Robust Monocular Detection of Independent Motion by a Moving Observer

Felix Woelk and Reinhard Koch

Christian-Albrechts-Universität zu Kiel,
Institute für Informatik und Praktische Mathematik,
Ohlshausenstr. 40, 24098 Kiel, Germany
{woelk, rk}@mip.informatik.uni-kiel.de

Abstract. A fast and robust algorithm for the detection of independently moving objects by a moving observer by means of investigating optical flow fields is presented. The detection method for independent motion relies on knowledge about the camera motion. Even though inertial sensors provide information about the camera motion, the sensor data does not always satisfy the requirements of the proposed detection method. The first part of this paper therefore deals with the enhancement of earlier work [29] by ego-motion refinement. A linearization of the ego-motion estimation problem is presented. Further on a robust enhancement to this approach is given.

Since the measurement of optical flow is a computationally expensive operation, it is necessary to restrict the number of flow measurements. The proposed algorithm uses two different ways to determine the positions, where optical flow is calculated. A fraction of the positions is determined by using a sequential Monte Carlo sampling resampling algorithm, while the remaining fraction of the positions is determined by using a random variable, which is distributed according to an initialization distribution. This approach results in a fixed number of optical flow calculations leading to a robust real time detection of independently moving objects on standard consumer PCs.[1][2]

1 Introduction

The detection of independently moving objects by an also moving observer is a vital ability for any animal. The early detection of an enemy while moving through visual clutter can be a matter of life and death. Also for modern humans it is useful, e.g. for collision prevention in traffic. Using the human head as an inspiration, a lightweight monocular camera mounted on a pan-tilt-unit (PTU) is chosen to investigate the environment in this application. The analysis of optical flow fields gathered from this camera system is a cheap and straight forward approach avoiding heavy and sensitive stereo rigs.

The method for the detection of independent motion used in this work relies on the knowledge of the camera motion. Even though inertial sensors provide information about this motion, the accuracy of these sensors does not satisfy the requirements of

[1] This work was supported by BMBF Grant No. 1959156C.

the used detection methods. To overcome this handicap, a novel algorithm for the refinement of the essential matrix is developed and compared against other approaches. Extensive studies about essential matrix estimation [9, 18, 7] and the closely related problem of fundamental matrix computation [26, 9, 15, 16, 27, 8, 32] exist. Further on there is a vast variety of literature about ego-motion estimation [23, 24, 4]. Two other papers from this workshop also deal with ego-motion refinement: Badino [1] fuses data from optical flow algorithm with depth data gathered from a mobile stereo platform for ego-motion estimation and Hengel [10] estimates the relative translational speed from dense optical flow fields.

Since determining highly accurate optical flow with subpixel precision is a computationally expensive operation, restrictions on the maximum number of optical flow computations have to be made in real time environments. The approach chosen in this work is inspired by [12] and determines the sample positions (i.e. points where optical flow will be calculated) partly by using a vector of random variables, which are distributed according to an initialization distribution function (IDF), and partly by propagating samples from the last time step using a sequential Monte Carle sampling resampling approach.

While a wide range of literature on the application of particle filters to tracking tasks [12, 13, 20] and lately on improvements on the particle filter to overcome the degeneracy problem [11, 17, 14, 30] exist, only little work has been done in the field of using such probabilistic techniques for the investigation and interpretation of optical flow fields: In [3] motion discontinuities are tracked using optical flow and the CONDENSATION algorithm and in 2002 Zelek [31] used a particle filter to predict and therefore speedup a correlation based optical flow algorithm.

In the following sections, the basic concept used for the detection of independent motion is explained first. Afterwards an algorithm for the estimation of the necessary camera motion parameters is presented. A sequential Monte Carle sampling resampling approach to speedup and stabilize the detection of independent motion is described next. Finally experiments on synthetic data are shown.

2 Detection of Independently Moving Objects

The basic concepts used for the detection of independently moving objects by a moving observer through investigation of the optical flow are introduced in this section.

Computation of the Optical Flow: A large number of algorithms for the computation of optical flow exist [2]. Any of these algorithms calculating the full 2D optical flow can be used for the proposed algorithm. Algorithms calculating the normal flow only (i.e. the flow component parallel to the image gradient) are, however, inappropriate. The optical flow in this work is calculated using an iterative gradient descend algorithm [19], applied to subsequent levels of an image pyramid.

Detection of Independent Motion: Each optical flow field resulting from a relative motion between the camera and a static object consists of a rotational part and a translational part (Fig. 1). The rotational part is independent of the scene geometry and

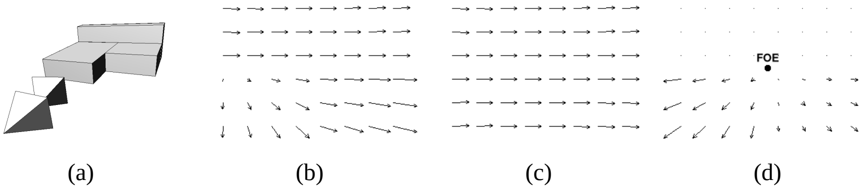

Fig. 1. Theoretical flow fields for a simple scene. The 3D scene is shown at (a). The scene consists of 3 blocks. The camera, displayed as small pyramids, translates towards the blocks while rotating around the y axis. The flow field F as induced by this movement is shown in (b). Its rotational component F_R (c) and translational component F_T (d) with the Focus of Expansion (FOE) are shown on the right.

can be computed from the camera rotation. Subtraction of the rotational flow field from the overall flow field results in the translational flow field, where all flow vectors point away from the focus of expansion (FOE), which can be calculated from the camera motion. With known camera motion, only the direction of the translational part of the optical flow field can be predicted. The angle between the predicted direction and the (also rotation corrected) flow calculated from the two images serves as a measure for independent motion [28] (Fig. 5). In this context, independent motion means motion inconsistent with the camera motion. This detection method requires the exact knowledge of the camera motion. In our approach, the camera motion (relative to the static scene) can be derived from rotation sensor and speed sensor data of the car, or it can alternatively be measured directly from the static scene [21]. A robust approach for the estimation of the necessary camera motion parameters is presented in the next section.

3 Essential Matrix Refinement from Image Point Correspondences

In this section a new iterative linearized method for the estimation of the essential matrix is proposed. First the basic concepts for the description of image relations, the fundamental matrix and the essential matrix, are introduced. After a brief review of 8-point and the Levenberg-Marquard (LM) algorithm, the new iterative linearized approach is derived in detail and compared against the two previous mentioned approaches. In the last part, an enhancement to the new approach is suggested to make it robust against gross errors in the image point correspondences. Large errors in image point correspondences may result from either measurement errors or from the presence of independent motion in the images.

The fundamental matrix F describes the relationship between two views of a static scene. Any point p in the first view is constrained to lie on the epipolar line l in the second view. The fundamental matrix F relates x to l via $l = Fx$. The well known fundamental constraint relates two corresponding image points $\overline{p}'$ and $\overline{p}$ through:

$$\overline{p}'^T F \overline{p} = 0 \quad (1)$$

Although the matrix F has 9 entries it has only 7 degrees of freedom (dof) [9] [3].

The essential matrix E describes the relationship between two views of a static scene if the calibration matrix K

$$K = \begin{pmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

consisting of the focal length f, the aspect ratio a, the skew s and the principal point $(c_x, c_y)^T$ is known. The fundamental constraint (eq. 1) then holds on normalized image coordinates $\overline{p} = K^{-1}p$:

$$p'^T E p = 0 \tag{3}$$

In contrast to F, the essential matrix E has only 5 dof (3 for the rotation and 2 for the translation direction between the two views). If the epipole and the rotation are known, the essential matrix can be computed as follows:

$$E = [e]_\times \cdot R = \begin{pmatrix} 0 & -1 & e_y \\ 1 & 0 & -e_x \\ -e_y & e_x & 0 \end{pmatrix} \cdot R \tag{4}$$

where $(e_x, e_y, 1)^T$ is the epipole in homogenous coordinates, [4] $[e]_\times$ denotes the 3 by 3 skew symmetric matrix with entries from the vector e and R is the rotation matrix. For a thorough discussion on multiple view geometry and projective geometry see [9].

3.1 Comparison of Different Approaches

8 Point: In this approach (which is described in detail in [9]), the 8 point algorithm is used to estimate a first approximation of the essential matrix E. A linear system A in the 9 unknown entries of the essential matrix $f = (E_{1,1}, E_{1,2}, \ldots, E_{3,3})^T$ is constructed using homogenized, normalized image point correspondences $p_n = (x_n, y_n, 1)^T$ and $p'_n = (x'_n, y'_n, 1)^T$ in the fundamental constraint (eq. 1):

$$A \cdot f = 0 \tag{5}$$

where each row A_n of A has the form

$$A_n = (x'_n x_n, x'_n y_n, x'_n, y'_n x_n, y'_n y_n, y'_n, x_n, y_n, 1) \tag{6}$$

After retrieving the least squares solution (subject to $||f||_2 = 1$) of the system (eq. 5), the SVD of the resulting matrix $E' = UDV^T$ is used to enforce the 5 dof constraint by setting the smallest singular value to zero and the remaining two singular values to their mean [5]. Updating the diagonal matrix D with the new singular values leads to D'.

[3] F is scale independent and has rank 2

[4] Homogenous coordinates are members of a projective space $\mathbb{P}$ (in this case $\mathbb{P}^2$). The projective space $\mathbb{P}^2$ is an extension of the Euclidean space $\mathbb{R}^2$.

[5] From the three singular values of an essential matrix, two are equal and the remaining is zero.

Recomputation of the essential matrix with the updated D' leads to the closest essential matrix $E'' = UD'V^T$ to E in Frobenius norm. The epipole e and rotation matrix R can be extracted from E'' by using the SVD of $E'' = UD'V^T$:

$$[e]_\times = UZU^T \qquad \text{and} \qquad R = UWV^T \text{ or } UW^TV^T \tag{7}$$

with the matrices

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \text{and} \qquad Z = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{8}$$

Levenberg-Marquard: In this approach the essential matrix is parameterized using the direction of translation (the angles of the spherical representation of the vector between the two camera centers) and an axis and angle representation of the rotation matrix R. In order to restrict the number of parameters to 5 (the dof of the essential matrix), the angle is encoded using the length of the axis. Two different error functions are investigated here:

- A: The geometric error function uses the sums of the distances between the image point and the corresponding epipolar line as error function.
- B: The algebraic error function uses the residuum resulting from the fundamental constraint (eq. 3) directly.

A starting approximation to the essential matrix needs to be known in both versions A and B of this approach.

Iterative Linearized: The true epipole e and the true rotation R are assumed to be known up to small errors Δe and ΔR:

$$e = e_c + \Delta e \qquad R = \Delta R \cdot R_c \tag{9}$$

The starting approximations e_c and R_c are calculated using the inertial sensors of the car (speed, yawrate, steering angle, pan angle and tilt angle) [28]. If only an approximate essential matrix is known, eq. 7 can be used to extract e_c and R_c from the essential matrix. In order to solve the problem with low computational cost, ΔR is approximated by a Taylor series expansion up to the linear term:

$$\Delta R \approx (I + [r]_\times) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{pmatrix} = \begin{pmatrix} 1 & -r_z & r_y \\ r_z & 1 & -r_x \\ -r_y & r_x & 1 \end{pmatrix} \tag{10}$$

where $r = (r_x, r_y, r_z)^T$ is a vector of the Euler rotation angles around the x, y and z axis resp., I is the identity matrix and $[r]_\times$ again describes a skew symmetric 3×3 matrix with entries from r. The linearization is only valid for small angles r_ν and small epipole error $\Delta e = (\Delta e_x, \Delta e_y, 0)^T$:

$$r_x, r_y, r_z \ll 1 \qquad \Delta e_x, \Delta e_y \ll 1 \tag{11}$$

Using eq. 9 and eq. 10 in fundamental constraint (eq. 3) results in:

$$\begin{aligned} e_y p'_{x,r} - e_x p'_{y,r} + e_x p_y - e_y p_x - e_x r_x - p'_{x,r} p_y + p'_{y,r} p_x + p'_{x,r} r_x + \\ p_x \mathit{\Delta} e_x r_z + p_y \mathit{\Delta} e_y r_z - e_x p'_{y,r} p_y r_x + e_y p'_{x,r} p_y r_x + p'_{y,r} p_x \mathit{\Delta} e_x r_y + \\ \mathit{\Delta} e_y (p'_{x,r} - p_x + p'_{x,r} p_y r_x) + \mathit{\Delta} e_y r_y (-p'_{x,r} p_x - 1) + \\ r_y (p'_{y,r} - e_y + e_x p'_{y,r} p_x - e_y p'_{x,r} p_x) + r_z (e_x p_x + e_y p_y - p'_{x,r} p_x - p'_{y,r} p_y) + \\ \mathit{\Delta} e_x (p_y - p'_{y,r} - r_x - p'_{y,r} p_y r_x) = 0 \end{aligned} \tag{12}$$

with the point correspondence between $p = (p_x, p_y, 1)^T$ and the rotated point $p'_r = R_c * p' = (p'_{x,r}, p'_{y,r}, 1)^T$.

Neglecting the terms with quadratic small values (e.g. $r_z \mathit{\Delta} e_x$) and using at least 5 point correspondences leads to a somehow simpler linear equation system in $\mathit{\Delta} e$ and r:

$$A \cdot \begin{pmatrix} \mathit{\Delta} e_x \\ \mathit{\Delta} e_y \\ r_x \\ r_y \\ r_z \end{pmatrix} = B \tag{13}$$

with $A = (A_1, A_2, \ldots)^T$ consisting of the rows A_i:

$$A_i^T = \begin{pmatrix} p_y - p'_{y,r} \\ p'_{x,r} - p_x \\ p'_{x,r} - e_x - e_x p'_{y,r} p_y + e_y p'_{x,r} p_y \\ p'_{y,r} - e_y + e_x p'_{y,r} p_x - e_y p'_{x,r} p_x \\ e_x p_x + e_y p_y - p'_{x,r} p_x - p'_{y,r} p_y \end{pmatrix} \tag{14}$$

and

$$B = \begin{pmatrix} e_x p'_{y,r} - e_y p'_{x,r} - e_x p_y + e_y p_x + p'_{x,r} p_y - p'_{y,r} p_x \\ \vdots \end{pmatrix} \tag{15}$$

Solving this linear system iteratively and updating the epipole and the rotation matrix

$$e_{c,k+1} = (\mathit{\Delta} e_{x,k}, \mathit{\Delta} e_{y,k}, 0) + e_{c,k} \qquad \text{with } e_{c,0} = e_c \tag{16}$$

$$R_{c,k+1} = R(r_k) \cdot R_{c,k} \qquad \text{with } R_{c,0} = R_c \tag{17}$$

from iteration k to $k+1$ quickly converges against the true solution. Hereby $R(r_k)$ denotes the rotation matrix as composed from the Euler angles contained in the vector r_k. In order to demonstrate the convergence of the algorithm empirically, the mean absolute rotation error $\mathit{\Delta} r_{\text{abs}}$ and the mean absolute epipole error δe_{abs} are used:

$$\mathit{\Delta} r_{\text{abs}} = \frac{1}{N} \sum_{n=1}^{N} ||r(R_c, k, n) - r_{t,n}||_2 \qquad \delta e_{\text{abs}} = \frac{1}{N} \sum_{n=1}^{N} ||e_{c,k,n} - e_{t,n}||_2 \tag{18}$$

where $r_{t,n}$ is the vector consisting of the Euler angles describing the true rotation of the nth run and $r(R_c, k, n)$ is the vector consisting of the Euler angles extracted from the rotation matrix $R_{c,k}$ of the nth run.

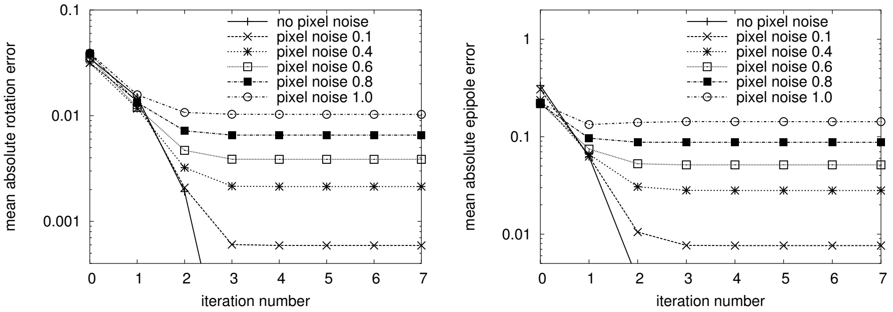

Fig. 2. Convergence of the iterative linearized essential matrix estimation approach on simulated point correspondences. The mean absolute rotation error $\varDelta r_{\text{abs}}$ is plotted against the iteration (left). In the right image, the mean absolute epipole error δe_{abs} is plotted against the iteration. Normal distributed noise with different variances has been added to each of the image point positions.

Figure 2 shows the mean absolute rotation error $\varDelta r_{\text{abs}}$ and the mean absolute epipole error δe_{abs} over 100 runs on simulated point correspondences. The experiment has been carried out several times varying the variances of normal distributed noise added to the image point positions. Fig. 2 shows that a final result is reached in the first 4 iteration steps, even though the quality of the result, especially of the epipole position, is not very high. Further investigations in the next section compare this result to the results from other estimation methods.

Comparison: The error after the final iteration of the proposed linearized iterative approach was compared to the other 3 algorithms (fig. 3) with varying additional noise on the image point positions. For this investigation the absolute mean rotational error $\varDelta r$ and the absolute mean epipole error δe are used:

$$\varDelta r = ||\frac{1}{N}\sum_{n=1}^{N} r(R_c, k, n) - r_{t,n}||_2 \qquad \delta e = ||\frac{1}{N}\sum_{n=1}^{N} e_{c,k,n} - e_{t,n}||_2 \tag{19}$$

Fig. 3 shows the superiority of the LM algorithm with the geometric error function. The accuracy for the epipole position and the rotation error which is achieved by the three algorithms using the algebraic error is within the same magnitude range.

3.2 Robust Essential Matrix Refinement

The algorithms for robust parameter estimation can be sorted into 4 different categories [26, 22, 33], namely:

1. Algorithms that use clustering techniques.
2. M-Estimators use an iterative re-weighting technique to achieve robustness, while using all available data.

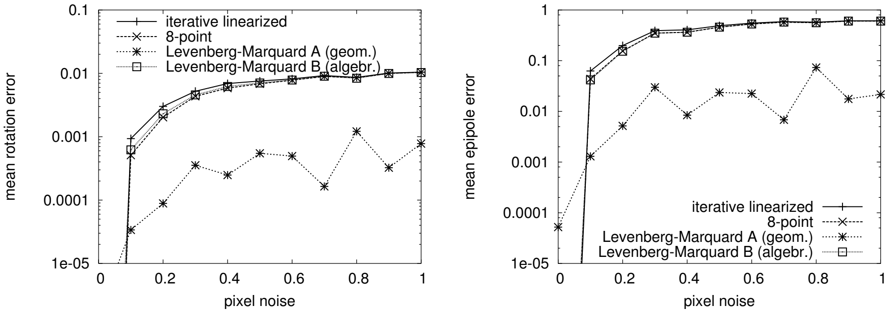

Fig. 3. Comparison of the accuracy of the 4 different approaches described in section 3.1. The mean rotational error Δr (left figure) and the mean epipole error δe (right figure) is plotted against the variance of the normal distributed noise added to the image point positions. The Levenberg-Marquard algorithm with the geometric error function shows superior results to the three other approaches. The error after the final iteration is used in the linearized iterative approach.

3. Case deletion diagnostic algorithms try to identify outliers and reject them from the computation.
4. Algorithms that use random sampling techniques to achieve a solution with a minimal data set.

In [26] a comparison between some of the above mentioned robust estimation methods has been carried out for the case of estimating the fundamental matrix. In this study the best random sampling algorithm (LMedS) shows superior results in comparison to the best member of the M-Estimator class, while the best method in case deletion diagnostics is only slightly worse than the LMedS algorithm. The RANSAC method performs only slightly worse than the LMedS method. In this system the MSAC algorithm, a method similar to RANSAC and LMedS, is chosen. Because the MSAC algorithm is an enhancement to the RANSAC algorithm and shares the main ideas, the RANSAC algorithm is described first:

RANSAC: The RANSAC algorithm [6] is robust in the case of data heavily corrupted with outliers. Assume that there are m data and a minimum of n of them is needed to estimate the parameter vector. The approach works as follows:

- Randomly select a minimum set of n data and extract the parameters x from them.
- Calculate the number k of data from the overall set supporting the parameters x with respect to a given threshold t over the residue r_i.
- If k is bigger than a given fraction, calculate the least squares solution from all data supporting x and exit with success.
- Repeat the above steps L times.
- Either use the parameters with the biggest support k, calculate the least squares solution and exit with success or exit with failure.

This is in fact a search for the solution that minimizes the cost function [25]

$$C = \sum_i \rho(r_i) \qquad \text{with} \qquad \rho(r_i) = \begin{cases} 0 & r_i^2 < t^2 \\ \text{const.} & r_i^2 \geq t^2 \end{cases} \tag{20}$$

The number of trials L needed to ensure at least one outlier free set of data with probability z can be calculated by

$$L = \frac{\log(1-z)}{\log(1-p^n)}, \tag{21}$$

where p is the expected outlier fraction in the data [6].

MSAC: In the RANSAC algorithm, the penalty for each datum is either 0 or some constant value. The residuum associated with the datum is only used to make a binary decision about the penalty. This undesirable situation can be resolved with no extra cost by replacing $\rho(r_i)$ by

$$\rho_2(r_i) = \begin{cases} r_i^2 & r_i^2 < t^2 \\ \text{const.} & r_i^2 \geq t^2 \end{cases} \tag{22}$$

in the cost function (eq. 20) of the RANSAC algorithm [25]. The cosine of the angle between the predicted and the measured flow is used as the residual value in this application.

4 Sequential Monte Carle Sampling Resampling

First the general concept of the CONDENSATION algorithm is summarized. The CONDENSATION algorithm is a particle filter and hence a Sequential Monte Carle Sampling Resampling SMCSR algorithm. Then the application of the SMCSR algorithm for detection of independent motion is described.

4.1 CONDENSATION

The CONDENSATION algorithm is designed to propagate any probability density function (pdf) over time. Due to the computational complexity of this task, pdfs are approximated by a set of weighted samples. The weight π_n is given by

$$\pi_n = \frac{p_z(s^{(n)})}{\sum_{j=1}^{N} p_z(s^{(j)})} \tag{23}$$

where $p_z(x) = p(z|x)$ is the conditional observation density representing the probability of a measurement z, given that the system is in the state x. $s^{(n)}$ represents the position of sample n in the state space.

Propagation: From the known a priori pdf, samples are randomly chosen with regard to their weight π_i. In doing so, a sample can be chosen several times. A motion model is applied to the sample positions and diffusion is done by adding Gaussian noise to each sample position. A sample that was chosen multiple times results in several spatial close samples after the diffusion step. Finally, the weight is calculated by measuring the conditional observation $p(z|x)$ and using it in eq. 23. The a posteriori pdf represented by these samples is acting as a priori pdf in the next time step. This iterative evaluation scheme is closely related to Bayes' law

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)} \tag{24}$$

where $p(z)$ can be interpreted as a normalization constant, independent of the system state x [12]. The sample representation of the a posteriori pdf $p(x|z)$ is calculated by implicitly using the a priori pdf $p(x)$ as the sample base from which new samples are chosen and the probability of a measurement $p(z|x)$ given a certain state of the system x (eq. 23). The sample representation of the a posterior pdf converges “almost sure” against the true a posteriori pdf when using an infinite number of samples [5].

Initialization: In order to initialize without human interaction a fraction of the samples are chosen by using a random variable which is distributed according to an initialization distribution in every time step. With a very high initialization fraction, the samples will most probably no longer represent the correct a posteriori pdf, even with a very large number of samples. The disturbance of the a posteriori pdf representation from the initialization samples can be reduced by using importance sampling [5]. Since this is not (yet) done, the algorithm will be called a sequential Monte Carlo sampling resampling algorithm rather than a particle filter. In the first time step, all samples are chosen using the initialization function.

4.2 Amplified Detection of Independent Motion

Since optical flow (OF) is computationally expensive, the number of OF measurements have to be restricted. However, when computing OF at sparse locations, one would like to capture as much flow on independently moving objects as possible. An adapted particle filter is chosen for this task. In this application the probability for a position belonging to an independently moving object is chosen as the pdf for the CONDENSATION algorithm, resulting in a state dimension of 2. A fraction of the samples are chosen by propagating samples from the last time step using the CONDENSATION approach. Hereby samples are chosen randomly with respect to their weight. Samples with a high weight (a high probability for an independently moving object) are chosen with a higher probability. In general these high weight samples are chosen multiple times, resulting in more samples in the vicinity of the old sample after the diffusion in the next time step. The remaining part of the samples are generated by using a random variable with a distribution depending on the image gradient. OF is measured at each sample position.

Measurement: The measurement at each sample position should represent the probability $p(x)$ that this sample is located on an independently moving object. Let α denote the angle between the predicted translational optical flow pointing away from FOE$_S$ and the rotation corrected OF vector pointing away from FOE$_M$ (see Fig. 5). $c_\alpha = \cos(\alpha)$ is used as a basis for the calculation of this probability [28]. The probability for an independently moving object $p_{c_i}(c_\alpha)$ in dependence of c_α is modeled as a rounded step function:

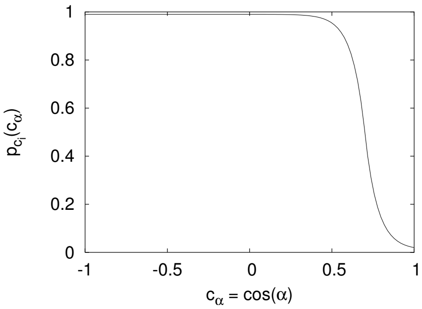

Fig. 4. The probability $p_{c_i}(c_\alpha)$ that a flow measurement is located on an independently moving object in dependence of $c_\alpha = \cos(\alpha)$ at a given inflection point $c_i = 0.7$.

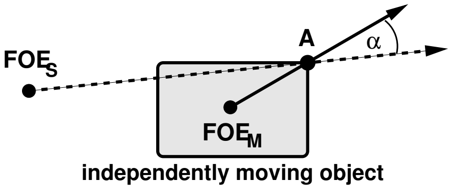

Fig. 5. Detection of moving object by the angle between the predicted flow direction (pointing away from FOE$_S$) and the measured flow direction (pointing away from FOE$_M$).

$$p_{c_i}(c_\alpha) = \begin{cases} e^{f(c_i)\cdot c_\alpha + \ln(0.5) - c_i \cdot f(c_i)} & \text{if } c_\alpha > c_i, \\ 1.0 - e^{-f(c_i)\cdot c_\alpha + \ln(0.5) + c_i \cdot f(c_i)} & \text{if } c_\alpha \leq c_i, \end{cases} \tag{25}$$

where $f(c_i) = \frac{ln(0.01) - \ln(0.5)}{1.0 - |c_i|}$ is a function of the inflection point c_i. Since it is not feasible to set probabilities to 1.0 or 0.0, $p_{c_i}(c_\alpha)$ is scaled and shifted to represent a minimum uncertainty. Fig. 4 shows $p_{c_i}(c_\alpha)$.

In the proposed algorithm, the inflection point is chosen automatically to be $c_i = \tilde{c}_\alpha - \sigma_{c_\alpha}$, where $\tilde{c}_\alpha$ is the median of all the cosine angles not detected as "moving" in the last time step, and σ_{c_α} is the variance of the c_α. Choosing c_i automatically has the advantage, that erroneous camera positions do not disturb the measurement. This only holds under the assumption that more than half of the flow vectors are located on the static scene.

Similar terms ensuring a minimum cornerness p_c (since OF can only be computed with spatial structure), a minimum flow length p_f (standing for the accuracy of the OF computation) and a minimum distance from the focus of expansion p_{FOE} (since errors in the FOE position influence the direction prediction for closer points more than for points further away) are introduced. The overall probability $p(x) = p(z|x)$ is then given by:

$$p(x) = p_{c_i}(c_\alpha) \cdot p_c \cdot p_f \cdot p_{FOE} \tag{26}$$

In order to further stabilize the result, spatial and temporal filters are applied to the resulting probability images [29].

5 Experiments

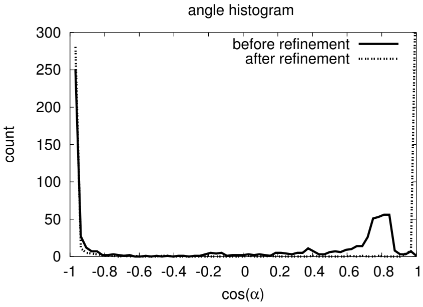

Fig. 6. Histogram over the cosine of the angles between the predicted and the measured flow vectors before and after refinement. The histogram after refinement shows that a lot more flow vectors have the expected value of 1.0, even though the flow vectors located on an independent moving object still have a significant angle to the predicted flow direction.

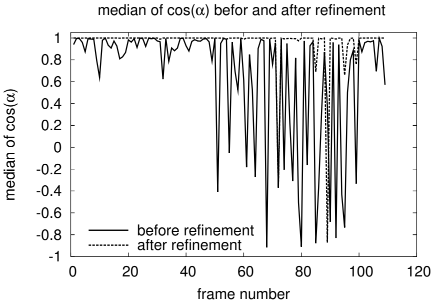

Fig. 7. The median of the cosine of the angles between the predicated and the measured flow vectors is evaluated before and after refinement over a synthetic image sequence.

To investigate the results of the essential matrix estimation process, the histogram over all angles between the predicted and the measured direction of the translational flow component is used (fig 6). Fig 6 shows that most flow vectors have the expected direction ($\cos(\alpha) \approx 1.0$) after the refinement process, in contrast to before. At the same time, the flow vectors lying on independently moving objects still have a significant angle to the predicted translational flow ($\cos(\alpha) \approx -1.0$). As a simple way to score these histograms, the median of all angles is chosen. The closer the median lies to the expected value of 1.0, the better is the given essential matrix under the assumption that more than half of the flow measurements are located on points belonging to the static scene. To test the algorithm a simulated street intersection was realized in VRML. Simple block models of houses, textured with real image data, are located on the corners of the intersecting street (Fig. 8). A model of a car was used as an independently moving object. Screenshots of a ride through this intersection provided the image data, while the sensor information was calculated from the known camera pose and parameters. The uncertainty of the camera pose was modeled as additional normal distributed noise on the camera position and rotation. In fig. 7 the median of the cosine of all angles between the predicted and the measured translational flow vectors is evaluated over the synthetic image sequence. In most of the images, the correct essential matrix is estimated (median = 1.0) while in some images an incorrect estimate of the essential matrix was estimated. This behavior results from the restricted number of tries in the MSAC algorithm, and vanishes (at the expense of additional computation time) when more tries are conducted.

The resulting detector output is shown in fig. 8. Points where the spatio-temporal filter output is above a certain threshold are marked with white blobs.

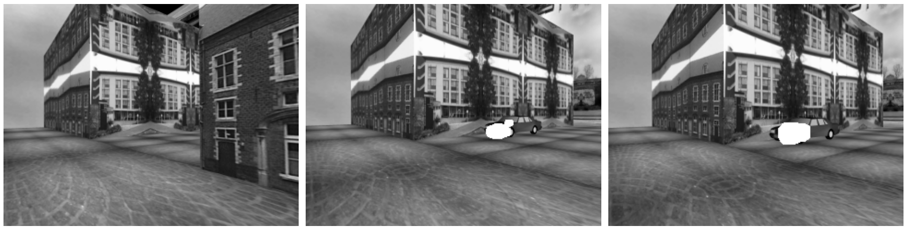

Fig. 8. Some images from the synthetic intersection sequence. The camera is moving on a straight line, while the car in the image is on a collision course. Points where the filter output is above a threshold of 0.35 are marked white.

6 Conclusions and Further Work

A fast and robust sequential Monte Carlo sampling resampling system for the detection of independently moving objects by a moving observer has been presented. The robustness against single measurement errors in the optical flow mainly results from the sequential Monte Carlo sampling resampling approach, while its robustness against erroneous camera pose data results from the chosen random sampling consensus approach. Experiments with synthetic were accomplished. Further work will include:

- experiments on real data
- run time optimization
- clustering of detected independently moving points
- investigation of the trajectory extraction possibility of moving objects
- enhanced fusion of inertial sensor information (speed, yawrate and steering angle) with image based measurements (optical flow from static scene).

References

1. H. Badino, “A Robust Approach for Ego-Motion Estimation Using a Mobile Stereo Platform.” *Proc. IWCM*, 2004.
2. J.L. Barron, D.J. Fleet, S.S. Beauchemin and T.A. Burkitt, “Performance Of Optical Flow Techniques”, *Proc. CVPR*, Vol. 92, pp. 236-242, 1994.
3. M.J. Black and D.J. Fleet, “Probabilistic Detection and Tracking of Motion Discontinuities.” *ICCV*, 1999.
4. A.R. Bruss and B.K.P. Horn, “Passive Navigation.” *CGIP*, 21:3-20, 1983.
5. A. Doucet, N.d. Freitas, N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.
6. M. Fischler and R. Bolles. “RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography.” *Commun. Assoc. Comp. Mach. 24* , 1981.

7. R. Hartley, “Computation of the essential matrix from 6 points.” http://rsise.anu.edu.au/ hartley/Papers/6pt/6ptnew.pdf
8. R. Hartley, “In defence of the eight-point algorithm.”, *IEEE TPAMI*, 19(6):580-593, 1997.
9. R. Hartley and A. Zisserman, *Multiple View Geometry*, Cambridge University Press, 2000.
10. A.v.d. Hengel, W. Chojnacki, M.J. Brooks, “Determining the Translational Speed of a Camera from Extended Optical Flow.” *Proc. IWCM*, 2004.
11. C. Hue, J.-P. Le Cardre and P. Perez, “Tracking Multiple Objects with Particle Filtering.” *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):791-812, 2002
12. M. Isard and A. Blake “Condensation – conditional density propagation for visual tracking.” *IJCV*, 29(1), pp.5-28, 1998.
13. M. Isard and A. Blake “ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework.” *ECCV*, vol. 1 893-908, 1998.
14. M. Isard, J. McCormick, “BraMBLe: A Bayesian Multiple-Blob Tracker.” *ICCV*, 2001.
15. K. Kanatani, “Optimal fundamental matrix computation: Algorithm and reliability analysis.” *SSII*, 2000.
16. K. Kanatani et al., “Fundamental matrix from optical flow: Optimal computation and reliability evaluation.” *JEI*, 2000.
17. Z.Khan, T. Balch and F. Dellaert, “An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets.” *ECCV*, 2004
18. H.C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections.” *Nature*, Vol. 293:10, 1981.
19. B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision” *Proc. DARPA IU Workshop*, pp. 121-130, 1981.
20. P. Perez et al., “Color-Based Probabilistic Tracking.” *ECCV*, 2002.
21. M. Pollefeys, R. Koch and Luc J. Van Gool, “Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters”, *IJCV*, 32(1):7-25, 1999.
22. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection.* John Wiley and Sons, 1987.
23. T. Suzuki and T. Kanade, “Measurement of vehicle motion and orientation using optical flow.” *IEEE ITS*, Tokyo, 1999.
24. T.Y. Tian, C. Tomasi and D.J. Heeger, “Comparison of robust approaches to ego-motion computation.” *CVPR*, 1996.
25. P.H.S. Torr and A. Zisserman, “MLESAC: A new robust estimator with application to estimating image geometry.” *CVIU*, 1996.
26. P.H.S. Torr and D.W. Murray, “Development and Comparison of Robust Methods for Estimating the Fundamental Matrix.” *IJCV*, 1996.
27. R.Y. Tsai and T.S. Huang, “The perspective view of three points.”, *IEEE TPAMI*, 6:13-27, 1984.
28. F. Woelk, S. Gehrig and R. Koch, “A Monocular Image Based Intersection Assistant.” *IEEE Intelligent Vehicles*, Parma, Italy, 2004.
29. F. Woelk and R. Koch, “Fast Monocular Bayesian Detection of Independently Moving Objects by a Moving Observer.” *DAGM*, pp:27-35, Tübingen, Germany, 2004.
30. J. Vermaak et al., “Maintaining Multi-Modality through Mixture Tracking.” *ICCV*, 2003.
31. J.S. Zelek, “Bayesian Real-time Optical Flow.” *Proc VI*, 2002.
32. Z. Zhang, “Determining the epipolar geometry and its uncertainty - a review.” *IJCV*, 27(2):161-195, 1998.
33. Z. Zhang, “Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting.” http://www-sop.inria.fr/robotvis/personnel/zzhang/Publis/Tutorial-Estim/Main.html, 1996.