# A Shadow Handler in a Video-based Real-time Traffic Monitoring System

M. Kilger

Siemens AG, Corporate Research and Development, ZFE ST SN 3
Otto-Hahn-Ring 6, D-8000 Munich 83

## Abstract

*A video-based system for traffic monitoring is presented. The objective of the system is to set up a high-level description of the traffic scene comprising the position, speed and class of the vehicles. Algorithms for detecting moving objects, separating the vehicles from their shadows, tracking and classification are presented. Especially the classification of the vehicles under sunny illumination conditions is very difficult, if the shadow isn't separated from the vehicles. Our novel approach for classification runs in real-time on a low-cost hardware. The shadow can be separated from the vehicle and the knowledge about the shape of the shadow can be efficiently used. The shadow analysis algorithm itself uses high-level knowledge about the geometry of the scene (heading of the observed road) and about global data (date and time).*

## 1. Introduction

Ever increasing traffic volumes demand more efficient and intelligent management and control strategies. As a prerequisite, on-line traffic data acquisition is necessary. Typical requirements are [8]:

- A wide range of view of at least 100m should be monitored,
- The speed and position of each vehicle should be reported exactly,
- Vehicles should be counted, tracked and classified.

Several authors [5]..[9] have presented similar work, but no one has yet met these requirements under difficult illumination conditions typically found on a sunny day.

In [5] the vehicles are counted and the position and the velocity of the vehicles are calculated. A wide area can be monitored and the algorithms run in real-time. However this special model-based approach makes classification extremly difficult. The approach described in [9] also doesn't classify the vehicles. Furthermore it requires a manual initialisation of the background image and the operating window. In [6] a quite different approach for detecting the vehicles is used. The whole region of interest is searched for regions of constant grey values. If this grey value is different from the one of the road, which was detected before, it is assumed that it is vehicle. The system detects vehicles well, but with this approach no real-time detection is possible with low-cost hardware until now. In [8] the vehicle is detected by subtracting two subsequent frames. The camera is positioned vertically above the road, therefore the range of view is limited. The detected vehicles are classified and their position and velocity calculated. In [7] the classification is made by matching the outlines of the detected vehicles with templates. The calculation is done in real-time by special purpose hardware.

Obviously the above mentioned work has only been applied to scenes with ambient illumination. No results have been reported for scenes with bright sunlight and corresponding shadows.

In general, the shape of the vehicles has been modelled by complex wire frame models, which don't allow for real-time processing on low-cost hardware. On the other hand classification by the bounding box is computationally feasible, but is only robust, if the vehicle can be reliably separated from its shadow.

In this paper a video-based traffic monitoring system is presented which includes a shadow handling algorithm. It is shown that traffic monitoring is possible even with low-cost hardware and under difficult illumination conditions. Furthermore it is also shown that a high-level description of the observed scene can be built up with the information obtained from all moving objects. From this description expected results of low-level image routines (such as tracks etc.) can be computed. The difference between the actual results and the expected results can be used to optimise low-level processing parameters or to select among different alternative routines, depending on the situation. Thus far the framework of this on-line

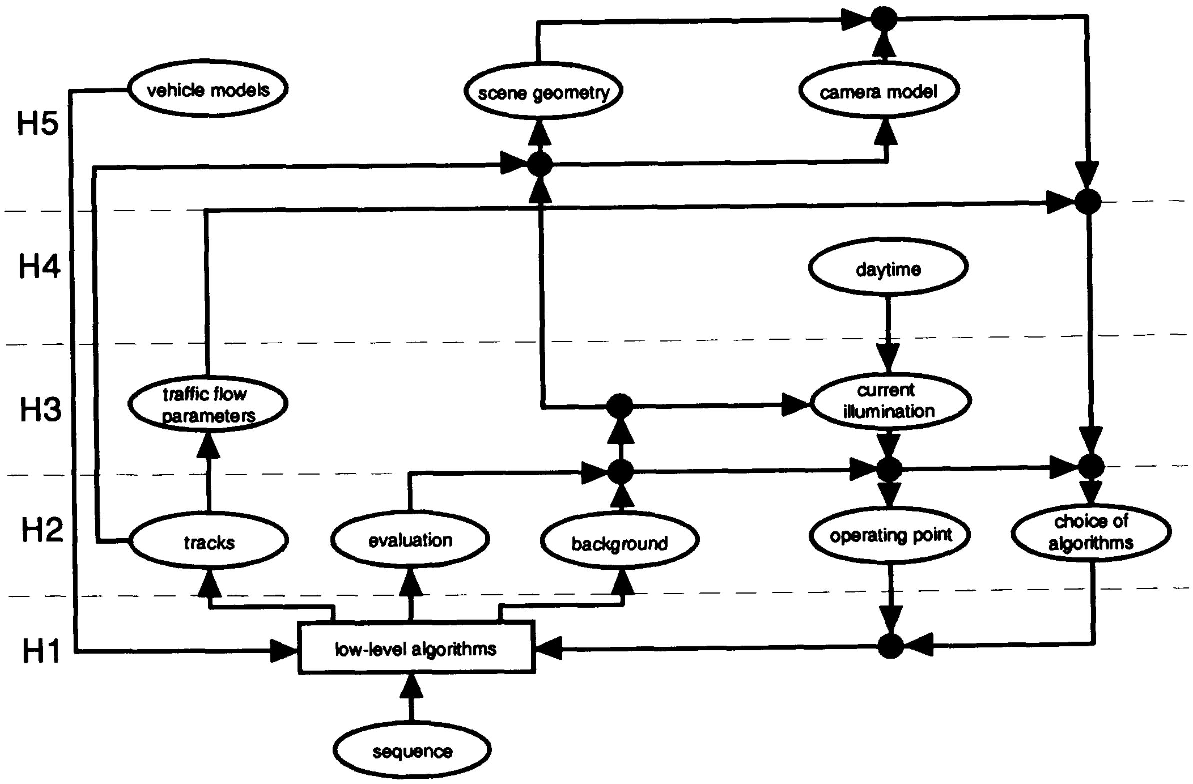


**Fig.1: system architecture**

adaptive system has been established and its validity has been demonstrated. This basic set of routines is now extended by the shadow handling algorithm.
In this paper the algorithms of the real-time traffic monitoring system are presented. Tests with the shadow analysis algorithm have yielded good results.
In chapter 2.1. the whole system architecture with its five levels is presented. In the following chapters the algorithms used, are discussed, in 2.2. the detection of the moving objects, in 2.3. the shadow analysis, in 2.4. the occlusion analysis, in 2.5. the tracking and in 2.6. the classification. In chapter 3. the implementation is introduced. In chapter 4. the results of the shadow handler applied to several image sequences are shown. In chapter 5. some conclusions are drawn and in chapter 6. an outlook for future work is given.

## 2. Algorithms in the Video-based System

### 2.1 System Architecture

The whole real-time video-based system is devided into five levels (shown in Fig.1). The lowest level H1 contains all the low-level image processing routines.
Their use is controlled by the algorithms of level H2. Therefore a special module *choice of algorithms* is defined. It chooses, depending on the situation, the best suited algorithm, e.g. the use of the shadow separation algorithm. All parameters, e.g. all threshold values, are saved in the *operating point* module. The results of the low-level algorithms are stored in the *tracks* module, where the class, position and speed of each vehicle are gathered. These results are continuously evaluated and if necessary the parameters of the algorithms are adapted. As a result the background image is obtained, which can be used for further static scene analysis.

In the next level H3 the traffic flow parameters are calculated. This level also contains the slowly changing parameters, like the *current illumination*. These *current illumination* parameters influence the *choice of algorithms* and the parameters of the *operating point*, where the fast changing parameters are stored.
A higher level H4 provides information such as the *daytime* (e.g. night or sunlight), which controls the *current illumination*.
In the highest level H5 the slowest or quasi static parameters are stored, like the *vehicle models*, the *scene geometry* or the *camera model*.
To demonstrate the efficient feedback from the high-level knowledge to the low-level algorithms, the shadow handler was implemented in the real-time system. The parameters of the system are adaptive and so ensure a more efficient calculation in further analysis stages.

## 2.2. Detection of Moving Objects

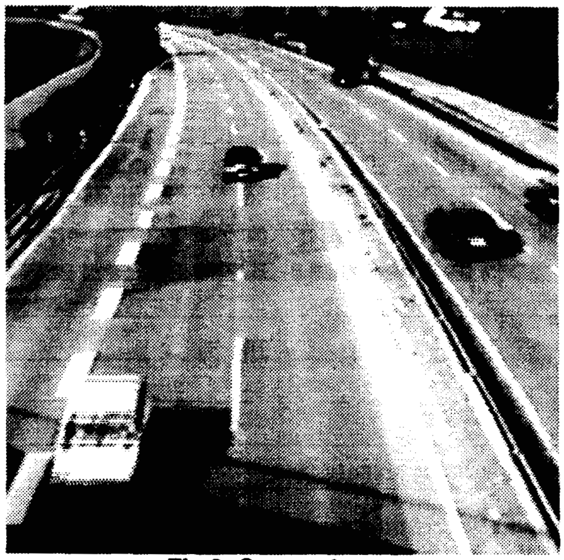

**Fig.2: Current image**

### 2.2.1. Building up a Binary Object Image

Several approaches for detecting moving objects have been discussed in the literature [5]..[9]. The simplest of all comprises the differencing of two successive frames [8]. All the motion is detected, but this algorithm lacks robustness [4]. Another approach is based upon the calculation of the optical flow [12], but it is computationally expensive and therefore not practicable for real-time systems based on a low-cost hardware.

A robust and a good approach is based upon subtracting the current image (shown in Fig.2) from the background image (shown in Fig.5) [1]..[4].
The equations for subtracting the background image from the current image are given by

$$|I_k(p) - B_k(p)| > T_k(p) \quad \Rightarrow \quad M_k(p) = 1 \quad (1)$$

$$|I_k(p) - B_k(p)| \le T_k(p) \quad \Rightarrow \quad M_k(p) = 0$$

where I is the current image, B is the background image, M is the object image and T is the threshold at time k and pixel p.
A threshold decision on the difference image is made which separates the image into regions of fast motion and slow, or no motion. The result is a binary object image as shown in Fig.3.

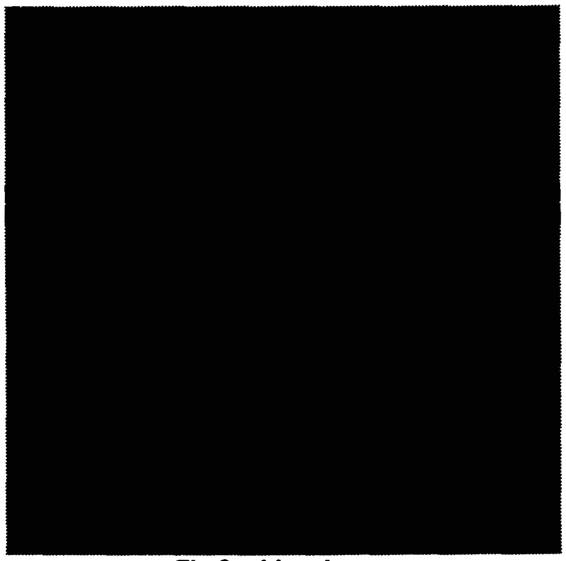

**Fig.3: object image**

However the quality of the object image depends critically on the chosen threshold value. If the threshold is too high, the image is too fragmented. In the opposite case, lots of uninteresting motion like the rustling of leaves will be detected. This problem can be overcome by continuously adapting the threshold T. This adaptation is performed by the *operating point* module of the system, where all parameters are stored. A high-level routine is set up to evaluate the quality of the current threshold value and adapts it, if necessary. Instead of one global threshold for the entire image several local thresholds are used which can be adapted more precisely to the current scene.

#### 2.2.2. Segmentation of the Object Mask

For each connected component (segment) in the image the bounding box, the area in the image and an estimation of the corresponding area in the world are computed. The latter is derived from the inverse perspective mapping under the assumption that vehicles move on height zero. The segmentation algorithm itself is controlled by the parameters provided by the *operating point* module. The bounding box turns out to be a sufficient description of the vehicles, if counting, tracking and a rough classification are required. Other authors use model-based approaches with wire-frame models [11], but they haven't met the real-time requirements yet.

#### 2.2.3. Segment filtering

The bounding boxes whose estimated area in world co-ordinates is too small and which are unlikely to correspond to a vehicle are removed (e.g. off the road, or behave irregularly). The result of the segmentation is shown in Fig.4. A black rectangle shows the detected moving objects.

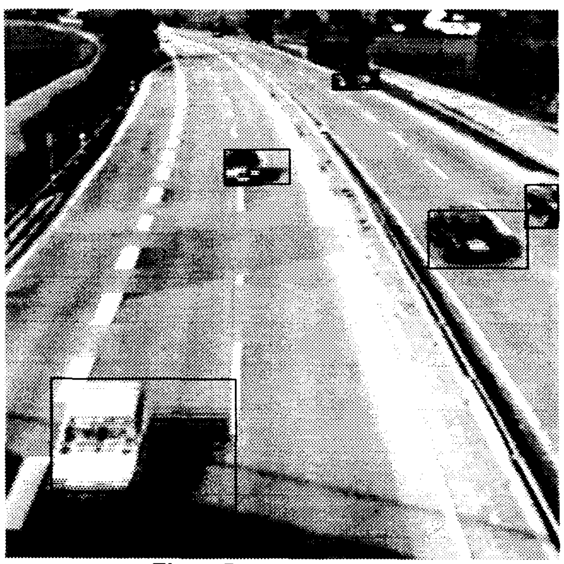

**Fig.4: Detected objects**

#### 2.2.4. Background Adaptation [4]

The background image (see Fig.5) is adapted continuously by masking the regions of motion, using recursive filtering with variable coefficients.
The equation for the background adaptation is given by

$$B_{k+1} = B_k + \left(\alpha_1 \times \left(1 - M_k\right) + \alpha_2 \times M_k\right)\left(I_k - B_k\right) \quad (2)$$

where the update coefficients $\alpha_1$, $\alpha_2$ start with a fixed value. These two coefficients are adapted together with the threshold value T of equ (1).

**Fig.5: Background image**

### 2.3. Occlusion Analysis

In heavy traffic the detection of a single vehicle is very difficult. Vehicles may partially occlude each other and can therefore be detected as a vehicle cluster. In order to count all vehicles and classify them, these clusters have to be resolved.
In this approach the first edges of the vehicles are extracted and matched with a geometric model. The bounding box turns out to be sufficient in this case. A detailed discussion can be found in [10].

## 2.4. Shadow Analysis

### 2.4.1. Shadow Detection Algorithm

The shadow detection algorithm is integrated into the hierarchical system architecture, see Fig.1. In level H4 the knowledge of the date and time is stored. This knowledge determines the plausibility of shadows occurring. If the time excludes the appearance of shadows (e.g. during the night), the detected objects do not have to be analysed for shadows which saves processing time. Therefore the module *choice of algorithms* excludes the use of the shadow separating algorithm.

If shadows are possible the detected moving objects must be analysed for typical attributes, such as an abnormal width or height. The expected direction and the possible shape of the shadow are used to direct the search. For detection of the shadow the shadow separating algorithm is applied.

In a set-up phase all possible directions (left hand side, right hand side and in front of the vehicle) are searched for shadows. If the shadow is detected, its direction together with the time of occurrence, is saved in the *scene geometry* module of level H5. This knowledge is now used for further analysis. Therefore it isn't necessary to look for shadows in each direction every time.

Once a shadow has been detected, the shadow detection continues for several frames in order to increase the plausibility of the shadows. In case of the plausibility being greater than a threshold, it is recorded in the *choice of algorithms* module. Now each frame is analysed for shadows until no shadow is detected for a certain period of time. If this happens the entry in the *choice of algorithms* is deleted again.

### 2.4.2. Shadow Separating Algorithm

For correct separation of the shadows from the corresponding objects, the knowledge about the direction of the shadow is used. This knowledge is obtained from the date, time (in H4) and heading of the road (in H5 *scene geometry*). Furthermore the knowledge about the motionless scene (in H2) is embedded in the algorithm. A real-time clock found nowadays in every PC-system, provides date and time. The heading of the road is obtained by the set-up phase and entered in the *scene geometry* module. The background image is provided in each frame by the detection and background update algorithm.

All analysis is based upon the bounding box of the detected moving object. Therefore the region of interest is limited, which saves processing time.

The knowledge about the expected shadow directs the search for shadows. It starts either on the left hand side, the right hand side or in front of the vehicle. Both lateral directions can be combined with the direction in front of the vehicle.

In case of the directions being to the side of the vehicle the vertical edge image of the current image is calculated. Otherwise the horizontal edge image is computed. The Sobel operator is used as edge detector. Thereafter the histogram of the image is computed. However for efficiency reasons the edge histograms are not computed over the whole detected moving region of the image. For instance, if it is known that the shadow of the vehicle is on its right hand side, the edge detection algorithm starts from that end of the segment and builds up the edge histogram of the current image.

In general it can be said that a shadow is edge-less, but the vehicles have significant edges especially at their borders. However in the case of the shadow falling on the middle line of the road, the edge histogram of the current image produces a lot of edges in the region of the shadow. Therefore the first significant edge of the vehicle cannot be detected and the correct shadow detection and separation is not possible. However these edges also occur in the background image. Therefore the corresponding edge histogram of the background image is calculated.

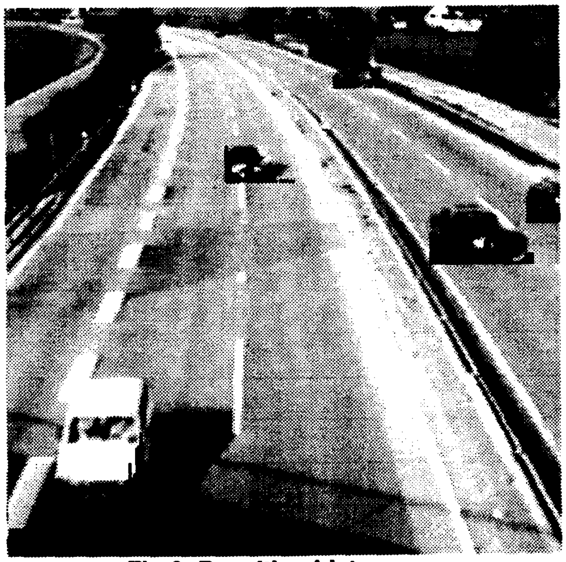

**Fig.6: Resulting histograms**

Subtracting the two histograms yields the resulting histogram (in Fig.6 the vertical edge histogram is overlaid on the current image) whose structure is significant for a detected vehicle. A threshold decision is made to obtain the significant first edges of the vehicle. This threshold is normalised by the height of the detected segment. In future this threshold should be adapted similarly to the parameters of the background adaptation. This adapted threshold could be entered in a high-level module and used for further analysis.

For efficiency reasons the algorithm is stopped after obtaining the first significant edge of the vehicle. Now it is assumed that this edge is the border of the vehicle, which is sufficient in most cases.

In Fig.7 the segments after shadow separation are shown. It can be seen that the shadows are separated well from the corresponding vehicles.

#### 2.4.3. Information Obtained from the Shadow Analysis

As a direct result of the shadow analysis, both the shape of the vehicle and the shape of the corresponding shadow are obtained. A correct classification of the vehicle can be made. Additionally the shape of the shadow can be used to calculate the height of the vehicle and improve classification. Furthermore a high-level geometric description of the scene can be built up. From the direction of the shadow, the heading of the observed road can be calculated and used for further analysis.

Finally all shadow analysis and camera parameters can be adapted by using the knowledge that vehicles cannot be deformed on their way from the background to the foreground of the image.

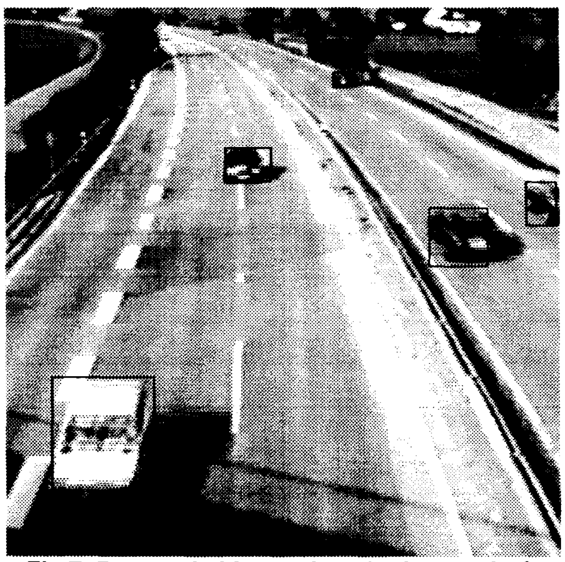

**Fig.7: Detected objects after shadow analysis**

### 2.5. Tracking

The main purpose of the tracking algorithm is to track the detected objects and to predict their respective positions based on a state model [13]. The results of this algorithm provide the following attributes for each detected object:

- Current position,
- Predicted position for the next image frame,
- Current speed,
- Predicted speed for the next image frame,
- Width and
- Plausibility.

Our implementation uses a constant velocity state model, where the middle of the front edge is tracked. The whole filtering and prediction is based on real world co-ordinates [2].

In Fig.8 the current image with overlaid track numbers is shown, where the track number identifies each vehicle. Thereafter a classification of each vehicle can be made.

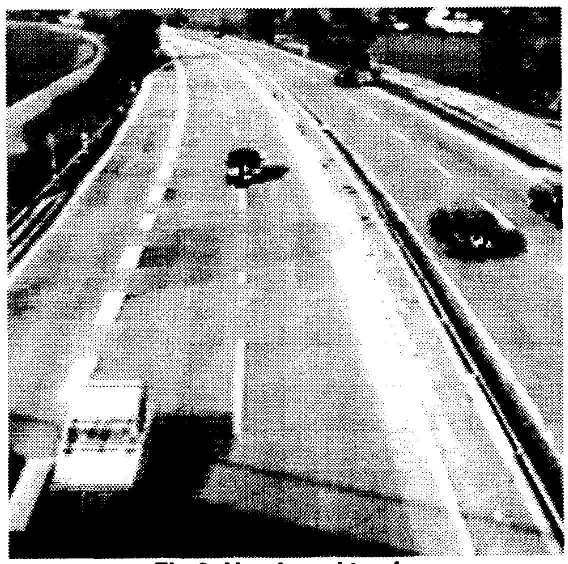

**Fig.8: Numbered tracks**

### 2.6. Classification

The vehicles have to be classified according to:

- trucks,
- cars,
- motorcycles, bicycles.

The selection of the features best suited for the classification depends on each situation. If the vehicle is moving towards the camera, the first feature to evaluate

is the width of the bounding box. This width is tracked over several frames and estimated.
If a rotation of the vehicle is detected the front edge is used as a suitable feature.
Another useful evaluation feature is the height and length of the detected object. Therefore the height of the bounding box is tracked by a Kalman filter.

## 3. Implementation

The high-level algorithms (tracking, classification and communication) are implemented on a PC with an Intel i486 processor. The low-level routines (detection and shadow separating algorithm) are implemented on a DSP (Motorola's DSP96002). The high-level algorithms are written in C and the time-critical low-level algorithms are written in the assembler of the DSP.
The resolution of the image is 256x256 pixels with 256 grey levels. The frame rate (with full shadow separation) is 5 Hz (without code optimisations).

## 4. Results

The algorithms were tested under normal traffic and daylight conditions for several image sequences lasting several hours. The vehicle detection rate was over 99%. In the set-up phase with no a priori knowledge about the observed scene the shadow detection rate was 95%. In this case the shadow was detected correctly. In the remaining 5% of the cases, the shadow was rejected after several minutes in 90% of these cases, because the shadow separating algorithm could not find shadows in this direction. The knowledge about the possible shadow during the daytime is saved and used for further shadow analysis. If shadow appears in these cases, which is normal for the monitoring system, the possible direction of the shadow is well-known. Therefore the detection rate increases to 98%. If the direction was found correctly the shadow could be separated from the vehicles in 90% of these cases. After filtering the width of the vehicle over time the classification success rate was more than 95%.

## 5. Summary and Conclusions

A real-time traffic monitoring system has been presented and its system architecture described. The algorithms for detection of moving vehicles, tracking and classification have been presented. In sunlight conditions the classification based upon the width of the detected moving objects wasn't satisfactory. Therefore a shadow handler was integrated into the system architecture. This shadow handler operates upon the current and the background image. It detects the edges of these images and calculates their histograms, based upon the regions of detected motion. It could be seen that the vehicle has significant edges, however the corresponding shadow is generally edge-less.
This shadow handling algorithm was tested on a image sequence of several hours. The results have shown that this algorithm has separated the objects from their shadows very robustly.

The bounding box, especially the width of this box turns out to be a good geometric model for classification of the vehicles in a real-time application. Furthermore it is shown that traffic monitoring is possible with a low-cost hardware.

## 6. Outlook

Work is currently in progress which extends the approach as follows:

- Switching between various algorithms depending on the illumination conditions,
- Adapting low-level image processing parameters by using the knowledge of high-level routines,
- Adapting high-level scene description parameters by using the information obtained from a sequence of images over several hours.
- Adaptation of the whole system (continuous parameter adaptation and choosing the best suited algorithms depending on the observed scene).

## 7. References


[1] M. Kilger: Video-based traffic monitoring. Proc. of 4th International Conference on Image Processing and Its Applications, pp 89-92, Maastricht/The Netherlands, Apr. 1992

[2] W. Feiten, A.v. Brandt, G. Lawitzky, I. Leuthäusser: A video-based system for extracting traffic flow parameters. Proc. 13th DAGM 1991, Munich, pp. 507-514 (in German)

[3] K.P. Karmann, A.v. Brandt: Moving object segmentation based on adaptive reference images. Proc. of EUSIPCO 1990, Barcelona

[4] R. Gerl: Detection of moving objects in natural enviroment using image sequences. 1990, Diplomarbeit TU Munich (in German)

[5] A.Bielik, T. Abramczuk: Real-time wide-traffic monitoring: information reduction and model-based approach. Proc. 6th Scandinavian Conference on Image Analysis. 1989, Oulu, Finland, pp. 1223-1230.

[6] S. Beucher, J.M. Blosseville, F. Lenoir: Traffic Spatial Measurements Using Video Image Processing. SPIE Vol.848 Intelligent Robots and Computer Vision: Sixth in a Series (1987), pp.648-655.

[7] A.D. Houghton, G.S. RHobson, N.L. Seed, R.C. Tozer: Automatic vehicle recognition. 1989, University of Sheffield, U.K.

[8] W. Leutzbach, H.P. Bähr, U. Becker, T. Vögtle: Development of a system for extracting traffic flow data. 1987, Technical Report, University of Karlsruhe (in German).

[9] N. Hoose, L.G. Willumsen: Automatically extracting traffic data from videotape using CLIP4 parallel image processor. Patt. Recog. Letters, 6, 1987, pp. 199-213

[10] Siemens AG: Patent pending: Algorithm for resolution of clusters , 6.4.1992. (in German)

[11] G.D. Sullivan, K.D. Baker: Model-based vision: using cues to select hypotheses. SPIE Vol.654 Automatic Optical Inspection (1986). pp.272-277.

[12] D. Koller, H.-H. Nagel: A robust algorithm for detecting and tracking of moving objects in image sequences. Proc. 12th DAGM 1990, Oberkochen-Aalen, Germany, pp.625-633. (in German)

[13] Y. Bar-Shalom, T.E. Fortmann. Tracking and Data Association. 1988 Academic Press.