

# An Iterative 5-pt Algorithm for Fast and Robust Essential Matrix Estimation

Vincent Lui

Department of Electrical & Computer  
Systems Engineering  
Monash University  
Clayton, Australia

Tom Drummond

---

## Abstract

This paper presents a novel algorithm for calculating epipolar geometry from 5 correspondences. The algorithm directly solves for the orientation of each camera relative to the baseline that separates them and is able to impose the half plane constraints that arise from the requirement that visible landmarks must be in front of both cameras. The algorithm is conceptually simple, and provides numerically stable solutions that are used as a hypothesis generator within RANSAC. It is significantly faster than existing methods and is comfortably able to provide frame-rate performance on real data.

## 1 Introduction

Relative pose estimation is an important problem in computer vision for applications such as structure from motion and augmented reality. The well-known essential matrix, a  $3 \times 3$  matrix which contains relative pose information, was first introduced by Longuet-Higgins [LH83]. This work led to the normalized 8-pt algorithm [N84] for uncalibrated cameras, and the 6-pt [L85] and 5-pt algorithms [L86] for calibrated cameras.

The 5-pt algorithm is the current state-of-the-art in relative pose estimation. It is a minimal-set direct solver which solves the essential matrix as a system of polynomial equations. Here we introduce an iterative method that provides improved performance. We make the following contributions:

- A novel geometrically meaningful error function which encodes half plane constraints arising from the visibility criterion [L86].
- Robust and real-time essential matrix estimation, with an average of 30fps.

Although direct solvers offer a closed-form solution, the use of high degree polynomials may lead to ill-conditioning and they are difficult to solve [L86]. Iterative solvers, on the other hand, are numerically stable. Although convergence is not guaranteed for iterative solvers, they are usually simpler and faster compared to direct polynomial solvers. Hence, when used within RANSAC, more hypotheses can be evaluated in the same block of time, resulting in improved performance and robustness.

## 2 Related Work

Philip [12] first introduced the 5-pt algorithm, solving the essential matrix using a 13<sup>th</sup> degree polynomial. Nistér’s 5-pt algorithm [13] improved the work of Philip by reducing the polynomial to a 10<sup>th</sup> degree polynomial. Since then, several alternatives which avoid using high degree polynomials have been proposed. These alternatives sacrifice speed for simplicity. [2, 9, 14] represented the polynomial constraints on the essential matrix in different ways: [14] used Gröbner bases, [9] used the hidden variable technique, and [2] represented the constraints as a cubic polynomial eigenvalue problem.

Batra *et al.* [10] estimated the essential matrix and translation vector simultaneously as a constrained optimization problem. Naroditsky *et al.* [15] introduced the 3-plus-1 algorithm, which uses 3 point correspondences and 1 directional correspondence to solve the relative pose problem. It is faster than the 5-pt algorithm and provides better rotational performance. However, the 5-pt algorithm performs better for sideways translational motion.

Recent approaches for estimating the essential matrix include [6, 8, 11]. The work in [6] and [11] parametrize the image plane as a unit sphere, similar to our work. Hartley and Kahl [6] searched for a rotation which minimizes the angular error between estimated and measured image points. However, this approach is exhaustive and is not suited for real-time applications. Lim and Barnes [11] used antipodal point correspondences from an omnidirectional camera to estimate the epipole location. Kneip *et al.*’s [8] approach uses an inertial measurement unit to obtain relative rotation whereas the translation between two cameras is computed iteratively using the constraint of the intersection between two epipolar planes. In contrast, our approach uses only visual information and computes both relative rotation and translation between two cameras using the angle between epipolar planes.

Iterative solvers for solving the essential matrix were proposed in [6] and [11], which used parametrizations that are different from the work in this paper. [6] and [11] uses a parametrization based on an algebraic error constraint, which does not enforce orientation consistency between points. By contrast, our parametrization, which is based on a geometric error, allows the incorporation of the half plane constraint and thus enforces orientation consistency between points.

### 2.1 Epipole Parametrization

Figure 1 shows the geometry of two views. A coordinate frame for the geometry is chosen such that the z-axis  $e_z$  joins the two camera centres. In this frame, vectors  $\hat{v}_i$  and  $\hat{v}'_i$  and  $e_z$  are coplanar. The goal of the iterative 5-pt algorithm is to find a rotation for each of the two cameras that maps from their internal coordinate frame to that of Figure 1. In order to do this, the image plane for each camera is represented as a unit 2-sphere. Image points in normalized camera coordinates  $[x, y, 1]^T$  are mapped to unit vectors that lie on the sphere by dividing by  $\sqrt{x^2 + y^2 + 1}$ , giving

$$\hat{u} = \frac{[x, y, 1]^T}{\sqrt{x^2 + y^2 + 1}}, \quad \hat{u} \neq -\hat{u}. \quad (1)$$

At each iteration, the normalized point correspondences  $\hat{u}_i \leftrightarrow \hat{u}'_i$  are left multiplied with the rotations  $R$  and  $R'$ , giving the rotated point correspondences

$$\hat{v} = R\hat{u}, \quad \hat{v}' = R'\hat{u}'. \quad (2)$$

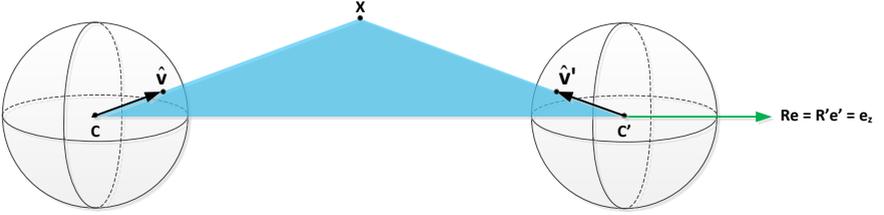


Figure 1: Visual concept of iterative 5-pt algorithm. Image planes are represented as unit spheres, constraining the epipole to lie on the sphere’s surface. Here, the rotated point correspondences lie on the same plane as the epipoles.

This rotates the two unit spheres and changes the direction of the epipoles. By rotating the unit spheres such that the z-axis  $e_z$  is aligned with the epipoles, i.e.  $e_z = Re = R'e'$ , the rotated point correspondences  $\hat{v}_i \leftrightarrow \hat{v}'_i$  become coplanar with the epipoles  $e, e'$ . The estimated epipole location is then given as

$$e = R^T e_z, \quad e' = R'^T e_z. \quad (3)$$

## 2.2 Error Function

If  $R$  and  $R'$  are incorrect then  $\hat{v}_i$  and  $\hat{v}'_i$  will not lie on a common plane with  $e_z$ . In particular there will be a non-zero angle between the plane defined by  $\hat{v}_i$  and  $e_z$  and the plane defined by  $\hat{v}'_i$  and  $e_z$ . The angle can be computed by projecting  $\hat{v}_i$  and  $\hat{v}'_i$  onto the x-y plane and computing the angle between the two vectors. This is analogous to looking at a plane in the direction of  $e_z$  (see Figure 2). The 3D unit vectors become 2-vectors  $p_i$  and  $p'_i$  on the plane, which can be obtained simply by discarding the third element in the 3D vectors.

The 2-vectors have angles  $A_i$  and  $A'_i$  respectively, relative to the x-axis  $e_x$  of the plane. Instead of testing whether the epipole lies on an allowed segment of a conic [24], we enforce the half-plane constraint by noting that the angles  $A_i$  and  $A'_i$  will be equal if and only if the back-projected rays from the image points lie on the same half-plane. We define the residual error for each rotated point correspondence as the angle between their projections onto the z-axis. Let  $[x_i \ y_i \ z_i]^T$  be the 3-vector of a rotated point  $\hat{v}_i$ . The angle of the rotated point relative to  $e_x$  on the 2D plane is

$$A(\hat{v}_i) = \text{atan2}(y_i, x_i), \quad (4)$$

and the residual error associated with each rotated point correspondence is

$$r_i = A(\hat{v}_i) - A(\hat{v}'_i), \quad (5)$$

taking care when  $A(\hat{v}_i)$  and  $A(\hat{v}'_i)$  are near  $\pm\pi$ . In other words, if  $A$  exceeds  $|\pi|$ , it has to be either added ( $A < -\pi$ ) or subtracted ( $A > \pi$ ) with  $2\pi$  to ensure that  $A$  stays within the range of  $\text{atan2}()$ . Point correspondences not lying on the same half-plane produce an angular difference greater than  $90^\circ$ .

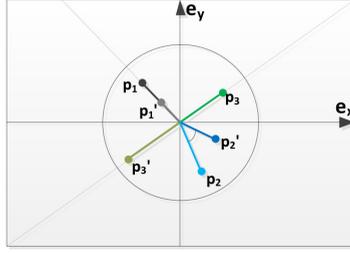


Figure 2: Rotated points orthogonally projected onto the plane tangential to the z-axis  $e_z$ . The point pair  $p_1 \leftrightarrow p_1'$  lie on the same radial line and has a zero residual error whereas  $p_2 \leftrightarrow p_2'$  has a non-zero residual error.  $p_3 \leftrightarrow p_3'$  satisfies the epipolar constraint but does not lie on the same half-plane, hence it does not have zero error.

### 2.3 Update of Rotational Parameters

The essential matrix has 5 degrees of freedom (DOF) but two rotations provide a total of 6DOF—the gauge freedom is the rotation of both cameras about the joint z-axis  $e_z$ . Thus, we constrain  $R'$  to have no component of rotation about its z-axis, giving 5 parameters  $\alpha_j$ , where  $\alpha_{1-3}$  represents  $R$  and  $\alpha_{4,5}$  represents  $R'$ . At each iteration, the rotational parameters are updated as

$$R^{k+1} = e^{\sum_{i=1}^3 \alpha_i G_i} R^k, \quad R'^{k+1} = e^{\alpha_4 G_1 + \alpha_5 G_2} R'^k. \quad (6)$$

In order to update the rotation parameters, the Jacobian for a rotated point pair is derived through partial differentiation of the residual error  $r_i$  relative to  $\alpha_{1-5}$ . Representing the projected point  $p_i$  as the 2-vector  $[x_i, y_i]$ , the partial derivatives of the angle  $A_i$  relative to  $x_i$  and  $y_i$  are given by

$$\frac{\partial A_i}{\partial x_i} = \frac{x_i}{x_i^2 + y_i^2}, \quad \frac{\partial A_i}{\partial y_i} = -\frac{y_i}{x_i^2 + y_i^2}. \quad (7)$$

Using the 3-dimensional Lie group,  $\text{SO}(3)$  [24], the exponential map parametrization is used to represent rotation through the generators

$$G_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, G_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, G_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

This allows the partial derivatives of  $x_i$  and  $y_i$  relative to  $\alpha_{1-5}$  to be derived easily, which are given as

$$\frac{\partial x_i}{\partial \alpha_j} = (G_j \hat{v}_i)[0], \quad \frac{\partial y_i}{\partial \alpha_j} = (G_j \hat{v}_i)[1]. \quad (8)$$

Equations 7-8 are combined using the chain rule to give the Jacobian

$$J = \frac{(x_i)((G_j \hat{v}_i)[1]) - (y_i)((G_j \hat{v}_i)[0])}{x_i^2 + y_i^2}. \quad (9)$$

Using the LM algorithm, the change to be applied to the rotational parameters  $\alpha_j$  is

$$\delta = (J^T J + \lambda I)^{-1} J^T r. \quad (10)$$

This process is repeated iteratively until the algorithm converges. The algorithm terminates early if the L2-norm of  $\delta$  is smaller than  $10^{-10}$ , or if the sum of squared errors is smaller than  $10^{-20}$ .

## 2.4 Recovery of Essential Matrix

The essential matrix is a cross-product between a translation and a rotation:

$$E = [\hat{t}_\wedge]R_E, \quad (11)$$

where  $\hat{t}$  is a translation,  $[\hat{t}_\wedge]$  is a skew-symmetric matrix, and  $R_E$  is a rotation matrix. The essential matrix is recovered through the two rotations  $R, R'$  and the second epipole  $e'$ . The translation  $\hat{t}$  is given by the second epipole  $e'$  whereas the rotation matrix is

$$R_E = R'^T R. \quad (12)$$

Noting that  $e = R^T e_z$  and  $e' = R'^T e_z$ , the essential matrix is recovered as

$$E = R'^T [e_z \wedge] R. \quad (13)$$

## 2.5 Approximation of atan2

The single most expensive part of the algorithm is performing  $\text{atan2}()$ . We use a quadratic approximation of the  $\text{atan2}()$  function [14], which gives a maximum error of 0.0038 radians:

$$\text{atan2}(y, x) = \begin{cases} \phi(s) & \text{if } y \geq 0, x \geq 0, \\ \pi - \phi(s) & \text{if } y \leq 0, x \geq 0, \\ -\pi + \phi(s) & \text{if } y \leq 0, x \leq 0, \\ -\phi(s) & \text{if } y \geq 0, x \leq 0, \end{cases} \quad (14)$$

where

$$s = (|x| - |y|) / (|x| + |y|), \quad \text{and} \quad (15)$$

$$\phi(s) = \frac{\pi}{4}(1 + s) + 0.273(1 - |s|). \quad (16)$$

## 2.6 Iterative 5-pt algorithm with RANSAC

The iterative 5-pt algorithm is used in conjunction with RANSAC [15] to make it robust to outliers. A number of hypotheses are generated, with each hypothesis initialized with the two rotation matrices  $R$  and  $R'$  being at identity. Each hypothesis is scored using the squared angular error function, i.e.  $(A(\hat{v}_i) - A(\hat{v}'_i))^2$ . We weigh each point pair according to the distance of the points to the epipole, which is simply the Euclidean distance from the origin of the 2D plane in Figure 2 to the 2-vector  $p_i$ . Points closer to the epipole are assigned a smaller weight as they have a smaller distance error for the same angular error. Let  $d_i$  and  $d'_i$  be the distance of a point pair to the epipole. Each point pair has a weight

$$w = \frac{2}{1/d_i^2 + 1/d_i'^2}. \quad (17)$$

The best hypothesis is then refined by extending the iterative 5-pt algorithm to use multiple points.

## 2.7 Triangulation

Detected inliers can now be easily triangulated by making use of cylindrical coordinates. The vectors  $\hat{v}_i$  and  $\hat{v}'_i$  can be transformed into cylindrical coordinates by dividing by  $\sqrt{x_i^2 + y_i^2}$  and  $\sqrt{x_i'^2 + y_i'^2}$  respectively:

$$\hat{v}_i / \sqrt{x_i^2 + y_i^2}, \quad \hat{v}'_i / \sqrt{x_i'^2 + y_i'^2}. \quad (18)$$

In this coordinate frame,  $\|x_i, y_i\| = 1$  and  $\|x'_i, y'_i\| = 1$ , so the inliers should have  $x_i = x'_i$  and  $y_i = y'_i$  since inliers have  $[x_i, y_i]$  pointing in the same direction as  $[x'_i, y'_i]$ . The disparity  $|z_i - z'_i|$  now encodes the cylindrical radial distance to the point, so setting

$$P_i = \frac{\hat{v}_i}{|z_i - z'_i|} \quad \text{and} \quad P'_i = \frac{\hat{v}'_i}{|z_i - z'_i|} \quad (19)$$

gives the coordinates of the triangulated point in the coordinate frame of Figure 1. Note that this calculation guarantees that  $p_i = p'_i \pm e_z$  as should be the case because the vector  $e_z$  represents the motion between the two cameras in the frame of  $\hat{v}_i$  and  $\hat{v}'_i$ . The coordinates of the triangulated point in the original camera frames can then be recovered by simply multiplying  $P_i$  and  $P'_i$  by  $R^T$  and  $R'^T$ :

$$Q_i = R^T P_i \quad \text{and} \quad Q'_i = R'^T P'_i \quad (20)$$

## 3 Results

The iterative 5-pt algorithm is evaluated using synthetic and real data. Firstly, the performance of the algorithm and its convergence radius are tested using synthetic data. Next, the timing performance of the iterative 5-pt algorithm is compared with Nistér's direct 5-pt algorithm [19] using both synthetic and real data.

### 3.1 Synthetic Data

250 synthetic point pairs are generated through a synthetic point cloud and a projection matrix. Using a camera with a  $67.38^\circ$  FOV and assuming that the first camera is at identity, the point cloud is projected onto the second camera. Outliers and noise can then be added to the point correspondences.

#### 3.1.1 Influence of Initial Guess

To test the influence of the initial guess of the epipole location on the iterative 5-pt algorithm, the initial guess of the epipole location is varied from being  $5^\circ$  to  $25^\circ$  away from the true epipole location. The iterative 5-pt algorithm is applied to 5 pairs of synthetic point correspondences for 10 iterations. The angular error between the true and estimated translation vector is computed, whereas rotational performance is measured using the Frobenius norm of the difference between the estimated and true rotation. The experiment is conducted twice, once without noise and the other with a random Gaussian noise of  $\sigma = 0.001$ . Figure 3 shows that both datasets follow the same trend, with the noisy dataset displaying slightly

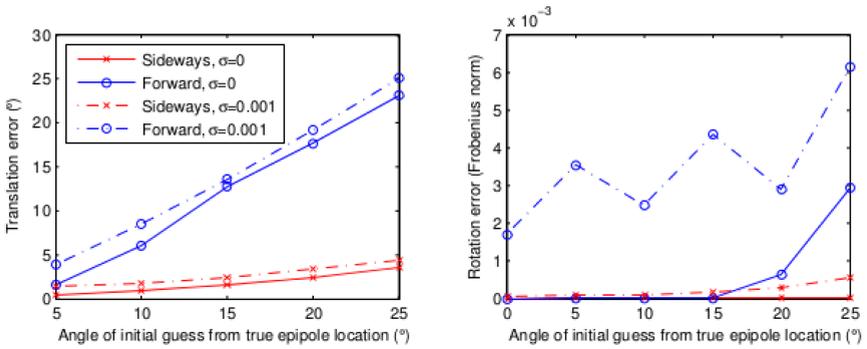


Figure 3: Influence of the initial guess of epipole locations on the performance of our iterative 5-pt algorithm with and without noise for forward and sideways motion.

larger errors. As the initial guess of the epipole location is slowly moved away from the true epipole location, forward motion performance degrades faster compared to sideways motion. However, in practice, as the iterative 5-pt algorithm is initialized with the two rotations being at identity, *i.e.* the two epipoles are initially at the  $z$ -axis, it more than compensates for this degradation, as shown in the real data experiments.

### 3.1.2 Timing

We compare the average time to generate a successful hypothesis between the iterative 5-pt algorithm and Nistér’s direct 5-pt algorithm for different outlier fractions. Firstly, 10,000 projection matrices are generated, each consisting of a random translation (up to 1 unit) and a rotation of up to  $30^\circ$  about one of the three Cartesian coordinate axes. For each projection matrix, a hypothesis is generated and the solution is iteratively refined using all the inliers found. To perform iterative refinement over all inliers, the iterative 5-pt algorithm is extended to a multi-point solver whereas an extension of Rosten *et al.*’s iterative 5-pt solver to multiple points [14] is used for the direct 5-pt algorithm. If the final number of inliers exceed 80% of the total number of inliers, the hypothesis is considered a success.

Let  $h$  be the number of successful hypotheses,  $t_{\text{avg}}$  be the average time to generate one hypothesis, and  $n$  be the average number of solutions generated for each hypothesis (for the iterative 5-pt algorithm,  $n$  is always 1). Assuming that the time to compute consensus,  $t_c$  is equal for both algorithms, the time to generate a successful hypothesis is

$$t_s = \frac{10000}{h}(t_{\text{avg}} + t_c n). \quad (21)$$

The results are shown in Figure 4, where the timing of a C implementation of the iterative 5-pt algorithm is measured on a 3.4 GHz machine. For the direct 5-pt algorithm, because we only have a MATLAB implementation of Nistér’s 5-pt algorithm<sup>1</sup>, we use this to calculate the success rate of hypotheses, but substitute timing for an optimised implementation of the algorithm reported by Naroditsky *et al.* [14] of  $13\mu\text{s}$  scaled down to  $11.47\mu\text{s}$  for  $t_{\text{avg}}$  as

<sup>1</sup><http://www.vis.uky.edu/stewe/FIVEPOINT/>

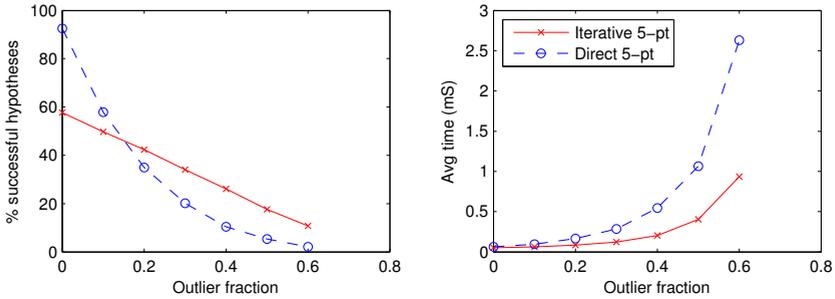


Figure 4: Left: The % of successful hypotheses with increasing outlier fraction. Right: Average time to generate a successful hypothesis with increasing outlier fraction.

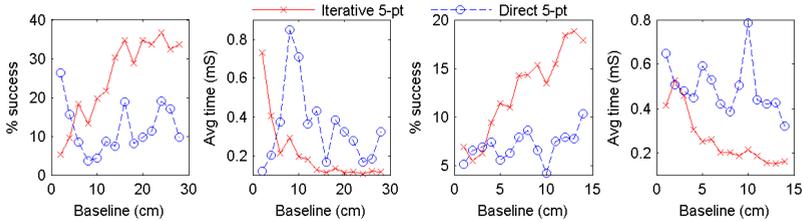


Figure 5: Graphs for experiment with real data and samples of real image sequences, showing % of successful hypotheses and the average time to generate a successful hypothesis for left: sideway motion, and right: forward motion

we are using a 3.4GHz machine by comparison to the 3GHz machine used to generate that timing.

The average time to compute consensus,  $t_c$  is about  $10.0\mu\text{S}$ . For low outlier fractions, the direct 5-pt algorithm records a greater % of successful hypotheses. However, there is an expensive cost associated with consensus checks, as the direct 5-pt algorithm generates an average of  $n = 4.5$  solutions for every hypothesis generated. Hence, the average time required by the iterative 5-pt algorithm to generate a successful hypothesis is smaller compared to the direct 5-pt algorithm. As the outlier fraction increases, the iterative 5-pt algorithm outperforms the direct 5-pt algorithm by a factor of 2.5.

### 3.2 Real Data

Two sets of real image sequences are generated, one for sideways and another for forward motion so that the direction of motion is known (see Figure 5). The camera used to capture the images is calibrated using an open-source MATLAB camera calibration toolbox<sup>2</sup>. There are 16 images in each dataset, and each image has a fixed baseline from the previous image

<sup>2</sup>[http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)

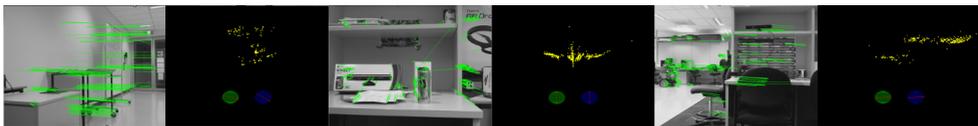


Figure 6: Inlier point matches from iterative 5-pt algorithm, and top view of triangulated inliers as well as the epipole directions.

in the sequence (2cm for sideways motion, 1cm for forward motion). Image pairs are formed between the first image and the  $N^{\text{th}} \in 2 \dots 16$  frame in the dataset, therefore slowly increasing the baseline. The scene depth varies from about 3m to infinity for sideways motion, and from about 1m to 1.5m for forward motion.

For each image pair, we detect features using Harris corners [9] and find 250 putative point correspondences using OpenCV’s implementation of the ORB (Oriented FAST & Rotated BRIEF) feature descriptor [18]. Two sets of matches are stored, one with all the 250 point correspondences whereas the other set filters out matches with a matching distance greater than a threshold of 40.

The essential matrix is estimated using standard RANSAC for both algorithms as this gives a familiar baseline for comparing timings. In practice, both algorithms can be used within more sophisticated RANSAC variants to further optimize computational efficiency. Samples of 5 point pairs are drawn randomly from the set of filtered matches, and consensus check is performed using the set of all matches. For each algorithm, only 1 hypothesis is generated in RANSAC. Iterative refinement is performed on all the inliers found so far, similar to the synthetic data experiment. We consider the hypothesis successful if the final estimated translation has an angular error less than  $5^\circ$ . For every image pair, the experiment is repeated 10,000 times so that the average time to generate a successful hypothesis can be computed.

The results (see Figure 5) show that the iterative 5-pt algorithm requires less time to generate a successful hypothesis compared to the direct 5-pt algorithm. For small baselines, the average time to obtain a successful hypothesis for the direct 5-pt algorithm is smaller compared to the iterative 5-pt algorithm. However, once the baseline increases, the iterative 5-pt algorithm dominates as pixel motion between the two views increases. For the plot on sideways motion, it can be seen that a small decrease in the percentage of successful hypothesis for the direct 5-pt algorithm leads to a spike in the average time to obtain a successful hypothesis due to the cost of consensus checks.

### 3.3 System Results

Figure 6 shows the inliers found using the iterative 5-pt algorithm, and the triangulated inliers viewed from the top. We use 1000 features here so that we have a denser reconstruction of the scene. On average, the timing breakdown of the system for 250 points is shown in Table 1.

## 4 Conclusion

We have presented a novel, iterative 5-pt algorithm to estimate the essential matrix robustly and in real-time. Using the parametrization presented in this paper, the inlier points can be

| Component                                | Time (mS) |
|--|-----------|
| Feature detection, extraction & matching | 12.7926   |
| Iterative 5-pt algorithm with RANSAC     | 7.1126    |
| Inliers refinement                       | 0.4157    |
| Drawing                                  | 1.9433    |

Table 1: Timing breakdown of live video implementation, averaged from 100 frames.

triangulated easily using cylindrical coordinates. We have also implemented a live video implementation of the iterative 5-pt algorithm, and showed that it can run at frame-rate.

## References

- [1] Dhruv Batra, Bart Nabbe, and Martial Hebert. An Alternative Formulation for Five Point Relative Pose Problem. In *Proc. WMVC*, pages 21–27, 2007.
- [2] Martin A Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Graphics and Image Processing*, 24(6), 1981.
- [3] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–152, 1988.
- [4] Richard Hartley. In defense of the eight-point algorithm. *TPAMI*, 19(6):580–593, June 1997.
- [5] Richard Hartley and Fredrik Kahl. Global Optimization through Rotation Space Search. *IJCV*, 82(1):64–79, January 2009.
- [6] Uwe Helmke, Knut Hüper, Pei Yean Lee, and John Moore. Essential Matrix Estimation Using Gauss-Newton Iterations on a Manifold. *IJCV*, 74(2):117–136, January 2007.
- [7] Li Hongdong and Richard Hartley. Five-Point Motion Estimation Made Easy. In *Proc. ICPR*, pages 630–633, 2006.
- [8] L Kneip, M Chli, and R Siegwart. Robust real-time visual odometry with a single camera and an imu. In *Proc. BMVC*, 2011.
- [9] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial Eigenvalue Solutions to the 5-pt and 6-pt Relative Pose Problems. In *Proc. BMVC*, 2008.
- [10] John Lim and Nick Barnes. Estimation of the epipole using optical flow at antipodal points. *CVIU*, 114(2):245–253, February 2010.
- [11] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–134, 1981.
- [12] Oleg Naroditsky, Xun S Zhou, Jean Gallier, Stergios I Roulmeliotis, and Kostas Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *TPAMI*, 34(4):818–24, April 2012.

- [13] David Nistér. An efficient solution to the five-point relative pose problem. *TPAMI*, 26(6):756–77, June 2004.
- [14] Johan Philip. A Non-Iterative Algorithm for Determining All Essential Matrices Corresponding to Five Point Pairs. *Photogrammetric Record*, 15(October):589–599, 1996.
- [15] Oscar Pizarro, Ryan Eustice, and Hanumant Singh. Relative Pose Estimation for Instrumented , Calibrated Platforms. In *Proc. DICTA*, pages 601–612, 2003.
- [16] Sreeraman Rajan, Sichun Wang, and Robert Inkol. Efficient Approximations for the Four-Quadrant Arctangent Function. In *Proc. CCECE*, pages 1043–1046. IEEE, 2006.
- [17] Edward Rosten, Gerhard Reitmayr, and Tom Drummond. Improved RANSAC performance using simple, iterative minimal-set solvers. 2010.
- [18] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proc. ICCV*, pages 2564–2571. Ieee, November 2011.
- [19] Henrik Stewénus, Christopher Engels, and David Nistér. Recent developments on direct relative orientation. *ISPRS*, 60(4):284–294, June 2006.
- [20] V Vadarajan. *Lie Groups, Lie Algebras, and Their Representations*. Springer-Verlag, 1974.
- [21] Tomas Werner and Oxford Ox. Constraint on Five Points in Two Images. In *Proc. CVPR*, pages 2–7, 2003.