Kalman Filter based Depth from Motion with Fast Convergence

Uwe Franke, Clemens Rabe
DaimlerChrysler AG
70546 Stuttgart
Email: {uwe.franke,clemens.rabe}@daimlerchrysler.com

***Abstract*— The extraction of depth is a prerequisite for many applications in robotics and driver assistance. Examples are obstacle detection, collision avoidance, and parking. This paper presents a new Kalman Filter based depth from motion approach. Thanks to multiple filters running in parallel the rate of convergence is significantly higher than in direct methods, especially if the vehicle drives slowly. A goodness-of-fit test fuses the states of the different filters in an optimum manner. In addition, this test allows to distinguish between static and moving obstacles.**

I. Introduction

Intelligent cars of the foreseeable future will be equipped with a camera for tasks such as lane departure protection and Night Vision. Several applications such as obstacle detection, obstacle avoidance, and parking require 3D information. If only one camera is available, depth must be estimated from the image sequence obtained while driving. Pollefeys [1] gives a comprehensive overview on structure from motion methods. Most of them have in common that they do not incorporate any a priori knowledge on the possible camera motion. Since the mentioned applications require 3D information at as many image points as possible in order not to overlook an important object, fast methods are needed.

If the camera motion is known, in vehicles we know at least speed and yaw rate, the 3D-position of stationary world points can be efficiently estimated by means of Kalman Filters. This requires tracking of image points through the sequence. The better the estimation of the 3D-positions, the faster the tracking. This raises the hope that a real-time estimation of depth is possible that is robust with respect to measurement noise.

Kanade [2] describes the usage of Kalman Filters for the ego-motion estimation and sketches the detection of obstacles with the same principle. However, it turns out that the rate of convergence of the suggested system is rather slow. If we want to realize a vision bumper as the basic sensor for the avoidance of stationary objects, we may loose too much time and driving distance, respectively.

In this paper we present a multiple filter approach that shows a significant convergence speed up. Initialized with different states, i.e. 3D-positions, their innovation errors are used as a goodness-of-fit test. This test acts as a weighting function that combines the states of the different filters. In addition, the test criterion allows to distinguish between stationary and moving objects.

The paper is organized as follows. Chapter 2 describes the principles of Kalman Filter based depth reconstruction, the plant model, and the measurement model. The usage of multiple differently initalized filters and their combination is presented in chapter 3. The accuracy of the proposed method is investigated in chapter 4. Chapter 5 shows how moving objects are detected. Results for real sequences are finally presented in chapter 6.

II. 3D from motion

In the following we use a right handed coordinate system with the origin at the road. The lateral x-axis points to the left, the height axis y points upwards and the z-axis representing the distance of a point is straight ahead. This coordinate system is fixed to the car, so that all estimated positions are given in the coordinate system of the moving observer. The camera is at $(x, y, z) = (0, height, 0)$.

A. System model

The movement of a vehicle with constant velocity v and yaw rate $\dot{\psi}$ over the time interval Δt can be described in this car coordinate system as

$$\Delta \underline{x}_c = \int_0^{\Delta t} \underline{v}(\tau)\, d\tau \tag{1}$$

$$= \frac{v}{\dot{\psi}} \begin{pmatrix} -\left(1-\cos\dot{\psi}\Delta t\right) \\ 0 \\ \sin\dot{\psi}\Delta t \end{pmatrix}. \tag{2}$$

The position of a static world point $\underline{x} = (X, Y, Z)^T$ after the time Δt can be described in the car coordinate system as

$$\underline{x}_k = R_y(\psi)\, \underline{x}_{k-1} - \Delta \underline{x}_c \tag{3}$$

with the rotational matrix around the y-axis $R_y(\psi)$. This yields to the discrete System model equation

$$\underline{x}_k = A_k \underline{x}_{k-1} + B_k v + \underline{w}_{k-1} \tag{4}$$

with the state transition matrix

$$A_k = \begin{pmatrix} \cos\left(\dot{\psi}\Delta t\right) & 0 & -\sin\left(\dot{\psi}\Delta t\right) \\ 0 & 1 & 0 \\ \sin\left(\dot{\psi}\Delta t\right) & 0 & \cos\left(\dot{\psi}\Delta t\right) \end{pmatrix} \tag{5}$$

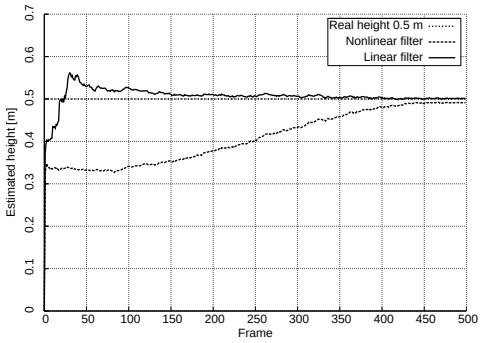


Fig. 1. Height estimation obtained by the linear and the non-linear version of the Kalman Filter

and the control matrix

$$B_k = \frac{1}{\dot{\psi}} \begin{pmatrix} 1 - \cos\left(\dot{\psi}\Delta t\right) \\ 0 \\ -\sin\left(\dot{\psi}\Delta t\right) \end{pmatrix}. \quad (6)$$

The noise term $\underline{w}$ is assumed to be a gaussian white noise with covariance matrix Q.

B. Measurement model

Image coordinates u and v of a feature are measured using an appropriate point tracker. In our current implementation we use a KLT-tracker [3]. Assuming a pin hole camera the nonlinear measurement equation for a point given in the camera coordinate system is

$$\underline{z} = \begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} Xf_u \\ Yf_v \end{pmatrix} + \underline{\nu} \quad (7)$$

with the focal lengths f_u and f_v. Although Kalman Filters can cope with non-linear measurement equations, it is beneficial to work with linear ones. Linearisation as described in [4] yields the linear measurement equation

$$\begin{aligned} \underline{z} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} f_u & 0 & -u \\ 0 & f_v & -v \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + Z\underline{\nu} \end{aligned} \quad (8)$$

with the state dependent noise $\underline{\nu}$. The measurement covariance matrix R is

$$R = Z^2 \begin{pmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{pmatrix}. \quad (9)$$

Figure 1 compares the behaviour of the linear and the non-linear Kalman Filter version. The filters estimate the height of a static world point using the same input data (tracked point) at low speed ($1\,\frac{\mathrm{m}}{\mathrm{s}}$). Please note that a correct estimation of the point's height is equivalent to the correct estimation of its distance for a known camera pitch angle. The linear filter shows a faster convergence especially if the initial estimate of the state vector is bad. In general, the linear filter is more robust due to the nature of the Kalman filtering scheme and is strongly recommended.

III. Multi Kalman Filter

The presented Kalman Filter estimates the world position of a static point in relation to the moving car. Before the filter can begin with its work, it has to be initialized. This initial guess will then be refined by the filter over time. If the car drives slowly, the observed optical flow is also small and the initialization turns out to be a crucial point.

The further the first guess deviates from the correct value, the longer it takes until the estimate is below a given error threshold. This is illustrated by figure 2. A point at $z = 20.0\,\mathrm{meter}$ and height $y = 0.75\,\mathrm{meter}$ is tracked and the estimated height is plotted for different initializations. We assumed the point to lie on the ground (zero height) or to be part of a wall at a distance corresponding to the assumed height. Although very large initial values of the P-Matrix are used, the speed of convergence is unsatisfactory low. In this simulation, 100 frames correspond to a driven distance of 4 meter.

As the main goal is to achieve a fast convergence that is almost independent of the initial choice, the question arises how to choose the optimal initialization. We will focus on this initialization problem in the following.

Our approach is to run multiple differently initialized Kalman Filters in parallel, estimating the world position using the same input data. We combine the estimated states of these filters in a proper manner. Assuming a limited initial state space, i.e. the tracked point is not below the road level and its height is limited, we initialize the filters on different heights including the boundaries. For example, this multi filter system consists of three Kalman Filters initialized on equidistant heights as show in figure 2. As it can be clearly seen, the filter with the largest initial deviation needs the largest amount of time to fall below a given error threshold.

How can we decide which state is the best?

One way is to calculate the distance between the real measurements and the predicted measurements using the Mahalanobis distance. Instead of the Euclidean distance it takes the different variances into account. It is defined as

$$D_M\left(\underline{z}, \underline{x}\right) = \left(\underline{z} - \underline{x}\right) \Sigma^{-1} \left(\underline{z} - \underline{x}\right)^\top \quad (10)$$

with the measurement $\underline{z}$, the predicted measurement $\underline{x}$ and the innovation covariance matrix Σ, which can be calculated using the measurement matrix H, the state covariance matrix P, and the measurement covariance matrix R at the iteration k:

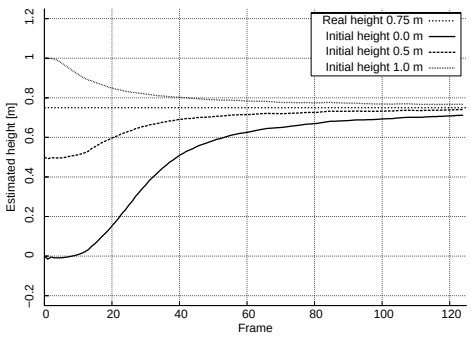
$$\Sigma = H_k P_k^- H_k^\top + R_k. \quad (11)$$

This is also known as the normalized innovation squared (NIS) [5].

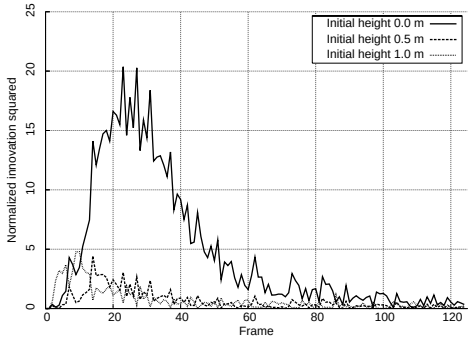
Alternatively, the probability density function, also called likelihood, defined as

$$f\left(\underline{z}|\mu\right) = \frac{1}{(2\pi)^{\frac{n_\mu}{2}} \left|\Sigma_\mu\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{z}-\underline{x})\Sigma^{-1}(\underline{z}-\underline{x})^\top} \quad (12)$$

can be used as an indicator to decide whether a given measurement $\underline{z}$ corresponds to a Kalman Filter model μ. This is



(a) Height estimation



(b) Normalized innovation squared

Fig. 2. Kalman Filter initialized on different heights

used for example in the IMM as described by Bar-Shalom [5]. However, the likelihood calculation tends to suffer from too small floating point data types.

Both criterions, normalized innovation squared and likelihood, can be used to select the best matching filter. In order to avoid the mentioned numerical problems, we base our decisions on the NIS-criterion. Figure 2(b) shows these normalized innovation squares for the three differently initialized filters. It is obvious that the initialization quality corresponds with the discrepancy in measurement space between the measured and predicted position.

Selecting one of the three (in general n) filter states as the correct one would ignore valuable information contained in the other filters. As described above we assume a limited state space with filters on the boundaries. The real state must then lie in between these boundaries and can be expressed as a weighted sum

$$\underline{\widetilde{x}} \quad = \quad \frac{1}{\sum \beta_i} \sum_{i=0}^{k} \beta_i \underline{x}_i. \tag{13}$$

The weights represent the matching quality of each Kalman Filter.

If the NIS-criterion is used, the weights are simply given by

$$\beta_i \quad = \quad \frac{1}{NIS_i}. \tag{14}$$

For the Likelihood criterion we have to use:

$$\beta_i \quad = \quad f\left(\underline{z}_k | i\right). \tag{15}$$

It is beneficial not to base the decision or weighting on the current measurement quality only, since this would lead to undesired effects due to measurement noise. This can be seen from figure 2(b), where the NIS of the worst filter is optimal at the start. Here the filter initialized on height 0.0 meter has a much greater measurement variance than the other filters due to the state dependent noise. Therefore, we apply a low pass filtering to the weights thus accumulating the errors over a certain time.

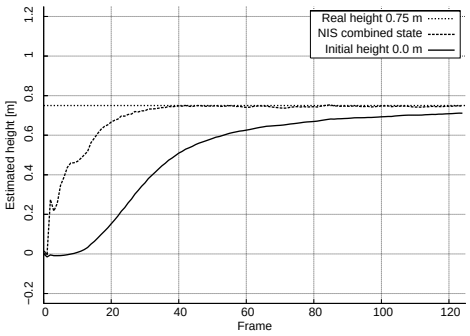


Fig. 3. Height estimation obtained by a single and the multi Kalman Filter system showing the improved rate of convergence

Figure 3 shows the result obtained by the above approach, if the three filters are initialized at zero, half and one meter. For comparison, a single filter initialized at zero height is considered, which seems to be the best initialization if we are interested in obstacle detection and do not have any additional knowledge. It can be seen that the multi filter approach converges at least three times faster than the simple one. A comparison with figure 2 reveals that the combined system shows a better performance than each of the three single filters.

IV. Accuracy Investigations

How well does the method perform if applied to images? In order to answer this question, we first consider the artificial scene shown in figure 4 including different static objects. We concentrate on the marked wall. Since we are interested in the convergence behaviour and not in the tracking aspect, we attached a high frequent texture to this obstacle.

Figure 5 gives a view from the side, while we are approaching the obstacle. At frame zero, the distance is 15 meter, the last frame 50 is taken at 13 meter. The upper left plot shows the estimated height over the distance for all tracked points after the fifth frame. The green (bright) positions are obtained by means of the multi filter approach; the red (dark) points are the results of a single filter initialized at zero height. While

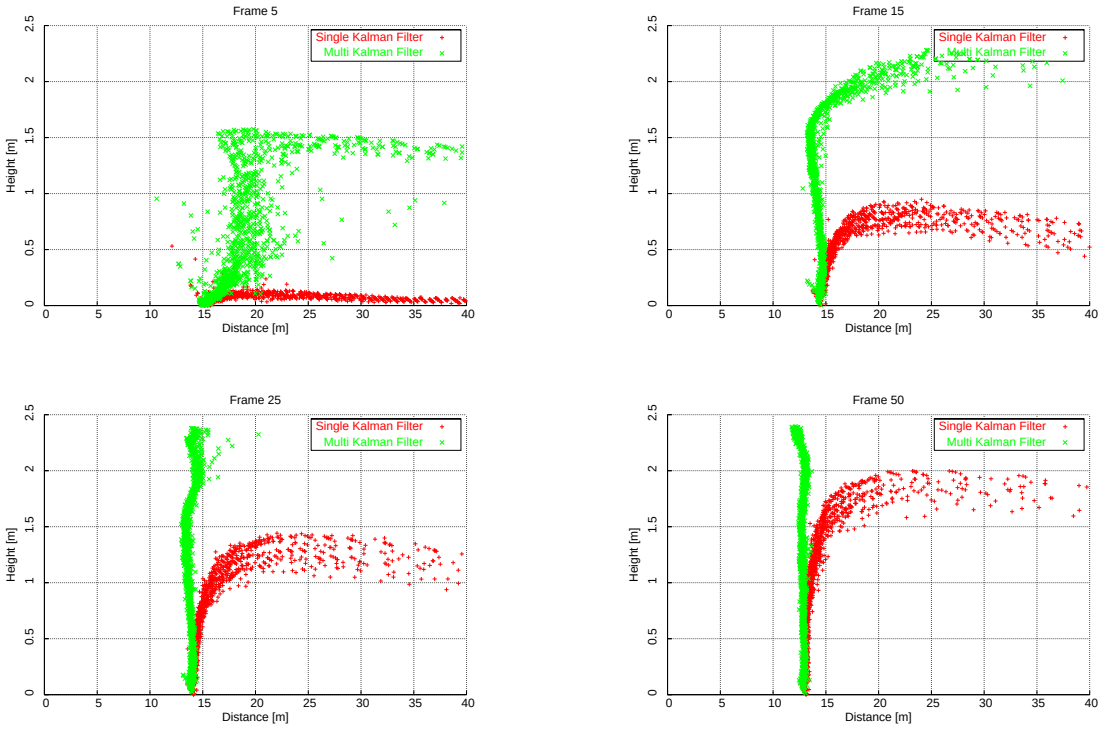


Fig. 5. Estimation results for a vertical plane at frame 5 (0,2 s), 15 (0,6 s), 25 (1,0 s) and 50 (2,0 s)

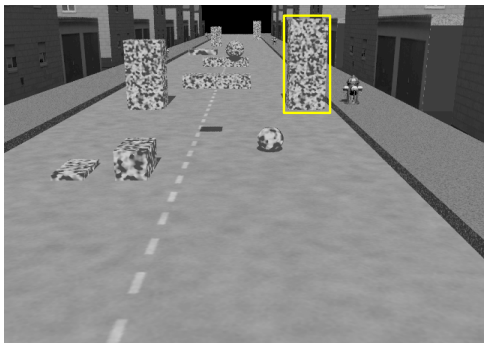


Fig. 4. First frame of a synthetic image sequence. The marked area shows the box used for convergence speed test

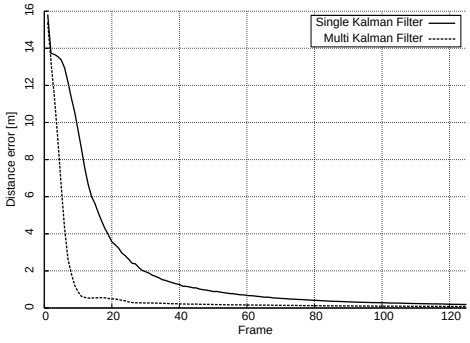


Fig. 6. Median of the absolute distance estimation error of a single and the multi Kalman Filter system

the single filter yields no usable result, the multi filter already gives a rough estimate of the world.

The plots taken after 15, 25 and 50 frames confirm the superiority of the fused filters. The depth accuracy is within 2 % at the final distance of 13 meter. Only for points close to the focus of expansion it takes the same time until the preferred variant yields good results. However, this is problem inherent; the positions of points close to the epipole cannot be recovered if the vehicle is moving along a translational path.

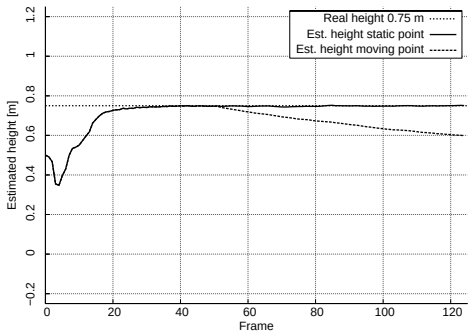
The overall improvement can be well seen if the distance error for all tracked points is considered. Figure 6 shows the median of the absolute distance error as a function of time. Again it becomes obvious that this error decreases much faster if the multi filter approach is used.

The results obtained for two real sequences will be given in chapter 6.

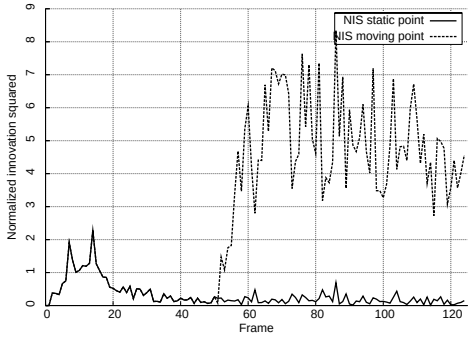
V. Detection of moving objects

So far we assumed the scene to be stationary. How do we have to deal with moving obstacles? It is clear that we have to segment those objects, otherwise we get erroneous results.

Fortunately, the NIS-criterion allows to decide for each point whether it fulfils the stationary-assumption or not. Thus laterally moving objects, e.g. pedestrians crossing the road, can be detected easily. Longitudinal motion, in general, can not be detected due to the well-known motion field ambiguity. Only if the obstacle changes its speed the detection becomes possible. To investigate this, we let the wall in figure 4 start to move at frame number 50 with $0.5\,\frac{\mathrm{m}}{\mathrm{s}}$ in longitudinal direction.



(a) Height estimation



(b) Normalized innovation squared

Fig. 7. Kalman Filter estimating the 3D position of a moving point

Fig. 8. First and last frame of the real sequence

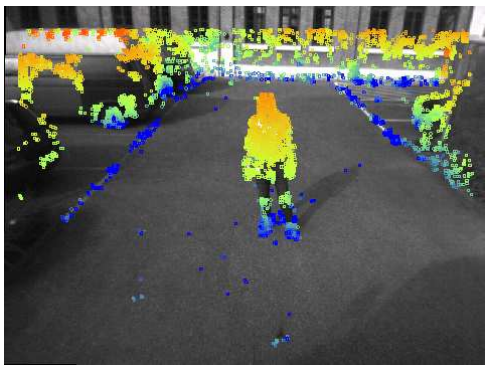


Fig. 9. Estimation result for the person after 10 frames

We consider one point at a height of 0.75 meter and compare the results with the case in which the wall remains stationary.

Figure 7 shows on the left the estimated height. Clearly, after around 20 frames the multi filter has converged to the correct state. When the motion starts, we get a wrong height, as expected. The right graph shows the NIS-values for both cases. No doubt, the motion change can be detected testing the NIS against a properly selected threshold.

VI. Real-life example

In this final section we considered two real-live examples. First, a sequence of 60 frames containing a still obstacle is investigated. Figure 8 shows the first and last frame of this sequence taken from a truck driving at about $1\,\frac{\mathrm{m}}{\mathrm{s}}$. The purpose of this experiment is to stop the vehicle in case of an obstacle. Due to this very slow speed, the lengths of the optical flow vectors are small and the integration performed by the Kalman Filter is of special importance.

In this example, 5000 points are tracked initially. After 10 frames, 4757 points remain, after 25 frames 4499 points are still alive, and after 50 frames 4164 points remain tracked. The used KLT tracker allows the detection of new points continuously, such that the number of tracked points can be kept at a sufficiently high level in practice. However, to analyse the stability of the KLT tracks we disabled this feature in this investigation.

Figure 9 shows the estimated heights after 10 frames (i.e. 40 cm driving distance) qualitatively. We illustrate the increase of height with the warmth of the color. Please note that for all points belonging to a specific objects the height increases with the lines, although there was no constraint put on this.

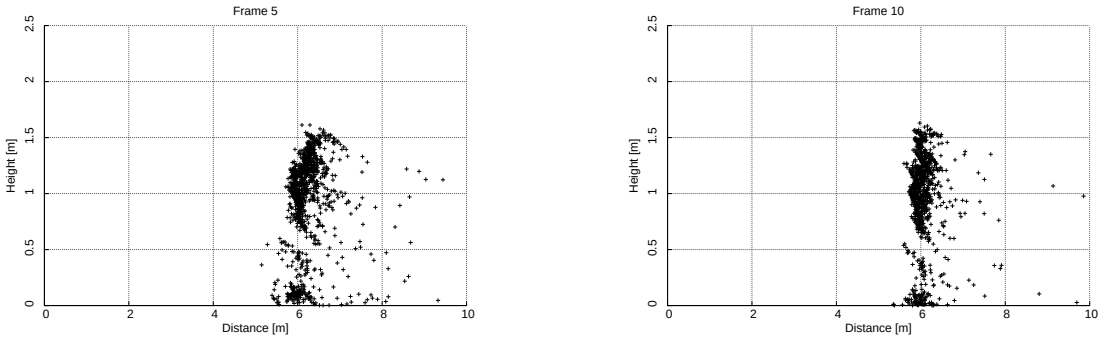


Fig. 10. Estimation result for the person at frame 5 (driven distance 20 cm) and frame 10 (40 cm)

Fig. 11. Free corridor detection

Figure 10 shows the 3D-reconstruction after 5 and 10 frames, respectively. These views confirm the simulation results already presented in figure 5. Indeed, three-dimensional plots show the precise reconstruction of the body. They even reveal that the left leg is closer to the camera than the right one.

The obtained depth accuracy is comparable to stereo results with a small base line. Consequently, the same modules used for segmenting stereo depth maps can be applied to the depth information obtained from monocular vision.

The second example shown in figure 11 is concerned with a typical construction site. Here, the actual lane is limited by beacons and other vehicles in a traffic jam. In contrast to the first example, our car is driving at about $10\,\frac{\mathrm{m}}{\mathrm{s}}$. The same colour scale as in the first example is used to encode the height of the tracked points.

After mapping the 3D-points into a bird's view depth map, the 3D-lane boundaries have been computed and remapped into the image. The red fences show the good results obtained by the presented depth-from-motion algorithm. This means that 3D road boundaries can be detected even if only one camera is installed in the vehicle.

VII. Conclusion

A robust way to solve the 3D-from-motion problem is the usage of Kalman Filters. The paper shows that the proper combination of multiple filters initialized with different states can speed up the rate of convergence by at least a factor of 3-5. If more than three filters are used, an even higher rate of convergence is possible. However, since we want to run the system for obstacle detection in real-time, we restrict ourselves to three filters per pixel. Using a fast implementation of the KLT-tracker, the 3D-positions of 1000 points can be calculated on a 3 GHz Pentium 4 at 16 Hz. This includes the calculation of the structure tensor and the selection of good points on images with VGA resolution.

The used error criterion additionally allows to check each tracked point whether it is in accordance with the stationary assumption. In this paper we exclude moving points from the analysis.

Throughout the paper the motion was assumed to be known. Speed and yaw rate are known from odometry with sufficient accuracy, small errors do not have serious effects. The key problem is pitching. Developments considering this problem are under way.

The described multi filter approach can be used in any Kalman Filter based structure from motion scheme in order to speed up the estimation. Moreover, the principle behind the presented scheme, i.e. the usage of parallel running filter with different initial states, can be transferred to many other Kalman Filter problems that suffer from slow convergence due to the initialization problem.

References

[1] M. Pollefeys, "Self-calibration and metric 3d reconstruction from uncalibrated image sequences," Ph.D. dissertation, Leuven, 1999.

[2] T. Suzuki and T. Kanade, *Method and apparatus for environment recognition*. U. S. Patent Office, Mar 18 2003, patent number 6,535,114 B1.

[3] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, April 1991.

[4] S. Carlsson, "Recursive Estimation of Ego-Motion and Scene Structure from a Moving Platform," *SCIA91*, pp. 958–965, 1991.

[5] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2002.