Adaptive methods of moving car detection in monocular image sequences

Włodzimierz Kasprzak
Institute of Control and Computation Eng., Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warszawa,
E-mail: W.Kasprzak@ia.pw.edu.pl

Abstract. Computer vision applications for traffic scene analysis and autonomous navigation (driver support) require highly sophisticated sensors and computation methods – they constitute a real challenge for image analysis systems. Common to both applications is the moving object detection/tracking task. In this paper we study this task on four different data abstraction levels: image segmentation, 2-D object tracking, model-based 3-D object tracking and many-object traffic scene description. Two meanings of the term "adaptive" are considered: learning algorithms or connectionist systems and recursive estimation for dynamic systems. Generally the first approach may be applied for low- and segmentation-level analysis of finite image sequences, whereas the second approach is suitable for 2-D and 3-D object tracking and estimation.

Key words: model-based analysis, object tracking, obstacle detection, recursive estimation, road following, traffic scene analysis, visual motion estimation.

1. Introduction

In the mid 80-s an intensive research work started in the field of autonomous navigation and visual driver support systems (under moving camera) [13, 30, 37, 42, 51] and in automatic traffic scene analysis systems (under stationary or active camera) [18, 29, 39, 41, 46, 61]. Although considerable results have been achieved, these problems still constitute very attractive and challenging application fields of image sequence analysis techniques.

The partly unknown ego-motion of the camera requires model-based approaches to the main two problems in such vision systems: (1) the *road following* and *ego-state tracking* task ([27]), and (2) the obstacle (on-road 3–D objects) detection task ([49]). There already exist solutions of the first task ([27, 40, 57, 59]) although they still seem not to be sufficiently robust. There is a problem of automatic orientation recognition (overcoming the camera nodding movement) and of recognizing the road if many obstacles exist in the scene.

A reliable detection and classification of obstacles in images of many–object scenes is still a challenging problem [33, 49, 51]. The complex nature of the subject makes it necessary to apply dynamic model–based analysis, which usually performs object

tracking and constrains the classes of recognized objects. But even a parallel tracking of many hypotheses is not sufficient for reliable object recognition, as tracking works in an object–centered manner. Only these image features are selected only, which support the given hypothesis. A robust recognition system is required, that not only tracks already known image features (due to manual or sub–manual initialization) but also makes automatic detection of new features and is able to adapt to changes of the road environment.

In this paper we review vehicle recognition techniques, that can be applied in automatic analysis of image sequences of traffic scenes. We study solutions to problems of: visual motion-based image segmentation, 2-D object tracking, model-based 3-D object tracking and many-object scene description.

2. Adaptive schemas

Two meanings of the term *"adaptive"* are considered in this work:

1. Artificial Neural Network(ANN)-learning based methods for the analysis of finite image sequences. These connectionists systems are especially suitable for solutions to low- and segmentation-level image analysis problems ([65]). In the learning phase the ANN is trained by providing known image data samples. In the analysis phase the ANN is stimulated by the unknown image data and its outputs correspond to the results of analysis task.
2. Recursive estimation of dynamic objects in the analysis of infinite image sequences. A recursive analysis mode is especially suitable for the analysis of a non-stationary scene environment (e.g. moving objects, moving camera). The appropriate solution of such analysis tasks requires the application of dynamic system theory in order properly to adapt (modify) the previous image results to current image analysis results ([17, 28]).

2.1. ANN for finite sequence analysis

Main types of behavior of ANNs, from the point of view of pattern recognition theory, were distinguished in the past:

1. associative memory - recall of patterns (e.g. Hopfield net [3]),
2. classification or recognition of patterns (e.g. multi-layer network with supervised back-propagating learning [8]),
3. clustering of patterns (e.g. Kohonen maps [15]),
4. feature extraction (e.g. Neocognitron [14]).

In scope of our application field - moving car detection in image sequences of traffic scenes, we shall give two exemplary applications of following neural networks: (a) feed-forward nets with supervised learning for image classification and feature detection (e.g. backpropagation net, discriminant analysis by supervised LVQ); (b) feedback nets with

unsupervised learning for associative memory and optic flow estimation (e.g. Hopfield net, relaxation networks).

Other two main categories of ANNs can also be applied in image analysis, i.e. feed-forward nets with unsupervised learning are used for filter-like mapping between vector spaces – image restoration, compression and clustering (e.g. ICA nets, PCA nets, SONNs), and feedback nets with supervised learning for correspondence problems – visual motion detection, arbiter of multiple detections or inconsistencies (e.g. Boltzmann machine).

1. Initialization: Application–dependent computation of initial estimation $\boldsymbol{s}^*(k_0)$ and covariance matrix $\boldsymbol{P}^*(k_0)$ of estimation error. Go to step 6.
FOR every next image $k > k_0$
2. Detection of new measurement $\boldsymbol{m}(k)$. The covariance matrix of the system noise $\boldsymbol{Q}(k)$ is also available.
3. Estimation of current gain $\boldsymbol{K}$ (2 versions).
4. State modification (innovation): $\boldsymbol{s}^*(k) = \boldsymbol{s}^+(k) + \boldsymbol{K}(k)\big\{\boldsymbol{m}(k) - \boldsymbol{H}(k)\boldsymbol{s}^+(k)\big\}$
5. Modification of matrix $\boldsymbol{P}$ (2 versions).
6. Prediction of next state: $\boldsymbol{s}^+(k+1) = \boldsymbol{F}(k)\ \boldsymbol{s}^*(k)$.
7. Prediction of next matrix $\boldsymbol{P}$: $\boldsymbol{P}^+(k+1) = \boldsymbol{F}(k)\boldsymbol{P}^*(k)\boldsymbol{F}^T(k) + \boldsymbol{Q}(k)$.
8. $k \leftarrow k+1$

Fig. 1. The recursive object state estimator.

2.2. Recursive estimator for object tracking

For the object tracking process in infinite image sequences usually a recursive estimator is used, like for example an extended Kalman filter (EKF) [17]. A general recursive estimator is given in (Fig. 1). From the state modification it is evident that a crucial role in the minimization of the estimated error is played by an appropriately designed $\boldsymbol{K}(t)$ matrix. But originally the gain matrix depended on the estimation error covariance matrix and on noise covariances only. There are two steps which we may be differently performed. In an EKF, the Kalman gain is estimated as (index k is omitted):

$$\boldsymbol{K}(k) = \boldsymbol{P}^*\boldsymbol{H}^T\left\{\boldsymbol{H}\boldsymbol{P}^*\boldsymbol{H}^T + \boldsymbol{R}\right\}^{-1}, \tag{1}$$

where $\boldsymbol{R}(k)$ is the covariance matrix of the measurement. Again in EKF, the modification equation of the error covariance matrix $\boldsymbol{P}$ is:

$$\boldsymbol{P}^*(k) = \boldsymbol{P}^+(k) - \boldsymbol{K}(k)\boldsymbol{H}(k)\boldsymbol{P}^+(k). \tag{2}$$

Hence, both steps in EKF are independent from current *tracking error*: $\boldsymbol{m}(k) - \boldsymbol{H}(k)\boldsymbol{s}^+(k)$. This requires the existence of a proper judgement scheme for measurement data in given

application. If such a scheme is not available, all the $n \times m$ parameters $K_{ij}(t) \in \boldsymbol{K}(t)$ for all t should be set by default, and this will usually result in object tracking errors. In the desired solution we need a direct inclusion of the tracking error in the gain estimation equation.

Hence, for estimation of the gain and the state covariance matrix a scheme is proposed, that is similar to the self–adaptation of learning rates in neural network learning. Let us assume, that measurement judgment can be expressed by the current tracking error: $\boldsymbol{e}(k) = \boldsymbol{m}(k) - \boldsymbol{H}(k)\boldsymbol{s}^{+}(k)$ with its covariance matrix $\boldsymbol{R}_e(k) = \mathrm{E}\{\boldsymbol{e}(k)\boldsymbol{e}^T(k)\}$. To assure a minimum error threshold the default matrix $\boldsymbol{R}(t)$, corresponding to measurement noise, is also added. Now the estimation of matrix $\boldsymbol{P}^*(k)$ is given as:

$$\boldsymbol{P}^* = (1-\delta)\boldsymbol{P}^{+}(k) + \delta\boldsymbol{P}^{+}\boldsymbol{H}^T(\boldsymbol{R}_e + \boldsymbol{R})(\boldsymbol{H}\boldsymbol{P}^{+}\boldsymbol{H}^T)^{-1} \tag{3}$$

and the estimation of a single gain element is:

$$K_{ij}(k+1) \quad = (1 - K_{ij}(k)\delta_1)K_{ij}(k) + \alpha K_{ij}(k)\textstyle\sum_l P_{il}(k)H_{ij}(k). \tag{4}$$

3. Segmentation-level moving object detection

3.1. Motion-based image segmentation

In case of image sequences from a stationary camera various robust visual motion detection and estimation methods are available [9]. Pixel-based estimation of *dense* visual motion is called *optical flow* detection [43]. The pixel motion can be applied for image segmentation into moving and non-moving regions. This type of segmentation is used very often for outdoor scene analysis, as the projected objects are relatively small and detailed object structure is not detectable ([44], [35]). This approach is especially suited for object detection if there is a homogeneous background.

For image sequence segmentation a (nearly) application-independent 2–D system module was developed by the author [62]. At first, this module contains mostly conventional methods for iconic processing, like: image normalization, edge- and region image detection, and motion mask estimation (Fig. 2(a)-(d)). Secondly, the image segmentation steps follow: a "free" region detection (large region in front of the camera), line segment detection, segment grouping (linking lines to regions), represented by closed boundary contours, and contour motion estimation (Fig. 2(e)-(h)).

3.2. ANN-technique of visual motion estimation

Let us consider the question if a visual motion detector can be helpful for segment classification. In a stationary camera case a positive answer can be given. Any visual motion detector can be applied to two or more consecutive images.

We propose a relaxation-like process in a feedback neural network, that can compute optical flow for two consecutive images. Let a network with $(n \times m \times k^2)$ output neurons be given, where the neuron $o_{(x,y,D)}$ represents the hypothesis, that the pixel (x, y) in the first image corresponds to pixel $(x + v, y + u)$ in the second image.

Let us denote by $D = fun(u, v)$ a linear mapping from motion vector (x, y), to an index D, where $\{v, u =< -k/2 + 0.5, k/2 - 0.5 >\}$. The relaxation rule is given as:

$$y^{(t+1)}_{(x,y,v)} = \sigma[\sum_{Exc.neighbors} y^{(t)}_{(xn,yn,vn)} - \eta \sum_{Inh.neighbors} y^{(t)}_{(xn,yx,vn)} + y^{(0)}_{(x,y,v)}]. \quad (5)$$

Here η is a relaxation constant and σ is a threshold function. The local excitatory neighborhood $S(x, y, D)$ and the local inhibitory neighborhood $P(x, y, D)$ are defined as:

$$S(x, y, D) = \{(a, b, c)|(a = x \ or \ a = x - v) \ and \ (b = y - v \ or \ b = y)\}, \quad (6)$$

$$P(x, y, D) = \{(a, b, c)|(|(a, b) - (x, y)| \leq N) \ and \ c = D\}. \quad (7)$$

Initially the network contains the cross-correlation of the images

$$y^{(0)}_{(x,y,v)} = I^{(0)}_{(x,y)} \ I^{(1)}_{(x+v,y)}. \quad (8)$$

Thus the initial state represents the full set of possible pixel motions. During learning this set is steadily reduced until a stable state of the network is reached. The motion of some pixel $(x1, y1)$ is represented by one particular active output neuron from the set:

$$O_{x1,y1} = \{o_{(x1,y1,i)}|i \in < 0, k^2 - 1 >\}.$$

Only a single neuron in such set should be active at the end of the relaxation process.

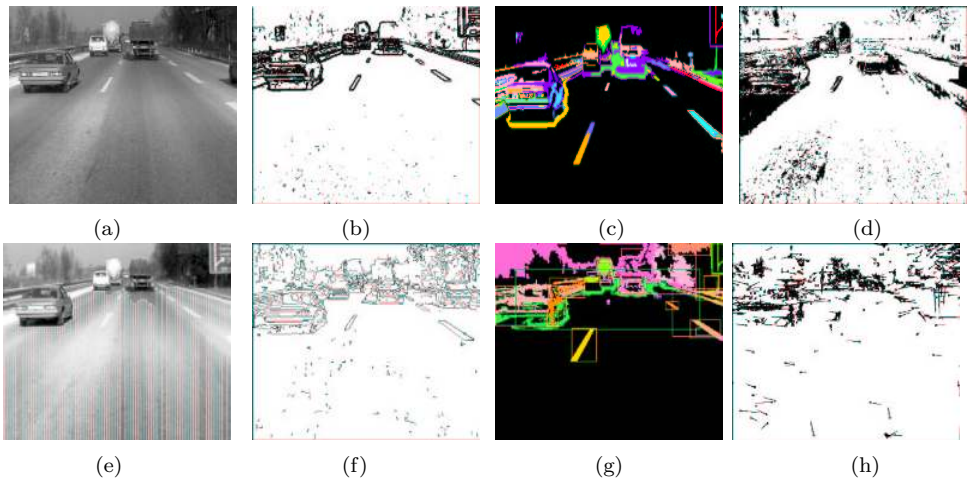
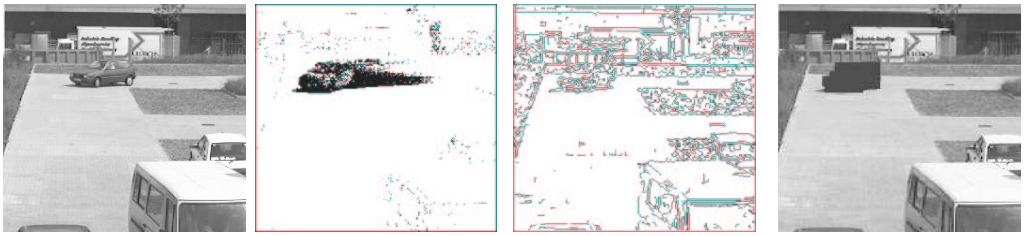


Fig. 2. Result samples of analysis steps from the 2–D module: (a) normalized image, (b) edge image, (c) region image, (d) the motion mask image, (e) free region, (f) line segments, (g) segment groups (contours), (h) contour motion vectors

As a result the optical flow is generated. In a simplified version of this method the visual motion is detected only, without estimating its direction and magnitude at each

pixel. The resulting so called *dynamic mask* image can additionally be tracked from image to image, giving the adaptive motion mask (Fig. 3). Now the image segments are classified into "moving" or "stationary" depending on the amount of covered "moving" pixel.



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

Fig. 3. Motion based segment classification in the stationary camera-case: (a) a normalized image, (b) the motion mask image, (c) segmented image, (d) moving objects, detected from grouping of moving segments.

3.3. The moving camera case

In a moving camera case due to partially unknown ego–motion and unknown environment the simple visual motion detection methods does not work properly (see Fig. 4). This is even more evident in case of a truly moving camera. From Fig. 5 it is clear that the optic flow in image regions representing moving obstacles and stationary surrounding area very frequently changes its sign, whereas the regions representing the road surface have approximately always the same number of upwards and downwards moving pixels - thus these regions seems to be more stable in the optic flow images. But in any case the obtained optic flow is not properly separating the stationary background from moving objects.

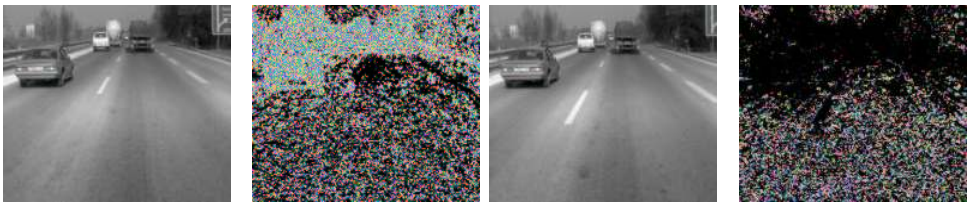
(a) (b) (c)

Fig. 4. Examples of wrong visual motion estimation if the stationary camera is performing an even small nodding movement: (a) one original image, (b) relatively proper visual motion, (c) relatively wrong visual motion after the camera has unpredictably moved.

3.3.1. Corrected visual motion under ego–motion

In order to overcame the influence of camera motion we developed a model-based scheme. Two development stages can be distinguished: visual motion in road plane (assuming known camera movement) (Fig. 6) and ego-motion corrected visual motion (moving camera with unknown nodding movement) (Fig. 7).

Let us first assume that the camera is moving with constant and known velocity and that no nodding movement occurs. This is approximately true for an indoor environment and for relatively slow moving car or robot. The camera is turned toward the ground, i.e. the vanishing point is located in the image plane over the visible image area. Hence the vanishing point can be fully back–projected onto the assumed ground plane (Fig. 6). As all points are assumed to be located on the ground, a transformation of the whole image into the "synthetic" road plane can be computed. Now a standard visual motion estimation can be applied for two consecutive ground images. This leads to motion vectors in the ground plane. The known ego-velocity of the vehicle provides a threshold for obstacle detection. If an obstacle is violating the planarity assumption it corresponds to high displacement values in the estimated motion field.



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

Fig. 5. Wrong optical flow in moving camera case due to unknown nodding camera movement: (a, c) images no. 14 and 20; (b, d) pixels with negative y-component of gradient-based optical flow.

In case of unknown camera nodding movement we try to recognize the instantaneous camera position, i.e. assuming that we have an outer orientation point or we can measure the relative position against the ground and/or horizon. Then we can eliminate the influence of the movement of camera between two consecutive images. Finally we establish a correspondence between current image and a corrected previous image, from which the influence of estimated current ego–motion is eliminated (Fig. 7) [60].

3.3.2. FOE-based segment classes

When the camera is performing a known translating movement and the scene is planar or sufficiently distant, then still the 2-D motion may be sufficient for moving object detection. The dynamic *focus of expansion* point FOE (or epipole), corresponding to egomotion, is estimated, which is defined as the image plane point in which the motion vectors of (hypothetical) stationary background points vanish. A differently than

the camera, moving object induces its own individual FOE point (denoted by C_FOE) (like for contour segments Fig. 8(d)). By comparing the general FOE point with image segment specific C_FOE points the image segment classification into "stationary' background and different "moving" objects is possible [63].

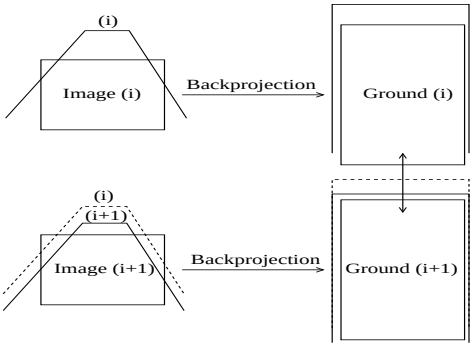


Fig. 6. The principle of visual motion in road plane under known ego-motion.

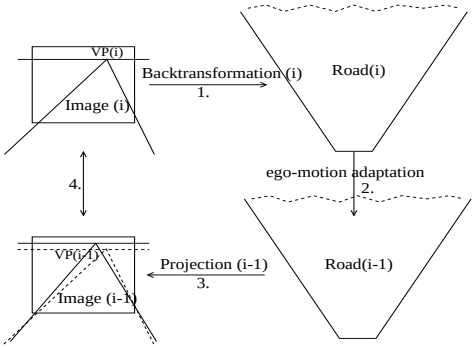


Fig. 7. The principle of visual motion with unknown ego–motion correction.

4. 2-D object tracking

The image segment detection in individual images may be extended in case of image sequence analysis to the segment tracking process. As motion data are immediately available from tracked segments, a so called "sparse" visual is the result of image segment tracking. In the past it was proposed to track different discrete image segments.

Let us detect a block of pixels that contains a 2–D texture in a first image. The corresponding problem for this block in the next image from given image sequence may be solved by *correlation methods* ([45]). The visual motion vector for this block is determined by tracking its position continuously from image to image.

In the *edge tracking* method ([21]), the image location and direction of a tracked edge is predicted. Around this expectation a search area is built. In this area it is searched for the best corresponding edge. Most often this requires the use of heuristic methods for pruning the correspondence space [52]. If point features, like corners, can be distinguished in the image, then their trajectory determination can give good results [26, 55].

The correspondence of more complex image features like, for example, *local symmetries* can also be searched for [34]. In this approach it is tried to detect and to track complex objects, like vehicles, as groups of pairs of symmetrical features. In similar approaches the correspondence and tracking of elements of a 2-D object model is per-

formed, under the assumption that nearly always a front or back view of a 3-D object is available only ([49, 33]). But the main goal of these methods is a stable detection and tracking of the image position of the object and not the estimation of its visual motion. Some dominant object points can define a 2-D contour hypothesis. This contour can change its shape in an elastic way during the tracking process (so called *active contour* tracking) [56].

4.1. 2-D object contour tracking

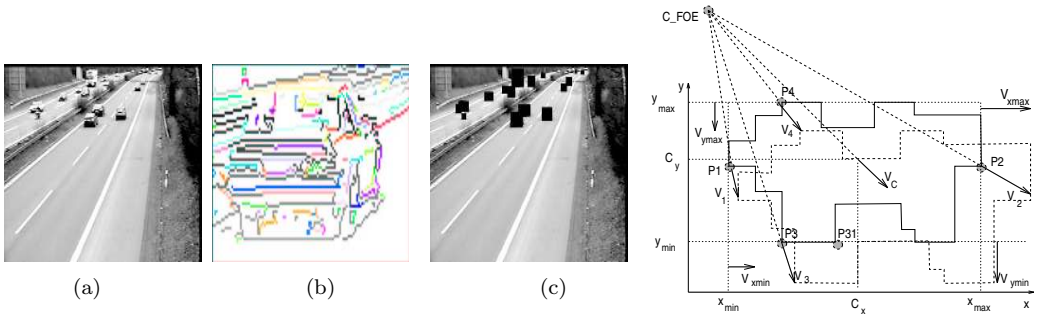
For outdoor scenes it was tried to detect short trajectories of many significant points of a ground plane moving object and to determine its 3–D structure ([2]) or at least its depth and motion ([41], [50]). In practise these applications require a stationary camera, a nearly orthographic projection and one large-sized object in the scene.

In [38] we represent vehicle objects, recognized as segment groups, by their closed boundary contours (Fig. 8). The state vector of a contour segment hypothesis consists of the geometric and dynamic features, i.e.:

$$\boldsymbol{s}_C(k) = [l(k), C_x(k), C_y(k), ..., v_x(k), v_y(k), v_z(k)]^T. \tag{9}$$

The measurement vector is of similar structure as the state vector, although only the position is measured directly. Dynamic features of a contour are obtained due to a short time correspondence of this contour found in up to N consecutive images ($N = 2 - 5$). Following dynamic contour features are available (Fig. 8(d)):

- visual motion of specific positions - mass center motion $v_C = (v_{Cx}, v_{Cy})$, boundary box motion $(v_{xmin}, v_{xmax}, v_{ymin}, v_{ymax})$, up to four point motions - $v^{P1} = (v_x^{P1}, v_y^{P1})$, $v^{P2} = (v_x^{P2}, v_y^{P2})$, $v^{P3} = (v_x^{P3}, v_y^{P3})$, $v^{P4} = (v_x^{P4}, v_y^{P4})$;
- length change rate – of contour length v_z^l, of diagonal distance v_z^d.



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

Fig. 8. Example of 2-D object tracking: (a) image from a sequence, (b) detection and classification of an image window with a single object, (c) 2-D object (contour) tracking, (d) visual motion of a contour.

In images of road scenes the contour motion in image plane corresponds to some object motion in the road plane. For example if a contour represents a moving car its approximated translational motion can be computed on the basis of estimated contour length change rate v_z, or on the different motion vectors v_x, v_y of contour points, respectively [62].

Under partially unknown camera ego-motion it is very difficult to perform a reliable tracking of contours corresponding to single image segments. For example, the stationary background consists usually of single segment contours. It was tested, that the estimation of the focus of expansion point FOE is of much worse performance than the detection and estimation of the road's geometry-based vanishing point [48]. In case of moving objects, a contour should correspond to a reliable detected segment group.

4.2. Image window classification

The ANN-learning scheme can be applied in the measurement step performed during 2-D contour tracking for each single object hypothesis. Let an image window of some normalized size is scanned to an input vector $\boldsymbol{Y}$. Let $\boldsymbol{D}$ be a projection matrix onto the *discriminant analysis* (DA) space:

$$\boldsymbol{Z} = \boldsymbol{D}\boldsymbol{Y}. \tag{10}$$

$\boldsymbol{Z}$ is a new feature vector from C classes, with class means located at $M_i, 1 = 1, 2, ..., C$.

In discriminant analysis we want to determine the projection matrix $\boldsymbol{D}$ that maximizes the ratio:

$$\frac{det(\boldsymbol{S}_b)}{det(\boldsymbol{S}_w)}. \tag{11}$$

In other words we want to maximize the between–class scatter $\boldsymbol{S}_b$, while minimizing the within–class scatter $\boldsymbol{S}_w$. It is known, that this ratio is maximized when the column vectors of projection matrix $\boldsymbol{D}$ are the eigenvectors of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$, associated with the largest eigenvalues. The scalar components in $\boldsymbol{Z}$ are feature values of given sample and the column vectors of $\boldsymbol{D}$ are the MDF feature vectors.

A standard numeric approach to DA would be very time consuming, as the vector sizes are very large. The approach usually requires matrix inversion of matrices with sizes equal to the size of a cross-correlation matrix of input vector $\boldsymbol{Y}$. An ANN-learning approach is preferred instead. For an adaptive solution of the DA problem we have developed the following adaptive algorithm, which can be called a *supervised LVQ* algorithm:

1. There exists N output neurons. Each output neuron i represents a class label C_i.
2. A learning sample p consists of an input vector $\boldsymbol{x}^p$ and its correct class label d^p.
3. Two winners are determined, the best one k and the second best l, by using distance measures between weights $\boldsymbol{w}_i$ of $i-$th output neuron and input $\boldsymbol{x}$:
$$|\boldsymbol{x} - \boldsymbol{w}_k| < |\boldsymbol{x} - \boldsymbol{w}_l| < |\boldsymbol{x} - \boldsymbol{w}_i|, \forall i, i \neq k, l. \tag{12}$$
4. The class labels C_k^p and C_l^p are compared with d^p and two weight update rules are used according to the following strategy:

(a) Let $o_k^p \neq d^p$ and $d^p = o_l^p$.
(b)

$$\text{If } |\boldsymbol{x}^p - \boldsymbol{w}_k| - |\boldsymbol{x}^p - \boldsymbol{w}_l| < \epsilon \quad \begin{array}{l} \text{then } \boldsymbol{w}_l(t+1) = \boldsymbol{w}_l(t) + \eta[\boldsymbol{s}(t) - \boldsymbol{w}_l(t)] \\ \text{and } \boldsymbol{w}_k(t+1) = \boldsymbol{w}_k(t) + \eta[\boldsymbol{s}(t) - \boldsymbol{w}_k(t)]. \end{array} \tag{13}$$

Hence, the vector $\boldsymbol{w}_l$ corresponding to the correct label is moved towards the input vector, whereas the vector $\boldsymbol{w}_k$ with the incorrect label, which may even be nearest the input, is moved away from it.

4.3. The road following/recognition task

A research topic closely related to autonomous navigation and obstacle detection is the road recognition problem [25, 27, 30]. The integration of systems for road recognition and traffic object recognition will lead to car driver assistance systems in relative near future [51].

Current development stage in road recognition includes the design of specialized hardware and software systems, that not only track the known state but also make automatic adjustments to lane and road type changes [59, 57]. They are also able to distinguish between a road lane marker and an obstacle [37]. An improved road recognition scheme was proposed by the author in [64]. The method recognizes the number of road lanes and estimates the width and curvature of the road. The ego-state consists of the camera orientation, of the camera location relative to the road center line and of the ego–velocities. In our approach to road curvature estimation it is required to monitor the velocity and the steering angle of the ego–car, i.e. in order to be able to estimate the rotational velocity of the ego–car.

5. Model-based 3-D vehicle tracking

By adapting control techniques a dynamic system with feedback for 3-D object tracking was first defined in [12]. Applications of this 3–D model based object tracking approach has been originally developed for flight and aerospace applications. It has later been transformed to applications in road scene analysis and for autonomous navigation in robotics. This approach is especially suitable for single object tracking if an exact object model is available, i.e. exact shape and dimensions.

A model based method for automatic satellite docking at space stations is proposed in [17]. The image measurement contains only a small number of significant points, which allows the recovery of the 3–D position of the station in space. The rotations are restricted to one plane and only the relative camera motion has to be recovered. In [28] the same application field is assumed as in the previous paper. A tracking method for known objects in space is described, where the number of degrees of freedom of object

motion is increased to six. The object can freely rotate and the camera is stationary. The measurement contains either points or edges, whereas the objects are defined either by wire frames or by points.

In [39] an approach to single vehicle tracking on the ground plane with a stationary camera is described. A parametric shape model (with 12 lengths) is applied that enables the modeling of different vehicle types. It is assumed that all the recognized objects are moving forward. The model edges are projected into the image plane and a match between them and the image edges is performed.

In [62] the single object tracking task is extended to many object recognition. The description of the developed 3–D vehicle recognition method starts with the object model specification, i.e. the state and measurement vectors for different specializations of the model. Then the particular object hypothesis tracking method, including measurement procedures, and the final object selection steps are described in detail.

5.1. The object model

An object hypothesis is specified by its class and its state vector $\boldsymbol{s}$:

$$\boldsymbol{s}(k) = [\boldsymbol{s}^d(k), \boldsymbol{\xi}(k)] \tag{14}$$

where $\boldsymbol{s}^d$ is the *trajectory vector*, that specifies the object position and motion on the ground plane, and $\boldsymbol{\xi}$ is the *shape vector*. Thus two trajectories are considered: the camera vehicle trajectory and the moving object trajectory. The two trajectories are approximated locally by circular arcs in the road plane.

A *trajectory* vector $\boldsymbol{x}(k)$ at time point t_k is a five–dimensional vector

$$\boldsymbol{s}^d(k) = [p_X(k), p_Z(k), \Theta(k), (V(k), \omega(k))]^T, \tag{15}$$

that consists of the position $(p_X(k), p_Z(k))$ and orientation Θ of the translational motion, and of the magnitudes $V(k)$ and $\omega(k)$ of translational and angular velocities, respectively.

The parameters of the shape vector of an object hypothesis are the width $Width$ and several parameters κ_i:

$$\boldsymbol{\xi}(k) = [Width(k), \kappa_1(k), \kappa_2(k), ..., \kappa_j(k)]^T \tag{16}$$

The number and meaning of the components κ_i depends on the object class and on the specialization of the shape representation. A general shape of a vehicle is modeled by two boxes with equal width, i.e. five shape parameters are necessary: $Width, Length_1$, $Length_2, Height_1$ and $Height_2$. Besides general shapes, which belong to the first specialization level of the object model, called the *object* level, two specialized model levels are defined: *object_shape* and *object_fine* (Fig. 9).

In the first (most general) case the 3–D object hypotheses are repeatedly generated and a matching between previous and new hypotheses takes place. The two shape parameters $height_2$ and $length_2$ are dependent and are assumed to be related to the independent parameters $height_1$ and $length_1$ by a constant. In the second case a goal

oriented 2–D measurement takes place for each previous object hypothesis. The matching takes place on the level of 2–D edges. The models of class *object_fine* are working with the edge based 2–D matching alone. Their goal is a detailed classification of the vehicle hypothesis, if the image dimension of the object allows it. This specialization level can contain such shapes like *car* or *truck*. The measurement step is similar to previous specialization level.

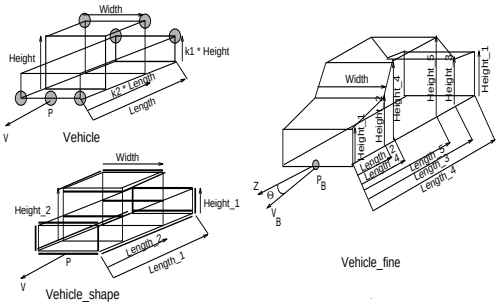


Fig. 9. Three vehicle model specializations.

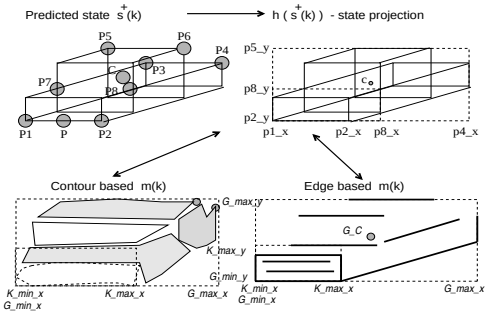


Fig. 10. The 2–D measurement via contours or edges (right).

5.2. The 2–D measurement

The matching of model feature with the next image features can be performed alternatively on two ways: the measured points are derived from contour groups or from line segment groups (Fig. 10). In both cases the state modification process is based on the differences between projected model points and significant points of measured data group. The projected model points are matched against the measured points from the vector:

$$\boldsymbol{m}(k)^T = [K_{minx}, K_{maxx}, G_{Cx}, G_{Cy}, G_{minx}, G_{maxx}, G_{miny}, G_{maxy}, K_x, K_y]^T \qquad (17)$$

Thus the 2–D measurement vector contains the x–components of the lower bounding box (K_{minx}, K_{maxx}), the gravity center $G = (G_x, G_y)$, the overall bounding box and the front location (K_x, K_y) of the smaller object box (Fig. 10).

Many alternative projections of the object hypothesis are generated in the image by varying the state parameter values (with exception of ω and V). The model alternatives are limited by several constraints – i.e. width, height or the width–to–height relation). For each alternative model k_a of a hypothesis k the distance $D_d(k_a)$ to the segment group d is calculated and the alternative k_a with a smallest distance $D_d(k_a)$ is selected. In Fig. 11 an example of three consecutive state projections and measurements are provided. The adaptation of the hypothesis on a real vehicle shape can be observed.

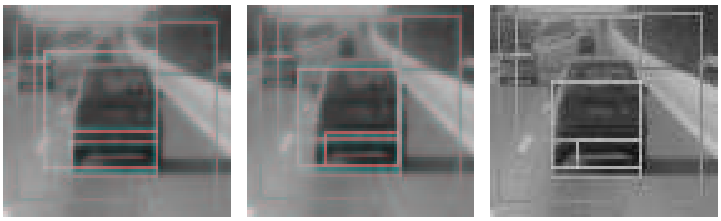


Fig. 11. Adaptation of the measured segments and the object hypothesis onto the real shape in consecutive images (the outer double box denotes the image search area, the smaller one the object prediction and the bright one comes from the measured data).

6. Many-object scene description

A paradigm for the selection of best hypothesis subset from the set of all generated hypotheses is required. In the simplest case the selection process deals with a set of (partially) competitive homogeneous hypotheses, i.e. all hypotheses represent objects belonging to the same representation level and no dependency relations like *is-part-of* or *is-specialization-of* appear between them. In the general case objects of complex structure, represented by a tree or graph, are assumed. The hypothesis selection process can be seen as a *matching problem* between the model structure and image data. Multiple hypotheses of all objects and its parts or specializations may be generated and the goal of such matching is to find a best (according to given judgement function) consistent subgraph (or subtree) of heterogeneous object hypotheses.

6.1. Hypothesis selection for simple objects

A hypothesis is either in its *tracking* or in one of its *recognition* phases. A tracking phase is given if the tracking time of this hypothesis is lower than T_{min} or its variance is greater than the $maximum\_var$. Otherwise the hypothesis is in one of its three recognition phases. These phases are closely related to the use of object specialization levels (from *Shape* over *Fine* to *Type*).

The consistency test takes place between pairs of hypotheses. If the tracking times of two competitive hypotheses are both larger than T_{min} or one of them is in the recognition phase (i.e. its tracking time is longer than some threshold time T_{min} and its combined variance is lower than some $Maximum\_Var$), then the consistency test among them is performed. After a hypothesis has been tracked successfully for some time its status changes to the recognition phase (Fig. 12).

6.2. Hypothesis selection for complex objects

Two general approaches for finding consistent analysis results could be be distinguished:

Fig. 12. Examples of selected vehicle hypotheses in three image sequences.

1. logical approach - using a truth maintenance system [7] and (b) a *central inference net* with *relaxation-like* or truth maintenance control (deterministic analysis control, one central inference net with analysis results, relaxation-like control for consistency detection [1]);
2. pattern recognition approach - by state space search [4] (non-deterministic analysis control with graph representation of the decision space, many local inference result nets, tree search for consistency detection [6]).

In the graph-search approach with each search graph node a local consistent inference subnet is associated. In the truth maintenance approach only one central inference net is given that may contain many inconsistencies. Let for example five competitive instances of some concept $Q(A)$ are generated. In the search approach this instantiation causes the generation of five competitive search tree nodes, where each of them contains one instance of $Q(A)$. In the truth maintenance approach five assumption nodes $A_1, ..., A_5$ and five justification nodes $J_1, ..., J_5$ are added to the central inference net. An assumption is contained in the net as long as one of its conclusions (justification nodes) is not proved to be wrong. All the justification nodes containing this wrong assumption can be immediately removed from the inference net.

In the context of image sequence analysis the control should satisfy two (to some extent contradictory) conditions: (1) it should avoid an exhaustive analysis flow by providing sufficient *selectivity* of analysis, it should provide a *stable* tracking of concept instances, i.e. the consistency decisions made for earlier image should not be changed during next image analysis.

In the context of dynamic analysis the consistency maintenance approach has an advantage over the tree search approach if stabilization power is considered. As the decision about competitive instances may be postponed to a later time (if necessary after several images) a parallel tracking of competitive hypotheses is always possible. Thus the tracked hypotheses are stabilized independently all the time. The selection decision made after such a long initialization phase is more proper than a decision made immediately after first appearance of an object hypothesis. At other side the selectivity of analysis is an important requirement for analysis of every image from the sequence. Here

the use of an optimal tree search approach is preferred, as it reduces the computational effort and provides a small number of alternative consistent solutions.

6.3. Graph search

After Nilsson [4] let us denote by the A^*-search the *best node* search algorithm in finite graphs. The A-algorithm employs an additive cost function:

$$f(n) = g(n) + h^*(n), \tag{18}$$

where $g(n)$ means the current traversal costs from start node to current node n, and $h^*(n)$ are (expected) remaining costs of the solution path from node n to some terminal node (heuristic part).

A^* is an optimal graph search algorithm but only under *consistent* (monotonic) heuristics [4, 6]. Some generalizations of the A-algorithm, with better behavior than A in the case of graphs with inconsistent or non-admissible heuristics were proposed, e.g. A^{**} [11], B C [5] and D [16].

The interchange between heuristic quality and search complexity is not elastic. Very high precision of heuristic is required in order to reach a polynomial complexity [6]. If fore some application the design of a "well" heuristic judgement is not possible, but one wants to keep the search complexity as low as possible, *sub-optimal* search methods (e.g. [6]) or *locally optimal* search (e.g. [24]) and *hybrid* methods (optimal search combined with depth-first search) (e.g. [19]) may be considered.

7. Summary

Presented computer vision research has to be perceived in a large context of transportation projects aiming to achieve improved traffic solutions by intensive application of information technology. For example more than 150 european companies and research institutes have cooperated in the research project EU45-PROMETHEUS between 1986 and 1994, continued by the project PROMOTE (Program for Mobility and Transportation in Europe) and national projects [20, 53]. The main project targets were:

Driver security: improving the field of view, supervising the dynamic of the car, controlling the distance of view and status of the driver, autonomous road following, collision avoidance.

Cooperated driving: communication between cars on roads, autonomous car driving, automatic warning.

Traffic control: mobile phone networks and GPS systems for efficient fleet management, digital road maps, on-road orientation systems, dissemination of multimedia travel and traffic information.

References

1979

[1] Haralick R., Shapiro L.: The Consistent Labeling Problem, Part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1(1979), No. 2, 173–184.

1982

[2] Dreschler L.S., Nagel H.-H.: Volumetric model and 3–d trajectory of a moving car from monocular TV-frame sequences of a street scene. *Computer Graphics & Image Processing*, vol. 20(1982), 199–228.

[3] Hopfield J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(1982), 2554-2558.

[4] Nilsson N.: *Principles of Artificial Intelligence.* Springer, Berlin etc., 1982.

1983

[5] Bagchi A., Mahanti A.: Search Algorithms under Different Kinds of Heuristics – a Comparative Study. *Communications of the ACM*, vol. 30(1983), No. 1, 1–21.

1984

[6] Pearl J.: *Heuristics. Intelligent Search Strategies for Computer Problem Solving.* Addison–Wesley, Reading, Mass., 1984.

1986

[7] De Kleer J.: An Assumption-Based TMS. *Artificial Intelligence*, vol. 28(1986), 127–162.

[8] Rummelhart D.E., McClelland J.L.: *Parallel Distributed Processing.* vol. 1 and 2, MIT Press, Cambridge MA, 1986.

1988

[9] Aggarwal J.K, Nandhakumar N.: On the computation of motion from sequences of images - a review. *Proceedings of the IEEE*, vol. 76(1988), No. 8, 917–935.

[10] Bagchi A., Sen A.K.: Average–Case Analysis of Heuristic Search in Tree-Like Networks. In: Kanal L., Kumar V. (Eds.): *Search in Artificial Intelligence*, Springer Series Symbolic Computation - Artificial Intelligence, Springer Vg., New York-Berlin-Heidelberg, 1988, 131–165.

[11] Dechter R., Pearl J.: The Optimality of A*. In: L. Kanal, V. Kumar (eds.), *Search in Artificial Intelligence*, Springer Vg., New York-Berlin-Heidelberg, 1988, 166–199.

[12] Dickmanns E.D.: Object recognition and real–time relative state estimation under egomotion. In: A.K. Jain (ed.), *Real–Time Object Measurement and Classification*, Springer, Berlin–Heidelberg etc., 1988, 41–56.

[13] Dickmanns E.D, Graefe V.: Applications of dynamic monocular machine vision. *Machine Vision and Applications*, vol. 1(1988), 241–261.

[14] Fukushima K.: Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, vol 1(1988), 119–130.

[15] Kohonen T.: *Self-Organization and Associative Memory.* Springer, New York-Berlin etc., 1988.

[16] Mahanti A., Ray K.: Network Search Algorithms with Modifiable Heuristics. In: Kanal L., Kumar V. (Eds.): *Search in Artificial Intelligence*, Springer Series *Symbolic Computation - Artificial Intelligence*, Springer Vg., New York-Berlin-Heidelberg, 1988, 200–222.

[17] Wuensche H.-J.: *Bewegungssteuerung durch Rechnersehen.* Springer, Berlin, 1988.

1989

[18] Bielik A., Abramczuk T.: Real-time wide-traffic monitoring: information reduction and model-based approach. *Proceedings 6th Scandinavian Conference on Image Analysis*, 1223–1230, Oulu, Finland, 1989. Pattern Recognition Society of Finland Press.

[19] Chakrabarti P.P., et al.: Heuristic Search in Restricted Memory. *Artificial Intelligence, 41*(1989), 197–221.

[20] Franke U.: PROMETHEUS – wissensbasierte Systeme eröffnen neue Perspektiven im Strassenverkehr, In: W. Brauer, C. Freksa (Eds.): *Wissensbasierte Systeme, 3. Internationaler GI–Kongress,*(Series: Informatik–Fachberichte, vol. 227), Springer, Berlin, 1989, 363–376.

1990

[21] Deriche R., Faugeras O.: Tracking line segments. *Image and Vision Computing*, vol. 8(1990), No. 4, 261–270.

[22] Fujimori T., Kanade T.: An approach to knowledge–bssed interpretation of outdoor natural color road scenes. In: Thorpe, C. (ed.), *Vision and Navigation: The Carnegie Mellon Navlab.*, Chapter 4, Kluwer Academic Publishers.
[23] Kluge K., Thorpe C.: Explicite models for robot road following. In: Thorpe, C. (ed.), *Vision and Navigation: The Carnegie Mellon Navlab.*, Chapter 3. Kluwer Academic Publishers.
[24] Korf R.E.: Real–Time Heuristic Search. *Artificial Intelligence*, vol. 42(1990), 189–211.
[25] Morgan A., Dagless E., Milford D., Thomas B.: Road edge tracking for robot road following: a real–time implementation. *Image and Vision Computing*, vol. 8(1990), No.3, 233–240.
[26] Salari V., Sethi I.K.: Feature point correspondence in the presence of occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12(1990), No. 1, 87–91.

1992

[27] Dickmanns E., Mysliwetz B.: Recursive 3d road and relative ego–state recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14(1992), No.2, 199–214.
[28] Gennery D.B.: Visual tracking of known three–dimensional objects. *International Journal of Computer Vision*, 7:243–270, 1992.
[29] Kilger M.: Video-based traffic monitoring. *Proceedings 4th Int. Conf. on Image Processing and Its Applications*, Maastricht, NL, 1992, 89–92.
[30] Masaki I.: *Vision-based Vehicle Guidance.* Springer, New York, Berlin–Heidelberg etc. 1992.
[31] Polk A., Jain R.: A parallel architecture for curvature–based road scene classification. In: Masaki, I. (ed.), *Vision–Based Vehicle Guidance.* Springer, New York etc., 1992, 284–299.
[32] Schaaser L., Thomas B.: Finding road lane boundaries for vision–guided vehicle navigation. In: Masaki, I. (ed.), *Vision–Based Vehicle Guidance*, Springer, New York etc., 1992, 239–254.
[33] Schwarzinger M., Noll D., v.Seelen W.: Object recognition with deformable models using constrained elastic nets. In: S. Fuchs and R. Hoffmann (Eds.), *Mustererkennung 1992*, (Series: Informatik aktuell), Berlin etc., Springer, 1992, 96–104.

1993

[34] Bernasch J., Koutny R.: Stabile Objektverfolgung und Detektion von nicht–vorhersagbarem Verhalten in komplexen Bildfolgen. *Mustererkennung 1993, Series: Informatik aktuell.* Springer–Verlag, Berlin Heidelberg New York, 1993, 19–26.
[35] Bothemy P., Francois E.: Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, vol. 10(1993), No. 2, 157–182.
[36] Chen X., Dagless E., Zhang S., Thomas B.: A real–time plan view method for following bending roads. *1993 IEEE Symposium on Intelligent Vehicles*, Tokyo, Japan, 1993, 219–224.
[37] Graefe V.: Vision for Intelligent Road Vehicles. *Proceedings, IEEE Symposium on Intelligent Vehicles*, Tokyo, 135–140.
[38] Kasprzak W., Niemann H.: Visual Motion Estimation from Image Contour Tracking, *Computer Analysis of Images and Patterns.* Proceedings, Lecture Notes in Computer Science, vol. 719 (1993), Springer, Berlin etc., Germany, 363–370.
[39] Koller D., Daniilidis K., Nagel H.-H.: Model–based object tracking in monocular image sequences of road traffic scenes. *Int. Journal of Computer Vision*, vol. 3 (1993), No. 10, 257–281.
[40] Pomerleau D.A.: Neural networks for intelligent vehicles. *Processdings of IEEE Conf. on Intelligent Vehicles'93*, IEEE Publ., 1993, 19–24.
[41] Tan T.N., Sullivan G.D., Baker K.D.: Recognizing objects on the ground plane. *Image and Computer Vision*, vol. 12(1993), No. 3, 164–172.
[42] Thorpe C. (ed.), *Vision and Navigation: The Carnegie Mellon Navlab.* Kluwer Academic, 1993.

1994

[43] Barron J.L., Fleet D.J., Beauchemin S.S.: Performance of optical flow techniques. systems and experiment. *International Journal of Computer Vision*, vol. 12(1994), No.1, 43–77.
[44] Bober M., Kittler J.: Estimation of complex multimodasl motion: an approach based on robust statistics and hough transform. *Image and Vision Computing*, vol. 12(1994), No. 10, 661–668.
[45] Eklund M.W., Ravichandran G., Trivedi M.M., Marapane S.B.: Real–time visual tracking using correlation techniques. In: *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, IEEE Computer Society Press, Los Alamitos, CA, USA, 1994, 256–263.
[46] Ferrier N.J., Rowe S.M., Blake A.: Real–time traffic monitoring. *Proceedings. Second IEEE Workshop on Applications of Computer Vision*, Los Alamitos, CA, USA, 1994, 81–88.

[47] Kasprzak W.: Road Object Tracking in Monocular Image Sequences Under Egomotion, *Machine Graphics and Vision*, vol. 3(1994), No. 1/2, 297–308, ICS PAS Publ., Warszawa, Poland.
[48] Kasprzak W., Niemann H.: Moving segment detection in monocular image sequences under egomotion. *Signal Processing VII: Theories and Applications*, EURASIP, Lausanne, 1994, 708–711.
[49] Regensburger U., Graefe V.: Visual recognition of obstacles on roads. *IROS '94. Proceedings of the IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems*, 980–987, Munich, Germany, 1994.
[50] Tan T.N., Sullivan G.D., Baker K.D.: Linear algorithms for multi–frame structure from constrained motion. In: E.R. Hancock (ed.), *BMVC94. Proceedings of the 5th British Machine Vision Conference*, Sheffield, U.K., BMVA Press, 1994, 589–598.
[51] Wetzel D., Niemann H., Richter S.: A robust cognitive approach to traffic scene analysis. *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, Los Alamitos, CA, USA. IEEE Computer Society Press., 1994, 65–72.
[52] Zhang Z.: Token tracking in a cluttered scene. *Image and Vision Computing*, vol. 12(1994), No. 2, 110–120.

1995

[53] – : Prometheus: results on roads., *Eureka News*, 1995, No. 27, European Service Network Publ., Brussels, pp.12.
[54] Kasprzak W.: Ground plane object tracking under egomotion. *International Archives of Photogrammetry and Remote Sensing*, vol. 30(1995), Part 5W1, 208–213, ISPRS Publ., Zurich, CH.

1996

[55] Cox I.J, Hingorani S.L.: An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18(1996), No. 2, 138–150.
[56] Denzler J., Niemann, H.: 3D Data Driven Prediction for Active Contour Models with Application to Car Tracking, *IAPR Workshop on Machine Vision Applications, MVA'96*, Proceedings. University of Tokyo, Nov. 1996, Japan, 204–207.
[57] Jochem T.M.: *Vision Based Tactical Driving*, Carnegie Mellon University, Ph.D. dissertation, CMU-RI-TR-96-14, January 1996.
[58] Kasprzak W., Niemann, H.: Applying a Dynamic Recognition Scheme for Vehicle Recognition in Many Object Traffic Scenes, *IAPR Workshop on Machine Vision Applications, MVA'96*, Proceedings. University of Tokyo, Nov. 1996, Japan, 212–215.
[59] Maurer M., Behringer R., Thomanek F., Dickmanns E.D.: A compact vision system for road vehicle guidance. *13th International Conference on Pattern Recognition*, Vienna, August 1996, 313–317.
[60] Niemann H., Kasprzak W., Weierich P.: Integrated motion and geometry based obstacle detection in image sequences of traffic scenes. In: J. G. Verly (ed.) *Enhanced and Synthetic Vision*, Proceedings of SPIE, 2736(1996), 228–239, SPIE Publ., Orlando, Fl., USA.
[61] Takahashi K., Kitamura T., et al.: Traffic Flow Measuring System by Image Processing, *IAPR Workshop on Machine Vision Applications, MVA'96*. Univ. of Tokyo, 1996, 245–248.

1997

[62] Kasprzak, W.: *Adaptive Erkennung von bewegten Objekten in monokularen Bildfolgen bei Eigenbewegung*. INFIX, St. Augustin, Germany, *DISKI* Series, vol. 172, 1997, 181 pages.

1998

[63] Irani M., Anandan P.: A Unified Approach to Moving Object Detection in 2D and 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20(1998), No. 6, 577–589.
[64] Kasprzak W., Niemann H.: Adaptive Road Recognition and Egostate Tracking in the Presence of Obstacles. *International Journal of Computer Vision*, Kluwer Academic Pub l., Boston/DordrechtŁondon, vol 28(1998), No. 1, 6–27.
[65] Leondes C.T.: *Image processing and pattern recognition*. San Diego, Academic Press, 1998. Series: Neural Network Systems Techniques and Applications, vol. 5.