ICIP2008 IEEE International Conference on Image Processing,
San Diego, California, U.S.A, October 12-15, 2008

EKF POSE ESTIMATION: HOW MANY FILTERS AND CAMERAS TO USE?

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

The Extended Kalman Filter (EKF) is suitable for real-time pose estimation due its low computational demand and ability to handle the nonlinear perspective camera model. There are many EKF based approaches in the literature; some are very recent while others exist for about two decades. These methods differ in two main aspects: the number and arrangement of cameras, and the number and usage of filters. In this work, we will compare these approaches using simulations and real experiments. As far as we know, it is the first attempt to do this with such details. We will show which is suitable under different motion patterns, and explain the effect of the bas-relief ambiguity upon the accuracy of the different approaches. Additionally, we will discuss how to solve the scale factor ambiguity, and suggest the best strategy to deal with the features fed to the filter.

Index Terms— Pose, EKF, multiple-cameras, bas-relief, scale

1. INTRODUCTION

Pose estimation is a crucial problem lasting-for-decades in computer vision as well as in various other fields. Its aim is to find the location and orientation of objects or cameras. The application range extends from mixed reality in movies to activity recognition [1], and guidance of bronchoscopic tracking [2]. Our motivation is to estimate the pose of a moving robot within an unknown indoor scene. Since this has to be done in real time, we need to adopt a recursive technique (working a frame-by-frame). The Extended Kalman Filter (EKF) is a good choice since its computational demand is low (e.g. compared to the particle filter). Additionally, it can handle the nonlinear perspective camera model (in contrast to the linear Kalman Filter (KF)). The EKF has been used in several ways to solve the pose estimation problem. These ways differ in two main aspects: firstly, the number and arrangement of cameras, and secondly, the number and usage of filters. For example, a single camera and one EKF for both pose and structure (sometimes iterated) are used in [3], [4], and in [5]. Alternatively, a single camera, one EKF for pose, and many EKFs for structure are used in [6]. In contrast, four cameras arranged in two back-to-back stereo pairs, one EKF for pose are used in [7] while the structure is obtained on-demand using triangulation. On the other hand, two cameras arranged as a one stereo pair, and one EKF for pose are used in [8] without solving for structure. In this work, we compare between the different approaches using the EKF for pose estimation. We focus on the approaches that handle structure especially that the model-less approach of [8] relies on calculating the trifocal tensor among successive frames which may suffer from degeneracy [9]. Moreover, it does not resolve the scale factor ambiguity since it aims at finding the camera projection matrix which is defined only up to a scale [9]. In particular, we compare four methods. The first uses a single camera, one EKF for pose, and many EKFs for structure as in [6]. The second uses a single camera and one EKF for both pose and structure as in [3], [4], and in [5]. The third uses four cameras arranged in two back-to-back stereo pairs, one EKF for pose as in [7]. The fourth is the same as the third however using only the stereo pair in front (we thank an anonymous reviewer of [7] for suggesting this comparison). As far as we know, it is the first attempt to do this in literature with such details. The main contributions of this paper are: comparing between different approaches of the EKF pose estimation, showing which is suitable under different motion patterns, explaining the effect of the bas-relief ambiguity upon the accuracy of the different approaches, and suggesting the best strategy to deal with the features fed to the filter. The rest of this paper includes: background, EKF solution approaches, experiments, and discussion and conclusions.

2. BACKGROUND

2.1. Feature Selection and Maintenance

It is indicated in [5] that when the zero-mean assumption for measurements is violated, the EKF settles at a biased estimate. In our experience, it is necessary to the stability of the filter to have the features selected uniformly around the mean (i.e. the image center at (0,0)). On the other hand, it is almost universal for disappearing features (e.g. due to occlusion) to be dropped from the filter. However, handling the newly appearing features differs from an implementation to another, as will be shown below.

2.2. Scale Factor and Bas-relief Ambiguities

There is a scale factor ambiguity related to the structure from motion (SfM) approaches [9] (we cannot get the actual 3D structure or translation but we get both up to a scale). As mentioned in [3], this free scale factor (if not normalized) is an extra degree of freedom along which the filter diverges. We will show in section 3 how the different approaches deal with this problem. On the other hand, the bas-relief ambiguity [10] becomes obvious when the depth variation of the scene is small, and the camera field of view is narrow leading to misinterpreting the small translation along one axis as a rotation around another and vice versa.

3. EKF SOLUTION APPROACHES

The EKF is a recursive estimator suitable for real-time applications. It estimates the output (a state space vector encoding either pose or structure OR both according to the approach used) based upon the inputs (2D pixel measurements of the tracked features in the current frame, the previous state space vector, and its covariance). After being initialized, the EKF undergoes two stages (prediction and update) for each time step. For more details, we refer the reader to

*Supported by direct Grants under Projects: Codes 2050350 and 2050410 from the Faculty of Engineering, CUHK, Shatin, Hong Kong.

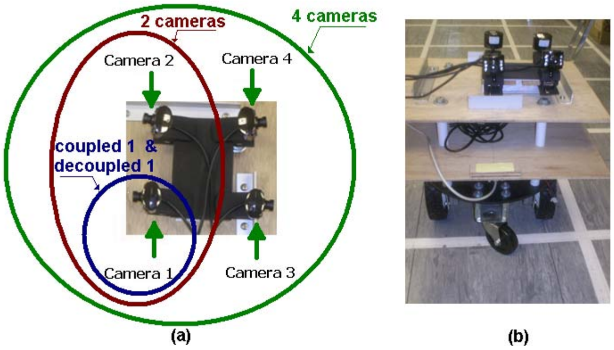

Fig. 1. (a) Top-down view: methods and corresponding cameras used (b) Frontal view of robot and camera setup.

[3], [5], and [11]. Moreover, the multiple-camera EKF formulation is found in [7] and with more details in [12]. The compared EKF approaches for pose estimation are explained below.

3.1. First Method: Decoupled 1

This method uses a single camera, one EKF for pose, and one EKF for each 3D feature (multiple EKFs for structure). The idea of using one EKF for each 3D feature is used for example in [13], however the recursive use of this method to get both pose and structure is introduced in [6]. This method follows an alternating fashion to get pose and structure (at each frame gets pose first using the EKF for pose then proceeds to the multiple EKFs for structure). Since this method decouples pose and structure and uses a single camera, we denote it as: (decoupled 1).

In [6] new features are used as soon as they appear without using subfilters. Additionally, it is assumed there that the scale factor ambiguity is resolved knowing the translation between the object and camera coordinate frames.

3.2. Second Method: Coupled 1

This method is denoted as: (coupled 1) since it couples pose and structure (using a single EKF for both), and uses a single camera. This method dates back to [3] however it still attracts recent research e.g. [11]. Additionally, there is a long sequence of uses in-between such as [14], [4], and [5].

In [5], each new feature is entered into a subfilter to absorb the initialization error and prevent it from propagating onto the original filter. While in [11] new features are added to the filter only if the number of visible features is below a certain threshold. On the other hand, the scale is set in [4] by setting the initial variance of the depth of the first feature to zero. In [5], the scale is associated with three reference features (e.g. the depth of their centroid). However, in [11], the scale is set using an initialization target (a black rectangle whose corners serve as features with known structure).

In addition to these differences, [3] and [11] use the quaternion to describe the rotation while in [5] the angle-axis representation is that used. However, what we need to emphasize here is that the same policy is adopted (i.e. coupling pose and structure using a single EKF and a single camera).

3.3. Third Method: 4 Cameras

This method is introduced in [7]. It uses four cameras arranged in two back-to-back stereo pairs. One EKF is used for pose while the structure is obtained on-demand using triangulation based on the filter output which guarantees the coupling between pose and structure. The zero-mean assumption for measurements is verified using a changeable set of features; in each frame the filter is fed with a number of features selected uniformly around the mean.

meth-	t_x	t_y	t_z	α	β	γ
od	mm	mm	mm	m rad	m rad	m rad
1^{st}	29.751	28.965	18.799	29.218	31.072	3.456
2^{nd}	61.603	69.949	48.339	76.444	70.398	60.432
3^{rd}	0.563	0.526	2.104	0.579	0.503	1.374
4^{th}	41.408	45.593	8.401	46.318	41.159	5.800

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

Stereo information is used to reject the outliers (away from their expected epipolar lines by e.g. 1.5 pixels). Newly appearing features are considered when the number of tracked ones for any stereo pair drops below a certain threshold. Stereo information is used also to resolve the scale factor ambiguity since the baseline of each stereo pair is known [10].

3.4. Fourth Method: 2 Cameras

This method is the same as the third however it is denoted as: (2 cameras) since it uses only the two cameras of the frontal stereo pair (composed of Camera 1, and Camera 2, see Fig. 1).

4. EXPERIMENTS

4.1. Simulations

A robot carrying a setup of four cameras was moved with random translations (t_x, t_y, and t_z) and with random rotation angles (α, β, and γ) in the direction of and around the X, Y, and Z axes respectively. The coordinate system origin coincided with the center of the first camera at the motion start with the Z axis perpendicular to the image frame. The translations were taken randomly from ±0.005 to ±0.015 meters, and the rotation angles were taken randomly from ±0.005 to ±0.02 radians. All cameras had a 6 mm focal length, a 640×480 resolution, and a stereo baseline ranging from 0.1 to 0.2 meters. A random noise was added to each feature point with a normal distribution of zero mean and 0.5 pixel standard deviation. The motion took place inside a spherical surface whose radius was one meter and whose center was coinciding with the origin of the coordinate system. The feature points were distributed randomly on the spherical surface. The total number of feature points, was 35,000. 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 (see below). For fair comparison, each section contained ten frames. We compared the methods mentioned above: decoupled 1, coupled 1, 2 cameras, and 4 cameras. Table 1 shows the average of 100 runs of absolute error in the six pose parameters for the four methods. All absolute errors are given per frame in milli-(meter/radian). To get them in percentage, they should be compared to the average sum of absolute translations in one run (1.00 meter), and the average sum of absolute rotation angles (1.25 radians).

4.2. Real Experiments

We carried out the comparisons for three motion patterns: pure translation, pure rotation, and mixed rotation and translation using two scenarios (the best and the worst). In the best scenario, we compare all the methods with all the motion sequence considered as a one section. This can be done because the motion of robots is usually uniform. On the other hand, in the worst scenario, we cut the motion sequence into multiple sections. For each section, the filters are restarted with fresh new features to study the cases where the number of tracked features becomes insufficient (due to large rotation or being small from the beginning). All the results of the best scenario and some of the worst scenario are shown in Fig. 2. The ground truth was provided by the computer controlling the robot, and the timing information is provided in Table 2.

Step	4 cameras	decoupled 1	coupled 1	2 cameras
initial	5.89062	0.174479	0.20312	2.78125
structure	(in	seconds	per	sequence)
tracking	0.18765	0.05250	0.05648	0.10598
features	(in	seconds	per	frame)
initial	0.01562	0.01562	0	0.01562
pose	(using Lowe's method, in seconds per sequence)			
inside	0.014100	0.04375	0.37179	0.00534
EKF	(in	seconds	per	frame)

Table 2. Average time inside the steps of the compared approaches (based on the best scenario of pure translation sequence, and using MATLAB-7.0.4 running on a machine with a 2.8 MHz Pentium processor, and 1.5 GB RAM). Frame means a frame from each camera.

5. DISCUSSION AND CONCLUSIONS

An advantage of 4 cameras and 2 cameras methods is that they verify the zero-mean assumption (see subsection 2.1) at every frame by using a changeable set of features around the mean. This cannot be done for decoupled 1 and coupled 1 which should ideally track the same set of features (whose structure has been already stabilized within the filter). In this case, unless the camera center moves along the Z axis, the zero-mean assumption will be soon violated. Additionally, tracking the same set of features will keep the key features which unify the scale as in [4], and [5] (note that the real scale is not recovered). The use of an initialization target in the first frame in [11] to recover the scale requires a manual intervention and may hide some features behind.

Newly appearing features can be dealt with instantly in 4 cameras and 2 cameras (stereo triangulation recovers the real scale). However, for decoupled 1 and coupled 1, subfilters should be used (see subsection 3.2). The subfilters are not used in [6] perhaps because the focus there is on one object whose translation from the camera center is known. However, subfilters have drawbacks; they reduce the longevity of features within the original filter, increase the computational demand, and may be useless when features are lost before being ready to enter the original filter. Dealing with newly appearing features as in [11] and [7] (see subsections 3.2 and 3.3) overcomes most of these drawbacks.

For fair comparison, we use more features for single camera methods (50-150) than we use for multiple-camera methods (35-50). Additionally, each time we restart the filter by moving back in time ten frames to offer single camera methods a chance to become stable before relying on their output (as in [6]). Furthermore, we use the Lowe's method (as used in [7] for multiple-camera methods) to initialize the pose in the decoupled 1 method based on the initial structure to relate them together and speed the processing by using good pose initial conditions.

In simulations, with all features on the unit sphere, setting the depth of the nearest feature to the image center in the first frame to one with zero initial variance fixes the scale factor to one and, accordingly, we can compare the approaches which recover the real scale with the approaches that only unify it. From Table 1, 4 cameras method clearly outperforms all others.

For the real experiments, up to a scale translational pose parameters are expected for single camera methods. From Fig. 2, 4 cameras and 2 cameras are close to the ground truth. The performance of the 2 cameras method is better than its performance in the simulations. The reason for this is that the simulations were carried out under harsh conditions which are likely to increase the bas-relief ambiguity since all features were distributed on a spherical surface with a small depth variation. This was done to fix the scale factor to one and accordingly be able to compare all methods.

On the other hand, decoupled 1 and coupled 1 always suffer from deviations from the ground truth especially for pose parameters that remain nearly constant (they could track pose parameters whose change is dominating e.g. t_z of rows 1 and 5 of Fig. 2, up to a scale factor, sometimes with a large offset, and usually tens of frames after initializing the filter). Additionally, cutting the sequence into multiple sections and restarting the filter again seems to be devastating to the single camera methods. In contrast, 4 cameras and 2 cameras can restart the filter at any time to acquire a fresh set of features with a minimum drift, and an efficient rejection of outliers on the per-frame-basis. Furthermore, a simple drift compensation approach can be implemented easily in 4 cameras method [7].

To sum up, 4 cameras method is accurate under all tested patterns of motion and in the presence of the bas-relief ambiguity. 2 cameras method is faster than the 4 cameras method and nearly as accurate provided that there is no bas-relief ambiguity in the scene. Single camera methods are suitable for tracking dominating smoothly changing parameters provided that there is no bas-relief ambiguity in the scene, and in this case, the decoupled 1 method is preferred for the sake of speed.

6. REFERENCES

[1] O.C. Jenkins, G. Gonzalez, and M. Loper, ," in *CVPR'06*. IEEE, vol. I, pp., 147–152.

[2] L. Rai, S. A. Merritt, and W.E. Higgins, ," in *CVPR'06*. IEEE, vol. II, pp., 2437–2444.

[3] T.J. Broida, S. Chanrashekhar, and R. Chellappa, "Recursive 3-D motion estimation from a monocular image sequence," *IEEE Trans. Aerospace and Electronic Systems*, vol. 26, no. 4, pp. 639–656, July 1990.

[4] A. Azarbayejani and A.P. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Trans. on PAMI*, vol. 17, no. 6, June 1995.

[5] A. Chiuso, P. Favaro, H. Jain, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Trans. on PAMI*, vol. 24, no. 4, pp. 523–535, April 2002.

[6] Y.K. Yu, K.H. Wong, and M. Chang, "Recursive three-dimensional model reconstruction based on Kalman filtering," *IEEE Trans. On Systems, Man, And Cybernetics*, vol. 35, no. 3, pp. 587–592, June 2005.

[7] M.E. Ragab, K.H. Wong, J.Z. Chen, and M.M.Y. Chang, ," in *ICIP'07*. IEEE, vol. VI, pp., 137–140.

[8] Y.K. Yu, *Model-less Pose Estimation*, Ph.D. thesis, The Chinese University Of Hong Kong, 2007.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*, Cambridge University Press, 2003.

[10] R. Szeliski, and S.B. Kang, "Shape ambiguities in structure from motion," *IEEE Trans. On PAMI*, vol. 19, no. 5, pp. 506–512, May 1997.

[11] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse, "Monoslam: Real-time single camera SLAM," *IEEE PAMI*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[12] M.E. Ragab, *Multiple Camera Pose Estimation*, Ph.D. thesis, The Chinese University Of Hong Kong, 2008.

[13] P.A. Beardsley, A. Zisserman, and D.W. Murray, "Sequential updating of projective and affine structure from motion," *International Journal of Computer Vision*, vol. 23, no. 3, pp. 235–259, 1997.

[14] J. Weng, N. Ahuja, and T.S. Huang, "Optimal motion and structure estimation," *PAMI*, vol. 15, no. 9, pp. 864–884, September 1993.

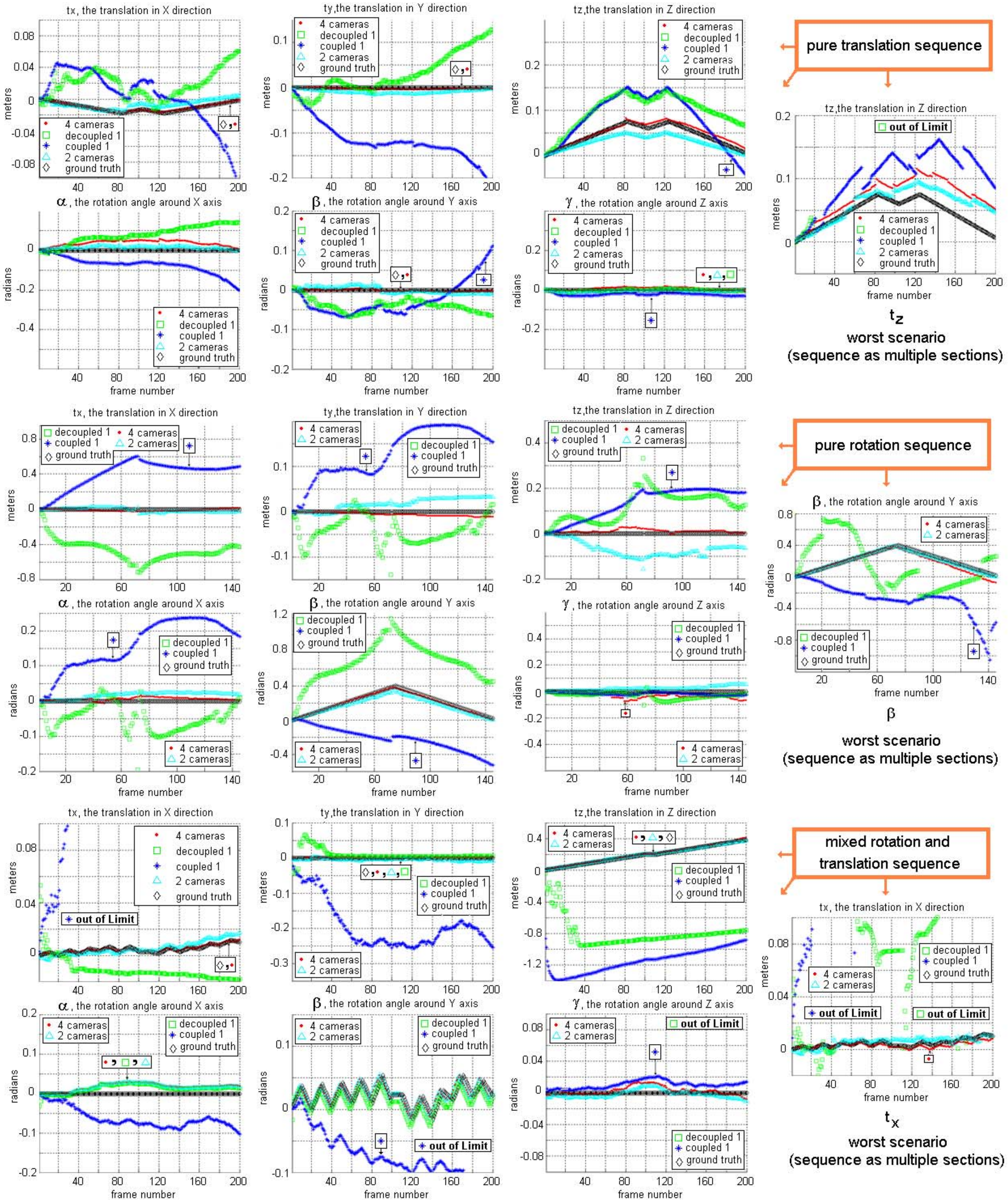

Fig. 2. Real experiments pose parameters, from top: rows 1 and 2 belong to pure translation sequence, rows 3 and 4 belong to pure rotation, rows 5 and 6 belong to mixed rotation and translation, and the rightmost column shows a pose parameter for each sequence belonging to the worst scenario (sequence as multiple sections).