ICIP 2007, the 14th IEEE International Conference on
Image Processing in San Antonio, Texas, September 16-19, 2007.

EKF BASED POSE ESTIMATION USING TWO BACK-TO-BACK STEREO PAIRS

M. E. Ragab, K. H. Wong, J. Z. Chen *

Computer Science & Engineering Department
The Chinese University of Hong Kong
Hong Kong

M. M. Y. Chang

Information Engineering Department
The Chinese University of Hong Kong
Hong Kong

ABSTRACT

In this work, we solve the pose estimation problem for robot motion by placing multiple cameras on the robot. In particular, we use four cameras arranged as two back-to-back stereo pairs combined with the Extended Kalman Filter (EKF). The reason for using multiple cameras is that the pose estimation problem is more constrained for multiple cameras than for a single camera. Back-to-back cameras are used since they provide more information. Stereo information is used in self initialization and outlier rejection. Different approaches to solve the long-sequence-drift have been suggested. Both the simulations and the real experiments show that our approach is fast, robust, and accurate.

Index Terms— Pose, EKF, multiple-cameras, stereo, drift

1. INTRODUCTION

To find the pose of an object is to get its position and orientation. It is a popular research problem, and is related to diverse areas such as: robotics, man-machine interaction, augmented reality (AR), and intelligent vehicle guiding [18]. Applications are abundant, for example, maintenance training by augmented reality [11], precise localization in industrial environments [20], and identifying large 3D objects [14].

In this work, we are interested in getting the ego-motion of a set of cameras atop a moving robot. As this problem needs to be solved in real time, we have to use recursive techniques such as the Kalman filter (KF) and the particle filter. Although the later is more advantageous in tracking continuous curves, such as hand contours, in dense visual clutter, it requires increasing the sample size and the computational cost to improve the performance [8]. However, KF and its variants such as the extended Kalman filter (EKF) are quite satisfactory in tracking feature corners among frames as in our case. In fact EKF or its iterated version (IEKF) has been used in diverse ways in the field of computer vision. For example, one filter is used for pose and 3D structure in [4], [17], [1], and [6]. Using one filter for both pose and structure guarantees that they are coupled however the length of the state space vector becomes large which may affect the filter stability [1]. Additionally, a separate filter is used for each 3D point in [3] and in [19] where another is used for the pose. This in fact improves speed but may lower the accuracy due to the decoupling of pose and structure.

Besides using multiple cameras in stereo rigs, they have been used in pose estimation primarily to resolve the bas-relief ambiguity [2]. In [5], [7], and [12], the multiple cameras are dealt with as a single generalized camera. Multiple cameras are used with KF or EKF, for example, in [15] and in [9] mainly as fixed cameras to estimate the pose of an object with a known CAD model.

In contrast, we use four cameras forming two stereo-pairs put back-to-back on a robot moving within the scene. The inputs to the system are the simultaneous frames taken by each camera, and the camera calibration. The output is the real time pose along a sequence of hundreds of frames. We use only one EKF for pose estimation. Whenever it is needed, the 3D structure of the features fed to the filter is calculated by triangulation based on the filter output which guarantees the coupling between pose and structure. The main contributions of our work are: (1) formulating the EKF implementation for the pose estimation of multiple moving cameras, (2) using a changeable set of features to avoid the effect of occlusion, and (3) comparing different approaches to solve the long-sequence-drift and using simple ways to reject the outliers. The rest of this paper includes: background, multiple camera model, proposed algorithm, simulations, real experiments, and is concluded by discussion and conclusions.

2. BACKGROUND

Using multiple cameras rather than a single camera is justified in [2], [12], and experimentally in [13]. Main reasons are that pose is much better constrained for multiple cameras, and the existence of ambiguous scenes. The back-to-back setting is motivated by the analysis of Fisher Information Matrix [12]. This work is an extension of [13]; the addition of two more cameras enables the system to use stereo information in self-initialization and outlier rejection.

To work in the Euclidean space, cameras need calibration.

*Supported by the Research Grant Council of HKSAR (Project No.: 4204/04E) and Faculty of Engineering, CUHK (Project Code: 2050350).

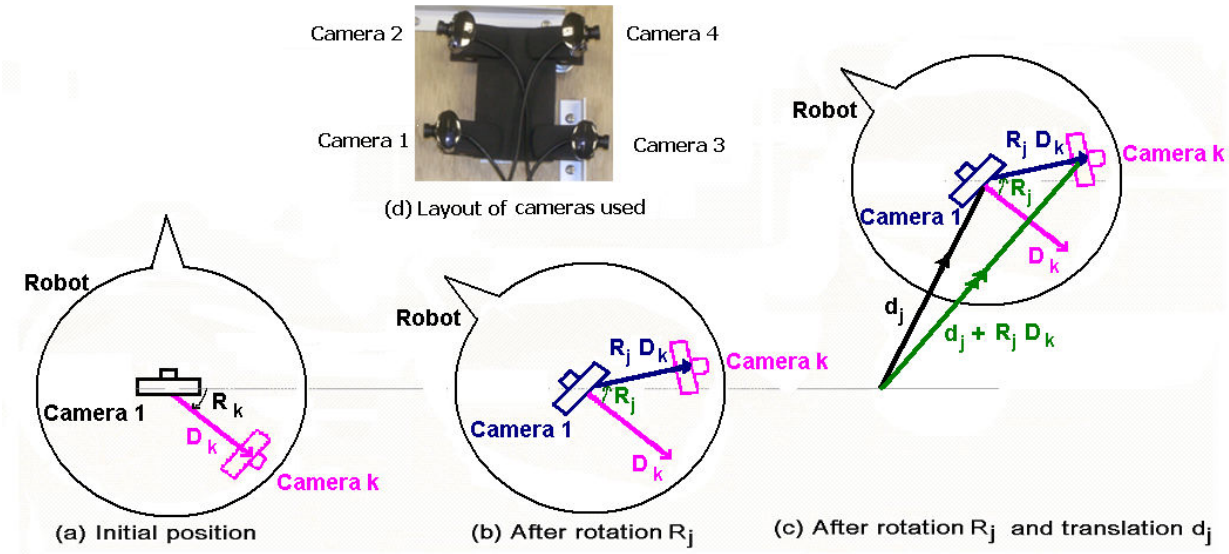

Fig. 1: Top-down view: effect of rotation and translation on the displacement of cameras (Camera *k* represents any of Camera 2, Camera 3, and Camera 4. Referred to the initial position of Camera 1, the rotation R_k and the translation D_k are fixed.)

Stereo calibration [21] can be used for stereo-pairs, or the fundamental matrix, F, can be obtained then the essential matrix, E, is decomposed to get the extrinsic parameters. In this case, the baseline between each stereo pair needs to be measured to resolve a scale factor. To relate the two stereo-pairs, any two back-to-back cameras can be externally calibrated using two parallel check-boards [13] or the 1-D calibration method [10].

3. MULTIPLE CAMERA MODEL

For the reference camera (camera 1), the camera coordinates, a (3×1) vector P_{ij}, of the 3D feature $M_i(3 \times 1)$ vector, at frame *j* is given by:

$$P_{ij} = R_j^T(M_i - dj) \tag{1}$$

Where R_j is the (3×3) camera rotation matrix at frame *j* (referred to the first frame), and d_j is the (3×1) camera translation vector at frame *j* (referred to the first frame). Any other camera, the k^{th} camera, which is rotated R_k, and translated D_k from the reference at the first frame has the following j^{th} frame camera coordinates of M_i:

$$P_{ijk} = R_k^T R_j^T(M_i - dj - R_j D_k) \tag{2}$$

In Fig.1, R_k, and D_k are shown for the k^{th} camera, while R_1 and D_1 are the identity matrix and the zero vector respectively (belonging to the first camera, reference position).

4. PROPOSED ALGORITHM

The core of this algorithm is the multiple-camera EKF. The state space vector:

$$s = [t_x\ \dot{t_x}\ t_y\ \dot{t_y}\ t_z\ \dot{t_z}\ \alpha\ \dot{\alpha}\ \beta\ \dot{\beta}\ \gamma\ \dot{\gamma}]^T \tag{3}$$

consists of the six pose parameters (Fig. 2), and their derivatives. The plant equation:

$$s_\tau = As_{\tau-1} + n_\tau \tag{4}$$

relates the current state s_τ to the previous, and the plant noise n_τ. The measurement equation:

$$I_\tau = h(s_\tau) + \eta_\tau \tag{5}$$

relates image features I_τ to the measurement noise η_τ and the state measurement relation:

$$h = \{\cdots, f_k[\cdots, \frac{P_{ijk}(1)}{P_{ijk}(3)}, \frac{P_{ijk}(2)}{P_{ijk}(3)}, \cdots], \cdots\}^T \tag{6}$$

using equations: (1), and (2) above and the focal lengthes f_k. These equations affect also the Jacobian calculation, $\partial h/\partial s$, in the EKF update step. More details can be found in [13].

Although the robot carrying the cameras is moving, they remain rigidly fixed to each other. Therefore, F is constant for each stereo-pair throughout the motion (assuming fixed intrinsic parameters). This fact is used to reject outliers. Initially features are matched between a stereo-pair and tracked from frame to frame of the same camera provided that they verify F of the stereo-pair (within 1.5 pixels of epipolar lines). The main steps of the algorithm are:

1. Find feature matches between each stereo-pair in the first frame then, triangulate them to get their 3D structure (with the first camera at the coordinates origin).
2. Track features to the second frame for each camera respectively. Knowing their 3D structure, get the pose of the first camera using Lowe's method ([16] with modifications of the Jacobian). Since we are close to the motion start, a few iterations (< 10) are enough. The aim of this step is to obtain the pose derivatives between the first two frames so as to start the EKF as accurately as possible (time required to set in work is minimized).
3. Track features to the next frame of each camera then, feed the measurements of a number of them (35-50 around the image center to verify the zero-average measurements [13]), their 3D structure, and the previous

state space vector and covariance to the multiple camera EKF. The output is the current state vector (required pose and derivatives), and the current state covariance.

4. Repeat step (3) to the end of the sequence. Each time use the output of the filter at the previous frame as input in the current frame together with the tracked features.
5. If the number of tracked features for any stereo pair drops below a certain threshold (35), return to the previous frame, go to step (1), and bypass step (2) since the state space vector of the previous frame was obtained from the filter. This step may introduce some drift (discussed below with suggested solutions).

5. SIMULATIONS

The set of cameras was moved randomly with translations from ± 0.005 to ± 0.015 meters and rotation angles ± 0.005 to ± 0.02 radians in the direction of and around the three axes inside a sphere centered at the origin with a one meter radius and 35,000 features disturbed randomly on its surface. Two stereo-pairs (as shown in Fig. 1) were formed by the four cameras with baselines ranging from 0.1 to 0.2 meters. All cameras have a 6 mm focal length and 640×480 resolution. Gaussian noise with zero mean and 0.5 pixel standard deviation was added to each image feature. A sequence of 100 frames was taken by each camera. Due to the motion randomness, the sequence should be divided into a number of sections (last step of the proposed algorithm above). For fair comparison, each section contained 10 frames. The results were accurate however we noticed some drift in the real experiments (section 6). Table 1 shows the average of 100 runs of absolute error in the six pose parameters for using EKF, IEKF only at the start of each new section (n.s.), IEKF for all frames, and EKF with drift compensation (cmp) described below. All absolute errors are given per frame in milli-(meters/radians). To get them in percentage, they should be compared to the average sum of absolute translations in one run (1 meter), and the average sum of absolute rotation angles (1.25 radians).

method	T_x mm	T_y mm	T_z mm	α m rad	β m rad	γ m rad
EKF	.963	.851	1.396	.659	.757	.551
IEKF n.s.	.888	.806	1.306	.528	.630	.508
IEKF all	.0643	.0752	.256	.0711	.0563	.287
EKF cmp	.838	.784	1.131	.641	.658	.576

Table 1: Average absolute error of pose values/frame (simulation)

6. REAL EXPERIMENTS

Four ordinary web cameras (shown in Fig. 1) with resolution 640×480 were used. They were calibrated (section 2 above). A sequence of 220 frames was taken simultaneously by each camera. The motion of robots is usually uniform, so in addition to the four methods compared in Table 1, we were able to run each sequence as one section using the EKF. The results obtained, the robot used, and samples of the sequences are shown in Fig.2. The ground truth was obtained from the computer controlling the robot motion.

7. DISCUSSION AND CONCLUSIONS

EKF with drift compensation used in the experiments above is simply calculating the pose of the last frame of a section twice (last of a section is first of the subsequent). Since they have to be the same, if there is drift, it will be subtracted from subsequent frames. Despite its simplicity, in simulation this method was next in performance to IEKF-for-all-frames with 10 iterations (marginally better results of IEKF-new-section in rotations suggest multiplying the rotational part of drift as a rotation matrix instead of subtracting it from the angles).

The matter is different in real experiments; both of the EKF with drift compensation and EKF-one-section are very close to the ground truth while the other three methods nearly coincide and suffer from obvious drift especially in case of t_y. The reason for this is that in real experiments the errors (mainly in triangulation) are caused by non-Gaussian noise which emerges from errors in camera calibration and lens distortion. Drift is noticed in [6] if any of the three reference features is occluded. Here, we use a flexibly changing set of features without any drawback of occlusion and the drift is encountered only when few features remain and starting a new section is a must which is rare for a uniform motion.

For MATLAB-7.0.4 running on a machine with a 2.8 MHz Pentium processor, and 1.5 GB RAM, the filter needs 14 ms on average to process each new frame (all cameras) which means theoretically it can process 71 frames/s. However, the bottleneck is in tracking and rejecting outliers (0.4 s), and in stereo-matching and triangulation (27 s usually once at the initial frame). Though not used here, 320×240 frames are expected to enhance the speed by nearly an order of magnitude. More is expected by using C and code optimization.

An essential difference between our solution and other uses of EKF [1], [3], [4], [6], [17], and [19] is that we do not enter the structure into the filter. This makes our filter faster, more stable (shorter state space vector [1]), and more flexible (any available features can be used). This is further justified by the fact that the pose is what we seek here; structure is calculated by triangulation based on the known pose of the initial reference frame then on demand if needed based on the output pose of the filter.

In the light of all this, the proposed algorithm verifies the accuracy, the stability, the flexibility, and the speed needed for the real time pose estimation. For future work, this solution will be compared in details with other approaches which enter the structure into the EKF.

8. REFERENCES

[1] A. Azarbayejani, and A.P. Pentland, "Recursive estimation of motion, structure, and focal length", *IEEE Trans. on PAMI*, 17(6):

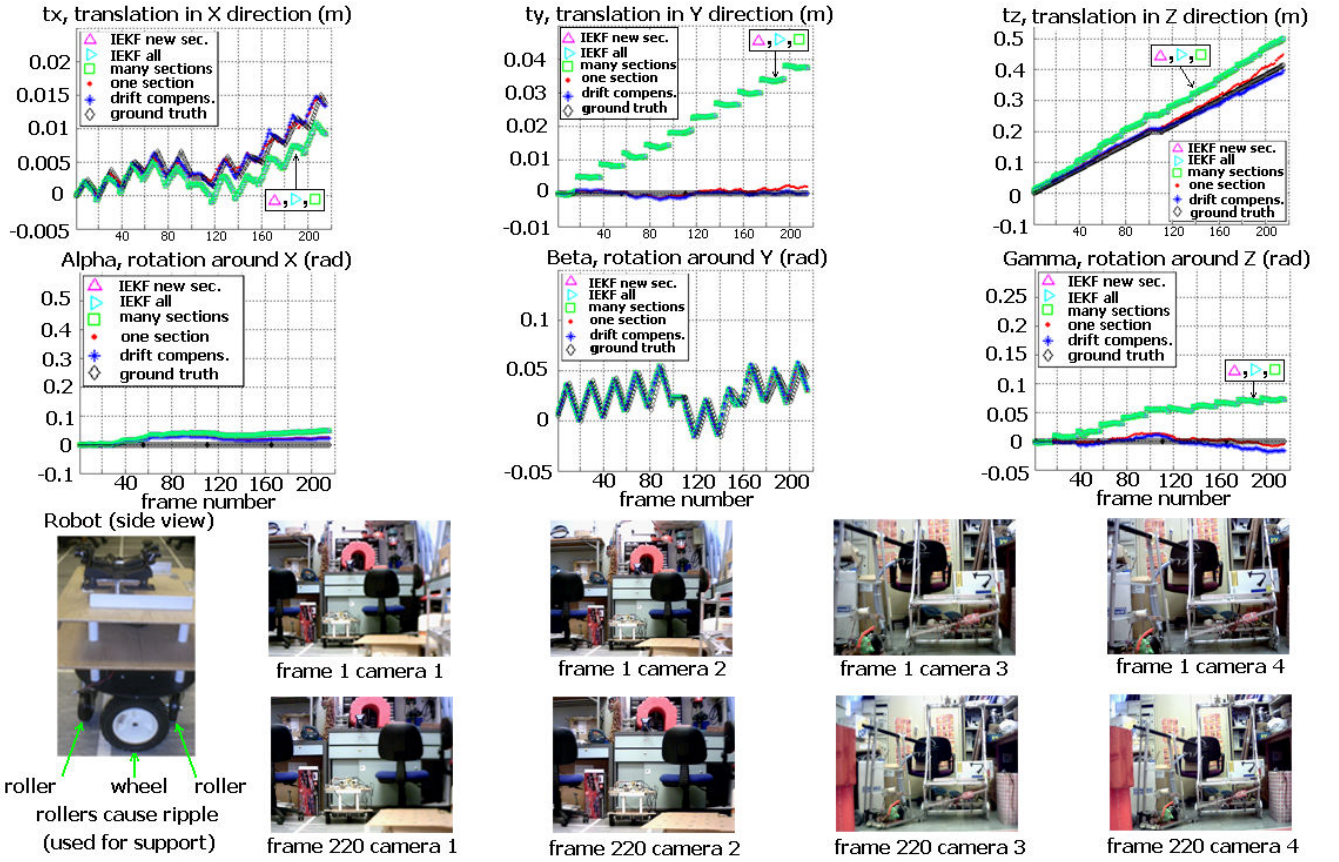

Fig. 2: Real experiment: first row (translations), second row (rotations), third row (first frame for each camera), fourth row (last frame for each camera), and bottom left (side view of the robot used). The curves: IEKF new sec., IEKF all, and many sections nearly coincide.

562-575, June 1995.

[2] P. Baker, C. Fermuller, Y. Aloimonos, and R. Pless, "A spherical eye from multiple cameras (makes better models of the world)", *Proc. CVPR*, 1: 576-583, 2001.

[3] P.A. Beardsley, A. Zisserman, and D. W. Murray, "Sequential Updating of Projective and Affine Structure from Motion", *IJCV*, 23(3): 235-259, 1997.

[4] T. J. Broida, S. Chanrashekhar, and R. Chellappa, "Recursive 3-D Motion Estimation from a Monocular Image Sequence", *IEEE Trans. Aerospace and Electronic Systems*, 26(4):639-656, 1990.

[5] W. Chang, and C. Chen, "Pose Estimation for Multiple Camera Systems", *Proc. ICPR*, 3: 262 - 265, 2004.

[6] A. Chiuso, P. Favaro, H. Jain, and S. Soatto, "Structure from Motion Causally Integrated Over Time," *PAMI*, 24(4): 523-535, 2002.

[7] M. Grossberg, and S. Nayar, "A general imaging model and a method for finding its parameters", *Proc. ICCV*, 2: 108-115, 2001.

[8] M. Isard, and A. Blake, "Contour tracking by stochastic propagation of conditional density", *Proc. ECCV*, 343-356, 1996.

[9] V. Lippiello, B. Siciliano and L. Villani, "Position and Orientation Estimation Based on Kalman Filtering of Stereo Images", *Proc. CCA*, 702-707, 2001.

[10] Medioni, G., and S.B. Kang, *Emerging Topics in Computer Vision*, NJ, USA, Prentice Hall, 2004.

[11] C. Nakajima, and N. Itho, "A Support System for Maintenance Training by Augmented Reality", *Proc. Image Analysis and Processing*, 158 - 163, 2003.

[12] R. Pless, "Using many cameras as one," *Proc. CVPR* , 2: 587-593, 2003.

[13] M. E. Ragab, and K. H. Wong, "Extended Kalman Filter Based Pose Estimation Using Multiple Cameras", Internal Report, CUHK, `http://www.cse.cuhk.hk/~khwong/papers.html`.

[14] S.D. Roy, S. Chaudhury, and S. Banerjee, "Recognizing Large Isolated 3-D Objects Through Next View Planning Using Inner Camera Invariants", *SMC-B*, 35(2): 282-292, 2005.

[15] D. C. Schuurman, and D. W. Capson, "Direct Visual Servoing Using Network-synchronized Cameras and Kalman Filter", *Proc ICRA*, 4: 4191-4197, 2002.

[16] Trucco E., and A. Verri, *Introductory Techniques for 3-D Computer Vision*, New Jersey, Prentice Hall, 1998.

[17] J. Weng, N. Ahuja, and T.S. Huang, "Optimal Motion and Structure Estimation," *PAMI*, 15(9): 864-884, 1993.

[18] M. Yang, Q. Yu, H. Wang, and B. Zhang, "Vision based Real-Time Pose Estimation for Intelligent Vehicles", *Intelligent Vehicles Symposium*, 262-267, 2004.

[19] Y. K. Yu, K.H. Wong, and M. Chang, "Recursive 3D Model Reconstruction Based on Kalman Filtering", *SMC-B*, 35(3): 587-592, 2005.

[20] X. Zhang, N. Navab, "Tracking and Pose Estimation for Computer Assisted Localization in Industrial Environments", *Applications of Computer Vision Workshop*, 214 - 221, 2000.

[21] http://www.vision.caltech.edu/bouguetj/calib_doc.