# Fast Monocular Bayesian Detection of Independently Moving Objects 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 algorithm for the detection of independently moving objects by an also moving observer by means of investigating optical flow fields is presented. 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 part of the positions is determined using a particle filter, while the other part of the positions is determined using a random variable, which is distributed according to an initialization distribution. This approach results in a restricted 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.

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 [8] 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 particle filter approach.

While a wide range of literature on the application of particle filters to tracking tasks [8, 9, 12] and lately on improvements on the particle filter to overcome the degeneracy problem [5, 6, 10, 15] 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 [2] motion discontinuities are tracked using optical flow and the CONDENSATION

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

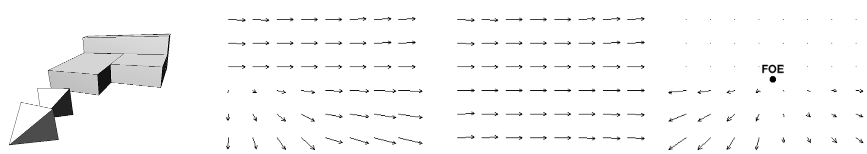
algorithm and in 2002 [16] used a particle filter to predict and therefore speedup a correlation based optical flow algorithm.

In the following sections, the basic concept used for the detection of independent motion is explained first. The particle filter system used to speedup and stabilize the detection of independent motion is developed next. Finally experiments with synthetic and real 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 [1]. 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 component parallel to the image gradient) are, however, inappropriate. The optical flow in this work is calculated using an iterative gradient descend algorithm [11], applied to subsequent levels of a Gaussian image pyramid.




(a) (b) (c) (d)

**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.

**Detection of Independent Motion:** Optical flow fields consists of a rotational part and a translational part (Fig. 1). The rotational part is independent of the scene geometry and 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 [14](Fig. 2). This detection method requires the exact knowledge of the camera motion. In our approach, the camera motion can be derived from rotation sensor and speed sensor data of the car, or it can alternatively be measured directly from the static scene [13].

## 3 Particle Filter

First the general concept of the CONDENSATION algorithm is summarized. Then the application of a particle filter for detection of independent motion is described.

### 3.1 CONDENSATION

The CONDENSATION algorithm is designed to handle the task of propagating 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 $\pi_n$ is given by

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

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 $\pi_i$. 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. 1. 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

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)} \tag{2}$$

where $p(z)$ can be interpreted as a normalization constant, independent of the system state $x$ [8]. The sample representation of the 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. 1).

**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. In the first time step, all samples are chosen in this manner.

### 3.2 Bayesian Detection of Independent Motion

First an overview over the proposed algorithm is given, then the algorithm is explained in detail.

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 using 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 possibility. 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.

**Modifications of the standard CONDENSATION algorithm:** A number of adaptations have been made to the CONDENSATION algorithm to ensure faster processing and optimization of the sample positions for the flow measurements:

- *Initialization Function:* The measurement of OF is only possible on regions with spatial structure. The lower eigenvalue of the structure tensor or "cornerness" [3] is chosen as initialization distribution density function (IDF). By placing samples randomly with respect to this IDF, the majority of the samples are located on positions with high "cornerness" and hence giving optimal conditions for the calculation of the OF. Due to the random nature of the sample placing, some samples are however placed in regions with lower spatial structure, giving less optimal conditions for OF calculation, but on the other hand allowing the detection of independently moving objects in these regions. Obviously, there has to be a lower bound on the minimum spatial structure necessary for OF calculation. To ensure a fast detection of moving objects, the fraction of samples positioned in this way is chosen to be as high as $0.7$. This high initialization fraction obviously disturbs the posterior pdf, but on the other hand improves the response time of the detector. The high fraction of samples generated by using the IDF also reduces the degeneracy problem of particle filters.
- *Discretizing of the State Space:* The sample positions are discretized, i.e. a sample cannot lie between pixels. This leads to the fact that multiple samples are located on the same location in state space, i.e. on the same pixel. Obviously only one expensive measurement of OF is necessary for all those samples located on the same pixel. This leads not to a reduction of the sample numbers, but only to a reduction of the necessary measurements and probability calculations (typically by 25%) and therefore speeds up the process.
- *Motion Model:* In the special case of applying Bayesian sampling to locations of OF measurements, no motion model of the underlying process is needed, because every measurement (i.e. optical flow = apparent motion of a point between two consecutive frames) represents the motion of the according sample itself. The new sample position can be predicted by using the old position and adding the OF measured at this position.
- *Non-Isotropic Diffusion:* In typical traffic situations, large portions of the images are very low structured (e.g. the asphalt of the road), therefore a modified diffusion step is used to increase the number of sample positions on structured image regions: A pointwise multiplication of the standard 2D Gaussian function with the cornerness in a window around the actual position is used as the diffusion density function. The window size is determined by the variances of the diffusion. Choosing randomly once with respect to this density results in the new 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 $\alpha$ 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. 2). For speed reasons, $c_\alpha = \cos(\alpha)$ is used as a basis for the calculation of this probability [14]. The probability for an independently moving object $p_{c_i}(c_\alpha)$ in dependence of $c_\alpha$ is modeled as a rounded step function:

$$p_{c_i}(c_\alpha) = \begin{cases} e^{f(c_i)\cdot c_\alpha + \ln(0.5) - c_i \cdot f(c_i)} & \text{if } c_\alpha > c_i, \\ 1.0 - e^{-f(c_i)\cdot c_\alpha + \ln(0.5) + c_i \cdot f(c_i)} & \text{if } c_\alpha \leq c_i, \end{cases} \tag{3}$$

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. 3 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 the all cosine angles not detected as "moving" in the last time step, and $\sigma_{c_\alpha}$ is the variance of the $c_\alpha$. 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 further points) are introduced. The overall probability $p(x) = p(z|x)$ is then given by:

$$p(x) = p_{c_i}(c_\alpha) \cdot p_c \cdot p_f \cdot p_{FOE} \tag{4}$$

**Spatio Temporal Filtering** In order to detect whether independently moving objects are present, the sampled observation density is investigated. An outlier observation density image is constructed by superimposing Gaussian hills with a given sigma for all sample positions. In order to further increase the robustness a temporal digital low pass filter is used on the outlier observation image density sequence. A user selected threshold on the output of this filter is used to mark independently moving objects.

## 4 Experiments

Experiments were carried out using synthetic images and sensor information as well as images and sensor data gathered with the Urban Traffic Assistant (UTA) demonstrator from the DaimlerChrysler AG [4].

**Simulated Data** To test the algorithm a simulated 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. 4). 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 parameters at the time of the screenshots. Fig. 4 shows some images from the simulated image sequence. Points where the spatio-temporal filter output is above $0.35$ are marked with white blobs. Only very few points are detected because the synthetic car color is uniform due to the simple texture model.

**Real Data** The setup of UTA [4] includes a digital camera mounted on a pan-tilt-unit (PTU), GPS, map data, internal velocity and yawrate sensors, etc. The fusion of GPS and map data will be used to announce the geometry of an approaching intersection to the vision system. The camera then focuses on the intersection. Using the known egomotion of the camera, independently moving objects are detected and the driver's attention can be directed towards them. Fig. 5 shows the results on a real world image sequence.

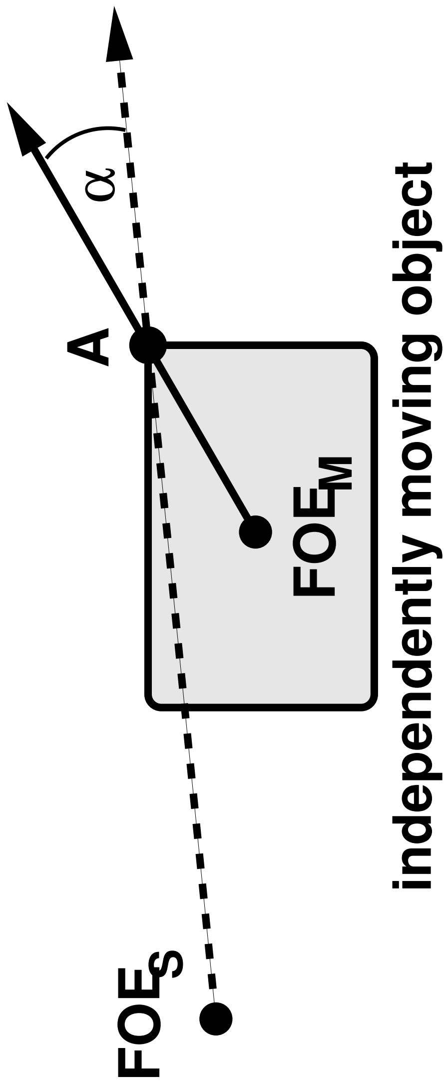



**Fig. 2.** Detection of moving object by the angle between the predicted flow direction (pointing away from FOE$_S$) and the measured flow direction (pointing away from FOE$_M$).

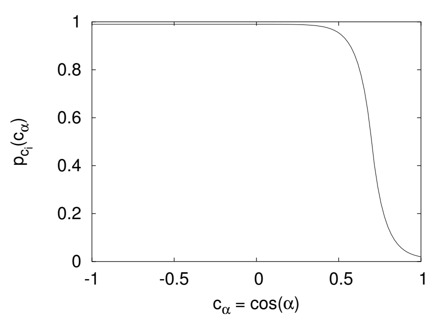



**Fig. 3.** The probability that a flow measurement is located on an independently moving object $p_{c_i}(c_\alpha)$ in dependence of $c_\alpha = \cos(\alpha)$ at a given inflection point $c_i = 0.7$.

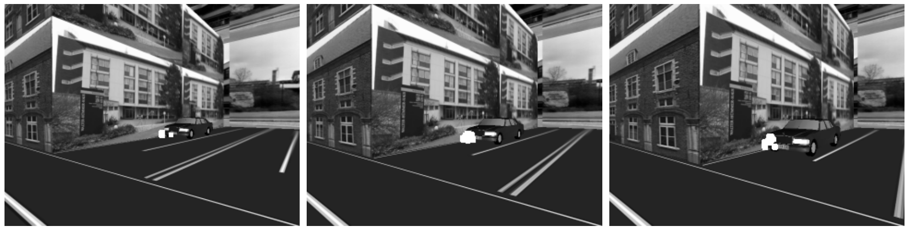


**Fig. 4.** Some images from the synthetic intersection sequence. The camera is moving on a straight line, while the car in the image is on a collision course. Points where the filter output is above a threshold of $0.35$ are marked white.

**Fig. 5.** Some images from a real intersection sequence. Points where the spatio-temporal filter output is above $0.35$ are marked white.

**Timing** The computation frequency is $18.2 \pm 2.0$ frames per second (fps) for the synthetic sequence and $20.0 \pm 2.4$ fps for the real world sequence. These timings were measured on a standard 2.4 GHz Pentium IV PC with an overall number of $1000$ samples. The optical flow used a pyramid of size $3$.

**Detection Rates** The detection rates and false positive rates were calculated on a pixel basis using a known image segmentation: For every pixel where the optical flow has been calculated, it is determined whether it is a false positive or a true detection, resulting in a detection rate of 100 % when every flow measurement on the moving object is detected as such by the algorithm. In the case of synthetic image data, the segmentation could be derived from the known 3D scene structure, in the case of the real world sequence, the image was segmented by hand. Several rectangles thereby approximated the moving object. Fig. 6 shows that a high detection rate combined with a low (nearly zero) false positive rate could be obtained with the chosen approach. The remaining false positive rate results from the spatio temporal filtering of the results. All false positives are located spatially very close to the moving object. Since the camera and the moving object were on collision course, independent motion was detected mainly at the object boundaries (car front). In the parts of the sequence where no moving object was visible the false positive rate stayed very low, causing no false object alarms. The evaluation of the real sequence showed essentially the same behavior and proves the robustness against noise.

## 5 Conclusions and Further Work

A fast and robust Bayesian based system for the detection of independently moving objects by a moving observer has been presented. The two advantages motivating the chosen approach lead to a very fast and robust algorithm:

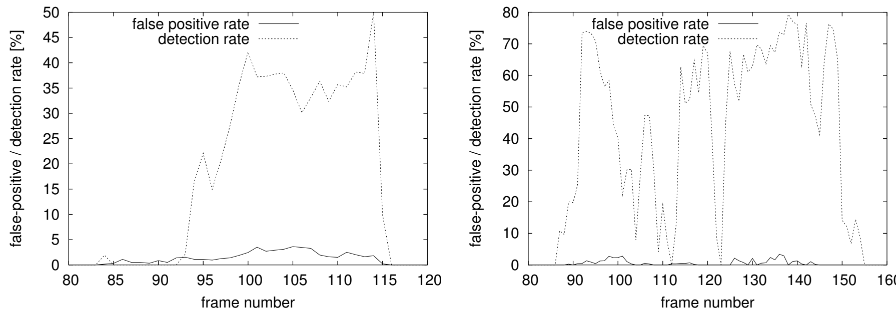



**Fig. 6.** False positive and detection rates for a synthetic (left) and real world (right) sequence. The ground truth image segmentation needed for obtaining these rates was known in the synthetic case and was generated by hand in the real world case. The moving object was approximated in the real world case by several rectangles. In the synthetic sequence (A) moving objects were visible between frame 72 and frame 117. In the real world image sequence (B), an image sequence of 80 images was evaluated. See text for further details.

1. By choosing the IDF to depend on the image gradient, most samples are positioned in high contrast regions resulting in optimal conditions for the calculation of optical flow. Because the IDF is however only the distribution function for the randomly chosen sample positions, their positions are not restricted to high contrast regions, but some of them are also positioned in lower contrast regions. This allows the detection of independently moving objects also in these lower contrast regions, while at the same time a maximum sample number and therfore a maximum computation time is guaranteed.
2. The use of the particle filter approach leads to a clustering of flow measurements in regions where independent motion was detected in the last time step. The surrounding flow measurements can be used to either confirm or reject the existence of an independently moving object by using a spatio-temporal filter.

Experiment with synthetic and real image data were accomplished. Further work should include:

– investigation of the trajectory extraction possibility of moving objects
– fast robust egomotion estimation refinement by fusing sensor information (speed, yawrate and steering angle) with image based measurements (optical flow from static scene)

## References

1. 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.
2. M.J. Black and D.J. Fleet, “Probabilistic Detection and Tracking of Motion Discontinuities.” *ICCV*, 1999.
3. W. Förstner,, “A feature based correspondence algorithm for image matching.”, *International Archives of Photogrammetry and Remote Sensing*, 26-3/3. pp. 150-166, 1986.
4. S. Gehrig, S.Wagner and U. Franke, “System Architecture for an Intersection Assistant Fusing Image, Map and GPS Information” *Proc. IEEE Intelligent Vehicles*, 2003.
5. 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

6. Z.Khan, T. Balch and F. Dellaert "An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets." *ECCV*, 2004
7. R. Hartley and A. Zisserman, *Multiple View Geometry*, Cambridge University Press, 2000.
8. M. Isard and A. Blake "Condensation – conditional density propagation for visual tracking." *IJCV*, 29(1), pp.5-28, 1998.
9. M. Isard and A. Blake "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework." *ECCV*, vol. 1 893-908, 1998.
10. M. Isard, J. McCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker." *ICCV*, 2001.
11. B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision" *Proc. DARPA IU Workshop*, pp. 121-130, 1981.
12. P. Perez et al., "Color-Based Probabilistic Tracking." *ECCV*, 2002.
13. 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.
14. F. Woelk, S. Gehrig and R. Koch, "A Monocular Image Based Intersection Assistant." *IEEE Intelligent Vehicles*, Parma, Italy, 2004.
15. J. Vermaak et al., "Maintaining Multi-Modality through Mixture Tracking." *ICCV*, 2003.
16. J.S. Zelek, "Bayesian Real-time Optical Flow." *Proc VI*, 2002.