

# An Alternative Formulation for Five Point Relative Pose Problem

Dhruv Batra<sup>1</sup>      Bart Nabbe<sup>1,2</sup>      Martial Hebert<sup>1</sup>  
dbatra@ece.cmu.edu    bart.c.nabbe@intel.com    hebert@ri.cmu.edu

<sup>1</sup>Carnegie Mellon University    <sup>2</sup>Intel Research Pittsburgh  
Pittsburgh, PA, 15213      Pittsburgh, PA, 15213

## Abstract

*The “Five Point Relative Pose Problem” is to find all possible camera configurations between two calibrated views of a scene given five point-correspondences. We take a fresh look at this well-studied problem with an emphasis on the parametrization of Essential Matrices used by various methods over the years. Using one of these parametrizations, a novel algorithm is proposed, in which the solution to the problem is encoded in a system of nine quadratic equations in six variables, and is reached by formulating this as a constrained optimization problem. We compare our algorithm with an existing 5-point method, and show our formulation to be more robust in the presence of noise.*

## 1. Introduction

Estimation of world co-ordinates of scene points (called “Scene Reconstruction”) and relative motion of camera (called “Ego-motion estimation”) from images is a classical problem with extensive literature in photogrammetry and (more lately) computer vision communities. It is well known that in the calibrated setting (when all intrinsic parameters of the camera are known a priori), given enough (at least five) point-correspondences between two views of a scene, the relative motion of the camera can be recovered. The “Five Point Relative Pose Problem” is to find all possible camera configurations between two calibrated views of a scene given *exactly* five point-correspondences, and was originally shown by Kruppa [3] to have 11 solutions. This was later improved by [1],[4],[5] to show that there are in fact only 10 possible valid solutions.

### 1.1. Previous Work

One of the earliest attempts at the solution to this problem, Kruppa’s method requires finding all intersection of two sextic curves and does not render itself to a feasible implementation. Over the years, several solutions have been proposed but few of them are suitable for numerical implementation. Triggs [6] builds a 60 x 60 sparse matrix, which

is later reduced to a 20 x 20 matrix, and then formulates the solution as an eigen decomposition of this reduced matrix. Philip [7] uses elimination to reach a 13<sup>th</sup> order polynomial, ten of which are the solutions to our problem. Nister [8] extends this idea to get a 10<sup>th</sup> order polynomial, all of which are the required solutions. Both these approaches work with a closed-form high-order (13th and 10th) uni-variate polynomial equation which encodes the solution to the problem. However, since no closed form solutions are known for polynomials with degree greater than 5, iterative numerical methods are used to converge to the solutions. Root finding for high order polynomials is an ill-conditioned problem [11], [12], where slightest perturbations in the coefficients of the equation can cause the solutions to not only change drastically but in some cases become complex. To get an intuition of how ill-posed this problem really is consider the following polynomial:

$$P_1(x) = x^4 + 4x^3 + 6x^2 + 4x + 1, \quad (1)$$

and a slightly perturbed version of this polynomial:

$$P_2(x) = x^4 + 4x^3 + 6x^2 + 4.01x + 1. \quad (2)$$

The result of the ‘solve’ command (in MAPLE<sup>©</sup>) on these are:

$$\text{roots}\{P_1(x)\} = ( -1 \quad -1 \quad -1 \quad -1 )^T, \quad (3)$$

$$\text{roots}\{P_2(x)\} = \left( \begin{array}{c} -1.34024 \\ 0.97500 - \iota 0.31723 \\ 0.97500 + \iota 0.31723 \\ -0.70975 \end{array} \right). \quad (4)$$

Although for low order polynomials such extreme behaviors is mostly limited to repeated roots, high order polynomials are inherently ill-posed for such analysis. As an extreme example, consider the infamous Wilkinson’s pathological polynomial [12]:

$$\begin{aligned} P(x) &= (x+1)(x+2)\dots(x+20) \\ &= x^{20} + 210x^{19} + \dots + 20! \end{aligned} \quad (5)$$

Although this polynomial has small and separated integral roots  $(-1, -2, \dots, -20)$ , its coefficients are very large. Consider the displacement its roots undergo when the coefficient of  $x^{19}$  is perturbed by  $2^{-23}$  (which is a change of  $5.66 \times 10^{-8}\%$ ):

$$\begin{pmatrix} -1.00000 & -2.00000 & -10.09527 \pm i 0.64350 \\ -3.00000 & -4.00000 & -11.79363 \pm i 1.65233 \\ -5.00000 & -6.00000 & -13.99236 \pm i 2.51883 \\ -6.99970 & -8.00727 & -16.73074 \pm i 2.81262 \\ -8.91725 & -20.84691 & -19.50244 \pm i 1.94033 \end{pmatrix} \quad (6)$$

We can only imagine the magnitude of errors that can creep into our calculations from more realistic errors. The situation is further worsened by the fact that the coefficients of these high order uni-variate polynomials are not directly estimated. These coefficients are derived by an elimination procedure from several low-order multivariate polynomials, further blowing up our estimates. It is for these reasons that the authors would prefer to work with a low-order system.

The rest of the paper is organized as follows: Section 1.2 introduces the notation and the problem definition. Section 2 describes our algorithm for estimating essential matrices, and Section 3 shows the results of a comparative evaluation of our algorithm with an existing 5-point method [13].

## 1.2. Problem Background and Formulation

Figure 1 shows a standard two view setup, where the second view is reached from the first one when the camera undergoes a rigid transformation (represented by a translation and a rotation). In the calibrated setting, since all intrinsic parameters of the camera are known, we can directly work with the normalized image plane instead of the retinal plane. Assuming the starting position of the camera to be the origin, and knowing the rigid transform it undergoes, the finite-projection center camera matrices can be represented as:

$$M_1 = K_1[I|0] \quad \text{and}, \quad (7)$$

$$M_2 = K_2[R|t]. \quad (8)$$

If  $\hat{\mathbf{p}}$  and  $\hat{\mathbf{p}}'$  are two image points (represented as homogeneous 3-vectors) looking at the same world point  $P$  (represented as homogeneous 4-vector), then the coplanarity constraint is represented as

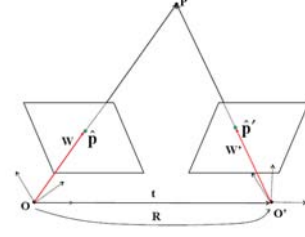
$$\hat{\mathbf{p}}^T E \hat{\mathbf{p}}' = 0, \quad (9)$$

where the Essential Matrix  $E$  is defined by

$$E = [t]_{\times} R \quad \text{and}, \quad (10)$$

$[t]_{\times}$  is the cross product matrix formed by vector  $t$ ,

$$[t]_{\times} = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} = T. \quad (11)$$



**Figure 1:** The normalized two view setup. One of the cameras is assumed to be at the origin.

Since the goal is to use the correspondence constraints to compute the essential matrix, it is important to realize that given enough constraints (at least 8) linear techniques exist [9] to reach the solution, but no linear methods are known when fewer than 8 correspondences are provided. It then becomes necessary to study the properties of essential matrices and use a ‘suitable’ parametrization to proceed. The following (previously proved) theorems are collated from [1],[5],[7] and [14]:

**Theorem 1** *The essential matrix is not full rank and has a one dimensional null space, that is*

$$\det(E) = 0 \quad \text{and}, \quad (12)$$

$$\mathcal{N}(E^T) \equiv t. \quad (13)$$

**Theorem 2** *The essential matrix has two equal non-zero singular values*

$$E \stackrel{svd}{=} U \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T. \quad (14)$$

**Theorem 3** *A nonzero  $3 \times 3$  matrix is an essential matrix if and only if it satisfies*

$$(EE^T)E - \frac{1}{2}\text{trace}(EE^T)E = 0. \quad (15)$$

**Theorem 4** *The necessary (but not sufficient) condition for a nonzero  $3 \times 3$  matrix to be an essential matrix is*

$$\frac{1}{2}\text{trace}(EE^T)^2 - \text{trace}[(EE^T)^2] = 0. \quad (16)$$

While each one of the above theorems lends itself to a particular parametrization of essential matrices, the next theorem establishes that they are all equivalent.

**Theorem 5** *A real  $3 \times 3$  matrix  $E$  is an essential matrix if and only if*

- *Parametrization 1: It can be decomposed as the product of a skew-symmetric and a rotation matrix,*

$$E = [t]_{\times} R, \quad (17)$$

and if and only if

- *Parametrization 2: It has one singular equal to zero and other two singular values equal,*

$$E \stackrel{\text{svd}}{=} U \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T, \quad (18)$$

and if and only if

- *Parametrization 3: It satisfies the following cubic constraints*

$$(EE^T)E - \frac{1}{2}\text{trace}(EE^T)E = 0, \quad (19)$$

and if and only if

- *Parametrization 4: It satisfies the following cubic and degree-4 constraints*

$$\det(E) = 0 \text{ and}, \quad (20)$$

$$\frac{1}{2}\text{trace}(EE^T)^2 - \text{trace}[(EE^T)^2] = 0. \quad (21)$$

Parametrizations 3 and 4 describe the manifold of essential matrices and have been extensively used in the past. Parametrization 3 was introduced by Demazure [4] who studied a sub-manifold ( $\mathfrak{M}$ ) of  $\mathbb{P}^8$  described by the 9 cubic constraints. Maybank [1] introduced Parametrization 4 and showed that equations (20) and (21) together describe the same manifold ( $\mathfrak{M}$ ) of essential matrices, under the assumption of real matrices. Philip [7] and Nister [8] use this parametrization and eliminate variables in nine cubic constraints to achieve a univariate (13<sup>th</sup> and 10<sup>th</sup>) polynomial equation. Parametrization 2 is generally used with SVD-based techniques [2],[10] to decompose an essential matrix into a rotation matrix and a translation vector.

## 2. Proposed Algorithm

Given the minimal case of five point-correspondences between two views (Figure 2), each of these correspondences result in an epipolar constraint (9), which is linear in the entries of  $E$ , and can be written as

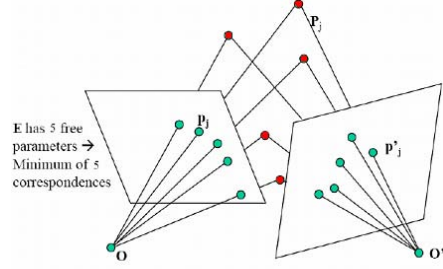
$$\hat{\mathbf{p}}_{vec}^T E_{vec} = 0, \quad (22)$$

where

$$\hat{\mathbf{p}}_{vec}^T \equiv [\hat{p}_1 \hat{p}'_1 \quad \hat{p}_2 \hat{p}'_1 \quad \hat{p}_3 \hat{p}'_1 \quad \hat{p}_1 \hat{p}'_2 \quad \hat{p}_2 \hat{p}'_2 \quad \hat{p}_3 \hat{p}'_2 \quad \hat{p}_1 \hat{p}'_3 \quad \hat{p}_2 \hat{p}'_3 \quad \hat{p}_3 \hat{p}'_3] \quad (23)$$

and,

$$E_{vec}^T \equiv [E_{11} \ E_{12} \ E_{13} \ E_{21} \ E_{22} \ E_{23} \ E_{31} \ E_{32} \ E_{33}], \quad (24)$$



**Figure 2:** Two views with five point-correspondences provided.

We can stack all 5 of the equations of the form (22) into a constraint matrix

$$A E_{vec} = 0, \quad (25)$$

where  $A$  is a  $5 \times 9$  matrix. Without the use of any other information, this system is under-constrained and thus the required essential matrix (in its corresponding vector form) lies in the 4 dimensional subspace of  $\mathbb{P}^8$ . This subspace is the null space of  $A$ , and the basis can be easily extracted using SVD. Thus

$$E_{vec} = \sum_{i=1}^4 \alpha_i \mathcal{X}_i, \quad (26)$$

where  $\{\mathcal{X}_1 \ \mathcal{X}_2 \ \mathcal{X}_3 \ \mathcal{X}_4\}$  is the basis for the right null space of  $A$ , and  $\alpha_i$  are the four (arbitrary) scalar multipliers.

Since the essential matrix has 5 degree of freedom (and we have already imposed 5 constraints), this system is *exactly* constrained. All that is required is for us to enforce that  $E$  is an essential matrix and that  $E_{11}, E_{12}, \dots, E_{33}$  are not independent variables. As we have already seen, this is usually done by describing the manifold of essential matrices in terms of entries in  $E$  (or equivalently  $\alpha_i$ ). We enforce  $E$  to be an essential matrix with the following constraints

$$t^T E = 0 \text{ and}, \quad (27)$$

$$\begin{aligned} EE^T &= ([t]_{\times} R)([t]_{\times} R)^T \\ &= [t]_{\times} R R^T [t]_{\times}^T \\ &= [t]_{\times} [t]_{\times}^T \\ &= T T^T. \end{aligned} \quad (28)$$

At this juncture, we would like to point out that the translation vector can be extracted from the essential matrix by straightforward SVD-based techniques [2], [10], and thus it is standard practice to solve for the essential matrix exclusively. While it may seem counter-intuitive to simultaneously solve for the essential matrix and the translation vector (which is perhaps why this has not been explored till

now), as we soon see, this allows us to system of quadratic equations, and it is this reduction in the order of equations that makes this a more stable formulation.

## 2.1. Formulating as a Constrained Optimization

Equations (27) and (28) together form a system of 9 homogeneous quadratic equations (3 from equation (27) and 6 from (28) in 7 variables:

$$x^T A_i x = 0, \quad (29)$$

where

$$i = 1, 2, \dots, 9 \text{ and,}$$

$$x^T = [\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ t_1 \ t_2 \ t_3]. \quad (30)$$

Thus the problem can finally be formulated as

$$\min \sum_{i=1}^9 \|x^T A_i x\|^2, \quad (31)$$

such that

$$x^T x = 1, \quad (32)$$

where equation (32) is necessary since the entire system in homogeneous co-ordinates and hence up to any arbitrary scale. Thus we find that we have effectively reduced our problem to that of a regular constrained minimization, which can be solved using standard techniques. The implementation described in this paper uses the MATLAB<sup>®</sup> Optimization Toolbox function ‘fmincon’, and is not followed by any root-polishing step. It should be noted here that the objective function is non-convex, and multiple seeds are required to converge to different minima. While this procedure does not guarantee convergence to global minima, in practice (as we show in our experiments), we can achieve a high confidence of finding the solution from a relatively small number of seeds.

## 3. Experimental Setup and Comparative Evaluations

The organization of this section is as follows: Section 3.1 describes the geometry of the synthetic data-sets used for quantitative analysis. We then compare the performance accuracies of our algorithm with an existing 5-point algorithm (the implementation for which is available at [13]), both without noise (in section 3.2) and with noise (in section 3.3). In section 3.4 we look at the convergence properties of our objective function.

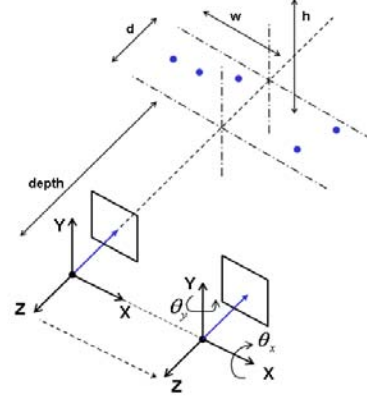


Figure 3: Geometry used to create test-set

### 3.1. Test Data Geometry

For quantitative analysis our experiments are performed on two test sets:

Set 1: In test set 1, the rotation matrix, translation vector and 5 world points are drawn from a uniform distribution.

Set 2: In test set 2, these quantities are drawn from a distribution centered around a reasonable geometry. The baseline between views is used as the reference unit and world points are chosen in a particular region of depths. This geometry is shown in Figure (3), and the parameters used are listed in table below:

Depth	$d$	$w$	$h$	$\theta_x$	$\theta_y$
$10t$	$5t$	$5t$	$5t$	$[-25^\circ \ +25^\circ]$	$[20^\circ \ 70^\circ]$

In all experiments a random rotation matrix and translation vector was sampled from the data-set to make an essential matrix. Five world points were also sampled and then projected using the two camera matrices to obtain correspondences, which were used for estimating the Essential Matrix via the two algorithms. Two different error metrics were used. The first was the residual

$$\min_i \left[ \min \left( \left\| \frac{\hat{E}_i}{\|\hat{E}_i\|} - \frac{E}{\|E\|} \right\|, \left\| \frac{\hat{E}_i}{\|\hat{E}_i\|} + \frac{E}{\|E\|} \right\| \right) \right], \quad (33)$$

computed over all the solutions returned. The second was the re-projection error, between the true projections of the world co-ordinates sampled:

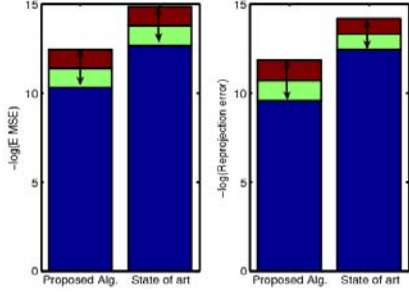
$$x_{true} = M_2 \text{ true } X_{world}, \quad (34)$$

and their estimates from our calculations:

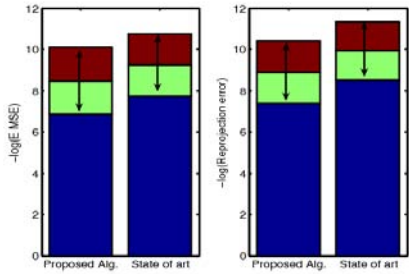
$$\hat{x} = \widehat{M}_2 X_{world}, \quad (35)$$

measured in pixel distances

$$\text{Error} = \sum_{5pts} \|x_{true} - \hat{x}\|, \quad (36)$$



**Figure 4:** Errors (in negative log) in data-set 1. Each experiment involves 200 data points. The arrows indicate the variance above and below the mean.



**Figure 5:** Errors (in negative log) in data-set 2. Each experiment involves 1000 data points. The arrows indicate the variance above and below the mean.

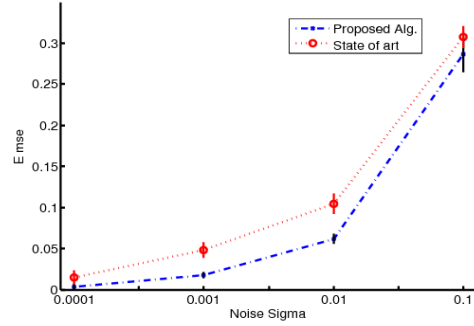
where  $M_2$  follows from equation (8).

### 3.2. Performance without noise

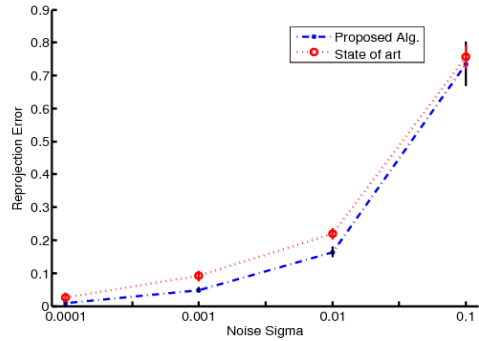
We would like to point out that without the presence of noise all 5-point methods would perform exactly the same in theory, and any differences seen in practice are due to implementation choices and resulting numerical issues (for example in our case, the precision of the stopping criterion, and the granularity of successive increments). It can be seen from Figure (4),(5) that without the presence of noise, as expected both 5-point methods perform very similar to each other. However, as we explore soon, they have different performances under noisy conditions.

### 3.3. Performance with noise

We study the performance of these algorithms under the presence of noise. To simulate tracking errors, Gaussian noise was added to the correspondences after projecting the true world points onto the two cameras. Figures (6),(7) show the behavior of these algorithms under noise on test-set 1, and we can see that our algorithm performs better under noisy conditions. The cases when [13] does not return any solution are not taken into consideration for computing the errors. Figures (8),(9) shows these error over test-set 2.



**Figure 6:** Test-set1: E MSE error versus Noise sigma. Each experiment involves 1000 data-points. Vertical lines at points denote confidence intervals at that sigma.



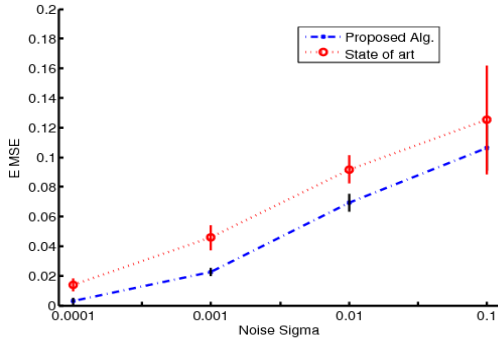
**Figure 7:** Test-set1: Re-projection error versus Noise sigma. Each experiment involves 1000 data-points. Vertical lines at points denote confidence intervals at that sigma.

### 3.4. Convergence Behavior

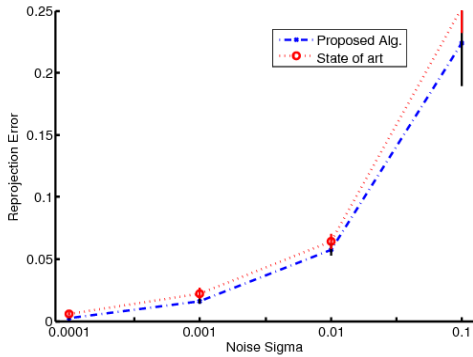
As we discussed in section 2, our algorithm requires random seeds as the starting point. It is important thus, to have an idea about the convergence, to study how the performance depends on the number of seeds. Figure (10) shows how frequently the true root of the system is reached as the number of seeds (drawn randomly with uniform prior) are increased. It can be seen that as the seeds reach as high as 15, the probability that the solution has been reached become more than 95 percent. Considering that in the worst case, there might be up to ten solutions to our system, this seems a reasonable number.

## 4. Conclusions

To summarize, we introduced an alternative formulation for estimating essential matrices in the minimal case of five points. We intuitively argued the stability of this method because of the reduction to a quadratic system of equations,



**Figure 8:** Test-set2: E MSE error versus Noise sigma. Each experiment involves 1000 data-points. Vertical lines at points denote confidence intervals at that sigma.



**Figure 9:** Test-set2: Re-projection error versus Noise sigma. Each experiment involves 1000 data-points. Vertical lines at points denote confidence intervals at that sigma.

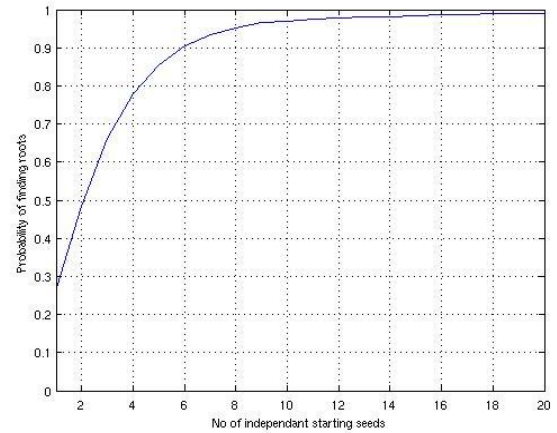
and showed the same to be true with experimental validation on synthetic data.

## Acknowledgments

We would like to thank Rahul Sukthankar, Siddhartha Srinivasa at Intel Research Pittsburgh, and Dave Tolliver, Fernando De la Torre at the Robotics Institute, CMU for helpful discussions and suggestions.

## References

- [1] O. Faugeras and S. Maybank, "Motion from point matches multiplicity of solutions," *International Journal of Computer Vision*, 4:225–246, 1990.
- [2] S. Maybank, "Theory of reconstruction from image motion," *Springer-Verlag, Berlin*, 1993.
- [3] E. Kruppa, "Zur Ermittlung eines Objektes aus zwei Perspek-



**Figure 10:** Curve showing the percentage of times the solution is found versus the number of seeds. Experiment performed on 2000 data-points.

- tiven mit innere Orientierung," *Sitz.-Ber. Akad. Wiss., Wien, Math.-Naturw. Kl., Abt. IIa*, (122):1939-1948, 1913.
- [4] M. Demazure, "Sur deux problemes de reconstruction," *Technical Report Rep. 882, INRIA, Les Chesnay, France*, 1988.
- [5] A. Heyden and G. Sparr, "Reconstruction from Calibrated Camera - a New Proof of the Kruppa-Demazure Theorem," *Journal of Mathematical Imaging and Vision*, 10:1-20, 1999.
- [6] B. Triggs, "Routines for Relative Pose of Two Calibrated Cameras from 5 Points," *Technical Report, INRIA, France*, 2000.
- [7] J. Philip, "A noniterative algorithm for determining all essential matrices corresponding to five point pairs," *Photogrammetric Record*, 15(88), pp.589-599, October 1996.
- [8] D. Nister, "An Efficient Solution to the Five-Point Relative Pose Problem," *IEEE Conference on Computer Vision and Pattern Recognition*, Volume 2, pp. 195-202, 2003.
- [9] H. Longuet-Higgins, "The reconstruction of a plane surface from two perspective projections," *Proc. Royal Society London*, vol. B227, pp. 399–410, 1986.
- [10] W. Wang and H. Tsui, "An SVD Decomposition of Essential Matrix with Eight Solutions for the Relative Positions of Two Perspective Cameras," *International Conference on Pattern Recognition*, Volume 1, p. 1362, 2000.
- [11] W. Press, B. Flannery, S. Teukolsky and W. Vetterling, "Numerical Recipes in C," *2nd edition, Cambridge University Press*, 1992.
- [12] F. Acton, "Numerical Methods That Work," *Washington: Math. Assoc. of America*, 1990.
- [13] H. Stewenius, C. Engels, and D. Nister, "Recent developments on direct relative orientation" *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284-294, June 2006. Code available at <http://vis.uky.edu/stewe/FIVEPOINT/>.
- [14] O. Faugeras, "Three-dimensional computer vision: a geometric viewpoint," *MIT Press*, 1993.