A comparison of visual feature tracking methods for traffic monitoring

Roman Peter Pflugfelder

Vienna University of Technology,
Pattern Recognition and Image Processing Group
email: romanp@prip.tuwien.ac.at
www: http://www.prip.tuwien.ac.at/˜romanp

Abstract. Many visual feature tracking methods have been proposed in the last decade. They all deal with pursuing objects of interest under different constraints like reliability, stability and real-time capability. The enormous progress in computer technology and a wide range of technical applications have led to an interest in visual feature tracking. The vast literature is the proof of intensive research. But papers that classify these methods and explain the advantages and disadvantages in the scope of the application are rare. This paper seeks to close this gap between general expert knowledge and the requirements of practitioners who want to use visual feature tracking for their own work. Therefore the paper presents the requirements of tracking in general and in particualar in traffic monitoring. It deals with three design decisions which engineers who want to build a tracker have to make. Different methods and models are explained in general. Three of them are compared for traffic monitoring.

1 Introduction

Visual feature tracking means pursuing image features or user-defined feature models between two consecutive images. This is done over a long sequence of images. These images are frames of a video or are captured in real-time by an acquisition system. The main problem is to match the feature in the first frame to a collection of measurements taken from the next. This matching problem is equal to the correspondence problem, which is well known in 3D computer vision.

The interest of researchers and practitioners in the field of visual feature tracking has grown rapidly in the last decade. First, the enormous progress in computer technology makes real-time and computational intensive applications like visual tracking possible. Second, a wide range of technical applications need visual tracking to solve specific problems. Automatic observation of buildings, garages and streets or modern access control systems using lip movements to identify people are possible applications. In medicine, scientists analyze the way people with artificial limbs walk. The aim of the research is to develop new therapeutic methods so that patients have quicker success in learning to walk after accidents. Also new interfaces for communication would be possible. Paraplegic people stand to gain from these developments. Robotics and production engineering are also aware of the potential role of visual tracking. Robots have been constructed which automatically recognize, pursue and manipulate objects. Thus, the next step towards more intelligent production processes can be taken. The paper confines itself to the field of traffic monitoring. Monitoring is able to measure the velocity of vehicles, the average velocity of the traffic, the state of the traffic flow (e.g. obstruction), the number of lane changes and more. Parameters like average velocity or number of vehicles per hour are also determinable by other types of sensors. However, until now, the recognition of more “symbolic” information like congestion, obstruction or lane changes has been impossible to perform. This flexibility and large potential to determine many different parameters is the advantage of visual tracking. Furthermore, the costs of installation and hardware are significantly lower than those of conventional sensor techniques.

The paper is divided into three sections. Section 2 is concerned with a general discussion of visual feature tracking methods. The requirements of tracking are mentioned. Sections 2.1, 2.2 and 2.3 show that engineers have to make three design decisions when building a tracking system for a specific application. Different approaches and models are examined within this scope. Section 3 looks more into tracking for traffic monitoring. It first presents the key requirements in that field (Section 3.1) and then points out advantages and disadvantages of some methods (Section 3.2). Lastly, section 4 compares the discussed methods and models and investigates general hypotheses with respect to traffic monitoring.

The paper does not examine tracking methods beyond visual feature tracking. Thus all methods based on a dense motion field like optical flow calculations, for example, are not within the scope of the discussion. Block or mesh tracking are also not dealt within this paper. All these methods are computationally expensive, because of their complexity and therefore are seldom used in applications with real-time constraints. A typical application for optical flow techniques is for instance video encoding.

2 Visual feature tracking

The aim of tracking is to pursue objects of interest through a long sequence of images. The requirements of the tracker are

1. reliability in detecting the objects of interest,
2. stability during the tracking,
3. accuracy in predicting the object's state (e.g. position, velocity in a specific frame),
4. real-time capability, which is a prerequisite in many applications.

Every engineer has to consider **three design decisions** to fulfill these requirements. First, well suited *features* have to be defined to reliably detect the objects of interest and then to guarantee a stable tracking. Next, the right choice of a *motion model* should enable the tracker to take new measurements of these features with high accuracy in an area of the next image. This area has to be as small as possible as the tracker works in real-time. Lastly, a *matching and state update algorithm* has to be chosen with respect to the application. It is the core task of every tracker to match measurements and predictions as accurately as necessary to ensure stability. Possible examples of these three properties are discussed in the next sections.

2.1 The choice of a feature

A taxonomy is given in figure 1. The literature distinguishes between image features and feature models. Image features are extracted by image processing methods. Shi and Tomasi [18] propose interest points with the property of high texturedness in their neighborhood. As they show, these points are corners in the image. Fusiello et al. [5] and Beymer et al. [2] also use this feature for tracking. Manku et al. [13] try the Harris detector for corner extraction. Hager and Toyama [9] support corners in their XVision environment.

Patches are highly related to the Shi, Tomasi interest points, because they should also exhibit the property of high texturedness. Otherwise they may not be found in the next image. The work of Hager and Belhumeur [8] describes a very efficient real-time patch tracking algorithm. Eklund et al. [3] have also designed a patch tracker.

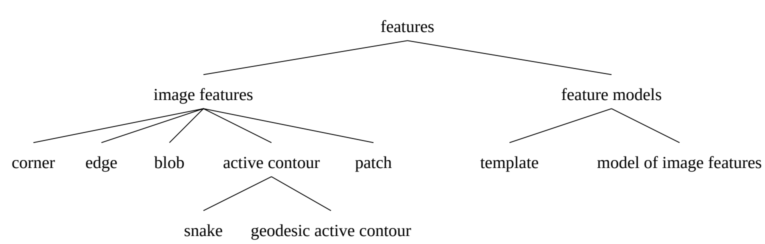

Figure 1. Feature models

Edges are not as popular as corners or patches due to the aperture problem. Thus motion can only be estimated perpendicular to the image gradient, if the area of concern is too small. Mostly, they are used as subfeatures of feature models. See [9][14] for details.

Blobs are an important group of image features. They are widely used in tracking applications due to the simplicity of the method and low computational expense. Background differencing is well known as a method of segmenting the image and getting the blobs for tracking. The static background is subtracted from every image. The blobs which survive are regions of motion [12]. Stauffer and Grimson [19] propose another method for segmenting blobs of motion and background. They model every pixel intensity in the image by a mixture of gaussians. Outliers can then be classified as foreground. These pixels form blobs for tracking. Garg et al. [6] consider the segmentation as a classification problem. They use a support vector machine to discriminate

between blobs and background. Lastly, Shi and Malik [17] present a completely different approach to motion segmentation. They construct a graph of pixels in their spatiotemporal vicinity. Partitions of the image with and without motion are found by using normalized cuts.
Active contours are the last group of image features to mention. Classical snakes are a well known representative [7][4] [11][9][12]. They are parametric curves and approximate the real contour of an object by minimizing an energy function. Paragios and Deriche [15] present level set geodesic contours for tracking. Geodesic active contours are comparable to snakes due to the fact that they do not depend on the curve parameterization. Level set theory allows the representation of the evolution of parametric curves as a three dimensional surface function. The evolution of curves basically indicates motion. The key point is that changes in topology of the curve are simply handled by the surface function. Both results are combined in a challenging parameter and topology-free model of active contours.
Beside image features, the second class of features are feature models. Feature models are based on image features. Templates and models of image features are two possible groups. The former can be a patch of edges or a hand-drawn model of an object while the latter is for instance a set or a graph of corners or edges. Matching in this class means achieving model correspondence. In the case of graph models, the image features and the graphs themselves have to match. Examples can be found in [14][10].

2.2 The choice of a motion model

Computer vision tasks are always time expensive. Tracking applications have to run in real-time in most cases. Therefore, the area where measurements are taken has to be limited in the next image. Motion models of the tracked object are used to overcome this problem.
The user can determine the model or it can be learned from training sequences [11]. The literature mentions two classes of motion models as figure 2 shows. Linear models are further divided into translational and affine models. The former consists of the translational vector and has two parameters of freedom. The latter is the most general linear model consisting of translational vector, rotation, scale and shear and has full six-parameter freedom. The second class are nonlinear models. They are rarely used because the displacement of the same feature is only a few pixels between two frames of an image sequence in most cases and the linear assumption of motion holds. Furthermore nonlinear models are expensive to compute and not adequate for real-time requirements.

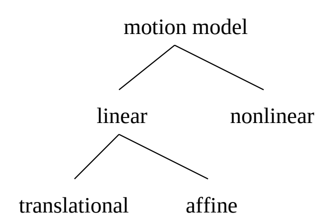

Figure 2. Motion models

2.3 The choice of a matching and state update model

Up to now, the questions of which features to pursue and where the measurements should be taken have been answered. The last stage of tracking is to perform the matching between the features and measurements and to examine how this state update happens. The state of a tracked object is normally a combination of position and velocity. As figure 3 shows, three classes of matching and update models exist.
Probabilistic methods model the state by a probability density function (pdf). The update is identified by deterministic drift, stochastic diffusion and reactive effect of measurements [11]. The first step determines the motion model, whereas the second models its uncertainty. Lastly, measurements are taken and influence the state pdf as well. The literature distinguishes between single and multiple hypothesis probabilistic matching and update models. The well known Kalman filter is a classical single hypothesis method. It means that at every time step, the method only represents one hypothesis of the state of a tracked object. The consequence is that the state pdf must be unimodal and is approximately a Gaussian. In many applications this assumption does not hold because there is high background clutter. Therefore many researchers have worked in the last years on multiple hypothesis methods which approximate a multimodal state pdf. This can be done by a mixture of gaussians like the extended Kalman filter, Probabilistic Data Association Filter (PDAF) or Joint PDAF (JPDAF) [1] with the general disadvantage that the data association problem is not avoided. A better idea is to collect samples of the multimodal state pdf. By generalizing the state pdf, the data association problem does not occur anymore. The Condensation algorithm is one implementation of this idea which uses factorized

sampling to obtain the examples which describe the pdf. Results show that this method is very promising with respect to background clutter [11].

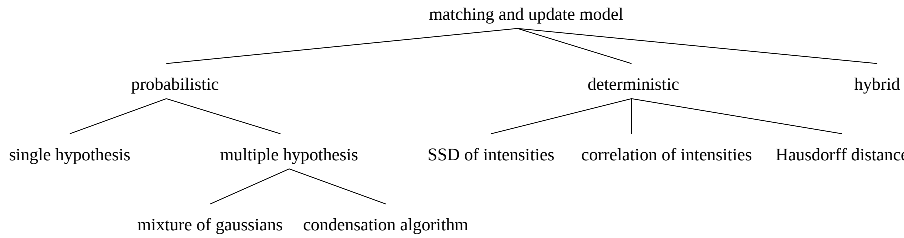

Figure 3. Matching and update models

Deterministic models are the second huge class of matching and update models. This class can be divided into the sum of squared differences of intensities (SSD), correlation of intensities and Hausdorff distance methods. The first is well known under the name Kanade-Lucas-Tomasi (KLT) tracker [18]. Patterns around corner points are extracted and matched in a search area in the next image. The search is guided by a specific motion model. Matching is done where the SSD of the intensities of the patch is a minimum. Hager and Belhumeur [8] have also developed a very efficient SSD tracker in which a reference template of a user-defined patch is stored. It is warped for every successive frame to match a target region. The new position of the patch is determined by the minimum of SSD of intensities between reference template and target region. The search is guided by a user-defined motion model. Instead of minimization of the SSD of intensities, correlation between patches is also mentioned in the literature. Methods are cross correlation like in stereo vision and correlation filters like the MINACE filter by Ravichandran [3]. A filter is synthesized from the reference patch and is then adaptively updated. In every successive frame the filter is used to find the highest correlation peaks and thus the new position. A different approach of deterministic models are distance measures between templates and patches of the current image. For example the Hausdorff distance is used by Huttenlocher et al. [10]. Two patches are considered sets of binary intensities. The distance is defined as the distance between the two sets. The minimum for a given patch and the whole image indicates the position of this patch.
Manku et al. [13] try to combine probabilistic and deterministic models. They use a Kalman filter for the first matchings and switch then to a deterministic matching by affine structure computation. For a given set of matched corner features the affine motion model is calculated. Matching is done by the Euclidian distance. If the distances are too large the algorithm switches back to Kalman filter matching.

3 Visual feature tracking for traffic monitoring

Traffic monitoring systems have to measure traffic parameters like average velocity or specific events like obstructions or lane changes of vehicles. In most cases these systems are supported by the operators of the traffic control center who have to observe the traffic. Because time can decide between life and death in case of accidents, computer-supported traffic monitoring is becoming more and more important. The next subsection deals with the requirements that most of traffic monitoring applications have. Then three different implementations are discussed.

3.1 Problems and requirements of traffic monitoring

Like in every application, traffic monitoring has specific problems and thus key requirements in addition to the requirements discussed so far:

Noise: Every computer vision application has to deal with image noise. It depends on the quality of the acquisition. Unfortunately, this is not the only noise. Light reflections of car-lights or the street lighting on vehicles also play an important role. They can disturb feature tracking or in bad cases lead to feature loss. Furthermore, different weather conditions like sunshine or rain strongly influence the tracking process. Noise influences all three design decisions and an engineer has to take that into consideration.

Occlusion: Roads or highways are observed by cameras mounted on bridges, walls of tunnels, buildings or special installations. It always leads to a perspective projection of the real world. Thus vehicles can occlude each other. Occlusion is divided into partial and total occlusion. The former means that vehicles are still visible while in the case of the latter, they completely disappear for a moment or period. Solutions to this problem strongly depend on which features are used for tracking. Generally, the occlusion problem is hard to handle.

Different kinds of vehicles: Vehicles are cars, trucks and bikes. There is a large variety in size and form of vehicles. This makes the choice of features difficult. Features must be invariant with respect to the characteristics of vehicles.

Use of existing cameras: Today, traffic is manually observed. For this purpose cameras have been installed. Unfortunately they are noisy with low resolution. This is the main cause of the first mentioned problem. On the basis of economic considerations the use of existing hardware is a key requirement.

Real-time capability: A huge amount of data has to be processed in real-time. Besides, time is crucial if accidents or traffic obstructions happen.

(a) Freeway scene in the morning (b) Tunnel portal scene

Figure 4. Two typical traffic images

All these problems have to be considered in relation to the actual application and its environment. Hence the actual requirements for the tracker can then be derived and design decisions have to be made. Figure 4 shows two typical images in the field of traffic monitoring. The next subsection shows three different implementations for the same purpose. They were all developed as part of a highway monitoring system, which should work in real-time and with all kinds of vehicles. There was no constraint regarding the use of existing cameras. Therefore the noise and occlusion problems are reduced. For instance total occlusion would be avoided by the careful installation of the camera.

3.2 Three different implementations

Koller et al. [12] use adaptive background differencing in every image to obtain blobs of motion. Then the convex polygon of every blob is extracted and approximated by a cubic spline. Snakes are also possible but not necessary because the contours of the blobs are known by the segmentation. An affine motion model is used. Two Kalman filters update the motion model and the shape of the spline. In order to get reduced noise sensitivity they assume that vehicles are well describable by cubic splines and motion is translational on a plane. Figure 5 shows the contours and tracked positions of the blob's center of gravities. Out of doors, light reflections are less important. The behavior of the system in several weather conditions is not mentioned by the authors, but it is known that background differencing is sensitive in that respect. The system has problems tracking the spline features

Figure 5. Koller's et al. approach: The black contours of the tracked blobs and their trajectories are shown.

due to occlusion of vehicles. Therefore an occlusion detection is performed. Overlapping regions are analyzed separately in the motion and shape estimation. Nevertheless, it can more or less handle partial occlusions, but with a high effort. Using the feature model, all possible kinds of vehicles should be able to be tracked by the system. Existing cameras on bridges can be used. The only assumption is that they are mounted high above the road so that no total occlusion of vehicles can happen. However, real-time requirements are not fulfilled at all. The software runs on a SparcStation 10 and processes 5 frames/sec for simultaneously tracking of about 10 vehicles. But a special hardware implementation using DSPs[1] promises near real-time performance.

After this first approach the same research group Beymer et al. [2] have used corner features for tracking. Figure 6 shows tracked features in a typical image. They saw in their first approach that active contours are not an appropriate feature to elude with occlusion. The motion model is the same as before. Kalman filters are also used to track the corners. The tracked corners are grouped together to increase robustness and to discriminate the vehicles. So a feature model (a graph to be precise) is defined upon corner features. The condition that two corners belong to the same object is an equal motion vector. The method has problems with abrupt changes in illumination due to noise or reflections. But it shows good performance in day/night situations and changes in weather. The approach is significantly less sensitive to the problem of partial occlusion. That is the main advantage to note. It also recognizes several vehicles and uses existing cameras. Near real-time performance is achieved by a DSP implementation. The performance of the tracker is 7.5 Hz in uncongested and 2 Hz in congested traffic. This reduction in speed does not of itself lead to a reduction in performance of the tracker, since vehicle speeds in congested traffic are reduced, and so the requirement for tracking rate is naturally reduced.

Figure 6. Beymer's et al. approach: The white crosses are the tracked features.

Ferrier, Rowe and Blake [4] implement another approach. Like Koller et al. they use B-splines to approximate the contour of every vehicle. They propose an affine shape model to describe the possible motions of the curve. The motion of the vehicles itself is described by a second order dynamics model in each of the six degrees of freedom of planar motion. It is updated by a Kalman filter. Because there is no segmentation during tracking, the measurement process consists of casting rays to the estimated curve as described. Edges on the rays indicate positions of the measurement. They note that the strength of the method is speed. It supports the tracking of at least 6 vehicles at 25Hz. Real-time is possible with quite modest hardware. The system is also robust against camera vibrations and it does not require camera calibration. Background differencing is not needed (except to initialize the tracker in a small area). The result is a robust system with respect to noise, reflections and weather conditions. The system is muted in that it only pursues individual vehicles. Furthermore, the partial occlusion problem is not solved by this method.

4 Discussion

As these three implementations show, corners, blobs, active contours and patches are often used as image features. Active contours are more robust against noise and reflections than corners. A reason is that they depend less on image intensities. Instead, corners as well as patches significantly lead to better tracking results in the case of partial occlusion of vehicles. As is shown in Beymer's et al. approach, the occluded parts of the splines have to be treated separately. It leads to higher complexity and thus to less execution time. Therefore Ferrier et al. have not treated the occlusion problem. They assure that the tracker only pursues features in an area where occlusions do not occur. Beymer et al. also show in their second approach that feature models can improve the stability of the tracker.

All three examples use an affine motion model. Normally, 5-10 Hz tracking frequence is too low to assure linear motion. Nevertheless, the model is sufficient, because vehicle's movements are restricted to a linear road-plane. Furthermore, they use a Kalman filter algorithm for matching and state update. Both model in their active

[1] Digital Signal Processors

contour approach the deformation of the contour by an affine model. Ferrier et al. mention that it is possible to learn this shape model a priori.
They do not use the "classical" Kalman filter but a modified second order autoregressive process for updating. Unfortunately, no comparison to the classical version is given. The matching is also done in different ways. While Beymer et al. use segmentation to get blobs and thus measurements for the new contour's position, Ferrier et al. use a modified edge detection. The advantage is mainly speed. The second approach of Beymer et al. also works with a "classical" Kalman filter. It does not need a specific shape model. Therefore the update process is simpler. Measurements are taken by the feature detector in the predicted area. Naturally, this fact favors the real-time capability of the approach.

requirements	contours	feature points
noise	+	-
occlusion	-	+
reliability	+	+
stability	+	-
real-time capability in		
light traffic	+	+
heavy traffic	-	+

As a summary, the opposite table shows the comparison of the contour and feature point approach. Generally, if occlusion is part of the problem as in traffic obstruction detection, feature points like corners or other interest points are preferable. It is better to work with contours if occlusion is negligible. The advantage is higher stability in noisy environments. It is also not necessary to consider more complex motion models than the affine model, as traffic motion is approximately linear, because motion is always restricted to the roadplane, or frame rates are relatively short. The reliability of detecting the vehicles is satisfactory in both implementations. As contours are a more robust description with respect to noise, contour tracking is more stable than pursuing feature points.
Edges as features for traffic monitoring are not mentioned in the papers discussed. Performing edge detection in a small window is impossible to carry out due to the aperture problem. Therefore computation has to be done in the whole image and this conflicts with real-time requirements. Blob tracking also fails while occlusion occurs [16]. As Beymer et al. show, feature models often improve the robustness of the tracker. The use of templates as feature models in a traffic monitoring application has not been mentioned so far. The Kalman filter is the widely used matching and update algorithm. From a statistical point of view it is an optimal estimator and therefore an optimal tracker. Unfortunately, this is only valid in undisturbed scenes. If high background clutter exists, multiple hypothesis methods are significantly better. Of course, this also happens in traffic monitoring, but up to now no approach in that direction has been proposed. Deterministic and hybrid methods are also seldom used, although SSD trackers should perform as well as the Kalman filter based corner trackers mentioned above.

5 Conclusion

This paper has discussed visual feature tracking methods in general and for traffic monitoring. Four general requirements for every tracker viz reliability in detection, stability in tracking, accuracy in prediction and real-time capability have been shown. Engineers have to take these requirements into account. They do this carefully by making three design decisions.
First, the best features have to be chosen which represent the objects of interest as well as possible. Several kinds of image features and feature models are available to fulfill this best choice.
Second, the question of where the tracker should search for the feature in the next image must be answered. Such a prediction of motion can be done by linear or nonlinear models. If the frame-rate of the image sequence is high enough so that motion of features from one to the next image is linear, affine or translational models are sufficient.
Lastly, the right algorithm for matching features and their measurements in the next image and updating the feature's state has to be chosen. Deterministic (e.g. SSD) and probabilistic (e.g. Kalman filter) methods are discussed in detail. The use of the right algorithm depends on the application and its requirements. Therefore, visual tracking has to be seen in relation to traffic monitoring.
Problems arise such as specific noise, occlusion of vehicles, different kinds of vehicles, the use of existing cameras and real-time capability. Such problems require specific algorithms to solve them. Therefore some of the methods mentioned above are discussed in that respect. To summarize this comparison it can be said that contours produce better results in handling noise than feature points like corners. The reason is that they are

not defined over a patch of brightness values. Therefore contour trackers are more stable than feature point trackers. On the other hand feature points are more robust in cases of vehicle occlusion and the computational expense in calculating the matching and state update is much less than with contours.

At this point it must be emphasized that a comparison of several visual feature tracking methods has not been done so far. This paper is the first one to present a taxonomy of visual feature tracking. It seeks to close this gap between general expert knowledge and the requirements of practitioners who want to use visual feature tracking for their own work.

Many models and methods are considered by this paper. Future investigations should complete the classification of different features, motion, matching and update models.

Acknowledgements

I want to thank Horst Bischof and Bernhard Strobl for their proof reading. I want also to thank Nabil Belbachir and Alexander Selb for helpful discussions. Amy Krois-Lindner supported me so much in improving my English and corrected this work - thank you! I am grateful for the financial support of the Austrian Research Center Seibersdorf (Helmut Schwabach).

References

[1] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, New York, 1999.

[2] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters.

[3] M. W. Eklund, G. Ravichandran, M. M. Trivedi, and S. B. Marapane. Adaptive visual tracking algorithm and real-time implementation. In *International Conference on Robotics and Automation*. IEEE, 1995.

[4] N. J. Ferrier, S. M. Rowe, and A. Blake. Real-time traffic monitoring. IEEE, 1994.

[5] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. *Pattern Analysis and Applications*, 2:312–320, 1999.

[6] A. Garg, I. Cohen, and T. S. Huang. Adaptive learning algorithm for svm applied to feature tracking. IEEE, 1999.

[7] S. Gil, R. Milanese, and T. Pun. Feature selection for object tracking in traffic scenes. In *SPIE International Symposium on Smart Highways*, Boston, Massachusetts, 1994.

[8] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[9] G. D. Hager and K. Toyama. Xvision: Combining image warping and geometric constraints for fast visual tracking. In *European Conference Computer Vision*, volume 2, pages 507–517, 1996.

[10] D. Huttenlocher, J. Noh, and W. Rucklidge. Tracking non-rigid objects in complex scenes. In *International Conference on Computer Vision*, pages 93–101.

[11] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. pages 5–28, 1998.

[12] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *European Conference Computer Vision*, pages 189–196. Springer Verlag, 1994.

[13] G. S. Manku, P. Jain, A. Aggarwal, and L. Kumar. Object tracking using affine structure for point correspondences.

[14] D. Noll, M. Werner, and W. von Seelen. Real-time vehilce tracking and classification.

[15] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *Transactions on Pattern Analysis and Machine Intelligence*, pages 100–114, 1999.

[16] R. P. Pflugfelder and H. Bischof. Tracking cars in tunnels. In *Czech Pattern Recognition Workshop*, pages 179–183, Praha, Czech, 2000.

[17] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *International Conference on Computer Vision*, volume 28, pages 5–28, 1998.

[18] J. Shi and C. Tomasi. Good features to track. In *Conference on Computer Vision and Pattern Recognition*, Seattle, 1994.

[19] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. 1999.

Roman Peter Pflugfelder.
is a graduate student of the PRIP Group. He received his Cand.-Ing. degree in Computer Science (comparable to BSc) in 1997. He has been worked for the Austrian Research Center (ARC) Seibersdorf since 1999 in the field of traffic monitoring. His research interests are computer vision, artifical intelligence and computer engineering.