.1 Antipodal+RANSAC

A minimal solver for RANSAC requires *two antipodal pairs* (or four points), where each pair gives rise to a plane constraining translation as discussed above. The solver only needs to find the intersection of the two planes, and its implementation is trivial.

The inlier-outlier segmentation of data points also occurs simultaneously. Any antipodal pair satisfying (1) to within some threshold is an inlier.

Our antipodal point and RANSAC algorithm perform faster than the usual five-point and RANSAC algorithm since fewer points are used for hypothesis generation (four versus five points). Further time savings are achieved due to the fact that the solver solves a simple, linear problem (in contrast, the nonlinear five-point algorithm needs to solve a tenth order polynomial).

3.1.2 Antipodal+Voting

Since translation estimation is a fairly low-dimensional problem (2D), we suggest that here, Hough-like voting may be an efficient algorithm for robust estimation. Voting has previously been used for motion estimation [40], [41], [42]. However, these generally vote in some higher dimensional space (at least 5D for estimating essential matrices). Hence, the large number of bins needed to discretize the solution space made previous methods susceptible to the problem of sparsity of points. Here, the decoupling effect of antipodal points enables us to work in a 2D solution space, for which voting is both effective and efficient.

Translation, t , is found from the intersection of two or more constraint planes, so voting to each plane gives a peak in the vote space which corresponds to a robust estimate of t . Since voting is computationally cheap, we use all of the constraint planes arising from every antipodal pair (no random sampling).

1. This actually introduces some coupling between t and R . However, the effect is not too significant and we made this design choice for efficiency purposes. In general, algorithms can be devised such that the two estimates are completely independent since the constraints themselves are decoupled.

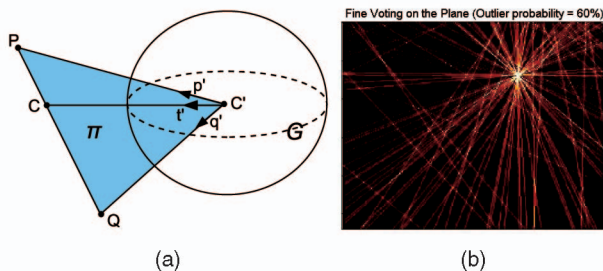


Fig. 5. (a) Epipolar plane Π with normal vector $\mathbf{p}' \times \mathbf{q}'$ intersects the surface of the image sphere at the great circle G . (b) Result of fine voting. The peak is obvious even with 60 percent outliers (this is a window of the vote space; many more outliers lie outside the window).

As in our previous work [35], [36], we represent the solution space of possible directions of translation as the surface of a unit image sphere centered on the camera center. The intersection of a plane passing through the sphere center with the sphere surface is a great circle. The problem of finding the intersection of planes then becomes one of finding the intersection of great circles on the sphere (see Fig. 5a).

For greater efficiency, we may further turn this problem into one of finding the intersection of multiple straight lines since the projection of a great circle onto a tangent plane of the sphere is a straight line. We then vote along these straight lines at progressively finer scales to refine the solution (Fig. 5b). We refer the reader to [35] for further implementation details as the algorithm is similar to a method we previously used in differential camera motion estimation.

Once again, a consequence of translation estimation is the segmentation of inliers from outliers in the data. A last linear refinement step is performed using these inliers in order to obtain a least-squares estimate of translation. This helps to mitigate the effects of voting bin quantization on accuracy.

3.2 Finding Rotation Linearly

Using only the inliers found when estimating translation with either the RANSAC-based or voting-based methods above, we now go on to perform rotation estimation.

From (2), one pair of inlier points would give a linear constraint on rotation. Hence, M inliers will give M linear constraints, giving a system of overconstrained equations which may be written in the form $A[\mathbf{R}_{vec}] = 0$, where $[\mathbf{R}_{vec}]$ is the vectorized form of the rotation matrix, \mathbf{R} . Matrix A has size $M \times 9$, while $[\mathbf{R}_{vec}]$ is a 9×1 vector.

A linear solution for \mathbf{R} requires $M \geq 9$. Equation $A[\mathbf{R}_{vec}] = 0$ can then be solved by the DLT algorithm [4], which involves performing an SVD, $A = UDV^T$. The least-squares solution is given by the column of V corresponding to the smallest singular value in D . A final nonlinear correction step ensures \mathbf{R} is a “proper” rotation by enforcing $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ and $\det(\mathbf{R}) = 1$.

In practice, there is a small probability of outliers slipping past the outlier rejection stage during translation estimation. This happens when the outlier is a “leverage point” that satisfies the translation constraint of (1) but does not satisfy the constraints of full camera motion. A practical system may need another RANSAC stage (or other robust algorithm) here, which should terminate quite quickly since the vast majority of outliers (usually all) has been removed.

Another practical issue is that certain singular values in matrix D (from the SVD) may sometimes be very small, such that under noise, it can be hard to identify the smallest singular value. To remedy this, we test all solutions corresponding to very small singular values by reconstructing several points, and picking the solution where all reconstructed points are in front of both cameras (similarly to the test applied to disambiguate twisted pairs in essential matrix motion recovery [4]).

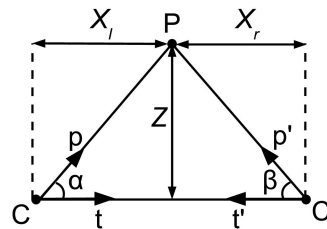


Fig. 6. Structure from \mathbf{t} and \mathbf{t}' (partial motion).

4 APPLICATION: STRUCTURE FROM PARTIAL MOTION (WITHOUT FINDING ROTATION)

We observe that relative scene structure can actually be found *without* computing rotation. In particular, we will show that recovering translation twice, that is, finding \mathbf{t} and \mathbf{t}' , is sufficient for structure computation.

Computing \mathbf{t} and \mathbf{t}' amounts to recovering rotation up to one degree of freedom, i.e., the cameras are free to rotate relative to each other about the straight line, CC' , joining the camera centers. However, given a point correspondence, we can assume that the rays \mathbf{p} and \mathbf{p}' intersect at world point P . This implicitly constrains the last remaining degree of freedom of rotation (without explicitly computing it).

Fig. 6 shows the geometry of the situation. We estimate \mathbf{t} from points which are antipodal in camera C , while \mathbf{t}' is found from points which are antipodal in camera C' . From basic trigonometry, $\tan(\alpha) = Z_l/X_l$ and $\tan(\beta) = Z_r/X_r$. Also, $X_r = |\mathbf{T}| - X_l$, where $|\mathbf{T}|$ is the baseline, the magnitude of which we set to unit. With $|Z_r| = |Z_l| = Z$, we have:

$$Z = \frac{\tan(\alpha)\tan(\beta)}{\tan(\beta) + \tan(\alpha)}, \quad (3)$$

where $\alpha = \text{acos}(\mathbf{t}^T \mathbf{p})$ and $\beta = \text{acos}(\mathbf{t}'^T \mathbf{p}')$ with all vectors being unit length. Then, the reconstructed world point, P , is $Z/\sin(\alpha)$ units away from camera C in the direction $\hat{\mathbf{p}}$.

This is reminiscent of (but different from) plane+parallax reconstruction methods [43], [44], where rotation and calibration need not be known since the plane+parallax constraints cancel them. Here, the camera is calibrated, and we use the angles between the rays and the epipoles for triangulation (where the angle between any two rays is rotationally invariant).

The practical implication is that this approach is better suited for parallelization (leading to speedier implementations on parallel hardware) since translation recovery (by voting) is a highly parallel process. The rotation stage, which is not as parallelizable, may be skipped altogether. Also, \mathbf{t} and \mathbf{t}' can be found simultaneously since each is not required to estimate the other. The method’s drawback is that, under noise, rays \mathbf{p} and \mathbf{p}' may not intersect exactly (likewise for rays \mathbf{t} and \mathbf{t}').

5 EXPERIMENTAL RESULTS

We now demonstrate that the antipodal-epipolar constraint and the algorithms based on it work robustly and accurately in practice. All algorithms were implemented in Matlab. Errors in the estimated motions are measured in units of degrees (e.g., angle between true and estimated translation directions). Motion recovery results are shown in Fig. 7 (simulations) and Fig. 9 (real image experiments). Figs. 10a, 10b, 10c, 10d, 10e, 10f, 10g, and 10h show reconstruction experiments. All plots show results *before* applying local, nonlinear refinement (e.g., bundle adjustment [45]), which can be used if greater accuracy is desired.

Results were benchmarked against five-point+RANSAC, which consists of the five-point algorithm of [12] within the RANSAC

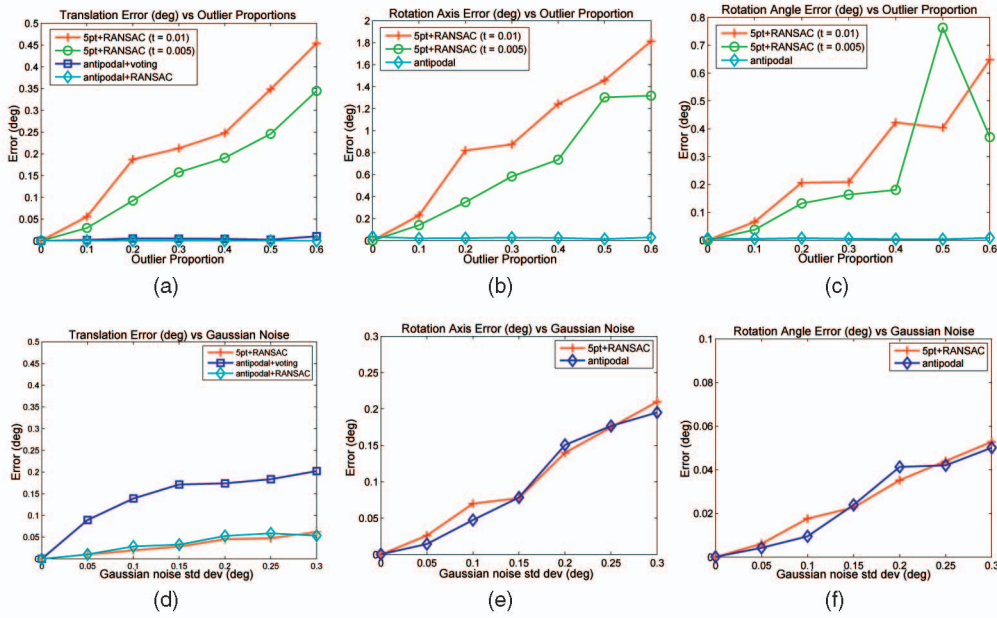


Fig. 7. (a)-(c) Errors under increasing outlier proportion. Our methods give only tiny increases in error compared to the benchmark approach. Here, t is the Sampson distance threshold used for five-point+RANSAC. (d)-(f) Performance degrades gracefully under increasing Gaussian noise.

code of [46]. It used adaptive sampling with probability $p = 0.99$ and Sampson distance thresholds of 0.01 and 0.005 (see [4] for details of RANSAC).

The antipodal+RANSAC method also used the RANSAC code of [46]. Adaptive sampling probability was $p = 0.99$ and the threshold used was 0.5 degrees (angle between translation and the antipodal-epipolar plane). Antipodal+voting performed coarse-to-fine voting in two stages, with the estimate taken as the center of mass of voting bin(s) with the highest votes.

5.1 Simulations

5.1.1 Method

A series of simulations tested performance under increasing outlier proportions and under increasing Gaussian noise. Two views of a 3D scene were generated with random translations and rotations between cameras. The translation magnitude varied randomly between 5 and 10 units; rotation angle varied randomly between 10 and 50 degrees, while distances to the randomly generated world points varied between 5 and 10 units. Results were averaged over 100 trials.

We tested on data contaminated with up to 60 percent outliers and on data under zero-mean Gaussian noise with standard deviations of up to 0.3 degrees (in a camera with viewing angle 60 degrees and a 640×480 pixel image, 0.3 degrees corresponds to around 2 to 3 pixels). This corresponds to realistic noise levels in feature matching methods (e.g., SIFT), which are subpixel accurate and typically exhibit errors of at most a couple of pixels for correct matches. Outliers consisted of randomly mismatched rays, while Gaussian noise was simulated by perturbing the directions of image rays according to a Gaussian distribution.

5.1.2 Robustness to Outliers

Both antipodal point algorithms were remarkably resistant to increasing outlier proportions. Figs. 7a, 7b, and 7c demonstrate that increasing the outlier proportions up to 60 percent caused no appreciable increase in the error of motion estimates from the antipodal+voting and antipodal+RANSAC methods. Over the same increase in outliers, the errors for five-point+RANSAC were still small but they increased more quickly than our methods.

The improved outlier robustness of our methods compared to five-point+RANSAC is not an unusual result since our methods perform robust estimation in a 2D space instead of a higher, 5D one. It is simply easier to cluster inliers from outliers in a lower dimensional space.

Also, since each constraint uses fewer points compared to five-point, the probability of obtaining a good constraint is significantly increased—which is beneficial for antipodal+RANSAC.

5.1.3 Runtime

It is important to note that the above results for the antipodal+voting method happened in *constant time*, regardless of outlier proportions (see Fig. 8). This is because it is viable to use all available constraints for the voting method (hence its runtime is linear in the number of constraints or number of antipodal pairs available).

Conversely, as outlier probability increases, five-point+RANSAC and antipodal+RANSAC will take longer to arrive at a good solution since RANSAC will test more hypothesis solutions in order to maintain a high chance of hitting on a good one (some RANSAC variants [33], [31], [32] bound or reduce the time taken to sample or score hypotheses, but this trades off accuracy and robustness for speed).

However, runtime for antipodal+RANSAC will increase at a slower rate compared to five-point+RANSAC since it uses fewer points and solves a much simpler intersection of two planes

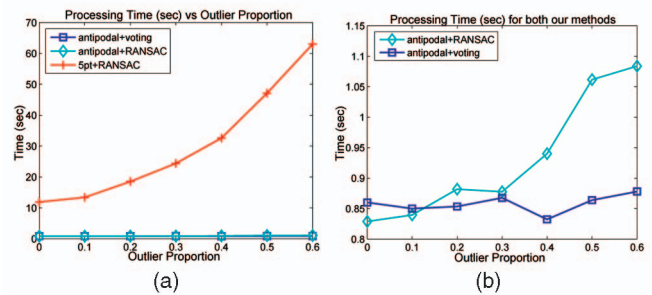


Fig. 8. (a) Runtime as outlier proportions increase. (b) Using a smaller y-axis scale (zooming in) to compare the two antipodal methods.

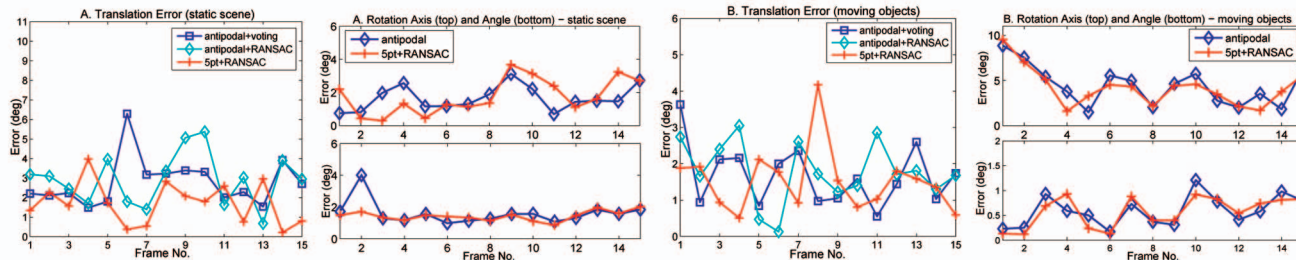


Fig. 9. Experiments on real images. (a)-(b) Image Sequence A—static scene, translation 4 cm, rotation 10 degrees per frame. (c)-(d) Image Sequence B—dynamic scene containing moving objects, translation 4 cm, rotation 4 degrees per frame.

problem. The numbers in the graphs are obviously implementation-dependent, but the *trend* will remain the same.

Another parameter influencing RANSAC processing time is the distance threshold used for distinguishing outliers from inliers. A stricter threshold will further reduce the errors in the estimate, e.g., in Figs. 7a, 7b, and 7c, reducing the distance threshold of five-point+RANSAC from 0.01 to 0.005 reduces the errors in general. However, RANSAC will then have to iterate many more times and take even longer to obtain the solution.

5.1.4 Gaussian Noise

Figs. 7d, 7e, and 7f show that all methods tested gave small errors, with performance degrading gracefully under increasing Gaussian noise. Fig. 7d shows that errors for the translation estimates of antipodal+RANSAC were similar to that of five-point+RANSAC. Translation errors from antipodal+voting were slightly higher (less than 0.2 degrees difference) due to quantization error from dividing the solution space into a finite set of voting bins; its performance may be improved by voting with finer resolutions (we used only two coarse-to-fine stages). The errors for rotation axis and angle in Figs. 7e and 7f were also comparable to five-point+RANSAC.

5.2 Real Image Sequences

5.2.1 Method

Real image sequences were obtained from a Ladybug omnidirectional camera [47]. This is a camera rig consisting of five cameras arranged in a ring configuration and one camera (which we did not use) pointing upward. Using the calibration supplied by the manufacturers, image points from the five cameras are mapped into a single coordinate frame.

In Image Sequence A, the camera was placed on the ground, where it translated by about 4 cm and rotated by 10 degrees between frames. The scene was static. The motion was estimated and compared with ground truth values (manually measured with ruler and protractor). Figs. 10b, 10c, 10d, 10e, and 10f show the images captured by the Ladybug omnidirectional camera rig at one time instance in sequence A. In another experiment, Image Sequence B, the camera translated by 4 cm and rotated by 4 degrees between frames; furthermore, the scene contained independently moving objects, leading to greater outlier probabilities. The fact that motion was planar, was *not* used by any of the algorithms to simplify estimation.

SIFT features (code from [48]) were used to find point correspondences. Points that were within 0.5 degrees of being antipodal were approximated as antipodal. In the real images, several hundred antipodal points could typically be found, which was many more than the minimum needed.

5.2.2 Motion Estimates

Figs. 9a and 9b show errors for Image Sequence A (static scene) while Figs. 9c and 9d show errors for Sequence B (contains moving

objects). In these experiments, all three methods performed comparably. The differences in their average errors were small—less than 1 degree—which is within the bounds of the accuracy of our ground truth measurements.

The accuracies of all methods were affected by practical issues such as the fact that the Ladybug camera rig is only approximately central—the centers of its constituent cameras don't exactly coincide. Also, our methods approximated nearly antipodal points to be antipodal. In spite of this additional source of error, our methods performed comparably to five-point.

The rotation in Sequence A (10 degrees) was larger than in Sequence B (4 degrees), resulting in better rotation estimates and worse translation estimates for A (and the reverse for results of B). Videos showing the experiments and recovered estimates are included as supplemental material, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.113>.

5.3 Structure Estimates

The structure from partial motion method described in Section 4 is compared with a standard linear triangulation method (see [4], [27] for details) in simulations under noise and in reconstructions of real scenes.

Under increasing Gaussian noise, the simulation results of Fig. 10a show that our partial motion method and linear triangulation performed comparably, with both giving small average reconstruction errors. Both methods used true camera motions and noisy correspondence data as inputs. Errors were measured as the euclidean distance between a reconstructed point and the (scaled) true world point.

We also performed reconstruction on Image Sequence A, and Figs. 10g and 10h show a bird's eye view of the reconstructed environment. Using measured ground truth motion as inputs, both methods gave very similar reconstructions.

6 DISCUSSION

6.1 Real-Time Implementation

The antipodal+voting algorithm is fundamentally parallel in nature and implementation on a GPU or other parallel hardware would see significant speedups. Furthermore, its constant-time performance under increasing outlier probability means speed is unaffected in complex scenes with large, variable outlier proportions.

6.2 Effect of Motion Size

Figs. 10i and 10j show the effect of different magnitudes of camera motion. In Fig. 10i, as the magnitude of translation increases, the translation is estimated with increasing accuracy. Conversely, in Fig. 10j, as the rotation angle becomes very small, rotation is increasingly difficult to recover. These effects are in accordance with the behavior of virtually all other motion estimation methods.

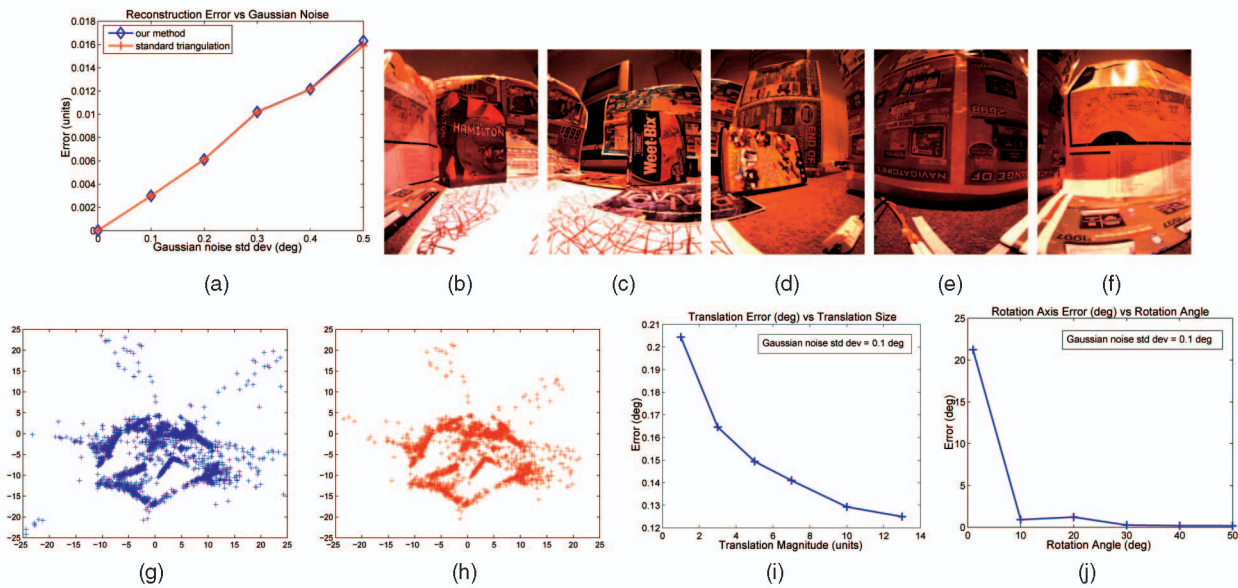


Fig. 10. (a) Gaussian noise simulation comparing our partial motion structure method with standard linear triangulation. (b)-(f) Example images from Image Sequence A taken by the five cameras on the Ladybug omnidirectional camera rig. (g)-(h) Top-down view of structure estimates for Image Sequence A using our method and using linear triangulation, respectively, with ground truth motion as the inputs. (i)-(j) Effect of different motion sizes on motion recovery.

6.3 Finding Antipodal Points Efficiently

In our experiments, we searched through all available correspondences for antipodes. For greater efficiency, we recommend avoiding this search by using DAISY features [49] instead. While SIFT features are computed only at certain positions in the image (at the local extrema in a stack of Difference-of-Gaussian images), DAISY features are designed to be computed anywhere in the image. Antipodal points can then be chosen in advance and the DAISY descriptor is computed at those points.

6.4 Deterministic Solutions

It is interesting to note that the solution given by RANSAC is random—i.e., 100 different runs can give 100 different solutions scattered around the “true” solution. This happens since sampling is random, so slightly different sets of inliers will be found each time [50]. On the other hand, the voting approach suggested here gives deterministic solutions since random sampling is not used.

7 CONCLUSION

We have presented novel constraints on relative camera motion by exploiting the geometry of antipodal points which are found in large FOV cameras. These constraints are linear and decoupled in the translational and rotational components of motion, thus simplifying the problem and leading to improved motion and structure algorithms. Future work will explore further applications of these constraints—e.g., rotation recovery is not strictly required for odometry, so skipping it will lead to speedier estimation.

Supplemental videos, which can be found on the Computer Society Digital Library at <http://doi.ieeeecomputersociety.org/10.1109/TPAMI.2010.113>, provide translation estimates and ground truth, marked in the video “Sequence A (static).mpg” corresponding to Fig. 9a. “Sequence B (moving objects).mpg” is the corresponding video for Fig. 9c. These show the view from the forward facing camera. Other camera views are included for the reader’s reference.

ACKNOWLEDGMENTS

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications, and the Digital

Economy, and the Australian Research Council (ARC) through the ICT Centre of Excellence Program. This research was also supported in part by ARC through its Special Research Initiative (SRI) in Bionic Vision Science and Technology grant to Bionic Vision Australia (BVA).

REFERENCES

- [1] K. Prazdny, “Egomotion and Relative Depth Map from Optical Flow” *Biological Cybernetics*, vol. 36, pp. 87-102, 1980.
- [2] J. Gluckman and S.K. Nayar, “Ego-Motion and Omnidirectional Cameras,” *Proc. IEEE Int’l Conf. Computer Vision*, 1998.
- [3] G.L. Mariottini and D. Prattichizzo, “Image-Based Visual Servoing with Central Catadioptric Camera” *Int’l J. Robotics Research*, vol. 27, pp. 41-57, 2008.
- [4] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2004.
- [5] H. Stewénius, C. Engels, and D. Nistér, “An Efficient Minimal Solution for Infinitesimal Camera Motion,” *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2007.
- [6] J.C. Bazina, C. Demonceaux, P. Vasseur, and I.S. Kweon, “Motion Estimation by Decoupling Rotation and Translation in Catadioptric Vision,” *Computer Vision and Image Understanding*, vol. 114, no. 2, pp. 254-273, 2010.
- [7] M. Antone and S. Teller, “Automatic Recovery of Relative Camera Rotations for Urban Scenes,” *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2000.
- [8] H. Longuet-Higgins, “A Computer Algorithm for Reconstruction of a Scene from Two Projections” *Nature*, vol. 293, pp. 133-135, 1981.
- [9] R. Hartley, “In Defence of the 8-Point Algorithm,” *Proc. Fifth Int’l Conf. Computer Vision*, pp. 1064-1075, 1995.
- [10] D. Nistér, “An Efficient Solution to the Five-Point Relative Pose Problem,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756-770, June 2004.
- [11] Z. Kukelova, M. Bujnak, and T. Pajdla, “Polynomial Eigenvalue Solutions to the 5-pt and 6-pt Relative Pose Problems,” *Proc. British Machine Vision Conf.*, 2008.
- [12] H. Li and R. Hartley, “Five-Point Motion Estimation Made Easy,” *Proc. Int’l Conf. Pattern Recognition*, pp. 630-633, <http://users.cecs.anu.edu.au/%7EHongdong>, 2006.
- [13] 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*, 2007.
- [14] R. Hartley and F. Kahl, “Global Optimization through Rotation Space Search” *Int’l J. Computer Vision*, vol. 82, no. 1, pp. 64-79, 2009.
- [15] Q.-T. Luong, R. Deriche, O.D. Faugeras, and T. Papadopoulos, “On Determining the Fundamental Matrix: Analysis of Different Methods and Experimental Results,” Technical Report RR-1894, INRIA, 1993.

- [16] Z. Zhang, "Determining the Epipolar Geometry and Its Uncertainty: A Review," Technical Report RR-2927, INRIA, 1996.
- [17] A. Jepson and D. Heeger, "Subspace Methods for Recovering Rigid Motion I: Algorithm and Implementation" *Int'l J. Computer Vision*, vol. 7, no. 2, pp. 95-117, 1992.
- [18] K. Kanatani, Y. Shimizu, N. Ohta, M.J. Brooks, W. Chojnacki, and A. van den Hengel, "Fundamental Matrix from Optical Flow: Optimal Computation and Reliability Evaluation" *J. Electronic Imaging*, vol. 9, no. 2, pp. 194-202, 2000.
- [19] C. Tomasi and J. Shi, "Direction of Heading from Image Deformations," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 422-427, 1993.
- [20] C. Fermüller and Y. Aloimonos, "Qualitative Egomotion" *Int'l J. Computer Vision*, vol. 15, pp. 7-29, 1995.
- [21] A.R. Bruss and B.K. Horn, "Passive Navigation" *Computer Vision, Graphics, and Image Processing*, vol. 21, pp. 3-20, 1983.
- [22] J.H. Rieger and D.T. Lawton, "Processing Differential Image Motion" *J. Optical Soc. Am. A*, vol. 2, no. 2, pp. 354-359, 1985.
- [23] K. Prazdny, "On the Information in Optical Flows" *Computer Graphics and Image Processing*, vol. 22, pp. 239-259, 1983.
- [24] J. Oliensis, "A Critique of Structure-from-Motion Algorithms" *Computer Vision and Image Understanding*, vol. 80, no. 2, pp. 172-214, 2000.
- [25] J. Oliensis and G. Yakup, "New Algorithms for Two-Frame Structure from Motion," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 737-744, 1999.
- [26] J. Oliensis, "Exact Two-Image Structure from Motion" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1618-1633, Dec. 2002.
- [27] R. Hartley and P. Sturm, "Triangulation" *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146-157, 1997.
- [28] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier, 1996.
- [29] 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, June 1981.
- [30] P.H.S. Torr and A. Zisserman, "MLE-SAC: A New Robust Estimator with Application to Estimating Image Geometry" *J. Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138-156, 2000.
- [31] J. Matas and O. Chum, "Randomized RANSAC with $T_{d,d}$ Test" *Image and Vision Computing*, vol. 22, no. 10, pp. 837-842, Sept. 2004.
- [32] O. Chum and J. Matas, "Optimal Randomized RANSAC," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 8 pp. 1472-1482, Aug. 2008.
- [33] D. Nistér, "Preemptive RANSAC for Live Structure and Motion Estimation" *Machine Vision Applications*, vol. 16, no. 5, pp. 321-329, 2005.
- [34] P. Rousseeuw, "Least Median of Squares Regression" *J. Am. Statistical Assoc.*, vol. 79, pp. 871-880, 1984.
- [35] J. Lim and N. Barnes, "Estimation of the Epipole Using Optical Flow at Antipodal Points," *Computer Vision and Image Understanding*, vol. 114, no. 2, pp. 245-253, 2010.
- [36] J. Lim and N. Barnes, "Directions of Egomotion from Antipodal Points," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2008.
- [37] C.X. Hu and L.F. Cheong, "Linear Quasi-Parallax SfM Using Laterally-Placed Eyes" *Int'l J. Computer Vision*, vol. 84, no. 1, pp. 21-39, 2009.
- [38] J. Lim and N. Barnes, "Estimation of the Epipole Using Optical Flow at Antipodal Points," *Proc. Workshop on Omnidirectional Vision Camera Networks and Non-Classical Cameras*, 2007.
- [39] I. Thomas and E. Simoncelli, "Linear Structure from Motion," technical report, IRCS, Univ. of Pennsylvania, 1994.
- [40] D.H. Ballard and O.A. Kimball, "Rigid Body Motion from Depth and Optical Flow" *Computer Vision, Graphics, and Image Processing*, vol. 22, pp. 95-115, 1984.
- [41] A. Makadia, C. Geyer, S. Sastry, and K. Daniilidis, "Radon-Based Structure from Motion without Correspondences," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2005.
- [42] R.J.M. den Hollander and A. Hanjalic, "A Combined RANSAC-Hough Transform Algorithm for Fundamental Matrix Estimation," *Proc. 18th British Machine Vision Conf.*, 2007.
- [43] B. Triggs, "Plane+ Parallax, Tensors and Factorization," *Proc. European Conf. Computer Vision*, pp. 522-538, 2000.
- [44] C. Rother and S. Carlsson, "Linear Multi View Reconstruction and Camera Recovery," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 42-50, 2001.
- [45] B. Triggs, P.F. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle Adjustment—a Modern Synthesis," *Proc. Int'l Workshop Vision Algorithms*, pp. 298-372, 1999.
- [46] P.D. Kovesi, "MATLAB and Octave Functions for Computer Vision and Image Processing," <http://www.csse.uwa.edu.au/~%7Epk/research/matlabfns/>, 2009.
- [47] Point Grey Research, <http://www.ptgrey.com>, 2009.
- [48] D. Lowe, SIFT Code: <http://www.cs.ubc.ca/~%7Elowe/keypoints/>, 2009.
- [49] E. Tola, V. Lepetit, and P. Fua, "A Fast Local Descriptor for Dense Matching," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2008.
- [50] H. Li, "Consensus Set Maximization with Guaranteed Global Optimality for Robust Geometry Estimation," *Proc. IEEE Int'l Conf. Computer Vision*, 2009.