ROBUST POSE ESTIMATION BASED ON SYLVESTER'S EQUATION: SINGLE AND MULTIPLE COLLABORATIVE CAMERAS

Chong Chen, Dan Schonfeld *

University of Illinois at Chicago
Dept. of Electrical and Computer Engineering

Magdi Mohamed

Motorola Labs
Physical and Digital Realization Research

ABSTRACT

A method is introduced to track the object's motion and estimate its pose from multiple cameras. Firstly pose estimation from one camera is explained. We show that pose estimation from the corresponding feature points can be formed as a solution to Sylvester's equation. Furthermore, we develop a distributed solution, which indicates that pose estimation from multiple cameras can be obtained from the linear combination of results from each single camera. We relay on the results from other cameras to improve the estimate of the first one, and vice versa. Finally, the computer simulation experiments demonstrate the superior performance of our algorithm in robustness.

Index Terms— Pose Estimation, Sylvester's Equation, Lagrange Method, Best Linear Unbiased Estimator (BLUE)

1. INTRODUCTION

Pose analysis aims to recover the object and camera poses and motion parameters from static images or video sequences. In the literature, the study of the orientation problem is mainly focused on investigation of the uniqueness or the number of solutions. In [1], a family of linear algorithms has been developed to provide a unique solution for the 4-point, 5-point and n-point camera pose estimation problem. Specially, the camera exterior orientation estimation problem is often referred to as the perspective-n-point (PnP) problem.

We rely on the feature-based approach in this paper, and directly estimate the 3D pose from 2D image sequences. Scale-Invariant Feature Transform (SIFT) [2] is used to extract corresponding feature points from image sequences. The pose estimation from corresponding points based on Singular Value Decomposition (SVD) techniques have been well established [3]. We will show that the pose estimation problem can be formed as a solution to Sylvester's equation, which can be solved with many methods, such as Kronecker Product approach [4]. We show that the proposed approach to the solution of Sylvester's equation is equivalent to the classical SVD method for 3D-3D pose estimation. However, whereas classical SVD cannot be used for 2D-2D pose estimation, our method based on Sylvester's equation provides a new approach to pose estimation from 2D image sequences.

In the multi-view case, Luong and Faugeras [5] show that the epipolar geometry can be summarized in one Fundamental matrix. Rother and Carlsson [6], based on a reference plane, develop a linear algorithm for computation of 3D points and camera positions from multiple perspective views by finding the null-space of a matrix built from image data using SVD. Fermuller and Aloimonos [7] show the ambiguities for motion estimation from one camera because of the limited field of view. In [8], a system, consisting of six cameras, is built to remove the inherent ambiguities of confusion between translation and rotation. However, this system does not use one pose estimation for all information from all cameras simultaneously [9]. Frahm et al. [9] combine all information of all cameras to estimate the pose of a multi-camera system. In contrast, we are estimating the object's pose, and by using all information from all cameras to give one pose estimate, we also research the relationship between this estimate with those from each camera separately. In our system, cameras can be arbitrarily placed, and we assume that they are fixed, and the orientations and translations between each other are known.

The rest of this paper is organized as follows: Section 2 illustrates the method of pose estimation based on Sylvester's equation from one view. It is extended to multiple views in Section 3. In Section 4, experimental results show the efficiency of our algorithm. Finally, in Section 5, we present a brief summary.

*Thanks Physical and Digital Realization Research CoE, Motorola Labs, for funding.

2. POSE ESTIMATION FROM ONE CAMERA

With the solution to Sylvester's Equation, we can obtain the pose change between two successive frames in a video sequences. The current pose is computed by accumulating all of the past results. We are assuming that the pose in the first frame is known. In the following, we only focus the motion between two sequential images.

2.1. Projection from 3D to 2D

For a point in the 3D real object, its positions before and after motion have the relationship as follows.

$$\begin{pmatrix} x_2^j \\ y_2^j \\ z_2^j \end{pmatrix} = R_{3\times 3} \begin{pmatrix} x_1^j \\ y_1^j \\ z_1^j \end{pmatrix} + T \tag{1}$$

where $R_{3\times 3}$ is a 3×3 rotation matrix, and T is a 3×1 translation matrix. $j, (j = 1, 2, \ldots)$ represents the j^{th} feature point. With the pinhole camera model, if (u_1^j, v_1^j) and (u_2^j, v_2^j) are the corresponding projection points in the image respectively, we have $x_k^j/z_k^j = u_k^j/f$ and $y_k^j/z_k^j = v_k^j/f, (k = 1, 2)$, where f is the focal length. Replacing the unnecessary parameters in the real model from the above two equations, we finally get

$$\begin{pmatrix} u_2^j \\ v_2^j \\ f \end{pmatrix} = \frac{z_1^j}{z_2^j} R_{3\times 3} \begin{pmatrix} u_1^j \\ v_1^j \\ f \end{pmatrix} + T' \tag{2}$$

where $T' = T \cdot f/z_2^j$ is the 3×1 translation matrix within the image. Here we assume $z_1^j/z_2^j = 1$. Because of the short interval between two video images (less than $1/24$ second), the motion in this period will be small enough correspondingly. A similar model, the weak projective camera model [10], is used in the 2D-3D case. Moreover, if only the first two rows of Eq. 2 are considered, we get

$$\begin{pmatrix} u_2^j \\ v_2^j \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \begin{pmatrix} u_1^j \\ v_1^j \end{pmatrix} + \begin{pmatrix} l_x \\ l_y \end{pmatrix} \tag{3}$$

where $l_x = r_{13}f + t'_x$ and $l_y = r_{23}f + t'_y$. Our problem is to estimate the rotation matrix and translation vector with this equation.

2.2. Lagrange Method

For all of the matched points from SIFT, if we take Least-Square Error with respect to the rotation and translation variables, the 2D-2D pose estimation problem is concluded as

$$\begin{aligned} min \quad & (U_2 - R_1P_1 - L_x)(U_2 - R_1P_1 - L_x)^T \\ + \quad & (V_2 - R_2P_1 - L_y)(V_2 - R_2P_1 - L_y)^T \end{aligned} \tag{4}$$

subject to

$$\begin{cases} R_1R_1^T + r_{13}^2 = 1 \\ R_2R_2^T + r_{23}^2 = 1 \\ R_1R_2^T + r_{13}r_{23} = 0 \end{cases}$$

where $R_{2\times2} = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix}$, which is actually the top-left four elements of $R_{3\times3}$, $P_2 = \begin{pmatrix} U_2 \\ V_2 \end{pmatrix} = \begin{pmatrix} u_2^1 & u_2^2 & \dots \\ v_2^1 & v_2^2 & \dots \end{pmatrix}$, $P_1 = \begin{pmatrix} u_1^1 & u_1^2 & \dots \\ v_1^1 & v_1^2 & \dots \end{pmatrix}$ and $L = \begin{pmatrix} L_x \\ L_y \end{pmatrix} = \begin{pmatrix} l_x & l_x & \dots \\ l_y & l_y & \dots \end{pmatrix}$. We can apply Lagrange method to solve the constrained optimization problem.

$$\begin{aligned} F \;=\; & (U_2 - R_1P_1 - L_x)(U_2 - R_1P_1 - L_x)^T \\ + \; & (V_2 - R_2P_1 - L_y)(V_2 - R_2P_1 - L_y)^T \\ + \; & \lambda_1(R_1R_1^T + r_{13}^2 - 1) + \lambda_2(R_2R_2^T + r_{23}^2 - 1) \\ + \; & 2\lambda_3(R_1R_2^T + r_{13}r_{23}) \end{aligned} \tag{5}$$

where λ_1, λ_2 and λ_3 are Lagrange multipliers. Then the partial derivatives of F are taken, for the rotation coefficients and translation coefficients respectively.

$$\begin{cases} \frac{\partial F}{\partial R_1} = (-U_2 + L_x)P_1^T + R_1P_1P_1^T + \lambda_1R_1 + \lambda_3R_2 = 0 \\ \frac{\partial F}{\partial R_2} = (-V_2 + L_y)P_1^T + R_2P_1P_1^T + \lambda_2R_2 + \lambda_3R_1 = 0 \end{cases}$$

$$\begin{cases} \frac{\partial F}{\partial L_x} = -U_2 + R_1P_1 + L_x = 0 \\ \frac{\partial F}{\partial L_y} = -V_2 + R_2P_1 + L_y = 0 \end{cases} \tag{6}$$

Since there is only one translation coefficient for all the feature points in a frame, we can take the same method as in [3], and L is quickly determined from Eq. 6, once the rotation parameters are known.

$$\begin{pmatrix} l_x \\ l_y \end{pmatrix} = \begin{pmatrix} \overline{u_2} \\ \overline{v_2} \end{pmatrix} - \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \begin{pmatrix} \overline{u_1} \\ \overline{v_1} \end{pmatrix} \tag{7}$$

where $\begin{pmatrix} \overline{u_1} & \overline{v_1} \end{pmatrix}^T = \begin{pmatrix} \sum_{j=1}^M u_1^j/M & \sum_{j=1}^M v_1^j/M \end{pmatrix}^T$ and $\begin{pmatrix} \overline{u_2} & \overline{v_2} \end{pmatrix}^T = \begin{pmatrix} \sum_{j=1}^M u_2^j/M & \sum_{j=1}^M v_2^j/M \end{pmatrix}^T$, the mean of the feature points in the image coordinates from the two sequential images, and M is the number of feature points.

If we define $\overline{P_2} = \begin{pmatrix} \overline{U_2} \\ \overline{V_2} \end{pmatrix} = \begin{pmatrix} \overline{u_2} & \overline{u_2} & \dots \\ \overline{v_2} & \overline{v_2} & \dots \end{pmatrix}$ and $\overline{P_1} = \begin{pmatrix} \overline{U_1} \\ \overline{V_1} \end{pmatrix} = \begin{pmatrix} \overline{u_1} & \overline{u_1} & \dots \\ \overline{v_1} & \overline{v_1} & \dots \end{pmatrix}$, $P'_2 = \begin{pmatrix} U'_2 \\ V'_2 \end{pmatrix} = \begin{pmatrix} U_2 - \overline{U_2} \\ V_2 - \overline{V_2} \end{pmatrix}$, $P'_1 = P_1 - \overline{P_1}$, $A = P'_1P_1^T$ and $B = P'_2P_1^T$, Eq. 6 can be simplified as

$$\begin{cases} -U'_2P_1^T + R_1A + \lambda_1R_1 + \lambda_3R_2 = 0 \\ -V'_2P_1^T + R_2A + \lambda_3R_1 + \lambda_2R_2 = 0 \end{cases} \tag{8}$$

which can be further combined in one matrix equation.

$$-B + R_{2\times2}A + \Lambda R_{2\times2} = 0 \tag{9}$$

where $\Lambda = \begin{pmatrix} \lambda_1 & \lambda_3 \\ \lambda_3 & \lambda_2 \end{pmatrix}$. Eq. 9 is Sylvester's equation, which is also called Lyapunov's Equation in the special case with $\Lambda = A^T$. As shown in [3], they also obtain the same equation as Eq. 9 in the 3D-3D case, and solve it with SVD method. However, in our 2D-2D case, the orthonormality constraints have been changed in the deduction, $R_{2\times2}R_{2\times2}^T = \begin{pmatrix} 1 - r_{13}^2 & -r_{13}r_{23} \\ -r_{13}r_{23} & 1 - r_{23}^2 \end{pmatrix}$, which is an arbitrary matrix, and SVD approaches cannot be applied. But it can be computed with the Kronecker Product method [4] or numerical calculations.

From the solution $\tilde{R}_{2\times2}$, we can finally extract $(r_x, r_y, r_z)^T$, which are the three angles around x, y and z axis respectively, with Euler's rotation theorem.

3. POSE ESTIMATION FROM MULTIPLE CAMERAS

For simplicity, we will restrict our discussion in this paper to only two views. However, the ideas presented here can be easily extended to multiple views.

3.1. Transformation Between Two Cameras

We firstly suppose that all the cameras could capture the same feature points in the object, but our conclusion is independent of this assumption. Assume we have the following relations

$$\begin{cases} Q_2^l = R_{3\times3}^l Q_1^l + T_{3\times1}^l \\ Q_2^r = R_{3\times3}^r Q_1^r + T_{3\times1}^r \end{cases} \tag{10}$$

$$\begin{cases} Q_2^r = R_{3\times3}^o Q_2^l + T_{3\times1}^o \\ Q_1^r = R_{3\times3}^o Q_1^l + T_{3\times1}^o \end{cases} \tag{11}$$

where Q represents the 3D points. The subscript 1 or 2 displays the coordinates before or after movement, and the superscript *l* or *r* stands for *left* camera or *right* camera. Therefore Q_1^l are the points in the coordinate system with respect to the *left* camera, and Q_1^r are the same points but measured by the *right* camera. 3×3 and 3×1 point out the dimensions of matrices. $R_{3\times3}^o$ and $T_{3\times1}^o$ are rotation and translation between two cameras, and both are known. From Eq. 10 and Eq. 11, we can obtain

$$\begin{aligned} Q_2^r &= R_{3\times3}^r Q_1^r + T_{3\times1}^r \\ &= R_{3\times3}^r R_{3\times3}^o Q_1^l + R_{3\times3}^r T_{3\times1}^o + T_{3\times1}^r \\ &= R_{3\times3}^o Q_2^l + T_{3\times1}^o \\ &= R_{3\times3}^o R_{3\times3}^l Q_1^l + R_{3\times3}^o T_{3\times1}^l + T_{3\times1}^o \end{aligned} \tag{12}$$

Because Q_1^l is arbitrary, for the rotation matrix, we have

$$R_{3\times3}^r = R_{3\times3}^o R_{3\times3}^l (R_{3\times3}^o)^T \quad (13)$$

$$R_{3\times3}^l = (R_{3\times3}^o)^T R_{3\times3}^r R_{3\times3}^o \quad (14)$$

And for the translation matrix, we get

$$T_{3\times1}^r = R_{3\times3}^o T_{3\times1}^l + T_{3\times1}^o - R_{3\times3}^r T_{3\times1}^o \quad (15)$$

With the weak projective camera model, we know $T'_{3\times1} = T_{3\times1} \cdot f/z_2$, where T' is the translation in the images plane. Accordingly, the 3D translation with Eq. 15 can be represented with 2D translation as

$$T'^r_{3\times1} \frac{z_2^r}{f^r} = R_{3\times3}^o T'^l_{3\times1} \frac{z_2^l}{f^l} + T_{3\times1}^o - R_{3\times3}^r T_{3\times1}^o \quad (16)$$

where $T'^r_{3\times1} = \begin{pmatrix} t'^r_x & t'^r_y & f^r \end{pmatrix}^T$, $T'^l_{3\times1} = \begin{pmatrix} t'^l_x & t'^l_y & f^l \end{pmatrix}^T$. Once the focus length f^r and f^l are known, (t'^r_x, t'^r_y) and (t'^l_x, t'^l_y) can be obtained with the same method in Section 2. Moveover, we can get z_2^r and z_2^l from Eq. 16, and the 3D translation for each camera can be computed with $T'_{3\times1} = T_{3\times1} \cdot f/z_2$.

3.2. Pose Estimation from Two Cameras

We will take the same approach as in Section 2.2, but for a clearer explanation, we assume that the translation vectors have been properly compensated.

Firstly, let us define $R^o = \begin{pmatrix} r_{11}^o & r_{12}^o \\ r_{21}^o & r_{22}^o \end{pmatrix}$, $R^l = \begin{pmatrix} R_1^l \\ R_2^l \end{pmatrix} = \begin{pmatrix} r_{11}^l & r_{12}^l \\ r_{21}^l & r_{22}^l \end{pmatrix}$, $R^r = \begin{pmatrix} R_1^r \\ R_2^r \end{pmatrix} = \begin{pmatrix} r_{11}^r & r_{12}^r \\ r_{21}^r & r_{22}^r \end{pmatrix}$, which are actually the top-left four elements of $R_{3\times3}^o$, $R_{3\times3}^l$ and $R_{3\times3}^r$. Denote P_2^l, P_1^l, P_2^r and P_1^r are points minus their mean in images, also the same as before. Then the pose estimation problem could be concluded as

$$\begin{aligned} min \quad & (P_2^l - R^l P_1^l)(P_2^l - R^l P_1^l)^T \\ + \quad & (P_2^r - R^r P_1^r)(P_2^r - R^r P_1^r)^T \end{aligned} \quad (17)$$

subject to

$$\begin{cases} R_1^l R_1^{lT} + r_{13}^{l2} = 1 \\ R_2^l R_2^{lT} + r_{23}^{l2} = 1 \\ R_1^l R_2^{lT} + r_{13}^l r_{23}^l = 0 \end{cases} \quad (18)$$

$$\begin{cases} R_1^r R_1^{rT} + r_{13}^{r2} = 1 \\ R_2^r R_2^{rT} + r_{23}^{r2} = 1 \\ R_1^r R_2^{rT} + r_{13}^r r_{23}^r = 0 \end{cases} \quad (19)$$

Eq. 18 and Eq. 19 are constrains for *left* and *right* camera separately.

The Lagrange Method is applied to solve the constrained optimization problem.

$$min \quad F_M = F^l + F^r \quad (20)$$

where the superscripts l and r are used to distinguish two views, and F is defined in Eq. 5, except that the translations have been compensated. Define $A^l = P_1^l P_1^{lT}$, $B^l = P_2^l P_1^{lT}$, $A^r = P_1^r P_1^{rT}$ and $B^r = P_2^r P_1^{rT}$.

3.3. Centralized Solution

Replace R^r in Eq. 20 with $R^r = R^o R^l (R^o)^T + H$, where $H = R^o R^l (R^o)^T + \begin{pmatrix} r_{13}^o \\ r_{23}^o \end{pmatrix} \begin{pmatrix} r_{31}^l & r_{32}^l \end{pmatrix} (R^o)^T$ $+ \begin{pmatrix} r_{11}^o r_{13}^l + r_{12}^o r_{23}^l + r_{13}^o r_{33}^l \\ r_{21}^o r_{13}^l + r_{22}^o r_{23}^l + r_{23}^o r_{33}^l \end{pmatrix} \begin{pmatrix} r_{13}^o & r_{23}^o \end{pmatrix}$, which is obtained from Eq. 13, and compute the partial differential.

$$\begin{aligned} \frac{\partial F_M}{\partial R^l} &= -B^l + R^l A^l + \Lambda^l R^l + R^{oT}[-B^r \\ &\quad +(R^o R^l R^{oT} + H)A^r + \lambda^r (R^o R^l R^{oT} + H)]R^o \\ &= [-B^l + R^l A^l + \Lambda^l R^l] \\ &\quad +R^{oT}[-B^r + R^r A^r + \lambda^r R^r]R^o \\ &= 0 \end{aligned} \quad (21)$$

Once Eq. 21 is solved, the object's pose in the *left* camera coordinate system is obtained, with the information from both *left* and *right* cameras.

3.4. Separate Solution

The partial differential of Eq. 20 is calculated.

$$\frac{\partial F_M}{\partial R^l} = -B^l + R^l A^l + \Lambda^l R^l = 0 \quad (22)$$

$$\frac{\partial F_M}{\partial R^r} = -B^r + R^r A^r + \Lambda^r R^r = 0 \quad (23)$$

where $\Lambda^l = \begin{pmatrix} \lambda_1^l & \lambda_3^l \\ \lambda_3^l & \lambda_2^l \end{pmatrix}$ and $\Lambda^r = \begin{pmatrix} \lambda_1^r & \lambda_3^r \\ \lambda_3^r & \lambda_2^r \end{pmatrix}$. Thus, for a multiple camera system, we can use a solution to Sylvester's equation to estimate pose for each camera, just as the single camera case.

We can prove that, if $\hat{R}^l$ is a solution for Eq. 22, the constructed $\tilde{R}^r = R^o \hat{R}^l R^{oT} + H$ is a solution for Eq. 23. Similarly, if $\hat{R}^r$ is a solution for Eq. (23), then $\hat{R}_{3\times3}^r$ can be obtained from $\hat{R}^r$. We can also prove that $(R_{3\times2}^o)^T \hat{R}_{3\times3}^r R_{3\times2}^o$ is a solution for Eq. (22), where $R_{3\times2}^o$ is the first two columns of $R_{3\times3}^o$. Hence both $\hat{R}^l$ and $(R_{3\times2}^o)^T \hat{R}_{3\times3}^r R_{3\times2}^o$ are solutions for Eq. (21), and we will find the best linear unbiased estimate from these two results in Section 3.5.

As a result, we can use the estimates of other views to compute the pose with regard to the first camera, even to a virtual camera. The problem of pose estimation from multiple views can be solved with the independent pose estimate from each individual view. The equation obtained for one camera is analogous to that of other cameras.

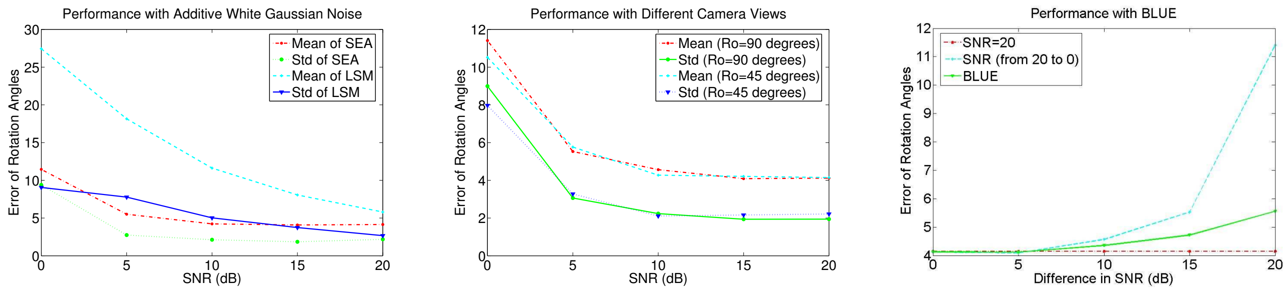
3.5. Best Linear Unbiased Estimator

Let us firstly rewrite the matrices $\hat{R}^l$ and $(R_{3\times2}^o)^T \hat{R}_{3\times3}^r R_{3\times2}^o$ into the vector form $\mathbf{s}_1$ and $\mathbf{s}_2$, and both of them are 4×1 vectors. Considering a general case, there are solutions $\mathbf{s}_i, i = 1, \ldots, N$, therefore we have

$$\mathbf{s} = \mathbf{I} \cdot \mathbf{s}_{true} + \mathbf{v} \quad (24)$$

where $\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \ldots & \mathbf{s}_N \end{bmatrix}^T$ is a $4N \times 1$ vector. $\mathbf{I} = \begin{bmatrix} I_{4\times4} & I_{4\times4} & \ldots & I_{4\times4} \end{bmatrix}^T$ is a $4N \times 4$ matrix, and $I_{4\times4}$ is a 4×4 identity matrix. $\mathbf{v}$ is a $4N\times1$ noise vector. $\mathbf{s}_{true}$ is the 4×1 vector, denoting the ground truth. Now we will find the Best Linear Unbiased Estimator (BLUE) $\tilde{\mathbf{s}}$ of $\mathbf{s}_{true}$ with $\mathbf{s}_i$.

Define $\mathbf{C}$ is the $4N\times4N$ covariance matrix of $\mathbf{v}$, and assume the cameras are independent of each other. Correspondingly, $\mathbf{C}$ is of the



(a) from the reference view under different input SNR

(b) in the transformed results from other cameras to the reference view

(c) with different SNR in two views

Fig. 1. Pose estimation results of the synthetic data.

form $\begin{bmatrix} C_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & C_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & C_N \end{bmatrix}$ where $\mathbf{0}$ is the 4×4 zero matrix, and $C_i, i = 1, \dots, N$ is the 4×4 covariance matrix for each camera. Then BLUE is

$$\tilde{\mathbf{s}} = (\mathbf{I}^T \mathbf{C}^{-1} \mathbf{I})^{-1} \mathbf{I}^T \mathbf{C}^{-1} \mathbf{s} = (\sum_{i=1}^{N} C_i^{-1})^{-1} \sum_{i=1}^{N} C_i^{-1} s_i \tag{25}$$

which provides the optimal fusion of the distributed solution. If the covariance matrix of the noise for each camera is identical, i.e. $C_i = C$, for $i = 1, 2, ..., N$, then the BLUE estimate is given by

$$\tilde{\mathbf{s}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i \tag{26}$$

4. EXPERIMENTAL RESULTS

We demonstrate the noise robustness of our algorithm with synthetic points, and the results are shown in Fig. 1, where (a) is especially for one-view case, and (b), (c) for multi-view case. Here we use 20 points and run 200 times.

In Fig. 1(a), we compare our Sylvester's Equation Algorithm (SEA) with the Least-Squares Method (LSM). The LSM is a method without considering the orthogonality constrains of a rotation matrix, and it deviates greatly as a high noise level presents. In Fig. 1(b), pose estimates from another two views, $R^o = 45(degrees)$ and $R^o = 90(degrees)$, are projected to the reference coordinate system, and no obvious difference is found with various views. Especially, no optimal viewpoint is found to give better results in our experiments. In Fig. 1(c), we fix SNR$= 20dB$ for the feature points in the first view, and decrease the SNR in the other view ($R^o = 90(degrees)$) from $20dB$ to $0dB$. BLUE has larger errors than the red curve, and it seems better to directly use the estimates from the first camera. But in real applications, we usually cannot distinguish which camera is better, and it will also change with time and the relative positions to the object. BLUE can improve the overall estimate from all observation of all cameras. More cameras generally mean more computation load. However, the increase in computation load is linear in our system.

5. CONCLUSION

This paper focuses on 3D pose estimation from multiple views. We use a solution to Sylvester's equation to estimate pose for each camera, and obtain a distributed solution which is fast and accurate. The best linear unbiased pose estimation is found to improve the results from one camera or estimate pose from a virtual camera.

6. REFERENCES

[1] L. Quan and Z. Lan, "Linear n-point camera pose determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 774–780, 1999.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] R. M. Haralick, C.-N. Lee H. Joo, V. G. Vaidya X. Zhuang, and M. B. Kim, "Pose estimation from corresponding point data," *IEEE Trunsactions on Systems, Man and Cybernetics*, vol. 19, no. 6, pp. 698–700, 1989.

[4] C F. Van Loan, "The ubiquitous kronecker product," *Journal of Computational and Applied Mathematics*, vol. 123, pp. 85–100, 2000.

[5] Q. T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *International Journal of Computer Vision*, vol. 17, no. 1, pp. 43–75, 1996.

[6] C. Rother and S. Carlsson, "Linear multi view reconstruction and camera recovery using a reference plane," *International Journal of Computer Vision*, vol. 49, no. 2-3, pp. 117–141, 2002.

[7] C. Fermuller and Y. Aloimonos, "Observability of 3d motion," *International Journal of Computer Vision*, vol. 37, no. 1, pp. 43–62, 2000.

[8] P. Baker, C. Fermuller, Y. Aloimonos, and R. Pless, "A spherical eye from multiple cameras (makes better models of the world)," in *IEEE Computer Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 576–583.

[9] J. M. Frahm, K. Koser, and R. Koch, "Pose estimation for multi-camera systems," in *Proceedings of DAGM*, 2004.

[10] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy, "Object pose: The link between weak perspective, paraperspective, and full perspective," *International Journal of Computer Vision*, vol. 22, no. 2, pp. 173–189, 1997.