

Extended Kalman Filter versus Newton-Lowe's Method for Robot Pose Estimation

Mohammad Ehab Ragab

Department of Informatics, Electronics Research Institute, Giza, Egypt

Article history

Received: 18-07-2015

Revised: 09-09-2015

Accepted: 29-10-2015

E-mail: mehab@hotmail.com

Abstract: In this work, we study the pose estimation problem of an autonomous mobile robot. Particularly, we compare the Extended Kalman Filter (EKF) to Lowe's method based on the iterative Newton's method for solving a system of nonlinear equations. Although the EKF is recursive which renders it suitable for the real-time problem at hand, Lowe's method has much less dimensionality. This is the motivation for comparing both approaches. We have used the stereo information for obtaining the 3-D structure and outlier rejection. This has provided an opportunity to weigh feeding both algorithms with single measurements (from one camera) against feeding them with pair measurements (from the stereo pair). We have studied the effects of using three ranges of the number of features and the longevity on the accuracy of the obtained pose parameters. Moreover, we have investigated the impact of the number of iterations on the accuracy of Lowe's method. An extensive set of simulations as well as real experiments using various motion patterns have been conducted. The main finding of this work is that Lowe's method (due to its low dimensionality) is much faster with approximately the same accuracy. Besides, it can recover from a situation which is close to singularity. On the other hand, the EKF makes better use of multiple camera measurements which allows a sustained performance even if one camera is off or occluded.

Keywords: Robot Navigation, Pose Estimation, Stereo, Extended Kalman Filter, Newton's Method

Introduction

The Pose estimation is a classic problem of computer vision. It solves for both the location and orientation (rotation) and can be classified into two types. The first is model-based, when the pose of an object in the scene is sought. The second is vision-based, when we are after the pose of camera or its mobile platform. The applications include obstacle detection (Panich, 2010a) and real time tracking of human face and gesture (Arulananth *et al.*, 2014). Additionally, pose estimation is indispensable for humanoid applications, autonomous robots, intelligent vehicles and man-machine interaction.

Pose estimation is interconnected with another classic computer vision problem, structure from motion. On one hand, to obtain the three-dimensional (3-D) structure of a number of features in the scene, we need to know their pose (or the camera pose) across multiple views. On the other hand, the pose is estimated according to the structure of features tracked from frame-to-frame. Bundle

Adjustment (BA) is the maximum likelihood solution for obtaining both pose and structure (Triggs *et al.*, 2000). It is a global optimization technique that aims at reducing the errors between the 2-D measurements and their corresponding 3-D features of the model. However, it is iterative and requires a good initialization. Above all, the whole set of features across the sequence of all frames should be fed simultaneously into the algorithm. This not only increases the dimensionality of BA, but renders it unsuitable for real-time applications as well.

In contrast, if our aim is to estimate the real-time pose of a mobile robot, we need to rely on recursive techniques working frame-to-frame. An optimal recursive linear estimator is the Kalman Filter (KF). However, to deal with the camera projective distortion, the Extended Kalman Filter (EKF) with the Jacobian of derivatives should be used (Chiuso *et al.*, 2002). Multiple cameras were used with the EKF for estimating the pose of a mobile robot in (Ragab *et al.*, 2007; 2008; Ragab and Wong, 2010). While in (Panich, 2010b), the

indirect KF was used (in simulations) to obtain the position of a mobile robot.

Another different approach is the renowned Newton's iterative method for solving systems of nonlinear equations. Lowe (1991) justified its use for model-based pose estimation in 2-D images. According to his work, although the projection from 3-D to 2-D is a nonlinear operation, it is a smooth and well-behaved transformation. Trucco and Verri (1998) derived the partial derivatives of the 2-D measurements with respect to the pose parameters. Their implementation was used in (Chang and Wong, 2005). In fact, the work in (Chang and Wong, 2005) extended the approach to obtain the structure besides the model-based pose. They initialized the structure orthographically (as if all 3-D features lied on a plane with a known constant depth). Accordingly, they estimated the pose iteratively. In the second iterative stage, the structure was refined using the estimated pose. The loop continued until the residual error fell below a certain threshold. However, the orthographic assumption required that the depth of the object to be considerably smaller than the distance between the object and camera (like the case they used of a flask on a turntable). It deserves mention that the Newton's method was used in computer vision in alternative ways. For example, Li and Hartley (2005) attempted to perform feature matching and pose estimation simultaneously using an alternating Newton iteration method. The matching was formulated as a nearest-matrix approximation problem and the orthogonality was made of. Additionally in (Baumann *et al.*, 2004), obtaining the pose was formulated as an estimation task for essential matrices. The problem was reformulated as a constrained optimization for a time-varying family of cost functions. Then, a Newton-type path following method was applied to asymptotically track the minima of the cost functions.

For more clarity, we will consider only the Lowe's application of the Newton's method as in (Lowe, 1991) and the implementations based on it as (Trucco and Verri, 1998; Chang and Wong, 2005). Additionally, from now on, we will refer to it as Lowe's method. The localization based on Lowe's method was combined with landmark-localization for better results in (Chen *et al.*, 2005). Both the KF and Lowe's method were used in (Saeedi *et al.*, 2003). The former was used to update the structure, while the latter was utilized in motion estimation.

In this study, we compare the EKF to Lowe's method for obtaining the pose of a mobile robot in an unknown scene. The motivation behind this is that although the latter is iterative, it has a lower dimensionality compared to the former recursive filter. Additionally, we study the effect of the number of features considered in both approaches (using three ranges). On one hand, the more the number of features, the more accurate the estimation in least-squares sense. On the other, as the number of features increases, the dimensionality increases as well

as the effect of radial distortion. Moreover, we investigate how the accuracy and speed are affected by the number of iterations of the Lowe's method. Since, we use a stereo pair of cameras for obtaining the structure and rejecting the outliers, we compare feeding the pose estimation algorithms with the measurements of only the reference camera to the double measurements of the stereo pair. To ensure the validity of our studies, we have carried them under different motion patterns.

Method

The camera layout used is shown in Fig. 1, where camera 1 and camera 2 form a stereo pair. The global coordinate system is attached to camera 1 (reference camera) at the initial position (i.e. at *frame 0*). While Camera 2 has its center displaced from camera 1 by the vector D_2 and is rotated by the rotation matrix, R_2 . During the motion, at any general frame (*frame j*), camera 1 is rotated by the rotation matrix, R_j , with its center translated by the vector d_j with respect to the reference coordinate system. Our task is to estimate the pose (d_j and R_j), or equivalently to find its six parameters (translation components in direction of the coordinate axes: t_{xj} , t_{yj} and t_{zj} (m)) and (rotation angles around the coordinate axes: α_j , β_j and γ_j (rad)). The camera coordinates for camera 1 is given by:

$$P_{ij} = g(R_j)(M_i - d_j) \quad (1)$$

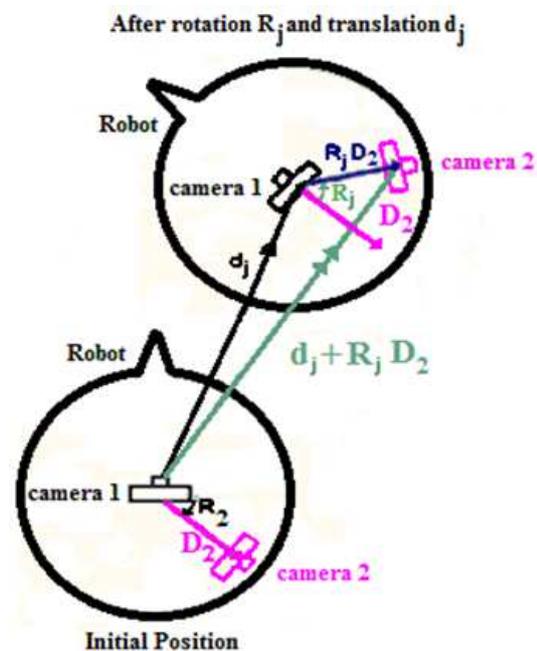


Fig. 1. The plant equation which relates the current state space the initial pose of each camera and the poses at a general frame, j . The pose we seek is rotation R_j and translation d_j (rotation R_2 is exaggerated for clarity)

Where:

M_i = A (3×1) vector defining the 3-D location of the feature i (seen by the camera) with respect to the reference coordinate system (m)

$g(R_j)$ = A function of the camera rotation (e.g. the rotation itself or its transpose (Ragab and Wong, 2009)).

Similarly, the camera coordinates of camera 2 is given by:

$$P_{ij2} = g(R_j)(M_i - d_j - R_j D_2) \quad (2)$$

According to Equations 1 and 2, the 2-D measurements in images (for camera 1 and camera 2 respectively) should be:

$$\left[\frac{K_1(1) \times P_{ij}(1)}{K_1(3) \times P_{ij}(3)}, \frac{K_1(2) \times P_{ij}(2)}{K_1(3) \times P_{ij}(3)} \right]$$

and

$$\left[\frac{K_2(1) \times P_{ij2}(1)}{K_2(3) \times P_{ij2}(3)}, \frac{K_2(2) \times P_{ij2}(2)}{K_2(3) \times P_{ij2}(3)} \right] \quad (3)$$

Where:

K_1 and K_2 = (3×3) matrices encoding the intrinsic parameters of each camera e.g. focal lengths and image centers (pixel)

$K_1(1)$ = First row of the matrix and $P_{ij}(1)$ is the first component of the vector (second and third rows and components are indicated as well with respective numbers)

EKF Implementation

The state space vector at frame j , s_j , consists of the pose parameters and their derivatives (velocities) in the form:

$$S_j = [t_{xj} \dot{t}_{xj} t_{yj} \dot{t}_{yj} t_{zj} \dot{t}_{zj} a_j \dot{a}_j \beta_j \dot{\beta}_j \gamma_j \dot{\gamma}_j]^T \quad (4)$$

where, The superscript T transforms the row to a column vector and other elements are described above.

The plant equation which relates the current state space vector s_j to the previous one s_{j-1} and the plant noise n_j assumed to be Gaussian is:

$$S_j = AS_{j-1} + n_j \quad (5)$$

where, A is a (12×12) matrix whose main diagonal elements are ones. The odd rows have a τ (equal to the time step between frames) just to the right of the main diagonal. In this way, a robot uniform motion of constant speeds is assumed.

The measurement equation relating the 2-D pixel locations of image features I_j and the state measurement relation $h(s_j)$ (given by Equation 3 above) assuming a Gaussian distribution for the measurement noise η_j :

$$I_j = h(s_j) + \eta_j \quad (6)$$

For each frame, the EKF predicts the state space vector based on the previous one and updates it (enhancing the prediction) based on the measurements and the calculations of the Jacobian of the state measurement relation with respect to the twelve elements of the state space vector. The main time-consuming step is calculating the Kalman gain since it includes an inversion of $((2 \times N) \times (2 \times N))$ matrix, where N is the number of features fed into the filter. More details about the EKF implementation can be found in (Chiuso *et al.*, 2002).

Lowe's Method Implementation

Here, we focus on the pose parameters (unlike Equation 4):

$$q_j = [t_{xj} t_{yj} t_{zj} a_j \beta_j \gamma_j]^T \quad (7)$$

The algorithm can be summarized as follows:

- At *frame 0*, all pose parameters are zeros as mentioned above, 2-D features are obtained and matched for the stereo pair (more details are below). The 3-D structure is acquired using triangulation
- Move to next frame, initialize rotation and translation to that of the previous frame, construct the measurement relation $h(q_j)$ (given by Equation 3 above)
- Compute the residual (difference between actual 2-D features and the measurement relation):

$$\varepsilon_j = I_j - h(q_j) \quad (8)$$

- Calculate the Jacobian, ϕ_j , which is a $((2 \times N) \times 6)$ matrix containing the partial derivatives of $h(q_j)$ with respect to the six pose parameters
- Solve the system of equations: □

$$\phi_j \Delta_j = \varepsilon_j \quad (9)$$

Where:

Δ_j = Vector of required variations in pose parameters to minimize the residual of Equation 8:

$$\Delta_j = [\delta t_{xj} \delta t_{yj} \delta t_{zj} \delta a_j \delta \beta_j \delta \gamma_j]^T \quad (10)$$

The standard solution of Equation 9 in least-squares sense is given by:

$$\Delta_j = (\phi_j^T \phi_j)^{-1} \phi_j^T \varepsilon_j \quad (11)$$

where, $()^{-1}$ is the matrix inversion operator.

- f) Update the pose using Δ_j obtained at Equation 11 (translations are added, while angles form a rotation matrix which is multiplied). Go to step (c), iterate to step (f) for a specified number of iterations or until the residual falls below a certain threshold
- g) While the tracked number of features is more than a certain threshold, go to step (b) (to process all frames till the end of sequence). Otherwise, go to step (a) to acquire new features to track (using the current frame and its pose)

In fact, the steps (a), (b) and (g) are common for Lowe's method and the EKF implementation (which is written concisely above). Additionally, the matrix inverted in Equation 11 is of size (6×6) which is much lower than that of the EKF implementation $((2 \times N) \times (2 \times N))$. Moreover, the number of required iterations (step (f)) is expected to be low since we always have the pose of the previous frame as a good initialization of the current.

Feature Management

The 2-D features mentioned above are small windows of pixels within the image frames characterized by having a corner property (high intensity gradients in both directions). For each camera, the features are obtained and tracked using the Kanade-Lucas-Tomasi (KLT) feature tracker. The stereo matches are based on a cross-correlation measure. Additionally, matches violating epipolar constraints are filtered out as outliers. The locations of features in the 3-D space are obtained using the triangulation. The features fed to the algorithms are chosen to be as evenly distributed as possible around the center of projection of each image. Accordingly, the set of features may vary from frame to frame. When the number of tracked features falls under a certain threshold, a new set of fresh features is obtained using the stereo matching as mentioned above. The number of features fed to the algorithms is a critical factor for the performance. As this number increases, the accuracy increases in least-squares sense. However, both the dimensionality of the problem and the effect of radial distortion will increase. Therefore, the performance will be checked using different numbers of features. As mentioned above, stereo is used on demand for obtaining the 3-D structure and to reject the outliers (in each frame). However, we have two 2-D measurements for each feature (one by each camera). Feeding the two measurements into the pose estimation algorithms will unfortunately double the dimensionality. Nevertheless, the question is: would this be justified by an increase in accuracy? The question will be answered below.

Experiments

Simulations

A stereo pair was put on a robot moving with random translations (t_x , t_y and t_z) and random rotation angles (α , β and γ) in the directions of and around the x , y and z axes respectively. Initially, the center of camera 1 (reference camera) coincided with the origin of the coordinate system with zero angles of rotation. The translations were taken randomly from ± 0.005 to ± 0.0225 m and the rotation angles were taken randomly from ± 0.005 to ± 0.03 rad. Both cameras had a 6 mm focal length and (640×480) resolution with a 0.1 m baseline in-between. A random noise was added to each feature point with a normal distribution of zero mean and a 0.5 pixel standard deviation. The motion took place inside a spherical shell whose outer radius was 1 m and inner was 0.667 m having 10,000 feature points distributed randomly within. A sequence of 100 frames was taken simultaneously by each camera. We ran the simulations 1000 times to estimate the pose using the EKF (measurement of a single camera and of the pair), the Lowe's method having 10, 20 and 30 iterations (single and pair measurements as well).

Real Experiments

We carried out the real experiments using a pair of calibrated Canon Power Shot G9 cameras with resolution (1600×1200) . The cameras were put parallel atop the robot used with the baseline equal to 14 cm. A sequence of more than 200 frames was taken simultaneously by each camera in an ordinary lab scene. The motion of the robot followed various patterns: a pure translation, a pure rotation and mixed rotation and translation. In addition to the eight variations of the algorithms we had in simulations, we tested the use of three sets of features. For the first set, we used all available features for each camera and did not acquire new features until their number dropped to 200. The second set used a number of features starting from 200 down to 140, while the third dealt with 100 down to 70 features. We considered also the effect of the longevity of features upon the accuracy and speed. We dealt with the sequence of frames either as one section or as five sections with fresh features for each new section.

Results

Average absolute errors for the pose in simulations are shown in Table 1. Each method is tested with various settings for the real experiments (as shown in Fig. 2).

Figure 3 compares the best overall performance for each method in real experiments under all studied motion patterns. The ground truth of Fig. 2 and 3 was obtained from the computer steering the robot. The timing information for the real experiments is shown in Table 2 (obtained using MATLAB on Intel Core i7 CPU with 1.73 GHz and 4GB RAM).

Table 1. Average absolute error of pose values per frame for simulations (rad/m)

Method	α	β	γ	t_x	t_y	t_z
EkfP ^a	0.0071	0.0195	0.0027	0.0207	0.0072	0.0057
EkfS ^b	0.0122	0.0227	0.0038	0.0238	0.0116	0.0066
LPit10 ^c	0.0398	0.0163	0.0111	0.0155	0.0355	0.0077
LPit20	0.0415	0.0163	0.0122	0.0152	0.0362	0.0083
LPit30	0.0439	0.0163	0.0138	0.0153	0.0378	0.0094
LSit10	0.0122	0.0091	0.0028	0.0089	0.0116	0.0028
LSit20	0.0122	0.0091	0.0028	0.0089	0.0116	0.0028
LSit30	0.0122	0.0091	0.0028	0.0089	0.0116	0.0028

- a. “Ekf” denotes using the EKF algorithm, “P” denotes measurements of the stereo pair
- b. “S” denotes taking measurements from a single camera (camera 1)
- c. “L” denotes Lowe’s method, “it10” denotes 10 iterations

Discussion

For the simulations, Table 1, the best overall performance is verified by Lowe’s method with single measurements (LS). Next is the EKF with pair measurements (EkfP), then come the EKF with single measurements (EkfS) and Lowe’s method with pair measurements (LP). Since Lowe’s method has a least-squares solution (Equation 11 above), it performs well with single measurements (where the measurements are distributed around the center of projection of one camera). In contrast, it is well-known that the EKF estimation capabilities improve when having more measurements (from both cameras). However, EkfS could sometimes verify the same accuracy as LS (for α and t_y). For LS, using ten iterations is adequate for convergence. Additionally, increasing the number of iterations for LP might even degrade the performance slightly. The reason for this is that LP is not in harmony with the least-squares solution as mentioned above.

Figure 2, for the real experiments depicts nearly the same results as the simulations. The aim of this figure is to find out the setting which verifies the best overall performance for each method. It is obvious, that dividing the sequence into five sections results in more accurate pose parameters than having only one. This proves that having fresh features more frequently reduces the tracking errors caused by the projective distortion affecting the feature windows. For most methods, having the range of feature numbers as “200-140” is the most accurate. This is the medium range which avoids the problems encountered by the two others. The smallest range, “100-70”, has all features in a small region around the image center of projection which is likely to cause the small aperture problem. In contrast, the largest range, “All-200”, includes all available features extending to the frame border which can suffer a lot of the radial distortion. Having ten iterations for Lowe’s method is adequate. Practically, fixing all other settings, the curves of 10, 20 and 30 iterations coincide. To use space

efficiently, we show in Fig. 2 four pose parameters and one motion pattern for each method. However, this represents the results well since each method has the same displayed behavior for the other two pose parameters (angles α and γ which are kept zeros throughout the real experiments) and for all motion patterns (which is elucidated in Fig. 3).

The best settings for all methods and motion patterns are compared in Fig. 3. For the pure translation pattern, “EkfP” is the closest to the ground truth which shows the accurate prediction of the filter in uniform translational motions. Then come very close both of “LS” and “EkfS” which nearly coincide. The angle β increases gradually up to 0.1 rad for most methods which indicates a possible slight slipping of the robot wheels. “LP” deviates for most pose parameters for the reason mentioned above in this section. For the pure rotation pattern, “LS” verifies the best performance which agrees with the results obtained so far. Additionally, the deviation of both “LS” and “LP” near the end of sequence remains limited. What might be unexpected is the divergence of both “EkfP” and “EkfS” near the middle of the sequence. This behavior is explained by having the blank-wall effect throughout tens of frames starting from the initial frame of the sequence. A large portion of each frame is occupied by a carton and a table surface without reliable features to track (as shown in Fig. 4). This leads to a singularity of the matrix inversion when calculating the Kalman gain. On the other hand, Lowe’s method could survive due to its low dimensionality and the collectivity of the least-squares (as shown in Equation 11). When starting the sequence after the blank-wall effect disappears (e.g. from *frame 70*), both “EkfP” and “EkfS” perform well (lowest row of Fig. 4). For the mixed motion pattern, all methods perform well for both t_z and β which dominate the change in pose parameters. “LP” deviates for both t_x , and t_y while the other methods remain close to the ground truth. “LS” and “EkfS” verify the best performance for t_x while “EkfP” is the most accurate for t_y .

From the timing information, in Table 2, we deduce that in most cases “sec5” takes longer time than “sec1”. Starting a new section repeatedly leads generally to having more features throughout each section with more calculations. This is especially obvious for the highest range of feature number “All-200”. For “EkfP” and “EkfS”, increasing the range of feature numbers raises dramatically the time which is dominated by a $((2 \times N) \times (2 \times N))$ matrix inversion as explained above. The complexity is approximately cubic which is clear when comparing the range “100-70” to its approximate double “200-140”. For “LS” and “LP”, fixing the number of iterations, the time averages remain close to each other throughout most of the ranges of feature numbers. The reason for this is explained in section 2.2 above. Having the highest range of features “All-200” with “sec5” is an exception which is clarified above in this paragraph.

Table 2. Average time per frame for real experiments (s)

Method setting	EkfP ^a	EkfS ^b	it10 ^d	LP ^c it20	it30	it10	LS it20	it30
100-70sec1 ^e	0.0429	0.0067	N/A ^g	N/A	0.0258	0.0043	0.0085	0.0177
100-70sec5 ^f	0.0305	0.0074	N/A	N/A	0.0175	0.0075	0.0080	0.0120
200-140sec1	0.2154	0.0373	0.0101	0.0193	0.0279	0.0055	0.0099	0.0181
200-140sec5	0.2425	0.0591	N/A	N/A	0.0276	0.0100	0.0075	0.0109
All-200sec1	3.4542	0.7529	0.0100	0.0190	0.0284	0.0044	0.0097	0.0142
All-200sec5	5.5511	0.9309	0.0113	0.0299	0.0802	0.0146	0.0328	0.0584

- a. Ekf^a denotes using the EKF algorithm, “P” denotes measurements of the stereo pair
- b. “S” denotes taking measurements from a single camera (camera 1)
- c. “L” denotes Lowe’s method
- d. “it10” denotes 10 iterations
- e. “100-70” denotes number of features per camera, “sec1” denotes taking all sequence of frames as one section
- f. “sec5” denotes dividing the sequence into five sections and starting each with new features
- g. “N/A” denotes not carrying out 10 and 20 iterations since even 30 iterations did not converge

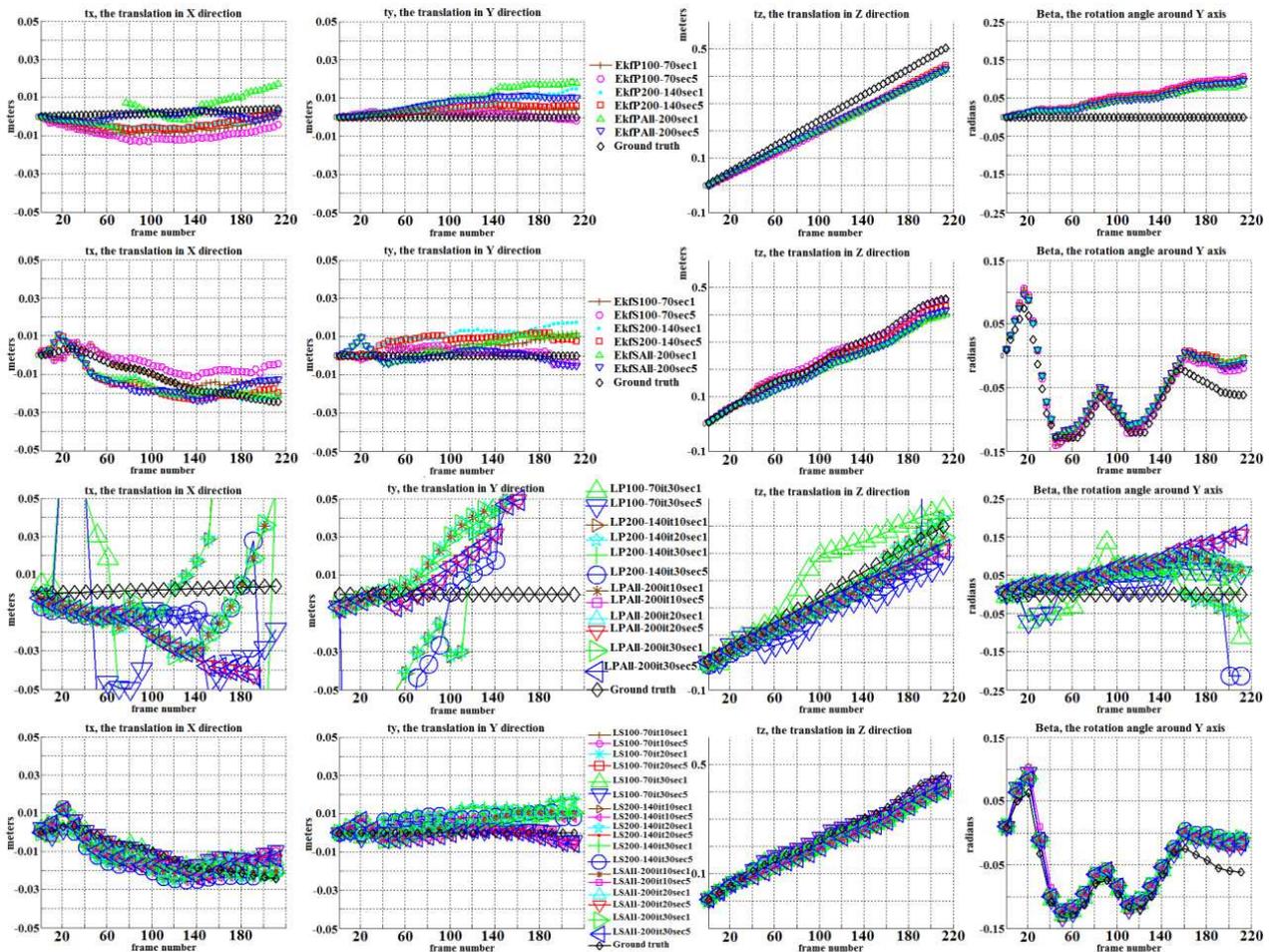


Fig. 2. Finding out settings of best overall performance. Four parameters are compared to the ground truth (t_x , t_y , t_z , and β the only angle varied during motion). Upmost row shows performance of EKF with pair measurements “EkfP” under pure translation pattern. Numbers e.g. “100-70” show the range of features used per camera, and “sec1” denotes having the sequence as one section in contrast to 5 sections “sec5”. Second row shows performance of EKF with single measurements “EkfS” under mixed motion pattern. Third row shows performance of Lowe’s method with pair measurements “LP” under pure translation motion pattern. The number of iterations is indicated e.g. “it10”. Smallest markers are used for “it10”, medium size is used for “it20”, and largest size is used for “it30”. Lowest row shows performance of Lowe’s method with single measurements “LS” under mixed motion pattern. Legends are put in the middle of rows with fewer entries for “LP” than “LS” (when “it30” does not converge, lower iterations are not carried out)

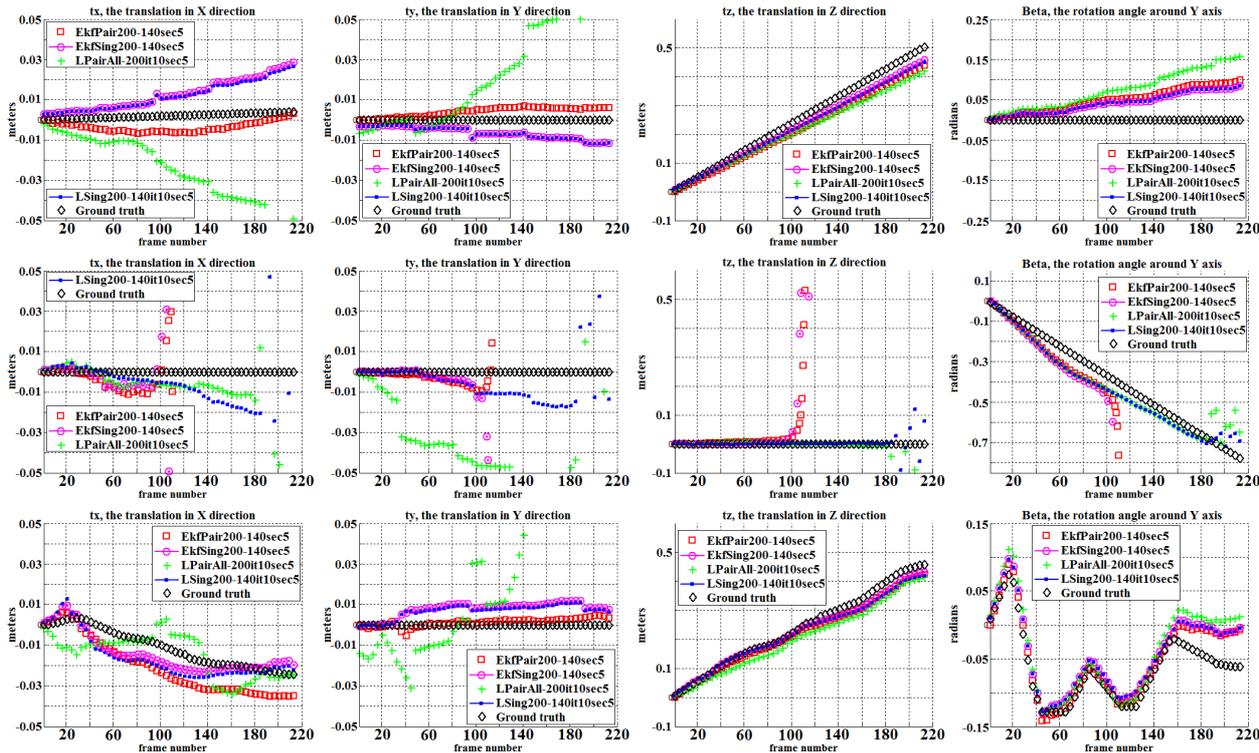


Fig. 3. Comparing all methods (setting of best overall performance). Four parameters are compared to the ground truth (t_x , t_y , t_z , and β the only angle varied during motion). Upmost row shows the performance under pure translation motion pattern. The second row shows the performance under pure rotation motion pattern while the lowest row belongs to the mixed motion pattern. Numbers e.g. “200-140” show the range of features used per camera, and “sec5” denotes dividing the sequence into 5 sections. “EkfS”: EKF with single measurements, “EkfP”: EKF with pair measurements, “LP”: Lowe’s method with pair measurements, and “LS”: Lowe’s method with single measurements. “it10”: using ten iterations

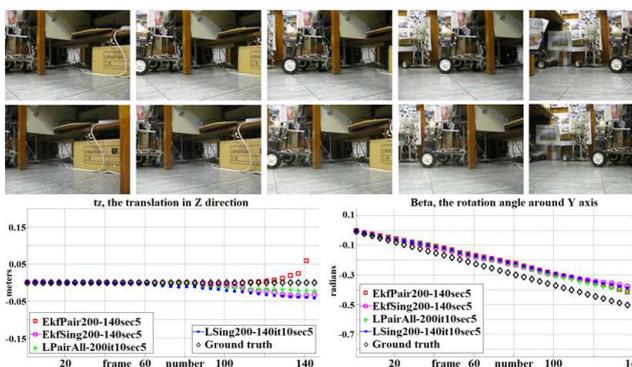


Fig. 4. Pure rotation, first and second rows: Frames 1, 25, 50, 100 and 200 for stereo. Large areas till frame 50 do not have enough features to track. Lowest row: Starting at frame 70, both “EkfP” and “EkfS” perform well as shown for t_z and β

Fixing the range of feature numbers, the time increases nearly linearly with the number of iterations (not exactly since some calculations are made outside the loop). Since “LP” considers double the measurements of “LS”, it generally requires double the time when all other settings are fixed. This is clear for the highest range of feature numbers “All-200” and “sec1”. We conclude this

section by comparing the time for the most accurate settings (shown in Fig. 3). Although “EkfP”, “EkfS” and “LS” are close in accuracy to each other, they have wide variations in speed. “LS” is the fastest of them all with a capability of handling 100 frames/s. “EkfS” can handle 16 frames/s while “EkfP” can handle only four frames/s. We should mention that there is a room for improvement using a faster processor and compiler (compared to that mentioned in section 4). Additionally, the time of feature detection and tracking is excluded. However, this task can be handed to a parallel processing unit.

Conclusion

In this study, we have probed the pose estimation of a moving robot in an unknown indoor scene. The stereo information is used for obtaining the 3-D structure and feature outlier rejection. We have compared Lowe’s method based on iterative Newton’s method for solving a system of equations to the EKF. Both have nearly the same accuracy. However, Lowe’s method has a lower dimensionality which results in more speed and more immunity against some singular settings (e.g. a blank-wall effect across large areas of the frames). We have found

that ten iterations are adequate for Lowe's method to converge. More speed can be gained by combining an error threshold with ten iterations as a maximum. In contrast to Lowe's method, the EKF is scalable with respect to the measurements. Taking the measurement of the same 3-D feature twice from the stereo pair increases the accuracy. Therefore, the EKF approach benefits from adding more cameras and can tolerate any of them being off or occluded. However, this comes at the expense of more processing time. An important finding of this work is that it is not always beneficial to increase the number of features fed to the pose estimation approaches. A range of features that avoids both the small aperture problem and the radial distortion would be more beneficial. Additionally, having moderate longevity of features by dividing the motion sequence into a suitable number of sections avoids much projective distortion of feature windows and leads to a more accurate estimation of pose parameters.

Acknowledgment

The author thanks the anonymous reviewers.

Ethics

The author has approved the manuscript.

References

- Arulananth, T.S., T. Jayasingh and S. Ravi, 2014. Tracking the position of multiple human gestures using contour mapping algorithm. *J. Comput. Sci.*, 10: 1994-2005. DOI: 10.3844/jcssp.2014.1994.2005
- Baumann, M., C. Lageman and U. Helmke, 2004. Newton-type algorithms for time-varying pose estimation. *Proceedings of Intelligent Sensors, Sensor Networks and Information Processing Conference*, Dec. 14-17, IEEE Xplore Press, Australia, pp: 155-160. DOI: 10.1109/ISSNIP.2004.1417454
- Chang, M.M.Y. and K.H. Wong, 2005. Model reconstruction and pose acquisition using extended Lowe's method. *IEEE Trans. Multimedia*, 7: 253-260. DOI: 10.1109/TMM.2005.843344
- Chen, Z., P. Pe, J. McDermid and N. Pears, 2005. Two-stage visual localisation: Landmark-based pose initialisation and model-based pose refinement. *Proceedings of International Conference on Intelligent Robots and Systems*, Aug. 2-6, IEEE Xplore Press, Canada, pp: 150-156. DOI: 10.1109/IROS.2005.1545555
- Chiuso, A., P. Favaro, H. Jain and S. Soatto, 2002. Structure from motion causally integrated over time. *IEEE Trans. Pattern Recognition Machine Intelligence*, 24: 523-535, DOI: 10.1109/34.993559
- Li, H. and R. Hartley, 2005. Feature Matching and Pose Estimation Using Newton Iteration. In: *Lecture Notes on Computer Science 3617*, Roli, F. and S. Vitulano, (Eds.), Springer-Verlag, Berlin, Heidelberg, ISBN-10: 3540288694, pp: 196-203.
- Lowe, D.G., 1991. Fitting parameterized three-dimensional models to images. *IEEE Trans. Patt. Analysis Machine Intelligence*, 13: 441-450. DOI: 10.1109/34.134043
- Panich, S., 2010a. Method of object detection for mobile robot. *J. Comput. Sci.*, 6: 1151-1153. DOI: 10.3844/jcssp.2010.1151.1153
- Panich, S., 2010b. Indirect Kalman filter in mobile robot application. *J. Math. Statistics*, 6: 381-384. DOI: 10.3844/jmssp.2010.381.384
- Ragab, M.E., K.H. Wong, J.Z. Chen and M.M.Y. Chang, 2007. EKF based pose estimation using two back-to-back stereo pairs. *Proceedings of the 14th International Conference on Image Processing*, Sept. 16-19, IEEE Xplore Press, USA, pp: 137-140. DOI: 10.1109/ICIP.2007.4379540
- Ragab, M.E., K.H. Wong, J.Z. Chen and M.M.Y. Chang, 2008. EKF pose estimation: How many filters and cameras to use? *Proceedings of the 15th International Conference on Image Processing*, Oct. 12-15, IEEE Xplore Press, USA, pp: 245-248. DOI: 10.1109/ICIP.2008.4711737
- Ragab, M.E. and K.H. Wong, 2009. Rotation within camera projection matrix using Euler angles quaternions and angle-axes. *Int. J. Robotics Automation*, 24: 312-318. DOI: 10.2316/Journal.206.2009.4.206-3174
- Ragab, M.E. and K.H. Wong, 2010. Multiple nonoverlapping camera pose estimation. *Proceedings of the 17th International Conference on Image Processing*, Sept. 26-29, IEEE Xplore Press, Hong Kong, pp: 3253-3256. DOI: 10.1109/ICIP.2010.5651178
- Saeedi, P., D.G. Lowe and P.D. Lawrence, 2003. 3D Localization and tracking in unknown environments. *Proceedings of International Conference on Robotics and Automation*, Sept. 14-19, IEEE Xplore Press, Taiwan, pp: 1297-1303. DOI: 10.1109/ROBOT.2003.1241771
- Triggs, B., P.F. McLauchlan, R.I. Hartley and A. Fitzgibbon, 2000. Bundle Adjustment-A Modern Synthesis. In: *Vision Algorithms: Theory and Practice*, Lecture Notes on Computer Science 1883, Triggs B., A. Zisserman and R. Szeliski, (Eds.), Springer-Verlag, Berlin, Heidelberg, ISBN-10: 9783540679738, pp: 298-372.
- Trucco, E. and A. Verri, 1998. *Introductory Techniques for 3-D Computer Vision*. 1st Edn., NJ: Prentice-Hall, Englewood Cliffs, ISBN-10: 0132611082, pp: 152.