in: Proc. 52nd Int. Scientific Colloquium (IWK), 2007, Technische Universitĺat Ilmenau, Germany, Vol. II, pp. 353-358, Universitĺatsverlag 2007

E. Einhorn / C. Schröter / H.-J. Böhme / H.-M. Gross

A Hybrid Kalman Filter Based Algorithm for Real-time Visual Obstacle Detection

1 Introduction and Related Work

Obstacle detection and collision avoidance must be considered important capabilities of mobile robots. Vision-based approaches provide a large field of view and supply a large amount of information about the structure of the local surroundings.

In this paper we present a sparse feature-based "shape-from-motion" approach for mobile robots which is applicable for collision avoidance, online map building and scene reconstruction in real-time. Our method processes a sequence of images which are taken by a single camera mounted on a mobile robot. In contrast to similar monocular shape-from-motion algorithms we combine two different approaches: A traditional motion stereo approach and a Kalman filter based algorithm for scene reconstruction. We show that the disadvantages of the traditional stereo approach are compensated by the Kalman filter and vice versa. Our special method of initializing the Kalman filter leads to a faster convergence compared to other plain Kalman based approaches. Moreover, we present a feature matching algortihm which is faster and more reliable than the widely-used KLT-Tracker in the domain of scene reconstruction. These image features are extracted using the "FAST" high-speed corner detector [8].

As we intend to use the reconstructed scene for obstacle detection and collision avoidance our camera is mounted in front of the mobile robot and tilted towards the ground. This results in two major problems we have to deal with:

1. The camera is moving along its optical axis. In a sensitivity analysis Matthies and Kanade [7] proved that when using forward motion shape-from-motion leads to higher uncertainties in the depth estimates. Compared to the ideal lateral camera translation parallel to the image plane - which is used in standard binocular approaches - the estimation must be applied over a long base distance in order to achieve the same accuracy.

2. Many objects are visible during a few frames of the captured image sequence only, while the robot is approaching these obstacles. Hence, most image features cannot be tracked over a large number of frames and the scene reconstruction algorithm must be able to provide a reliable estimate by using a few image measurements only.

To overcome the first problem Matthies et al. [7] suggest scene reconstruction using Kalman filters, since they can integrate the depth and scene information over a long base distance. Consequently, many shape-from-motion solutions that have been researched and published in recent years are based on Kalman filtering [10, 5, 1, 7]. Since Kalman filter based methods solve the reconstruction problem in an iterative manner the speed of convergence depends

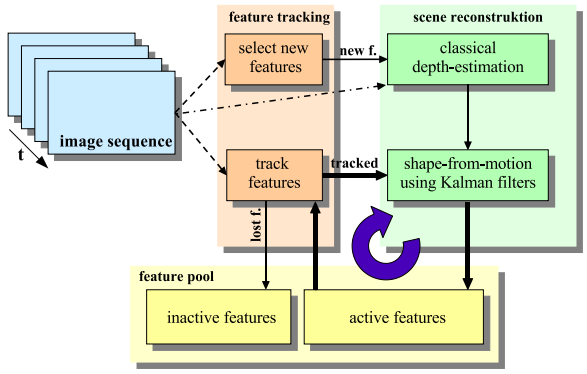
on the choice of the initial estimate which is used for the initialization of the Kalman filter. If unfavourable initial estimates are used, Kalman filter based approaches tend to suffer from a low speed of convergence, i.e. several iterations must be processed to get a reliable estimation of the obstacle positions. Unfortunately, as stated above, most image features cannot be tracked over many frames and it is not possible to compute enough iterations.

To prevent this problem our hybrid algorithm combines two completely different approaches for scene reconstruction: A Kalman filter based method for scene reconstruction and a "classical" correlation-based depth estimation approach.

The depth estimation algorithm is used to compute a reliable initial estimate for the Kalman filter, which then will refine the estimate and recover the three-dimensional model. We show that this novel kind of initialization leads to a significantly better convergence of the filter.

2 Scene-Reconstruction

Since depth estimation and scene reconstruction using Kalman filtering are common techniques in computer vision they will not be described in detail here. Further information can be found in [3, 10, 5, 1, 9].



The figure on the left illustrates the complete architecture of our algorithm. Our motion stereo approach is inspired by the publication of Bunschoten and Kröse [3], where a multi-baseline depth estimation algorithm for panoramic image data is presented. Based on their work we have developed a similar correlation-based algorithm for projective cameras. To obtain the image data we work with a single projective camera mounted on our mobile robot to capture not only a sequence of images - each taken from a different pose (i.e. position and orientation) during the robot's locomotion - but also the corresponding odometry data measured by the robot drive. Hence, for each image of the sequence the approximate position of the camera is known, including uncertainty in odometry measurements from systematic and non-systematic errors.

To correct these errors we use correspondences our feature tracker has found over the frames of the image sequence to estimate the pose of the camera. Since the translation vector of the camera movement can only be computed up to a scale, we are content with estimating the angle of roll and the pitch of the camera, since inaccuracies in the orientation of the camera cause the largest error in the scene reconstruction. Starting with values provided from the robots odometry both angles are varied using Gauss-Newton iteration in order to minimize the Sampson error, which is defined by the used image point correspondences and the fundamental matrix.

The estimated depth is then used to compute the approximate 3D position of the feature in the real scene. This position is used as a reliable initial estimate for the Kalman filtering, which then will refine the estimate and recover the three-dimensional model. In contrast to [5, 1] where one single Kalman filter with a large state vector is used to recover the 3D-positions of all features (model points), we use a separate filter for each feature point. According to [10] this leads to a linear space and time complexity in terms of the number of features while the loss in accuracy is small. Similar to [10] we choose the 3D position of the feature point as state vector $\mathbf{X} \in \mathbb{R}^3$ which is to be estimated. Using this estimated 3D position of each scene point an a-priori estimate of its image position can be computed by projecting it onto the image surface of the camera. The observed measurement is the position of the real image point in the current image which is provided by a feature tracker, that tracks each image point over consecutive frames. With each new frame the tracked features will pass through this Kalman filter cycle and their 3D positions will be estimated more precisely in each iteration.

3 Feature Tracking

In order to track the image features over several frames, we apply a feature matching algorithm. First we select the image features independently in each frame using the FAST corner detector [8]. Similar to the IPAN feature tracker [4] corresponding features are matched in subsequent frames then. While the IPAN tracker solves a pure motion correspondence problem by using three consecutive frames and solely kinematic constraints, we only use two frames. To eliminate the resulting ambiguities we additionally take the image similarity into account.

Let I_{t-1} and I_t be two consecutive frames of the image sequence. In order to find the correspondences $\mathbf{x}_{t-1}^{(i)} \leftrightarrow \mathbf{x}_t^{(i)}$ between the previously selected image features of both frames, possible hypotheses of matching points are chosen first. Each hypothesis $h = (\mathbf{x}_{t-1}^{(i)}, \mathbf{x}_t^{(j)})$ consists of a pair of two potentially matching points $\mathbf{x}_{t-1}^{(i)}$ and $\mathbf{x}_t^{(j)}$ of the frames I_{t-1} and I_t. To reduce the number of hypotheses we use a maximum speed constraint, i.e. we only choose pairs of image points that satisfy $\left|\mathbf{x}_{t-1}^{(i)} - \mathbf{x}_t^{(j)}\right|_2 \leq r_{max}$, where r_{max} defines the maximum speed, at which a point can cross the frame within the image sequence.

For each hypothesis $(\mathbf{x}_{t-1}^{(i)}, \mathbf{x}_t^{(j)})$ we compute a cost function which is defined by the following weighted sum: $\text{cost}(\mathbf{x}_{t-1}^{(i)}, \mathbf{x}_t^{(j)}) = w_1 c_1 + w_2 c_2 + w_3 c_3$, where $c_1 = \left\|\mathbf{x}_t^{*(i)} - \mathbf{x}_t^{(j)}\right\|_2^2$ is the squared euclidean distance between the image point $\mathbf{x}_t^{(j)}$ and the predicted feature position $\mathbf{x}^{*(i)}$. The latter can be computed as $\tilde{\mathbf{x}}^{*(i)} = \mathbf{P}_t \tilde{\mathbf{X}}_t^*$ using the reconstructed 3D position $\tilde{\mathbf{X}}_t^*$ of the feature which was estimated so far and the corresponding camera projection matrix $\mathbf{P}_t$, which is computed from the corrected odometry data of the robot. Since we perform an initial depth estimation as described in the previous section, an estimate of the 3D position is already available for newly selected features. For features that have been tracked over several frames

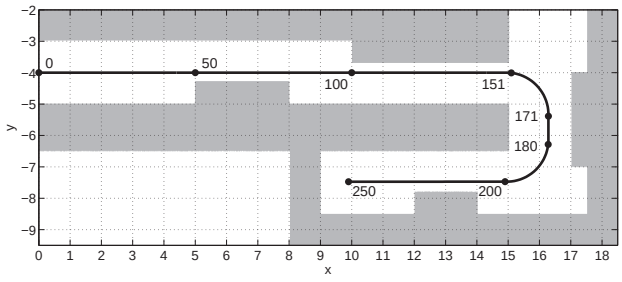
more accurate estimations of the 3D positions were computed by the Kalman filters and their location in the current frame can be predicted more precisely.

Additionally, corresponding image points must satisfy the epipolar constraint, hence an image point $\mathbf{x}_t^{(j)}$ that corresponds to $x_{t-1}^{(i)}$ is located on or near the epipolar line that is induced by $x_{t-1}^{(i)}$. The distance of the image point $\mathbf{x}_t^{(j)}$ from that epipolar line can be computed as follows: $c_2 = \frac{\left|\tilde{\mathbf{x}}_t^{(j)\top}\mathbf{F}\tilde{\mathbf{x}}_{t-1}^{(i)}\right|}{\sqrt{(\mathbf{F}\tilde{\mathbf{x}}_{t-1}^{(i)})_1^2+(\mathbf{F}\tilde{\mathbf{x}}_{t-1}^{(i)})_2^2}}$, where $\mathbf{F}$ is the corresponding fundamental matrix which again is computed using the robot's odometry. Alternatively, the Sampson distance [6] could be used which is, however, computationally more complex.

As stated above we also use a similarity constraint to eliminate ambiguous matchings. For each pair of potentially matching points $\mathbf{x}_{t-1}^{(i)}$ and $\mathbf{x}_t^{(j)}$ we compute the similarity of their neighborhood patterns. Again we use the SAD as measure of correlation: $c_3 = \mathrm{SAD}_W(\mathbf{x}_{t-1}^{(i)}, \mathbf{x}_t^{(j)})$. The weights of the above cost function must be chosen empirically. We use $w_1 = 1$, $w_2 = 3$ and $w_3 = 20$. From all hypotheses those with minimal matching costs are chosen by a greedy algorithm. Hypotheses whose costs are larger than a certain threshold are rejected. An appropriate threshold depends on the image data. Finally, all chosen hypotheses represent the corresponding image points.

4 Results

In order to make a quantitative analysis and to be able to compare our hybrid approach with others we have rendered a sequence of a synthetic scene consisting of 250 frames and their corresponding ground truth depth images using the raytracer POV-Ray[1]. The ground truth depth images are used to measure the tracking error and the error of the reconstructed 3D model. We use realistic textures and add some gaussian image noise. To simulate odometry errors and the sway of the camera we add gaussian noise to the camera position and orientation while rendering the images.



The figure on the left shows a top view of the synthetic scene where the camera trajectory is plotted and its position at certain frames is marked. In figure 1 and 2 the guided feature matching algorithm we have proposed in this paper and Birchfield's implementation of the KLT feature tracker [2] are compared. Because of the guided matching, our greedy feature linking algorithm has a smaller tracking error. Although the tracking error of the KLT tracker can also be reduced if guided tracking is used and the tracker is provided with the predicted feature locations as described in the previous section, the runtime of the KLT tracker remains a problem for realtime applications. Using the proposed feature matching

[1] `http://www.povray.org/`

algorithm we were able to reduce the runtime that is needed for feature tracking dramatically as shown in the second diagram of figure 2. Using 200 features per image the hybrid scene reconstruction and the feature linking can be computed in just 20 ms per frame. Hence we can process up to 50 frames per second.

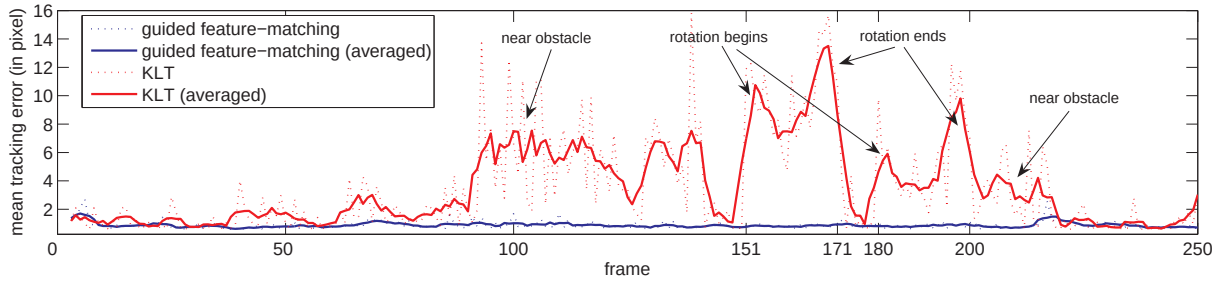


Figure 1: Mean tracking error for each frame of the synthetic image sequence. Due to the large optical flow while the camera is rotating and approaching near obstacles the tracking error of the KLT tracker becomes larger while with guided matching it remains small.

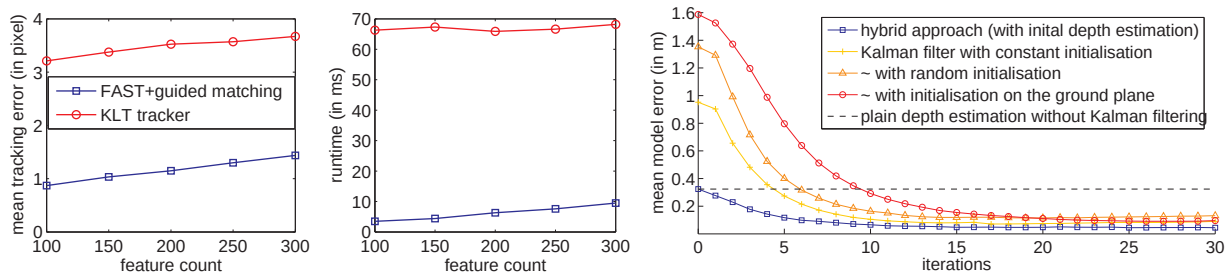
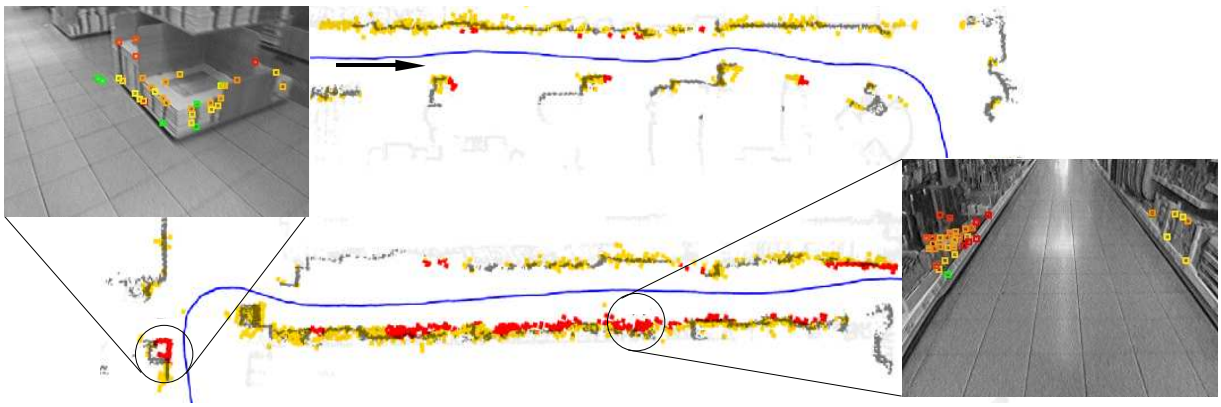


Figure 2: **left:** Mean tracking error averaged over all frames of the synthetic sequence for different feature counts. **middle:** Runtime that is needed for feature selection and feature tracking depending on the number of features that are selected in each frame. The time was measured on a Pentium 4 with 3.4 GHz. **right:** comparison of different methods for initializing the Kalman filters.

In the right diagram of figure 2 it can easily be seen that the hybrid algorithm presented in this paper converges faster than plain Kalman filter based aproaches which use simple heuristics for choosing the initial estimates. Less iterations and therefore less images are necessary to obtain a reliable scene reconstruction.



The above figure shows a map which was created while the robot was moving through a real indoor environment. The estimated positions of the features are visualized using red and orange dots. The estimated z-coordinate is used only to determine if a point belongs to an obstacle or if it lies on the floor, i.e. if the z-coordinate of a point is less than a certain threshold of 0.1 m it is regarded as belonging to the ground plane and not included in the map. The gray map in the background was built using a laser range finder and is used as reference. The accuracy of the map which was built using our approach is similar to the laser-built reference map. Moreover, our visual method is able to detect some obstacles which are not "visible" to the laser because they are too small and lie beneath the laser range finder. Those obstacles were labeled manually and are highlighted by the red color. Additionally, the corresponding camera images are shown for two of these obstacles. It can easily be seen that one part of the left obstacle is not included in the laser map, since it is too small and located below the laser plane. This would have led to a collision if solely laser based navigation had been used. Using our hybrid approach for visual obstacle detection instead, this obstacle can be detected very well.

References

[1] A. J. Azarbayejani, T. Galyean, B. Horowitz, and A. Pentland. Recursive estimation of CAD model recovery. In *Proceedings of the 2nd CAD-Based Vision Workshop*, Champion, PA, 1994.

[2] Stan Birchfield. Deriviation of kanade-lucas-tomasi tracking equation. January 20 1997.

[3] R. Bunschoten and B. Kröse. Robust scene reconstruction from an omnidirectional vision system. *IEEE Transactions on Robotics and Automation*, 19(2):351–357, 2003.

[4] Dmitry Chetverikov and Judit Veresty. Tracking feature points: a new algorithm. In *Proc. of 14th International Conference on Pattern Recognition*, pages 1436–1438, Brisbane, Australia, 1998.

[5] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 4*, Apr. 2002.

[6] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0-521-54051-8, second edition, 2006.

[7] Larry Matthies, Takeo Kanade, and Richard Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, 1989.

[8] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, May 2006.

[9] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[10] Y. Yu, K. Wong, and M. Chang. Recursive 3d model reconstruction based on kalman filtering. *SMC-B*, 2004.

Author Information:

Dipl. Inf. Erik Einhorn, Dipl. Inf. Christof Schröter,
Dr. Hans-Joachim Böhme, Prof. Dr. Horst-Michael Groß
Neuroinformatics and Cognitive Robotics Lab,
Faculty of Computer Science and Automation,
Ilmenau Technical University, POB 10 05 65, 98694 Ilmenau
Tel: +49 3677 69 1306
E-mail: `Erik.Einhorn@tu-ilmenau.de`