CHALMERS

Road traffic analysis based on visual information from video

Master of Science Thesis in the Master Degree Programme
Communication Engineering

MIKHAIL BOLBAT

Department of Signals and Systems

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2013
Report No. EX023/2013

Road traffic analysis based on visual information from video

Master of Science Thesis
in the Master Degree Programme
Communication Engineering

MIKHAIL BOLBAT

Examiner and supervisor: Prof. IRENE Y.-H. GU

Department of Signals and Systems

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden, 2013

Report No. EX023/2013

Contents

Acknowledgements

I am thankful to Irene Gu and Yixiao Yun for the visual materials and software they provided, and for their advice and discussions.

1. Introduction

The introduction of computer vision systems into the area of road traffic management opens wide opportunities for making the roads safer and more convenient. Possible applications for such systems can include:
– smart control over traffic lights using a feedback information on the traffic situation from a traffic analysis system;
– detection of violations of the traffic rules, as well as accidents and other emergency situations on the road;
– on-board vehicular safety systems for prevention of accidents and mistakes in driving;
– information systems for drivers providing real-time data on congestions and other problems on the roads;
– applications for journey time estimation depending on the current traffic situation;
– gathering long-term statistics on traffic situation in different time of the day, days of the week, seasons etc. for planning of urban development.

For these purposes, various traffic analysis systems are being developed and implemented now. The aim of this thesis project is to study the applicability of different image analysis and pattern recognition methods for the task of detection and tracking of vehicles on the road based on grayscale video sequences taken in different situations (static or moving camera, daytime vs. nighttime conditions, etc.).

The main elements of a basic road traffic analysis system are the following:
– detection of new vehicles that appear on the road;
– tracking of the vehicles movement;
– conversion from image to world coordinates in order to be able to obtain the results in a practically useful form.
From the data generated by these elements, statistical information on the traffic is extracted.

This thesis work is described in the following order. In Part 2, the state of the art in the traffic analysis methods is briefly described. In Part 3, the methods used in the conducted experiments are explained in details. In Part 4, the description of the traffic analysis mechanism, both for the cases of static and moving camera, is provided. In Part 5, the conducted experiments with real video sequences and their results are described. Finally, the conclusion on the work is given in Part 6.

2. Previous work

Object detection and tracking using the information from video sequences has been a topic of intensive research for the recent two decades. In this part of the work, the methodology developed for the different stages of the process is briefly summarized. In Chapter 2.1, methods for object detection used for identification of vehicles on the road are described. In Chapter 2.2, techniques for tracking the object movement are given. Finally, in Chapter 2.3, the possible approaches for the task of finding the correspondence between the object coordinates in the image and in the real world are mentioned.

2.1. Detection of vehicles

One of the possible approaches aims to detect moving objects on a static background (background subtraction). The background can be extracted by median filtering of each pixel location in time domain using a sufficiently large number of consecutive frames [28]; however, usually it requires for the camera to be perfectly stable. M.Piccardi [29] mentions the methods that build a statistical model of gray levels at each pixel location. A more sophisticated method developed by P.M.Jodoin et al. [5], [22] analyzes the statistics of frequency of gray level changes at each pixel position in order to decide whether this pixel belongs to the background or a moving object.

In cases where a static background cannot be directly estimated due to significant camera movement, the methods of camera motion estimation and compensation are used. M.Munderloh et al. [25] proposes a technique that uses a grid of tracked points for the camera motion estimation and compensation. T.Kanade et al. [27] applies RANSAC and Markov model to distinguish between trajectories of background and foreground points in a video sequence taken with a moving camera; this is the used for image segmentation between background and foreground objects. R.Dahyot [28] proposes an iterative method that estimates the camera translation and changes of the camera parameters between frames instead.

Different approaches to the problem of detection use a model of the relevant type of objects. Such models can be global, i.e. describing the properties of an object as a whole, or based on some local features of the type of objects that needs to be detected. Zhang et al.[32] proposes an object detection method using a set of gray level histograms. M.Murshed et al.[34] uses a method based on edge detection and analysis of edge displacement between consecutive frames. For identifying a relevant object in the image, many methods apply various kinds of feature points, such as histograms of oriented gradients (HOG) proposed by N.Dalal and B.Triggs in [12], scale-invariant feature transform (SIFT) developed by D.Lowe [30], or speeded-up robust features (SURF) developed by H.Bay et al.[33].

Some methods actually detect particular details specific for vehicles (wheels, symmetric rectangular elements, etc.). T.Kanade and H.Schneiderman [31] proposed a method of detection of objects by their specific parts using the wavelet transform. N.Kanhere in [6] used Haar-like features to detect rectangular elements of vehicles.

Detection methods based on some local details or feature points of relevant objects usually include an object model describing co-occurrence of the features for more reliable detection [35].

2.2. Tracking of vehicles

The most straightforward approach to tracking the vehicles is the template-based tracking, which search for a correspondence with a sample image of the object. D.Mohr and G.Zachmann [41] use a tracker based on the silhouette shape matching. F.Jurie and M.Dhome [40] proposed a method that uses splitting the object template into sub-templates. A.Cavallaro et al.[21] developed an algorithm that splits each object into segments and uses an individual template for each segment. This, however, can pose a problem of distinguishing between fragments of objects, single objects and close group of objects; this problem is addressed in [36].

When the objects to track are represented as a set of specific details or feature points, it opens an opportunity for tracking based on the feature point correspondence. C.Buarque et al. in [38] uses this approach with SIFT, and Hu Shuoa et al. in [39] – with SURF feature points. A.Ladikos et al. [23] combines the template-based method with feature points obtained with a corner detection algorithm.

Another approach to object tracking is to model the object with its global characteristics and search for an image region with the best possible match to these characteristics. D.Comaniciu et al.[19] developed a tracking procedure based on the mean shift algorithm and the object representation as a color histogram. B.de Villiers et al.[47] improved this algorithm by adding a more sophisticated object representation and a trajectory prediction with Kalman filter for better occlusion handling.

On the other hand, the tracker can simultaneously consider several hypotheses on the object trajectory and make a final decision between them later. Y.Rui and Y.Chen [45] proposed the particle filter tracking algorithm that generates multiple hypotheses on the object location ("particles"), checks their correspondence with the object model, and make decisions to drop improbable hypotheses and use only highly probable ones for predicting the further object movement. Ch.Yang et al. [46] developed a more sophisticated version of the particle filter method: it uses a double representation of the object – in a color histogram and edge orientation histogram form. Ch.Chang et al.[44] combined the mean shift and particle filter ideas in their tracking algorithm. The particle filter method also allows to use a "track before detect" approach, as in [37] by D.J.Salmond.

Some alternatives to these common approaches also exist. A.Yilmaz et al. developed several versions of an algorithm tracking the contours of objects [48][49]. A.Bhattacharyya et al.[42] tracks vehicles on the road using optical flow.

2.3. Conversion between image and world coordinates

The conventional approach to the task of finding the correspondence between the image and world coordinates includes camera calibration, i.e. finding the intrinsic and extrinsic parameters of the camera. This can be done using a two- or three-dimensional reference object, as proposed by S.Upadhyay et al. in [24]. N.Kanhere [6] describes a method of camera calibration using known details of the scene.

For a more general task of scene reconstruction, I.Gordon and D.Lowe [43] build a 3D model of the scene by combining information from different frames taken with a moving camera and using SIFT features to establish correspondence between the scene details in different frames.

An alternative approach is proposed by P.Melnyk and R.Messner in [50], who used the log-polar transform to mitigate the perspective instead.

The possible problem of non-linearities in the correspondence between the world and image coordinates and camera distortion compensation is studied in [7].

3. Methods used in the thesis work

This part provides a detailed description of the techniques used in this thesis work for extracting the information on road traffic. In Chapter 3.1, the method of describing the candidate objects is explained. Chapter 3.2 provides the theory of how the decision on whether a candidate object can be relevant is made. Chapter 3.3 describes the mechanism of object tracking. After this, the mathematics of correspondence between the image and the real-world coordinates is given; in Chapter 3.4, conversion from the world to image coordinates is described, and Chapter 3.5 shows the solution of the inverse problem for the particular case of a 2-dimensional road plane in the world coordinates.

3.1. Histogram of oriented gradients (HOG) as an object descriptor

In this thesis work, for the static camera scenarios, background subtraction using statistical filtering was implemented, which is a simple method explained in Chapter 4.2 within the description of the whole algorithm of image processing and analysis for the static camera. In the moving camera case, a more sophisticated mechanism including an SVM classifier based on HOG as object features is needed, which is described below.

The histogram of oriented gradients (HOG) is a method of object shape representation used for object detection. It divides the image of the object into a predefined number of regions (cells) and builds a histogram of image gradient orientations for each cell. Then the cells are combined into larger regions (blocks), and gradient orientation histograms for each block are built, normalized by average gradient magnitude in order to minimize the influence of contrast differences between different images of the same type of objects [12, p.2].

For computing a histogram of oriented gradients, first the gradient magnitude $\rho(x, y)$ and gradient direction $\omega(x, y)$ are calculated for all pixel locations (x, y) of the object image:

$$\rho(x, y) = \sqrt{D_X^2(x, y) + D_Y^2(x, y)} \qquad (3.1)$$

$$\begin{cases} \omega(x, y) = \operatorname{atan} \dfrac{D_Y(x, y)}{D_X(x, y)} \\ -\pi \le \omega < \pi \end{cases} \qquad (3.2)$$

where D_X and D_Y are the image gradients in the horizontal and vertical direction, respectively.

Every pixel location (x, y) is assigned to a corresponding cell. If the size of the object image is $x_{MAX} \times y_{MAX}$ pixels, and the cell grid consists of $N_X \times N_Y$ cells, the grid coordinates of the cell (C_X, C_Y) to which a particular pixel location (x, y) belongs, are:

$$C_X(x, y) = \left\lfloor \frac{(x - 1)\, N_X}{x_{MAX}} \right\rfloor + 1 \qquad (3.3)$$

$$C_Y(x, y) = \left\lfloor \frac{(y-1)\, N_Y}{y_{MAX}} \right\rfloor + 1 \qquad (3.4)$$

For each cell, a histogram of gradient directions is built. For the number of bins N_B, the bin b to which a particular direction ω corresponds, is calculated the following way:

$$b(x, y) = \left\lfloor \frac{(\omega(x, y) + \pi)\, N_B}{2\,\pi} \right\rfloor + 1 \qquad (3.5)$$

and then the values of the cell histogram $\boldsymbol{H}_C$ for every cell (i, j) and bin k are computed as an average gradient magnitude within the cell for the directions corresponding to this bin [4, p.4]:

$$\boldsymbol{H}_C(i, j, k) = \sum_{x, y} \frac{\rho(x, y)}{A_{ij}} \left| \begin{array}{l} C_X(x, y) = i \\ C_Y(x, y) = j \\ b(x, y) = k \end{array} \right. \qquad (3.6)$$

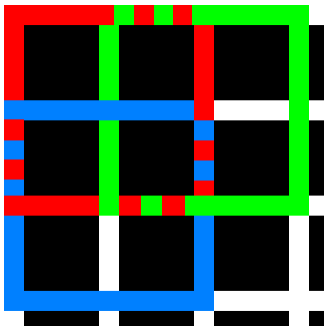


Fig. 3.1. Combining cells into blocks for HOG.

where A_{ij} is the area (number of pixels) of the cell (i, j).

As soon as the histograms for individual cells are made, they are combined together to form block histograms. For the blocks consisting of $N_I \times N_J$ cells each, a cell C_{ij} with the grid coordinates (i, j) is included into a block B_{pq} with the grid coordinates (p, q) if

$C_{ij} \in B_{pq}$ if:

$$(p \geq i) \cap (p \leq i + N_I - 1) \cap (q \geq j) \cap (q \leq j + N_J - 1) \qquad (3.7)$$

and the block histogram is formed as a concatenation of cell histograms for all cells within the block (see Fig. 3.1) [4, p.4]:

$$\boldsymbol{H}_{B0}(p, q) = \begin{bmatrix} \boldsymbol{H}_C(p, q) & \boldsymbol{H}_C(p+1, q) & \dots & \boldsymbol{H}_C(p+N_I-1, q) \\ \boldsymbol{H}_C(p, q+1) & \boldsymbol{H}_C(p+1, q+1) & \dots & \boldsymbol{H}_C(p+N_I-1, q+1) \\ \dots & & & \\ \boldsymbol{H}_C(p, q+N_J-1) & \boldsymbol{H}_C(p+1, q+N_J-1) & \dots & \boldsymbol{H}_C(p+N_I-1, q+N_J-1) \end{bmatrix} \qquad (3.8)$$

where
$\boldsymbol{H}_{B0}$ is the unnormalized block histogram,
(p, q) are the grid coordinates of the block,
N_I and N_J are the number of cells in the block in the horizontal and vertical direction, respectively.

Finally, each block histogram is normalized in order to suppress the influence of contrast differences and noise [11, p.5]:

$$\boldsymbol{H}_B(p, q) = f_N(p, q)\, \boldsymbol{H}_{B0}(p, q) \qquad (3.9)$$

where
$\boldsymbol{H}_{\mathbf{B}}$ is the resulting normalized block histogram,
f_{N} is the normalization factor [15, p.2]:

$$f_{\mathrm{N}}(p, q) = \sqrt{\frac{1}{\left\| \boldsymbol{H}_{\mathbf{B0}}(p, q) \right\|^2 + K\, \mu\, \{\boldsymbol{H}_{\mathbf{C}}\}}} \qquad (3.10)$$

where
$\left\| \boldsymbol{H}_{\mathbf{B0}}(p, q) \right\|$ is the norm of the unnormalized histogram for the (p, q) block,
$\mu\, \{\boldsymbol{H}_{\mathbf{C}}\}$ is the mean of the whole cell histogram for all cells,
$K = 0.1$ is the proportionality coefficient.

3.2. Support vector machine (SVM) as an object classifier

In practice, for the task of object detection in images, the HOG descriptor is combined with a classifier algorithm, which makes a decision on whether a particular object candidate can be an object of the relevant type. In this thesis work, the Support Vector Machine (SVM) classifier is used for this purpose. The SVM assigns any given object to either of two classes: +1 or –1. The classification mechanism is based on training samples, which have their class known, and tries to separate them with a hyperplane drawn at the maximum possible distance from the nearest samples (see Fig. 3.2a). This means that the hyperplane (shown in solid gray line at Fig. 3.2) has the following equation [17, p.2]:

$$\boldsymbol{w}^{T} f(\boldsymbol{X}_{\mathrm{i}}) + a = 0 \qquad (3.11)$$

where
$\boldsymbol{X}_{\mathrm{i}}$ is the vector of object features (the HOG in our case);
$f(.)$ is the function that maps the object features vector onto a feature space where the two classes of objects are linearly separable;
$\boldsymbol{w}$ is the normal vector to the hyperplane;
a is the offset constant depending on the coordinate system.

For all training samples, the following condition holds [17, p.2–4]:

$$c_{\mathrm{i}} (\boldsymbol{w}^{\mathrm{T}} f(\boldsymbol{X}_{\mathrm{i}}) + a) \geq 1 \qquad (3.12)$$

where c_{i} is the class where the sample $\boldsymbol{X}_{\mathrm{i}}$ belongs to (either $c_{\mathrm{i}} = 1$ or $c_{\mathrm{i}} = -1$). The samples for which the equality in (3.12) holds are called the support vectors.

The distance from a point $\boldsymbol{X}_{\mathrm{i}}$ in the feature space to the separating hyperplane is [17, p.3]:

$$d(\mathrm{Xi}, (\mathrm{w}, a)) = \frac{| \boldsymbol{w}^{\mathrm{T}} f(\boldsymbol{X}_{\mathrm{i}}) + a |}{\left\| \boldsymbol{w} \right\|^2} \geq \frac{1}{\left\| \boldsymbol{w} \right\|^2} \qquad (3.13)$$

Therefore, as it can be seen at Fig. 3.2a, the distance between the margins (shown in dashed gray lines) of different classes is [17, p.3]:

$$d_M = 2 \min \{d(X_i, (w, a))\} = \frac{2}{\|\boldsymbol{w}\|^2} \qquad (3.14)$$

However, in some practical cases the separating hyperplane does not exist (see Fig. 3.2c). In order to handle such situations, the classifier is designed to minimize the probability of classification error instead. In this case, the condition (3.12) has the following form [17, p.3]:

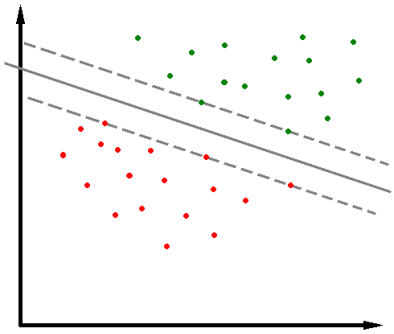


Fig. 3.2a. SVM classifier for a case when the samples can be linearly separated.

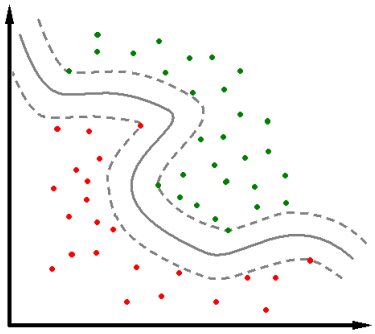


Fig. 3.2b. SVM classifier for a case of sample groups separable with a mapping function.

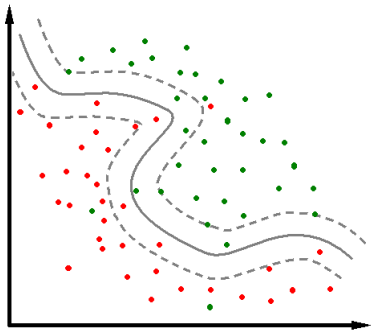


Fig. 3.2c. SVM classifier using a mapping function for inseparable groups of samples.

$$\begin{cases} c_i (\boldsymbol{w}^T f(\boldsymbol{X}_i) + a) \geq 1 - \varepsilon_i \\ \varepsilon_i \geq 0 \end{cases} \qquad (3.15)$$

where ε_i is the error for the case $\boldsymbol{X}_i$; it also needs to be mentioned that this situation leads to a classification error only if $\varepsilon_i \geq 1$, due to the nonzero distance between the margins of the areas in the feature space that the classifier intends to separate with a hyperplane.

This means that the task of building an SVM classifier can be formulated as an optimization problem [17, p.3]:

$$\begin{cases} d_M = \dfrac{2}{\|\boldsymbol{w}\|^2} \to \max \quad \left(\text{or } \dfrac{\|\boldsymbol{w}\|^2}{2} \to \min\right) \\ C \sum\limits_i \varepsilon_i \to \min \end{cases} \qquad (3.16)$$

that needs to be solved under the (3.15) constraints. (C is a constant regulating the degree of importance of avoiding errors in the classification of the training samples.) The Lagrangian for this problem has the form [17, p.3]:

$$\Lambda(\alpha_i, \lambda_i) = \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_i \varepsilon_i - \sum_i \alpha_i (c_i (\boldsymbol{w}^T f(\boldsymbol{X}_i) + a) - 1 + \varepsilon_i) - \sum_i \lambda_i \varepsilon_i \qquad (3.17)$$

where $\alpha_i \geq 0$ and $\lambda_i \geq 0$ are the Lagrange multipliers.

Therefore [17, p.3–4]:

$$\frac{\partial \Lambda}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_i \alpha_i c_i f(\boldsymbol{X}_i) = 0 \qquad (3.18)$$

$$\frac{\partial \Lambda}{\partial a} = -\sum_i \alpha_i c_i = 0 \qquad (3.19)$$

$$\frac{\partial \Lambda}{\partial \varepsilon_i} = C - \alpha_i - \lambda_i = 0 \qquad (3.20)$$

which also means that $\varepsilon_i = 0$ if $\alpha_i < C$.

This leads to [17, p.4]:

$$\boldsymbol{w} = \sum_i \alpha_i c_i f(\boldsymbol{X}_i) \qquad (3.21)$$

$$a = \frac{1}{N_{SV}} \sum_i (c_i - \boldsymbol{w}^T f(\boldsymbol{X}_i)) \Big|_{0 < \alpha_i < C} \qquad (3.22)$$

Therefore, the decision function for classifying any new HOG vectors will be [17, p.4]::

$$c(\boldsymbol{X}) = \text{sign}\,(\boldsymbol{w}^T f(\boldsymbol{X}) + a) = \text{sign}\left(\sum_i \alpha_i c_i f(\boldsymbol{X}_i)^T f(\boldsymbol{X}) + \frac{1}{N_{SV}} \sum_j (c_j - \right.$$

$$\left. - \sum_i \alpha_i c_i f(\boldsymbol{X}_i)^T f(\boldsymbol{X}_j)) \Big|_{0 < \alpha_j < C} \right) \qquad (3.23)$$

Now it is obvious that an explicitly stated mapping function $f()$ is not necessary. It is sufficient to formulate a kernel function $F()$ that makes an inner product between the mapping function of different arguments [17, p.2–4]:

$$F(\mathbf{X}_i, \mathbf{X}_j) = f(\boldsymbol{X}_i)^T f(\boldsymbol{X}_j) \qquad (3.24)$$

Several types of kernel functions exist [16, p.3]; in this thesis work, the Gaussian kernel is used:

$$F(\boldsymbol{X}_i, \boldsymbol{X}_j) = \exp(-\gamma \left\| \boldsymbol{X}_i - \boldsymbol{X}_j \right\|^2) \qquad (3.25)$$

where γ is a constant that needs to be adjusted to the particular classification problem, as well as the C. As recommended in [18, p.5], they are chosen between the values equal to 2 in a power of an integer number by analyzing the performance of the classifier at all combinations of these parameters within the selected range.

3.3. Kernel-based mean shift tracker

In this thesis work, for the static camera case a specially developed variant of template-based tracker was implemented, which is described in Chapter 4.2. For the moving camera, the mean shift tracker described in [19] was used. In this algorithm, the sample of the object is resized to a fixed size, which is equal to the size of the kernel [w_K, h_K]; it sets greater weights to the central part of the object [19, p.3]. In this thesis work, the Epanechnikov kernel is used [19, p.5]:

$$K(x, y) = 1 - \frac{(x - w_K / 2)^2 + (y - h_K / 2)^2}{(w_K / 2)^2 + (h_K / 2)^2} \qquad (3.26)$$

For the resized picture of the detected object, a gray level histogram G_S is built. Each pixel of the resized sample is assigned to a particular bin b:

$$b(x, y) = \left\lfloor g(x, y)\, N_B \right\rfloor + 1 \qquad (3.27)$$

where g is the gray level, and N_B is the number of bins in the histogram. The value of each histogram bin is calculated according to the weight that the kernel sets to each pixel:

$$G_S(i) = \frac{\sum_{x,y} K(x, y) \,\Big|\, b(x, y) = i}{\sum_{x,y} K(x, y)} \qquad (3.28)$$

During the object tracking process, this histogram is used as a template for comparison with such histograms of the object built for new locations being considered. As a criterion of similarity, the Bhattacharyya coefficient is used [19, p.3]:

$$B(G, G_S) = \sum_i \sqrt{G(i)\, G_S(i)} \qquad (3.29)$$

where G is the grey level histogram for the possible new position of the object.

While processing a new video frame, for each object present by the moment, an initial gray level histogram G_0 is built according to (3.27)–(3.28), assuming that the location of the object has not changed since the previous frame.

For the purpose of practical calculation of the Bhattacharyya coefficient, its approximation based on the Taylor series expansion is used [19, p.4]:

$$B(G, G_S) = \sum_i \sqrt{G(i)\, G_S(i)} \approx \frac{1}{2} \sum_i \sqrt{G_0(i)\, G_S(i)} + \frac{1}{2} \sum_i G(i) \sqrt{G_S(i) / G_0(i)} \qquad (3.30)$$

where G is the gray level histogram for a possible new location of the object being considered.

Only the second term in (3.30) needs to be maximized, since the first term does not depend on G. This means that a weight depending on G_S and G_0 is assigned to each bin of G [19, p.4]:

$$w(i) = \begin{cases} 0 & \text{if } G_0(i) = 0 \\ \sqrt{G_S(i) / G_0(i)} & \text{otherwise} \end{cases} \tag{3.31}$$

After these initialization procedures, the mean shift algorithm is applied to approach to a new location of the object [19, p.4]:

$$[du, dv] = \frac{\sum_{x,y} [(x - w_K / 2), (y - h_K / 2)]\, w(b(x, y))}{\sum_{x,y} w(b(x, y))} \tag{3.32}$$

where
$b(x, y)$ is the index of the histogram bin corresponding to the gray level of the pixel at the $[x, y]$ coordinates of the object sample,
$[du, dv]$ is the found shift of the object coordinates $[u, v]$ from the previous location.

Then $[u_o + du, v_o + dv]$ are assigned as the new values of the object coordinates $[u_o , v_o]$, G is assigned as the new value of G_0, and the calculations of (3.31) – (3.32) are repeated until the stop condition of the algorithm is fulfilled [19, p.4].

3.4. Camera self-calibration based on known details of the scene for conversion from world to image coordinates

Camera calibration is the problem of finding the parameters of a camera that produce the given mapping from the 3D world coordinates (x, y, z) to the 2D image coordinates (u, v). In this thesis work, camera calibration was done using the method described in [6], with some necessary corrections. This method is based on using some a priori information on the scene geometry, which can be known in advance. Such information includes:
– parallelism or perpendicularity of particular lines in the world coordinates,
– known real-world distances between some distinct points visible in the image.

The relationship between the 3D world coordinates (x, y, z) and the 2D image coordinates (u, v) depends on the optical parameters of the camera and its placement in the scene. In homogeneous coordinates, it can be represented the following way [7, p.1]:

$$s\,[u, v, 1]^T = \boldsymbol{K}\,\boldsymbol{R}\,[x, y, z, 1]^T = \begin{bmatrix} s_U f & \xi & s_U \tau_U \\ 0 & s_V f & s_V \tau_V \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{bmatrix} \cdot [x, y, z, 1]^T \tag{3.33}$$

where:

– s is a scaling coefficient;

– ***K*** is a matrix of intrinsic parameters of the camera, which include:
  – focal length f,
  – the width and height of a pixel – s_U and s_V,
  – the skew factor ξ, representing the degree of non-perpendicularity between the u and v axes of the image,
  – translation vector $[\tau_U , \tau_V]^T$ between the center of the image and the point of intersection of the main optical axis of the camera with the image plane;

– ***R*** is a matrix of extrinsic parameters of the camera; it consists of:
  – rotation matrix (r_{11}, r_{12} ... r_{33}),
  – translation vector $[t_X, t_Y, t_Z]^T$ between the zero point of the world coordinates and the camera center [7, p.1].

The elements of the rotation matrix (r_{11}, r_{12} ... r_{33}) are combinations of trigonometrical functions of the three rotation angles:
– pan angle θ – rotation around the vertical axis;
– roll angle ψ – rotation around the optical axis of the camera,
– tilt angle φ – rotation in the plane drawn through the optical axis and its projection onto the road plane (see Fig. 3.3) [6, p.61].

The number of parameters necessary to find can be reduced by making some realistic assumptions on the camera:
– the camera optics is supposed to have no defects that could introduce geometrical distortions into the image ($\xi = 0$),
– the camera is supposed to be mounted horizontally ($\psi = 0$),
– the main optical axis of the camera projects into the center of the image (τ_U, = 0, $\tau_V = 0$),
– pixels of the output image are perfectly square and are used as a measurement unit ($s_U = 1$, $s_V = 1$).

Taking into account also that the translation vector $[t_X, t_Y, t_Z]^T$ can be expressed via the camera height h and the rotation angles, only one intrinsic (f) and three extrinsic (φ, θ, h) parameters remain to be estimated**.** The matrix of intrinsic parameters turns to have a simple form [6, p.62]:

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.34)$$

The extrinsic parameters depend on the world coordinate system and, therefore, can be derived by turning from one world coordinate system to another step-by-step and introducing the corresponding modifications into the ***R*** matrix (see Fig. 3.3 for the coordinate systems used).

In the (X_0, Y_0, Z_0) coordinate system, centered in the camera center, $\varphi = 0$, $\theta = 0$, $h = 0$, and the matrix of the extrinsic parameters of the camera is a trivial identity matrix concatenated with a zero translation vector:

$$\boldsymbol{R_0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (3.35)$$

The (X_1, Y_1, Z_1) coordinate system introduces the angle φ between the road plane and the camera

optical axis:

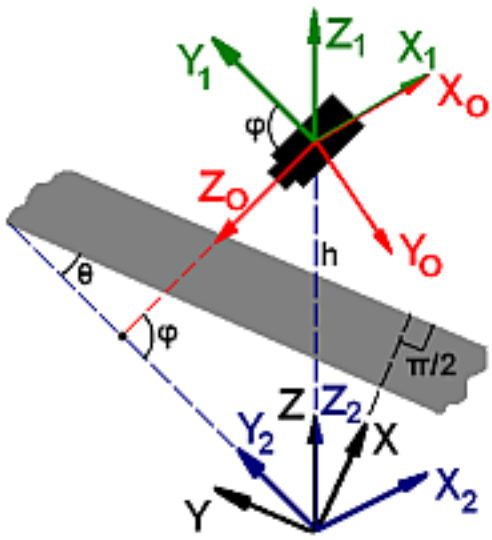


Fig. 3.3. The camera position over the road plane and the coordinate systems used.

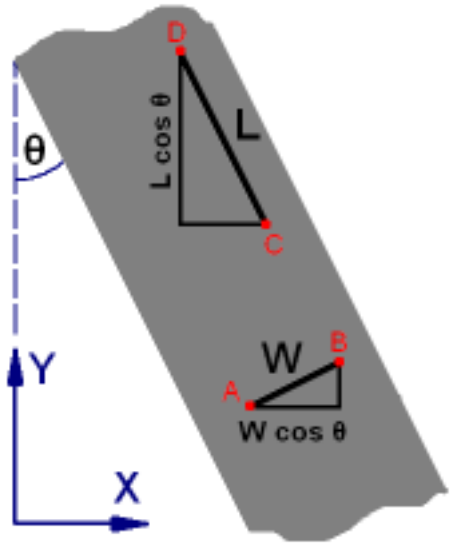


Fig. 3.4. The known distances used for camera calibration.

$$\boldsymbol{R}_1 = \boldsymbol{R}_0 \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi+\pi/2) & -\sin(\varphi+\pi/2) & 0 \\ 0 & \sin(\varphi+\pi/2) & \cos(\varphi+\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin\varphi & -\cos\varphi & 0 \\ 0 & \cos\varphi & -\sin\varphi & 0 \end{bmatrix} \qquad (3.36)$$

Then, the coordinate system is translated by the camera height h; this is the transition from (X_1, Y_1, Z_1) to (X_2, Y_2, Z_2):

$$\boldsymbol{R}_2 = \boldsymbol{R}_1 \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin\varphi & -\cos\varphi & h\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi & h\sin\varphi \end{bmatrix} \qquad (3.37)$$

In this coordinate system (X_2, Y_2, Z_2), the equation of (3.33) turns to the following form [6, p.61]:

$$s\,[u, v, 1]^{\mathrm{T}} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin\varphi & -\cos\varphi & h\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi & h\sin\varphi \end{bmatrix} \cdot [x, y, z, 1]^{\mathrm{T}} \qquad (3.38)$$

The θ angle is not included into the matrix yet; on this stage it remains as the angle between the projection of the main optical axis of the camera onto the road plane and the road traffic direction [6, p.61].

The remaining unknown parameters of the camera can be found using the data on the scene known in advance:
– road marking lines, and their parallelism or perpendicularity,
– the width of each lane,
– the length of the dashes in dashed road marking lines,
– possibly also the distances between some other details on the ground.

Detection of road marking elements was beyond the scope of this thesis work. It can be done using Canny edge detection method in combination with Hough transform and a priori information on the road marking standards [1, p.44–54]. As soon as the road marking lines are detected, the vanishing point of the lines heading along the road can be found in the image.

Let the image coordinates of this vanishing point be denoted as (u_0, v_0). This point corresponds to a point in infinity lying straight along the direction of the visible part of the road. As it can be seen in Fig. 3.3, its homogeneous coordinates in the (X_2, Y_2, Z_2) coordinate system are [6, p.63]:

$$V = [-\tan\theta,\ 1, 0, 0\ ]^T \qquad (3.39)$$

Applying (3.38) to these coordinates produces the following equations for the φ and θ angles:

$$\tan\varphi = -\frac{v_0}{f} \qquad (3.40)$$

$$\cos^2\theta = \frac{f^2 + v_0^2}{f^2 + u_0^2 + v_0^2} \qquad (3.41)$$

Then, by inserting the image coordinates of known points on the road surface (see Fig. 3.4) into (3.38), we get two additional equations:

$$\frac{h\,u_A}{(v_A - v_0)\cos\varphi} + W\cos\theta = \frac{h\,u_B}{(v_B - v_0)\cos\varphi} \qquad (3.42)$$

$$\frac{h\,(f^2 - v_C\,v_0)}{f\,(v_C - v_0)} + L\cos\theta = \frac{h\ (f^2 - v_D\,v_0)}{f\,(v_D - v_0)} \qquad (3.43)$$

Solving the equation system of (3.40), (3.41), (3.42) and (3.43) yields the necessary parameters of the camera: its focal length f, its height over the road surface h, and the angles φ and θ characterizing the position of the camera optical axis relatively to the traffic direction. In a case when the camera height h is known, only one known distance on the road surface (either W or L) is sufficient.

Finally, the transition from (X_2, Y_2, Z_2) to (X, Y, Z) coordinate system is made in order to include the θ angle into the camera matrix :

$$\boldsymbol{R} = \boldsymbol{R_2} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ -\sin\theta\sin\varphi & -\cos\theta\sin\varphi & -\cos\varphi & h\cos\varphi \\ \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi & h\sin\varphi \end{bmatrix} \qquad (3.44)$$

where $\boldsymbol{R}$ is the resulting matrix of the extrinsic parameters of the camera introduced in (3.33).

3.5. Planar homography for conversion from image to world coordinates

The camera calibration procedure described in Chapter 3.4 allows to locate the point in the image corresponding to the given point on the road plane. In some cases, the inverse problem needs to be solved: finding a road plane correspondence for any image plane point. For such situations, the method of planar homography can be used.

First, the correspondence for at least 4 points needs to be known in advance [20, p.3]. It is obtained by inverse mapping of a set of arbitrarily chosen points (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) onto the (u, v) image plane using the camera matrices $\boldsymbol{K}$ and $\boldsymbol{R}$ known from (3.34) and (3.44) and setting the vertical coordinate z to zero:

$$[s\,u_i,\ s\,v_i,\ s]^T = \boldsymbol{K}\,\boldsymbol{R}\,[x_i, y_i, 0, 1]^T \qquad (3.45)$$

Of the set of points (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), neither three can be located at a single straight line [20, p.3]. It is better to choose them as the angles of a square, for example [–100, –100], [–100, 100], [100, –100], and [100, 100].

In order to increase the accuracy of homography estimation, both the image plane and the road plane coordinate system undergo the Hartley normalization. This is done transforming them into the new coordinate systems (u', v') and (x', y'), such that:
– the zero points of the new coordinate system is located at the centroid of the set of points;
– the average distance from the zero point to each of the points is equal to $\sqrt{2}$ [20, p.4–6].

$$[x', y'] = \left([x, y] - [\mu\{x\}, \mu\{y\}]\right) \frac{\sqrt{2}}{\mu\{d_{xy}\}} \qquad (3.46)$$

where
$\mu\{x\}$, $\mu\{y\}$ are the mean values of the coordinates x and y, respectively;
$\mu\{d_{xy}\}$ is the mean distance from the points in the (x, y) coordinate system to their centroid:

$$\mu\{d_{xy}\} = \mu\left\{\sqrt{(x - \mu\{x\})^2 + (y - \mu\{y\})^2}\right\} \qquad (3.47)$$

Therefore, the transformation matrix $\boldsymbol{T}_{xy}$ necessary for the Hartley normalization of the coordinates:

$$[x', y', 1]^T = \boldsymbol{T}_{xy}\,[x, y, 1]^T \qquad (3.48)$$

appears to be the following:

$$\boldsymbol{T}_{xy} = \begin{bmatrix} \dfrac{\sqrt{2}}{\mu\{d_{xy}\}} & 0 & -\dfrac{\sqrt{2}\,\mu\{x\}}{\mu\{d_{xy}\}} \\ 0 & \dfrac{\sqrt{2}}{\mu\{d_{xy}\}} & -\dfrac{\sqrt{2}\,\mu\{y\}}{\mu\{d_{xy}\}} \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.49)$$

and, similarly, for

$$[u', v', 1]^T = \boldsymbol{T}_{uv}\,[u, v, 1]^T \qquad (3.50)$$

we get

$$\boldsymbol{T}_{uv} = \begin{bmatrix} \dfrac{\sqrt{2}}{\mu\{d_{uv}\}} & 0 & -\dfrac{\sqrt{2}\,\mu\{u\}}{\mu\{d_{uv}\}} \\ 0 & \dfrac{\sqrt{2}}{\mu\{d_{uv}\}} & -\dfrac{\sqrt{2}\,\mu\{v\}}{\mu\{d_{uv}\}} \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.51)$$

where

$$\mu\{d_{uv}\} = \mu\left\{\sqrt{(u - \mu\{u\})^2 + (v - \mu\{v\})^2}\right\} \qquad (3.52)$$

After the Hartley normalization is done, a homography matrix $\boldsymbol{H}'$ for the normalized coordinates is estimated, such that

$$[s\,x', s\,y', s]^T = \boldsymbol{H}'\,[u', v', 1]^T \qquad (3.53)$$

or, in the form of equations with individual coefficients of the $\boldsymbol{H}'$ homography matrix:

$$s\,x' = h'_{11}\,u' + h'_{12}\,v' + h'_{13} \qquad (3.54)$$

$$s\,y' = h'_{21}\,u' + h'_{22}\,v' + h'_{23} \qquad (3.55)$$

$$s = h'_{31}\,u' + h'_{32}\,v' + h'_{33} \qquad (3.56)$$

where
s is a proportionality coefficient;
h'_{ij} is the element from the i-th row and j-th column of the $\boldsymbol{H}'$ matrix.

Excluding the s, the equations of (3.54) – (3.56) can be transformed into

$$-h'_{11}u' - h'_{12}v' - h'_{13} + h'_{31}u'x' + h'_{32}v'x' + h'_{33}x' = 0 \qquad (3.57)$$

$$-h'_{21}u' - h'_{22}v' - h'_{23} + h'_{31}u'y' + h'_{32}v'y' + h'_{33}y' = 0 \qquad (3.58)$$

which produces the following matrix equation for the selected set of points (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) in the Hartley-normalized coordinate system [20, p.3]:

$$\begin{bmatrix} -u'_1 & -v'_1 & -1 & 0 & 0 & 0 & u'_1x'_1 & v'_1x'_1 & x'_1 \\ 0 & 0 & 0 & -u'_1 & -v'_1 & -1 & u'_1y'_1 & v'_1y'_1 & y'_1 \\ -u'_2 & -v'_2 & -1 & 0 & 0 & 0 & u'_2x'_2 & v'_2x'_2 & x'_2 \\ 0 & 0 & 0 & -u'_2 & -v'_2 & -1 & u'_2y'_2 & v'_2y'_2 & y'_2 \\ -u'_3 & -v'_3 & -1 & 0 & 0 & 0 & u'_3x'_3 & v'_3x'_3 & x'_3 \\ 0 & 0 & 0 & -u'_3 & -v'_3 & -1 & u'_3y'_3 & v'_3y'_3 & y'_3 \\ -u'_4 & -v'_4 & -1 & 0 & 0 & 0 & u'_4x'_4 & v'_4x'_4 & x'_4 \\ 0 & 0 & 0 & -u'_4 & -v'_4 & -1 & u'_4y'_4 & v'_4y'_4 & y'_4 \end{bmatrix} \cdot \begin{bmatrix} h'_{11} \\ h'_{12} \\ h'_{13} \\ h'_{21} \\ h'_{22} \\ h'_{23} \\ h'_{31} \\ h'_{32} \\ h'_{33} \end{bmatrix} = 0 \qquad (3.59)$$

By solving the (3.59), the coefficients h'_{ij} for the Hartley-normalized homography matrix $\boldsymbol{H}'$ are found.

Finally, the transition from the Hartley-normalized coordinate systems (u', v') and (x', y') to the real image coordinates (u, v) and road plane coordinates (x, y) is done. From (3.48), (3.50) and (3.53) we get:

$$\boldsymbol{T}_{xy}\,[s\,x,\, s\,y,\, s]^T = \boldsymbol{H}'\,\boldsymbol{T}_{uv}\,[u,\, v,\, 1]^T \qquad (3.60)$$

Therefore

$$[s\,x,\, s\,y,\, s]^T = \boldsymbol{T}_{xy}^{-1}\,\boldsymbol{H}'\,\boldsymbol{T}_{uv}\,[u,\, v,\, 1]^T = \boldsymbol{H}\,[u,\, v,\, 1]^T \qquad (3.61)$$

where $\boldsymbol{H} = \boldsymbol{T}_{xy}^{-1}\,\boldsymbol{H}'\,\boldsymbol{T}_{uv}$ is the homography matrix for direct conversion from the image to the road plane coordinates [20, p.5].

4. Thesis work description

4.1. General notes

Approaching the task of road traffic analysis differs depending on whether the video was taken with a static or moving camera. The case of the static camera is simpler for the following reasons:
1) In such a case, a static background can be extracted and used for distinguishing between static objects, which are ignored, and moving objects, which are the only ones that are proceeded.
2) A static camera has a permanent location within the scene; therefore, the problem of camera self-calibration needs to be solved only once to find the relationship between any point on the road surface and the corresponding point in any frame of the video sequence.

Proceeding a video taken with a moving camera poses some additional challenges for the image analysis algorithms:
1) Static and moving objects cannot be distinguished in advance; therefore, any object that may be relevant for the road traffic analysis task needs to be taken into consideration.
2) Camera calibration needs to be done repeatedly, in order to cope with the shift of the camera position within the scene.
3) The self motion of the camera is required to be tracked as well, because the correspondence between the different world coordinate systems drawn for different frames of the video sequence needs to be established.

Here, the block diagrams with necessary explanations are provided both for the static and moving camera case. Then, detailed descriptions of the whole image processing algorithm are given: in Chapter 4.2 for the static camera scenarios, and in Chapter 4.3 for the moving camera.

4.1.1. The static camera case

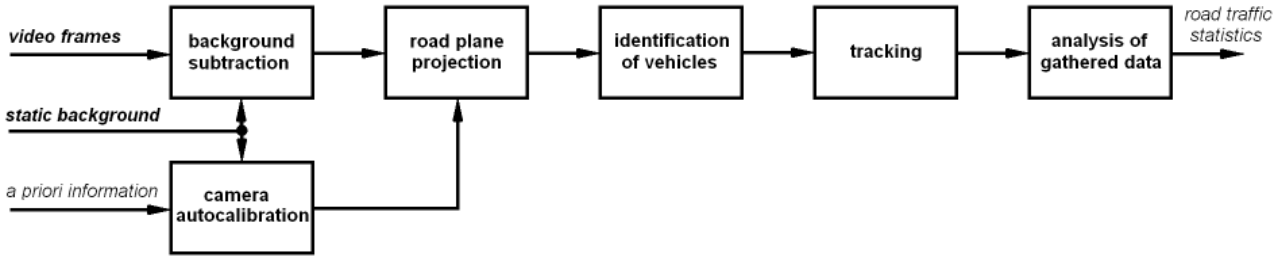


Fig. 4.1. The stages of road traffic analysis for the static camera case.

For the static camera case, the following stages of image processing are required (see Fig. 4.1):
– establishing the correspondence between the image coordinates and the world coordinate system (this needs to be done only once in such a case);
– distinguishing between moving and static objects;
– tracking of the objects movement;
– searching whether any new objects have appeared;
– analysis of the gathered data and extracting statistical information on road traffic from it.

4.1.2. The moving (dynamic) camera case

The case of a dynamic camera is more complicated, because a correspondence between three coordinate systems needs to be established:
– the image coordinate system;
– the relative world coordinate system for each frame, which puts the camera into the zero point of the coordinates;
– the absolute world coordinate system, which keeps the same coordinates for each static detail of the scene regardless of any movement of the camera.

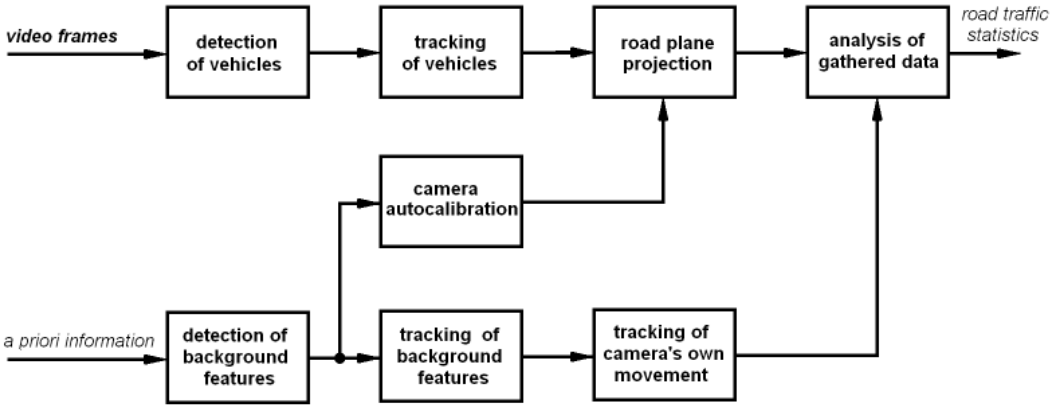


Fig. 4.2. The stages of road traffic analysis for the moving camera case.

This requires for the moving camera case to include some additional stages of image processing compared to the scenarios with a static camera (see Fig. 4.2):
– searching for new objects in a video frame (in the image coordinates),
– tracking of the known objects (in the image coordinates),
– camera calibration (establishing the relationship between the image and relative world coordinates for each frame);
– projection from the image plane to the road plane (shifting to the relative world coordinates);
– searching for background features, such as road marking elements (in the relative world coordinates);
– tracking the background features (which helps to draw the correspondence between the relative world coordinates of consecutive frames);
– tracking the camera's own movement (in order to convert the location of the objects from the relative to the absolute world coordinates);
– analysis of the gathered data (converted to the absolute world coordinates) and extracting statistical information on road traffic from it.

4.2. Road traffic analysis from video taken with a static camera

In this chapter, a description of all stages of image processing and analysis for the static camera case is provided in the way as they were implemented in the experiments made for this thesis work. Section 4.2.1 describes the procedure of distinguishing the moving objects from the static ones. In section 4.2.2, the way of conversion between the road plane and the image coordinates used in the experiments is explained. Section 4.2.3 tells about the object tracking mechanism used for the static camera scenarios. In Section 4.2.4, identification of the relevant objects (vehicles) among all

moving objects is discussed. Finally, Section 4.2.5 explains how the statistical parameters of road traffic are calculated using the data gathered at all the previous stages.

4.2.1. Detection of moving object by background subtraction

Detection of vehicles in a video sequence recorded by a static camera requires to distinguish truly moving foreground objects from apparent motion of background details due to camera pose instability, wind, etc. This can be achieved with various methods of background subtraction [2, p.3], [3, p.4–5], [5, p.1–2]. Their principle is based on comparing each frame of the video to a sample of the background of the scene (an image containing only the static details) and making a decision on each pixel of this frame, whether it belongs to the background or to a moving (foreground) object.

In this work, statistical filtering is used for distinguishing between foreground and background objects. A frame with no vehicles present in the area being analyzed is used as a sample of the background. For each pixel P (u_p, v_p) in this background image, a window of observation is set:

$$R = \begin{cases} u \in [\, u_p - \Delta u_{MAX} \,,\, u_p + \Delta u_{MAX}] \\ v \in [\, v_p - \Delta v_{MAX},\; v_p + \Delta v_{MAX}] \end{cases} \qquad (4.1)$$

The size of the window (Δu_{MAX}, Δv_{MAX}) is equal to the supposed maximum displacement of image details due to camera instability.

For all pixels inside *R,* a local histogram is built. Gray level values are linearly quantized with the number of quantization levels equal to the number of bins in the histogram (N_B) and assigned to the corresponding bin:

$$b = g_Q = \left\lceil \frac{g + \Delta g}{g_{MAX} + \Delta g} \cdot N_B \right\rceil \qquad (4.2)$$

where g is the gray level, g_{MAX} is the maximum gray level, Δg is the minimum resolution between gray levels, g_Q is the quantized gray level value, and b is the histogram bin to which it is counted.

This histogram reflects the assumed probabilities of gray levels in the position of $P(u_p, v_p)$ if the camera instability lies within the assumed limits ($\pm \Delta u_{MAX}$, $\pm \Delta v_{MAX}$) while no moving objects are passing through this point. The frequencies of gray levels in the window of observation are weighted with a Gaussian kernel:

$$\boldsymbol{H}_{[u,v]}(b) = \frac{\sum_{\Delta u} \sum_{\Delta v} \boldsymbol{K}(\Delta u, \Delta v) \Big|_{g_Q(\Delta u, \Delta v) = b}}{\sum_{\Delta u} \sum_{\Delta v} \boldsymbol{K}(\Delta u, \Delta v)} \qquad (4.3)$$

where $\boldsymbol{H}_{[u,v]}(b)$ is the estimated probability for the gray level at the (u,v) image coordinates to fall into the histogram bin b, and $\boldsymbol{K}(\Delta u, \Delta v)$ is the Gaussian kernel for the observation window:

$$\boldsymbol{K}(\Delta u, \Delta v) = k \exp\left(-\frac{\Delta u^2}{2\sigma_u^2} - \frac{\Delta v^2}{2\sigma_v^2} \right) \qquad (4.4)$$

where k is selected so that

$$\sum_{\Delta u=-\Delta u_{MAX}}^{+\Delta u_{MAX}} \sum_{\Delta v=-\Delta v_{MAX}}^{+\Delta v_{MAX}} \boldsymbol{K}(\Delta u, \Delta v) = 1 \qquad (4.5)$$

and σ_u and σ_v are selected so that, with the practical accuracy of computation,

$$\boldsymbol{K}(\pm\Delta u_{MAX}, \pm\Delta v_{MAX}) \approx 0 \qquad (4.6)$$

For each frame of the video, the gray level of every pixel is compared to the histogram for this pixel; if the estimated probability of such gray level value at this position in the image lies below the threshold, the pixel is classified as a part of a foreground object:

$$\boldsymbol{J}(u,v) = \begin{cases} \boldsymbol{I}(u,v) & \text{if } \boldsymbol{H}_{[u,v]}(g_Q(\boldsymbol{I}(u,v))) < \Theta \\ 0 & \text{otherwise} \end{cases} \qquad (4.7)$$

where $\boldsymbol{I}(u,v)$ is the gray level value of the pixel of the original frame at the (u,v) image coordinates, $g_Q(\boldsymbol{I}(u,v))$ is the quantized value of $\boldsymbol{I}(u,v)$, Θ is the threshold (selected to be between 0.001 and 0.05, depending on the video image quality and visibility conditions), and $\boldsymbol{J}(u,v)$ is the gray level value of the output statistically filtered image at the (u,v) image coordinates.

In order to improve the quality of the extracted foreground image, postprocessing needs to be done. It includes the following operations:
– removing overly small objects, which probably resulted from local gray level fluctuations;
– aggregating nearly located fragments into larger objects;
– for daytime videos, a shadow removal procedure is required in some cases;
– for nighttime videos, it is necessary to reduce the light reflections off the road surface.

4.2.2. Conversion from image to road plane coordinates

The camera calibration is made using the method described in Chapter 3.4. Given the image coordinates of a set of straight lines that are parallel in the real world, and the image coordinates of two pairs of points with known real-world distances between them, it produces the matrices of extrinsic ($\boldsymbol{K}$) and intrinsic ($\boldsymbol{R}$) camera parameters:

$$s\,[u, v, 1]^T = \boldsymbol{K}\,\boldsymbol{R}\,[x, y, z, 1]^T =$$

$$= \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ -\sin\theta\sin\varphi & -\cos\theta\sin\varphi & -\cos\varphi & h\cos\varphi \\ \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi & h\sin\varphi \end{bmatrix} \cdot [x, y, z, 1]^T \qquad (4.8)$$

which allow for transition from the 3D world coordinates (x, y, z) to the 2D image coordinates (u, v), but not vice versa. However, the task of vehicle traffic analysis can be reduced to the analysis of movement of the object projection on a 2D plane approximating the road surface. This projection can be generated for every video frame based on the results of the camera calibration, using inverse

mapping and setting the value of the vertical coordinate (z) to be constant (usually $z = 0$):

$$[s\,u,\ s\,v,\ s]^{\mathrm{T}} = \boldsymbol{K}\,\boldsymbol{R}\,[x, y, 0, 1]^{\mathrm{T}} = \boldsymbol{P}\,[x, y, 0, 1]^{\mathrm{T}} \qquad (4.9)$$

where $\boldsymbol{P} = \boldsymbol{K}\,\boldsymbol{R}$ is a full camera matrix (4.8).

Since the matrices $\boldsymbol{K}$ and $\boldsymbol{R}$ have been found at the camera calibration stage, for each point (x, y) of the road plane the corresponding image coordinates (u, v) are calculated by applying (4.9) and dividing the resulting vector $[s\,u, s\,v, s]^{\mathrm{T}}$ by its third element s:

$$s\,u = x\,f\cos\theta - y\,f\sin\theta \qquad (4.10)$$

$$s\,v = -x\,f\sin\theta\sin\varphi - y\,f\cos\theta\sin\varphi + f\,h\cos\varphi \qquad (4.11)$$

$$s = \sin\theta\cos\varphi + \cos\theta\cos\varphi + h\sin\varphi \qquad (4.12)$$

The resulting road plane projection image is generated using inverse mapping.

A road plane projection of the static background image is also generated the same way. In addition, it is used for predicting the vehicle movement direction. For this purpose, the trajectory of the road is approximated with a 4th power polynomial, as proposed in [1, p.14]. This is done by selecting a group of points $[u_i, v_i]$ lying on the same road marking line, which follows the road trajectory, and projecting them onto the road plane coordinate system using (4.9) and inverse mapping. The resulting set of points $[x_i, y_i]$ on the road plane is used to produce a polynomial approximation of the road trajectory in the (X, Y) coordinate system:

$$x = P_4(y) = k_4\,y^4 + k_3\,y^3 + k_2\,y^2 + k_1\,y + k_0 \qquad (4.13)$$

By this approximation, the angle of traffic direction β is estimated for each point $[x, y]$ on the road surface:

$$\beta(x,y) = \lim_{\Delta y \to 0} \operatorname{atan} \frac{\Delta y}{P_4(y+\Delta y) - P_4(y)} \qquad (4.14)$$

which in practice is approximately calculated the following way:

$$\beta(x,y) \approx \operatorname{atan} \frac{1}{P_4(y+1) - P_4(y)} \qquad (4.15)$$

– for the lanes where traffic is directed off the camera, and

$$\beta(x,y) \approx \operatorname{atan} \frac{1}{P_4(y+1) - P_4(y)} + \pi \qquad (4.15)$$

– for the lanes where traffic is directed towards the camera.

4.2.3. Template-based tracking

The generated road plane projection of both the full original video frame and the output of its

statistical filtering are the input of the tracking algorithm. All comparisons and calculations are made in the road plane coordinate system, and the result is the coordinates of all tracked objects in the road plane projection.

Known objects are proceeded in the order of their distance from the camera (nearest objects first). This helps to manage the cases of occlusion correctly, as the occluder, which is always closer to the camera than the occluded object, is processed first. Each object is searched within the specified search area:

$$S = \begin{cases} x \in [\, x_P + \Delta x_{MIN},\ x_P + \Delta x_{MAX} \,] \\ y \in [\, y_P + \Delta y_{MIN} ,\ y_P + \Delta y_{MAX} \,] \end{cases} \qquad (4.17)$$

checking all possible coordinate shifts from Δx_{MIN} to Δx_{MAX} and Δy_{MIN} to Δy_{MAX} around the predicted location (x_P, y_P) of the object for the current frame. For every possible coordinate shift $(\Delta x, \Delta y)$ the image of the object is supposed to be such part of the full frame in the road plane projection that is covered by the object template shifted by $(\Delta x, \Delta y)$ from the predicted coordinates of the object:

$$\boldsymbol{I}_{OS(\Delta x,\, \Delta y)}(x, y) = \boldsymbol{I}(x, y) \text{ sign } \boldsymbol{T}(x+\Delta x, y+\Delta y) \qquad (4.18)$$

where $\boldsymbol{I}_{OS(\Delta x,\, \Delta y)}$ is the supposed image of the object for the coordinate shift $(\Delta x, \Delta y)$, $\boldsymbol{I}(x, y)$ is the full image in the road plane projection, and $\boldsymbol{T}$ is the object template. The possibility for the object coordinates to be located at the $(\Delta x, \Delta y)$ is assessed with the following cost function that needs to be minimized:

$$C(\Delta x, \Delta y) = D_1(\Delta x, \Delta y) - a\, D_2(\Delta x, \Delta y) - b\, D_3(\Delta x, \Delta y) \qquad (4.19)$$

where:

the arguments of the cost function, $(\Delta x, \Delta y)$ are the coordinate shift within the search area S;

D_1 is the sum of absolute differences between the object template and the assumed image of the object in its new position:

$$D_1 = \frac{\sum_x \sum_y |\boldsymbol{T}(x+\Delta x, y+\Delta y) - \boldsymbol{I}(x,y)| \,\Big|_{\boldsymbol{T}(x+\Delta x,\, y+\Delta y) \neq 0}}{A_T} \qquad (4.20)$$

($\boldsymbol{I}$ is the road plane projection of the current frame, $\boldsymbol{T}$ is the template of the object, A_T is the total area of the template).

D_2 is the share of the area of the segments in the foreground image that intersect with the object template in its assumed position:

$$D_2 = \frac{\sum_x \sum_y \text{sign}(\boldsymbol{J}(x, y)) \,\Big|_{\boldsymbol{T}(x+\Delta x,\, y+\Delta y) \neq 0}}{A_J + 1} \qquad (4.21)$$

(***J*** is the part of the road plane projection of the foreground objects relevant for the vehicle being searched, and A_J is its area. ***J*** contains only the segments that have an intersection with the vehicle template shifted onto the predicted position of the vehicle. A segment ***Q*** in the road plane projection of the foreground image belongs to ***J*** if:

$$Q \in J \text{ if } \Big(\sum_x \sum_y \boldsymbol{T}_k(x,y)\,\boldsymbol{Q}(x,y) > 0 \Big) \cap \Big(\sum_{i=1}^{k-1} \sum_x \sum_y \boldsymbol{T}_i(x,y)\,\boldsymbol{Q}(x,y) = 0 \Big) \qquad (4.22)$$

where $\boldsymbol{T}_i$ is the template of i-th object, and k is the index of the object currently being searched for).

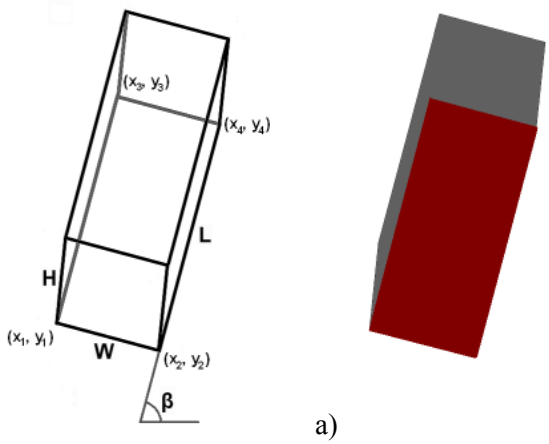
D_3 is the sum of absolute differences in gray levels between the assumed image of the object and the background image:

$$D_3 = \frac{\sum_x \sum_y |\boldsymbol{I}(x,y) - \boldsymbol{B}(x,y)| \Big|_{\boldsymbol{T}(x+\Delta x, y+\Delta y)\neq 0}}{A_T} \qquad (4.23)$$

(***B*** is the road plane projection of the background image; if the instability of the camera pose is significant, the average gray level of the road surface $g_{R.AVG}$ is inserted into the formula instead):

$$D_3 = \frac{\sum_x \sum_y |\boldsymbol{I}(x,y) - g_{R.AVG}| \Big|_{\boldsymbol{T}(x+\Delta x, y+\Delta y)\neq 0}}{A_T} \qquad (4.24)$$

In experiments, the following values for the weighting coefficients appeared to produce the best results: $a = 1$, $b = 3$.



a) b)

Fig. 4.3.
a) Approximation of the vehicle shape with a rectangular cuboid. *W*, *L* and *H* are the width, length and height of the vehicle, respectively, β is the movement direction.
b) Projection of the shape from (a) onto the road plane. The area really occupied by the vehicle is shaded in red.

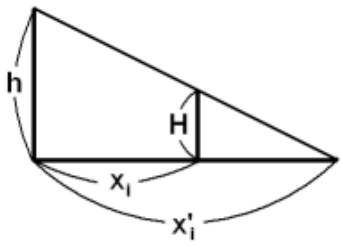


Fig. 4.4.
Proportions for calculating the x'_i coordinates

The search procedure aims to find the minimum of the cost function value for all coordinate shifts (Δx, Δy) within the search area S. The found optimal coordinate shift from the predicted position of the object is added to its predicted road plane coordinates:

$$[x_{(N),}\ y_{(N)}] = [x_{P\,(N),}\ y_{P\,(N)}] + \arg\min_{\Delta x,\ \Delta y} C(\Delta x, \Delta y) \qquad (4.25)$$

([$x_{P\,(N),}$ $y_{P\,(N)}$] are the predicted coordinates of the object, [$x_{(N),}$ $y_{(N)}$] are its found coordinates).

In order to update the object template, first the template shape, which changes slowly from frame to frame, is updated. The 3-dimensional shape of the vehicle is approximated with a rectangular cuboid (see Fig. 4.3a).

The coordinates of the lower corners of this cuboid: (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4,) are obtained from the new coordinates of the vehicle knowing its width W, length L, and movement direction β; the latter is known from (4.15), (4.16). For example, if (x_1, y_1) is used as the vehicle coordinates, the coordinates of other corners can be found the following way (see Fig. 4.3):

$$[x_2,\ y_2] = [x_1, y_1] + [W \sin\beta, -W \cos\beta] \qquad (4.26)$$

$$[x_3,\ y_3] = [x_1, y_1] + [L \cos\beta,\ L \sin\beta] \qquad (4.27)$$

$$[x_4,\ y_4] = [x_3, y_3] + [W \sin\beta, -W \cos\beta] \qquad (4.28)$$

For the road plane coordinates of the projection of the upper corners, we get (see Fig. 4.4):

$$\frac{x'_{i,} - x_{i,}}{H} = \frac{x'_{i,}}{h} \qquad (4.29)$$

Therefore, the road plane coordinates of the projection of the upper corners are found as:

$$[x'_{i,}, y'_{i}] = \begin{cases} [x_{i,}, y_i]\dfrac{h}{h-H} & \text{if } H < h \\ [\infty, \infty] & \text{otherwise} \end{cases} \qquad (4.30)$$

where [$x_{i,}$, y_i] are the tracked coordinates of the lower corners of the object, H is the object height, h is the camera height known from the camera calibration stage, and [$x'_{i,}$, y'_{i}] are road plane coordinates of the projection of the upper corners of the rectangular cuboid approximating the object shape. Using (4.26)–(4.30), the shape of the object template is updated (see Fig. 4.3b).

If the correspondence between the template and the image of the object at the road plane projection of the current frame is good ($C < 0.1$ can be a criterion), the part of the road plane projection of the frame with the calculated shape from the found location becomes a new template of the object. If the correspondence is far from being perfect, a new template includes a copy of the old one to the greatest possible extent. This is done by pasting the old template onto the road plane projection of the frame at the found object position and extracting the calculated new template shape from this synthesized image:

$$\boldsymbol{T}(x, y) = \begin{cases} \boldsymbol{E}(x, y)\, \boldsymbol{I}(x, y) & \text{if} \quad C < 0.1 \\ \boldsymbol{E}(x, y)((1 - \boldsymbol{E_0}(x, y))\, \boldsymbol{I}(x, y) + \boldsymbol{T_0}(x, y)) & \text{otherwise} \end{cases} \tag{4.31}$$

where $\boldsymbol{T_0}$ is the old template of the object, $\boldsymbol{E_0}$ is the shape of $\boldsymbol{T_0}$, and $\boldsymbol{E}$ is the shape generated for the new template using (4.26) – (4.30).

As soon as the true coordinates of a vehicle in the road plane coordinate system are found, the estimation of its average speed is updated using its value at the previous frame and the coordinate shift from the previous to the current frame:

$$s_{(N)} = \frac{s_{(N-1)}\,(N - N_0 - 1) + \sqrt{(x_{(N)} - x_{(N-1)})^2 + (y_{(N)} - y_{(N-1)})^2}}{N - N_0} \tag{4.32}$$

where $[x_{(i)}, y_{(i)}]$ are the object coordinates at i-th frame, N is the number of the current frame, N_0 is the number of the frame where the object was initially detected, and $s_{(i)}$ is the estimated absolute value of the average speed of the object by i-th frame. Here, the speed is expressed in coordinate system units per interval between frames (not in kilometers per hour). This estimated speed value is used to predict the object location in the next frame:

$$[x_{PR.(N+1)},\ y_{PR.(N+1)}] = [x_{(N)}, y_{(N)}] + [s_{(N)} \cos \beta,\ s_{(N)} \sin \beta] \tag{4.33}$$

where $[x_{(N)}, y_{(N)}]$ are the current coordinates of the object, $s_{(N)}$ is the estimated absolute value of the object speed, β is the traffic direction in the $[x_{(N)}, y_{(N)}]$ point of the road known from (4.15) and (4.16), and $[x_{PR.(N+1)},\ y_{PR.(N+1)}]$ are the predicted coordinates of the object for the next frame.

4.2.4. Identification of new vehicles by analyzing the shape of their road plane projection

As soon as all correspondences between the known vehicles and the detected foreground objects in the frame being processed are found, new vehicles are searched by proceeding the remaining unidentified foreground objects. First, their geometrical shape is analyzed in order to detect cases of fragmentation of an object into several segments, and to exclude irrelevant objects. This is done by assuming each segment to be an object in itself and drawing the shape of the road plane projection for such tentative object using (4.26) – (4.30) (see Fig. 4.5). At least two such shapes are generated: for the object width, length and height being equal to their minimum and maximum. If any one of the shapes overlaps another segment, that segment is considered to be covered with the segment being analyzed.

The result of testing all segments is accumulated in coverage matrices for the minimum and the maximum object size: the rows correspond to the segments been analyzed, and the columns – to the segments they cover. An example of coverage matrix for the situation shown at Fig. 4.5 will be:

$$\boldsymbol{M} = \begin{Vmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{Vmatrix} \tag{4.34}$$

If two segments reciprocally cover one another, it is treated as only the one located closer to the camera covers the other one:

$$m_{j,i} = 0 \quad \text{if} \quad (m_{i,j} = 1) \cap (j > i) \qquad (4.35)$$

where $m_{i,j}$ is the element of $\boldsymbol{M}$ from i-th row and j-th column.

From the coverage matrix, groups of segments that can be object candidates are found. The criteria are the following:

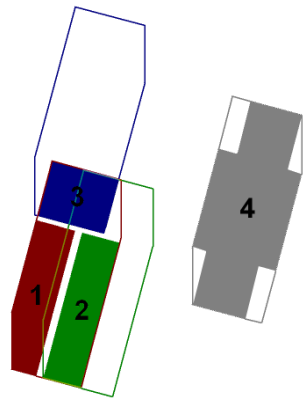


Fig. 4.5. Grouping foreground segments into object candidates.

1) there must be a segment j that is not covered by any other one:

$$\sum_i m_{i,j} = 0 \qquad (4.36)$$

2a) if this segment does not cover any other one, it is considered a group in itself:

$$G_i = \{i\} \quad \text{if} \quad \sum_j m_{i,j} = 0 \qquad (4.37)$$

where G_i is the group of segments based on i-th segment;

2b) if the segment covers some other segments, they all form a group together with this segment:

$$k \in G_i \quad \text{if} \quad (i = k) \cup (m_{i,k} = 1) \qquad (4.38)$$

3) in any case, the total area of the group needs to be greater than the minimum that depends on the particular visibility conditions (day or nighttime, camera height over the road surface, etc.) :

$$\sum_{k \in G} A_k \geq A_{MIN} \qquad (4.39)$$

where A_k is the area of k-th segment, and A_{MIN} is the minimum area for a group of segments to be considered a new object.

After this, the object size is estimated from the group of segments attributed to the newly detected object. First, all the dimensions of the object (width, length and height) are assumed to have their maximum values known from a priori data on objects of interest. A 2D road plane projection for such maximum-size object is made for the found location of the candidate object. Then, each of the dimensions of the object (the width first, then the length, and finally the height) is gradually decreased until the value is reached at which the projection no longer covers the whole group of segments that forms the candidate object. Therefore, the previous value of the dimension being measured is assumed to be the value of this dimension for the newly detected object.

As soon as the object is measured, a shape for the object template at the found location is generated from the estimated object size, using (4.26)–(4.30). Finally, a template for the new object is extracted from the road plane projection of the current frame by using the generated template shape as a mask, as it is done in (4.31) for $C < 0.1$.

4.2.5. Rectification of the results

After obtaining the whole trajectory of an object within the field of view, the data on the object speed can be refined. First, the vehicle trajectory is approximated with a 4th power polynomial, as proposed in [1, p.14]:

$$x = P_4(y) = k_4\, y^4 + k_3\, y^3 + k_2\, y^2 + k_1\, y + k_0 \qquad (4.40)$$

Then the length of this polynomial curve is used for a more accurate estimation of the speed of the vehicle using piecewise linear approximation of the obtained trajectory of the vehicle (this procedure removes most of noise in coordinates):

$$s \approx \frac{\kappa \sum \sqrt{\Delta x^2 + \Delta y^2}}{(N_E - N_0)\,\tau} \approx \frac{\kappa \sum_{y=y_{MIN}+1}^{y_{MAX}} \sqrt{|P_4(y) - P_4(y-1)|^2 + 1}}{(N_E - N_0)\,\tau} \qquad (4.41)$$

where:
y_{MIN} and y_{MAX} are the minimum and maximum values of the y coordinate of the vehicle trajectory, respectively;
$P_4(y)$ is the x coordinate value corresponding to y according to the 4th power polynomial approximation of the trajectory (4.40);
N_0 and N_E are the numbers of the first and the last frame where the vehicle was observed, respectively;
τ is the time interval between two consecutive video frames, in hours;
κ is the scaling coefficient between the real distances and the road plane projection image coordinates, in kilometers per pixel;
s is the final estimation of the average speed of the vehicle, in kilometers per hour.

4.3.Road traffic analysis from video taken with a moving camera

In this chapter, a description of all stages of image processing and analysis for the moving camera case is provided in the way as they were practically implemented in this thesis work. Section 4.3.1 explains the procedure of detecting new vehicles on the road. Section 4.3.2 describes the implementation of the object tracking algorithm. In Section 4.3.3, the problem of conversion between the different world coordinate systems and the image coordinates for different frames of the video is discussed. Section 4.3.4 tells about the possible way of tracking the own movement of the camera. Finally, Section 4.3.5 explains how the statistical parameters of road traffic are calculated using the data gathered at all the previous stages.

4.3.1. Detection of new vehicles using HOG-based SVM

For the purpose of vehicle detection in the moving camera scenario, a whole video frame (or, at least, the parts of it that can realistically contain a vehicle on the road) needs to be scanned. In this thesis work, this was made with a support vector machine classifier using histograms of oriented gradients as object features (see the description in Chapter 3.1 and 3.2). Several variations of the HOG method exist. They differ in the size (fixed or variable) and shape (rectangular or not) of the cells, and in the definition of gradient (which can be signed or unsigned, and calculated using

different filters: Sobel, Prewitt, simple difference, etc.) [11, p.3-4], [13, p.2-4]. The version used in this thesis work implements the HOG with rectangular cells of variable size and signed gradients calculated from a difference image:

$$D_X(x, y) = g\,(x+1, y) - g\,(x, y) \qquad (4.42)$$

$$D_Y(x, y) = g\,(x, y+1) - g\,(x, y) \qquad (4.43)$$

where
$g\,(x, y)$ is the gray level at (x, y) coordinates of the image,
$D_X(x, y)$ is the horizontal difference image,
$D_Y(x, y)$ is the vertical difference image.

The size of the HOG grid used for detecting vehicles was selected to be $N_X = 6$ cells in horizontal and $N_Y = 4$ cells in vertical direction. The block size was 2×2 cells.

For the practical purpose of detecting vehicles in a video sequence, the SVM classifier was trained with a set of positive (images of vehicles) and negative (images of other types of objects that can appear at the scene) samples, and the γ and C parameters were adjusted. For this thesis work, $C = 2^{13}$ and $\gamma = 2^{13}$ produced the best results.

The SVM using HOG as object features, though it demonstrates reasonably good performance, cannot guarantee absolutely reliable detection, especially in the regard of avoiding false positives. Therefore, in order to increase the robustness of vehicle detection, a voting mechanism was implemented. It uses the experimentally observed fact that true positive cases produce multiple overlapping detections, while false positives occur randomly.

For the purpose of detecting new vehicles, every 5th frame of the video is scanned with a bounding box shifting along its horizontal and vertical axes $[u, v]$ with a fixed step $[D_u, D_v]$, trying all possible values of $u = k_u D_u$, $v = k_v D_v$ as the top left corner coordinates of a candidate object, where k_u and k_v are integer numbers. The size of the bounding box $[x, y]$ also varies with a fixed $[D_x, D_y]$, step, so all sizes within the limits of possible size of the object in the image, from (x_{MIN}, y_{MIN}) to (x_{MAX}, y_{MAX}), such as $x = x_{MIN} + k_x D_x$, $y = y_{MIN} + k_y D_y$, are checked under additional constraints on the shape of a possible object:

$$1 \le x/y \le 2 \qquad (4.44)$$

The results of evaluating the hypotheses on the size and location of the object within the frame are stored in an accumulator matrix $\boldsymbol{Q}$ with the size equal to the one of the image. If a rectangular part of the frame having the top left corner coordinates $[u, v]$ and size $[x, y]$ is classified by the SVM as a possible vehicle, the accumulator values for the area occupied by it are incremented:

$$\boldsymbol{Q}(u\,..\,u+x-1,\, v\,..\,v+y-1) = \boldsymbol{Q}(u\,..\,u+x-1,\, v\,..\,v+y-1) + \boldsymbol{I}_{x,y} \qquad (4.45)$$

where $\boldsymbol{I}_{x,y}$ is a matrix of x columns and y rows, filled with all 1s. The found values of the coordinates and size $[u, v, x, y]$ for each object candidate classified by the SVM as a vehicle are appended to the hypotheses matrix:

$$\boldsymbol{R} = \begin{bmatrix} u_1,\ v_1,\ x_1,\ y_1 \\ u_2,\ v_2,\ x_2,\ y_2 \\ \dots \\ u_n,\ v_n,\ x_n,\ y_n \end{bmatrix} \qquad (4.46)$$

where $[u_i,\ v_i]$ are the coordinates of the top left corner, and $[x_i,\ y_i]$ is the object size for i-th hypothesis.

After entire frame has been scanned, it is checked whether the maximum of the accumulator $\boldsymbol{Q}$ exceeds the threshold Θ for the number of overlapping detections sufficient to consider a new vehicle been detected. If $\max\{\boldsymbol{Q}\} > \Theta$, the size of the object is calculated as a median of the sizes for all hypotheses:

$$[x_o, y_o] = [\ \text{median}\{[x_1, x_2, \dots x_n]\}, \text{median}\{[y_1, y_2, \dots y_n]\}] \qquad (4.47)$$

Then the coordinates of the object are found as a point where the bounding box of the (x_o, y_o) size makes the maximum sum of the accumulator:

$$[u_o, v_o] = \arg\max \left\{ \sum_{i=u_o}^{u_o+x_o-1} \sum_{j=v_o}^{v_o+y_o-1} \boldsymbol{Q}(i,j) \right\} \qquad (4.48)$$

According to these coordinates and size, a sample of the detected object is extracted from the video frame. For the tracking purpose, the sample of the object is resized to the size of the kernel $[w_K, h_K]$ used in the tracking algorithm; in this thesis work, $w_K = h_K = 100$ was set.

4.3.2. Tracking vehicles using the kernel-based mean shift algorithm

Vehicles were tracked using the kernel-based mean shift tracking procedure based on [19], as described in Chapter 3.3. Given the 64-bin gray level histogram, which was used as the object template, and the image coordinates of the object in the previous frame, this algorithm iteratively searches for the best possible correspondence of the object sample with a region in the current frame, until the stop condition is fulfilled. In this thesis work, the stop condition was the following:

$$\sqrt{du^2 + dv^2} \le 0.1 \cup n_{it} = 20 \rightarrow \text{Stop} \qquad (4.49)$$

where
$[du, dv]$ is the change of the object position since the previous iteration of the mean shift algorithm, in image coordinates $[u, v]$;
n_{it} is the number of proceeded iterations of the mean shift algorithm.

An additional aspect that needs to be taken into account in the task of object tracking in image coordinates is the change of the visible size of the object. In order to properly manage this, the size of the sample of the object has to be variable. This is implemented by running the mean shift tracking algorithm several times with the size of the object slightly expanded or reduced (compared to the previous frame) and choosing the size that produces the greatest Bhattacharyya coefficient [19, p.5]. In this thesis work, while proceeding each video frame the following five options are tried for the object size and coordinates:

$$\boldsymbol{R} = \begin{bmatrix} u_o, & v_o, & x_o, & y_o \\ u_o+0.05x_o, & v_o+0.025y_o, & 0.9x_o, & 0.95y_o \\ u_o+0.025x_o, & v_o+0.05y_o, & 0.95x_o, & 0.9y_o \\ u_o-0.05x_o, & v_o-0.025y_o, & 1.1x_o, & 1.05y_o \\ u_o-0.025x_o, & v_o-0.05y_o, & 1.05x_o, & 1.1y_o \end{bmatrix} \quad (4.50)$$

where
$\boldsymbol{R}$ is the array of possible options for the initial coordinates and size of the object,
(u_o, v_o) are the object coordinates at the previous video frame,
(x_o, y_o) is the object size at the previous frame.

4.3.3. The problem of conversion from image to world coordinates in the moving camera case

When the image coordinates of an object are found, they need to be converted into the world coordinates. In general, the problem of camera calibration in the moving camera case is solved using the principles applied in the case of static camera. From the image coordinates of a vanishing point formed by a set of parallel lines directed along the road, and the distances between some points on the road surface known from a priori information on the scene (road marking standards etc.), the system of equations (3.40)–(3.43) is built to find the camera parameters:

$$\tan\varphi = -\frac{v_0}{f} \quad (4.51)$$

$$\cos^2\theta = \frac{f^2 + v_0^2}{f^2 + u_0^2 + v_0^2} \quad (4.52)$$

$$\frac{h\,u_A}{(v_A - v_0)\cos\varphi} + W\cos\theta = \frac{h\,u_B}{(v_B - v_0)\cos\varphi} \quad (4.53)$$

$$\frac{h\,(f^2 - v_C\,v_0)}{f\,(v_C - v_0)} + L\cos\theta = \frac{h\,(f^2 - v_D\,v_0)}{f\,(v_D - v_0)} \quad (4.54)$$

where
f is the focal length of the camera;
h is the camera height over the road;
φ is the angle between the main optical axis of the camera and its projection onto the road plane;
θ is the angle between the projection of the main optical axis onto the road surface and the traffic direction;
(u_0, v_0) are the image coordinates of the vanishing point formed by the continuation of the road boundaries into infinity;
W is the known distance between the points A and B on the road plane with the image coordinates (u_A, v_A) and (u_B, v_B) respectively, located on a straight line perpendicular to the traffic direction;
L is the known distance between the points C and D on the road plane with the image coordinates (u_C, v_C) and (u_D, v_D) respectively, located on a straight line parallel to the traffic direction;
ψ, the rotation angle of the camera around its main optical axis, is assumed to be zero.

For the case of a camera mounted on a moving vehicle, only f remains constant during the camera movement, while the other parameters are changing:
h and φ fluctuate around their mean values, and ψ fluctuates around zero, because the vehicle is

shaking during its movement;
θ slowly deviates from its original value and can change significantly with time;
the coordinates of the points on the road surface are changing, previously known points go outside of the view, and new points need to be detected in order to keep on updating the camera parameters, as well as for tracking the own movement of the camera.

The deviation of camera parameters can be detected by producing a road plane projection image using the camera matrix for the previous video frame and analyzing the resulting defects:
– If the parallel road boundaries and marking lines tend to intersect in the road plane projection image, this means that the vertical coordinate of the vanishing point, v_0, needs to be corrected until the lines at the road plane projection become parallel; it affects φ and h.
– If the parallel lines remain parallel, but appear to turn to the right or left in the road plane projection image, this shows that the horizontal coordinate of the vanishing point, u_0, needs to be corrected; it affects θ.
– If the distances between known feature points on the road surface become equally longer or shorter in the road plane projection image, this means that h has changed.
– If the distances between known feature points on the road surface change unequally in different parts of the road plane projection image, while parallel lines remain parallel, this indicates a non-zero ψ.
Examples of defects caused by inaccuracies in estimation of the camera parameters are shown at Fig. 4.6.

Since in the moving camera case the roll angle ψ cannot be forced to zero or compensated for, it needs to be included into the camera matrix:

$$\boldsymbol{P} = \boldsymbol{P}_{(\psi=0)}\,\boldsymbol{R}_{\psi} = \begin{bmatrix} f\cos\theta & -f\sin\theta & 0 & 0 \\ -f\sin\theta\sin\varphi & -f\cos\theta\sin\varphi & -f\cos\varphi & fh\cos\varphi \\ \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi & h\sin\varphi \end{bmatrix} \cdot \begin{bmatrix} \cos\psi & 0 & -\sin\psi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\psi & 0 & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} f\cos\theta\cos\psi & -f\sin\theta & -f\cos\theta\sin\psi & 0 \\ -f\sin\theta\sin\varphi\cos\psi - f\cos\varphi\sin\psi & -f\cos\theta\sin\varphi & f\sin\theta\sin\varphi\sin\psi - f\cos\varphi\cos\psi & fh\cos\varphi \\ \sin\theta\cos\varphi\cos\psi - \sin\varphi\sin\psi & \cos\theta\cos\varphi & -\sin\theta\cos\varphi\sin\psi - \sin\varphi\cos\psi & h\sin\varphi \end{bmatrix} \quad (4.55)$$

where
$\boldsymbol{R}_{\psi}$ is the rotation matrix for the roll angle ψ;
$\boldsymbol{P}_{(\psi=0)} = \boldsymbol{K}\,\boldsymbol{R}$ is the camera matrix for $\psi = 0$ derived as in (3.33) – (3.44);
$\boldsymbol{P}$ is the full camera matrix including all the rotation angles (φ, θ, and ψ).

The updated camera matrix for each video frame is stored to be used later for extracting statistical information on the traffic.

Similarly to the static camera case, in the moving camera scenario a road plane projection image is generated from the video frame using the inverse mapping method based on the estimated camera matrix and setting the vertical coordinate z to zero:

$$[s\,u,\ s\,v,\ s]^{\mathrm{T}} = \boldsymbol{P}\,[x, y, 0, 1]^{\mathrm{T}} \qquad (4.56)$$

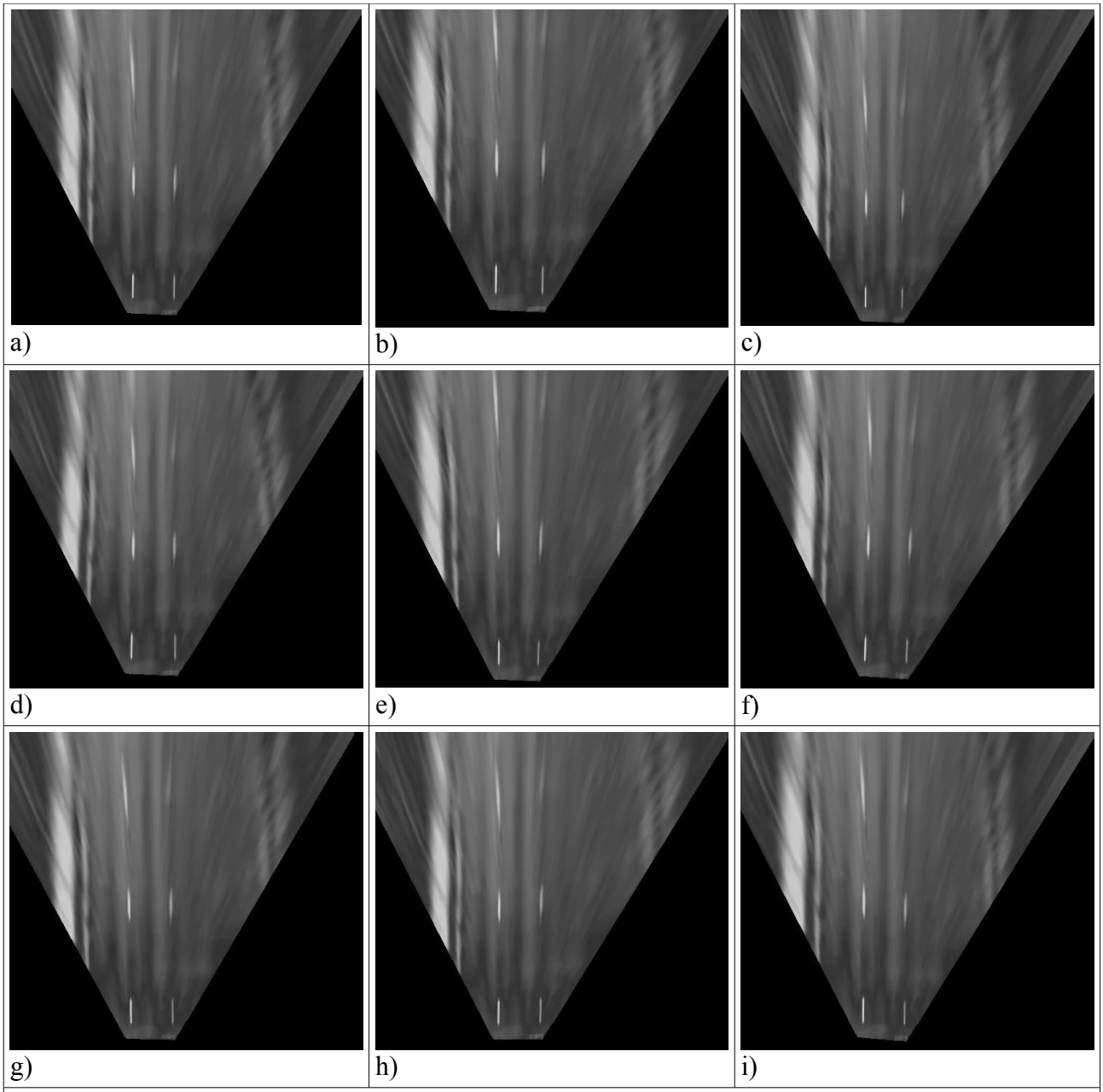


Fig. 4.6. The types of defects in the road plane projection image caused by inaccuracies in the estimated camera parameters:
a) a road plane projection made with the correct camera matrix;
b) h assumed to be higher than its true value;
c) h assumed to be lower than its true value;
d) the vanishing point mistakenly located above its true position;
e) the vanishing point mistakenly located below its true position;
f) the vanishing point mistakenly located to the left from its true position;
g) the vanishing point mistakenly located to the right from its true position;
h) $\psi < 0$ while assumed to be zero;
i) $\psi > 0$ while assumed to be zero.

So for each (x, y, 0) point on the road plane, the corresponding image location (u, v) is found.

The difference from the static camera case is that the used road plane coordinate system is relative and specific for the particular video frame. In order to set the correspondence between this relative coordinate system and the absolute world coordinates, tracking of static details of the scene and estimation of the camera's own movement are needed, which is described below.

In order to amplify the features to search for, a difference image of the video frame being proceeded and the average gray level of the road surface is generated. First, the gray level m of the road surface is estimated as a median of the part of the frame showing the road surface:

$$m = \text{median}\{\boldsymbol{I}(S)\} \qquad (4.57)$$

where
$\boldsymbol{I}$ is the road plane projection of the frame being proceeded,
S is the region of $\boldsymbol{I}$ that contains the road surface.

Then a difference image $\boldsymbol{J}$ is calculated:

$$\boldsymbol{J} = |\, \boldsymbol{I} - m \,| \qquad (4.58)$$

In this difference image, the details of road marking (in most cases, the ends of dashes in dashed lines) are tracked using simple template tracking mechanism.

$$[x_p[n],\ y_p[n]] = [x_p[n-1],\ y_p[n-1]] + \arg\min_{\Delta x,\ \Delta y} \sum_x \sum_y |\boldsymbol{T}(x+\Delta x, y+\Delta y) - \boldsymbol{J}(x,y)| \qquad (4.59)$$

where
$\boldsymbol{T}$ is the template of the feature being tracked,
$[x_p[i],\ y_p[i]]$ are the relative world coordinates of the feature p at i-th frame.

Using the difference image generated in (4.58), new features of the background, of which the ends of dashes in dashed road marking lines are the most important, are being detected using sample images of such road marking elements and checking their correspondence with the regions of a road plane projection of the video frame being proceeded.

$$\frac{\sum_{x,y} |\boldsymbol{J}(x \ldots x+x_{maxT}-1,\ y \ldots y+y_{maxT}-1) - \boldsymbol{T}|}{x_{maxT}\ y_{maxT}} < \Theta \qquad (4.60)$$

where
$\boldsymbol{J}$ is the difference image between the road plane projection of the frame being proceeded and the average gray level of the road surface, calculated in (4.58);
$\boldsymbol{T}$ is the template of the road marking element being searched for;
(x_{maxT}, y_{maxT}) is the size of $\boldsymbol{T}$;
Θ is the threshold for considering the $\boldsymbol{J}(x \ldots x+x_{maxT}-1,\ y \ldots y+y_{maxT}-1)$ to be the road marking element $\boldsymbol{T}$.

When a region of the size (x_{maxT}, y_{maxT}) satisfying (4.60) is found in $\boldsymbol{J}$, it is added to the list of

background features used for establishing correspondence between the relative world coordinate systems of different video frames and tracking the camera's own movement.

4.3.4. Tracking the own movement of the camera

Given a set of static points located on the road plane with their image coordinates $[u_p[n], v_p[n]]$ in the current frame n, their coordinates $[x_p[n], y_p[n]]$ in the relative world coordinate system specific for this frame can be found using the procedures described in Section 4.3.3; the same was done for the previous frame $(n-1)$. The own movement of the camera causes these static details of the scene to change their position in the road plane projection image. For the points located on the road plane, this shift of their coordinates is equal by the absolute value to the displacement of the camera, but has the opposite direction:

$$[x_p[n], y_p[n]]^T - [x_p[n-1], y_p[n-1]]^T = -\left([x_c[n], y_c[n]]^T - [x_c[n-1], y_c[n-1]]^T\right) \qquad (4.61)$$

where $(x_p[i], y_p[i])$ and $(x_c[i], y_c[i])$ are the relative world coordinates in the road plane projection image of the i-th frame for the feature point p and the camera, respectively.

In order to minimize the errors occurring due to inaccuracies in the coordinates of the feature points, the update procedure for the camera coordinates uses the median of the coordinate shifts of all the feature points:

$$x_c[n] = x_c[n-1] - \text{median}\{x_1[n] - x_1[n-1], x_2[n] - x_2[n-1], \ \ldots x_N[n] - x_N[n-1]\} \qquad (4.62)$$

$$y_c[n] = y_c[n-1] - \text{median}\{y_1[n] - y_1[n-1], y_2[n] - y_2[n-1], \ \ldots y_N[n] - y_N[n-1]\} \qquad (4.63)$$

where
$(x_j[i], y_j[i])$ are the relative world coordinates in the road plane projection image of the i-th frame for the j-th feature point,
$N > 0$ is the total number of feature points on the road surface currently being tracked.

Since in the moving camera case the vehicles have been tracked in the image coordinates, a mechanism for conversion from the image plane to the road plane coordinate system for the stored coordinates of vehicles is needed. For this purpose, planar homography is used, as described in Chapter 3.5. It produces a homography matrix $\boldsymbol{H}$ setting the correspondence between the two planar coordinate systems (u, v) and (x, y) in the following form (3.61):

$$[s\,x, s\,y, s]^T = \boldsymbol{H}\,[u, v, 1]^T \qquad (4.64)$$

Using (4.64), the tracked image coordinates for each object are converted to the relative world coordinates in the road plane projection for the particular frame. Then, using the fact that in the relative world coordinates the camera center is located at the zero point, the conversion to the absolute world coordinates is made by adding the own coordinates of the camera obtained from (4.62) – (4.63):

$$[x_A[n], y_A[n]]^T = [x[n], y[n]]^T + [x_c[n], y_c[n]]^T \qquad (4.65)$$

where
$[x[n], y[n]]$ are the relative world coordinates of the object at the n-th video frame obtained from

(4.64);
$[x_c[n], y_c[n]]$ are the absolute world coordinates of the camera at the n-th video frame obtained from (4.62) – (4.63);
$[x_A[n], y_A[n]]$ are the absolute world coordinates of the object at the n-th video frame.

4.3.5. Rectification of the results

As soon as the absolute world coordinates of a vehicle are found for each frame where it was observed, then its trajectory and speed can be estimated using the methods applied for the static camera case. The trajectory is approximated with a 4th power polynomial, as proposed in [1, p.14]:

$$x = P_4(y) = k_4 y^4 + k_3 y^3 + k_2 y^2 + k_1 y + k_0 \qquad (4.66)$$

Then the length of this polynomial curve is used for estimating the speed:

$$s \approx \frac{\kappa \sum \sqrt{\Delta x^2 + \Delta y^2}}{(N_E - N_0)\,\tau} \approx \frac{\kappa \sum_{y=y_{MIN}+1}^{y_{MAX}} \sqrt{|P_4(y) - P_4(y-1)|^2 + 1}}{(N_E - N_0)\,\tau} \qquad (4.67)$$

where:
y_{MIN} and y_{MAX} are the minimum and maximum values of the y coordinate of the observed vehicle trajectory, respectively;
$P_4(y)$ is the x coordinate value corresponding to y according to the 4th power polynomial approximation of the trajectory (4.66);
N_0 and N_E are the numbers of the first and the last frame where the vehicle was observed, respectively;
τ is the time interval between two consecutive video frames, in hours;
κ is the scaling coefficient between the real distances and the road plane projection image coordinates, in kilometers per pixel;
S is the final estimation of the average speed of the vehicle, in kilometers per hour.

5. Experiments and results

In this part, the different scenarios used for testing the image processing and analysis algorithms described in Part 4 are presented. Chapters 5.1 to 5.3 describe the three scenarios with a static camera. In Chapter 5.4, the performance of the methods used in the static camera scenarios is assessed. Chapter 5.5 shows the moving camera scenario, and in Chapter 5.6, some observations on the moving camera case are made.

5.1. Scenario 1: Video from a stable static camera, taken at daytime

The video sequence from [8] was used for this scenario. During the estimation of camera parameters, a non-horizontality of the camera with $\psi \approx -3.5°$ was found; it was compensated by rotating all frames by $-\psi$, as recommended in [6, p.59]. The camera calibration was done in assumption that the true distance between the points A and B (see Fig. 5.1) is equal to 6 m, and the distance between B and C is 10 m.

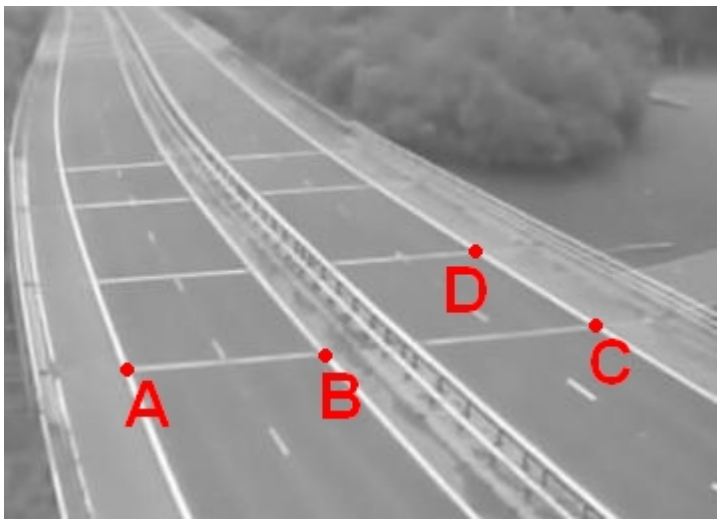
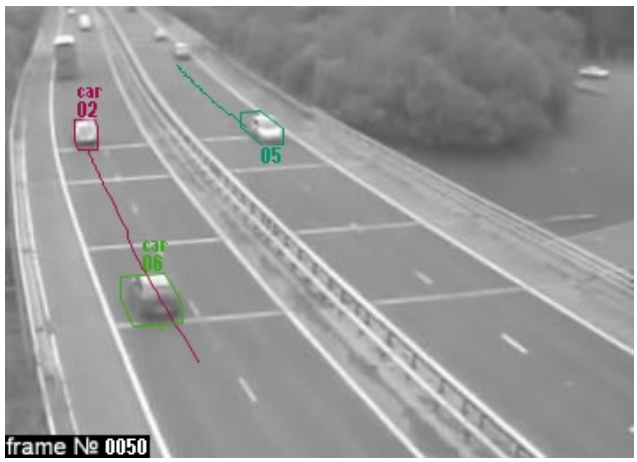


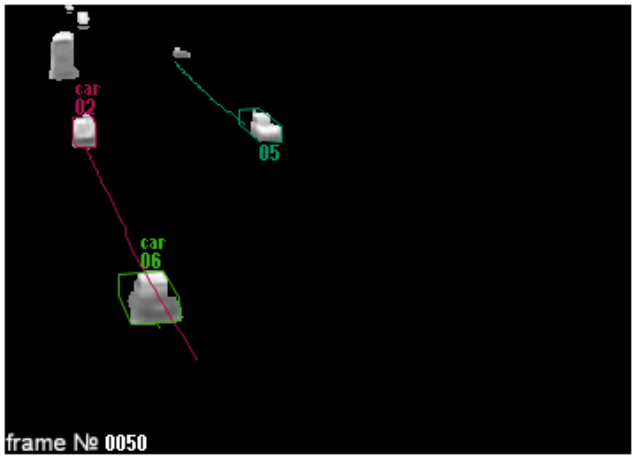
Fig. 5.1. The background image used for camera calibration in scenario 1; the distances between A and B and between C and D were assumed to be known.

The resulting camera parameters appeared to be the following:
camera focal length f = 129.6 m;
camera height over the road surface: h = 13.4 m;
tilt angle φ = 12.2°,
pan angle θ = 13.7°.

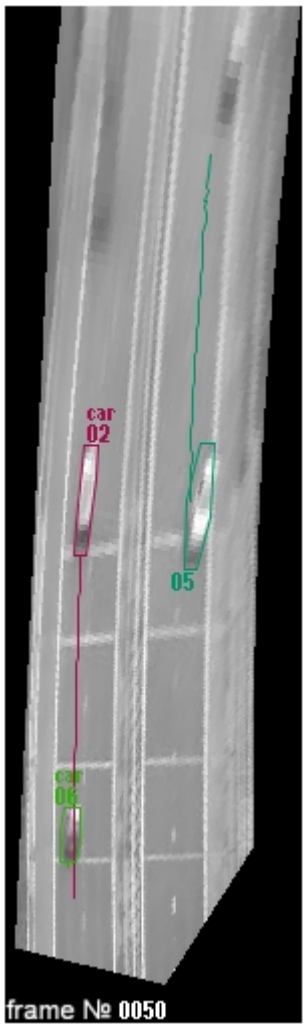
Using these parameters, a sequence of 1551 frames was proceeded. 56 vehicles passed through the tracking area in this video sequence, all of them were successfully detected, no misdetection of irrelevant objects was observed.



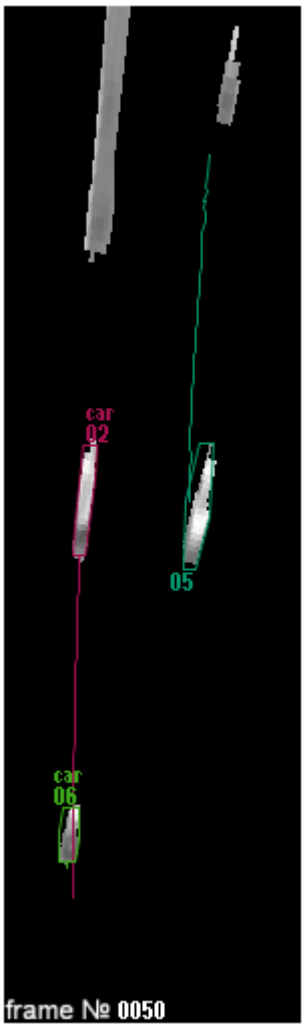
a)



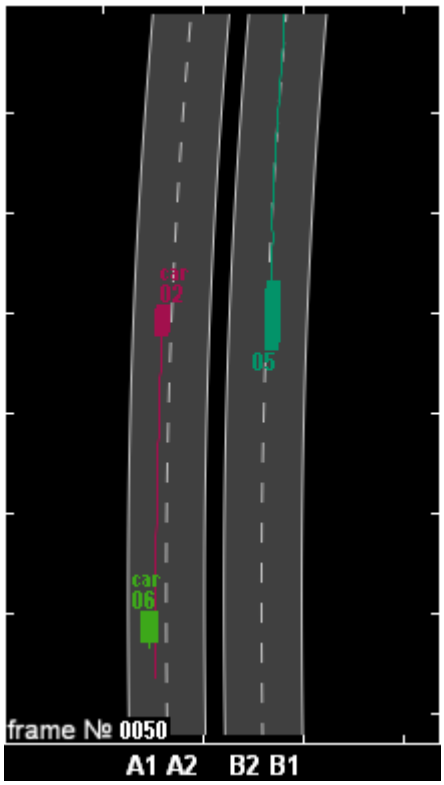
b)



c)



d)



e)

Fig. 5.2. Results for scenario 1.
a) Video frame No.50 from [8] with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.

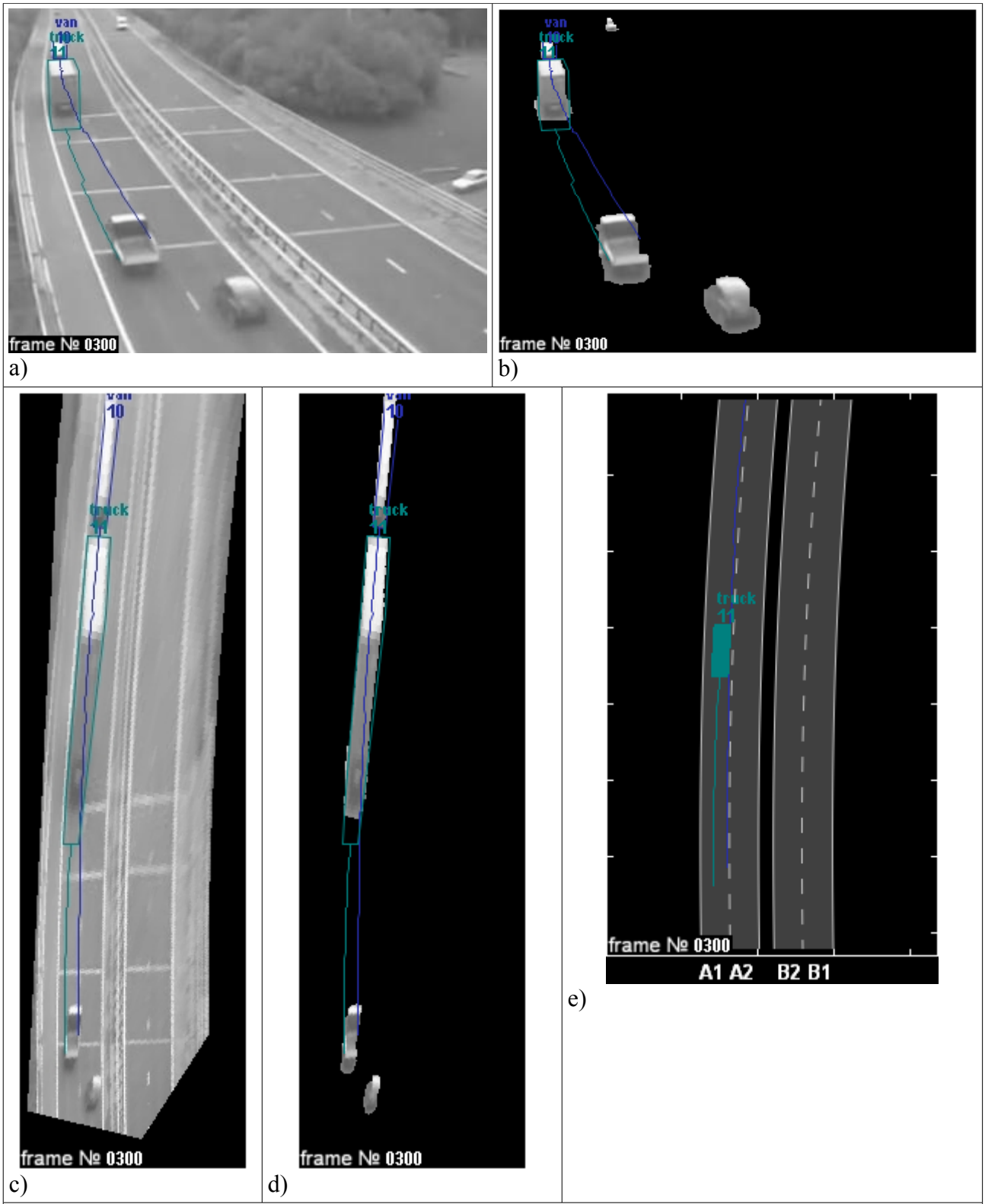
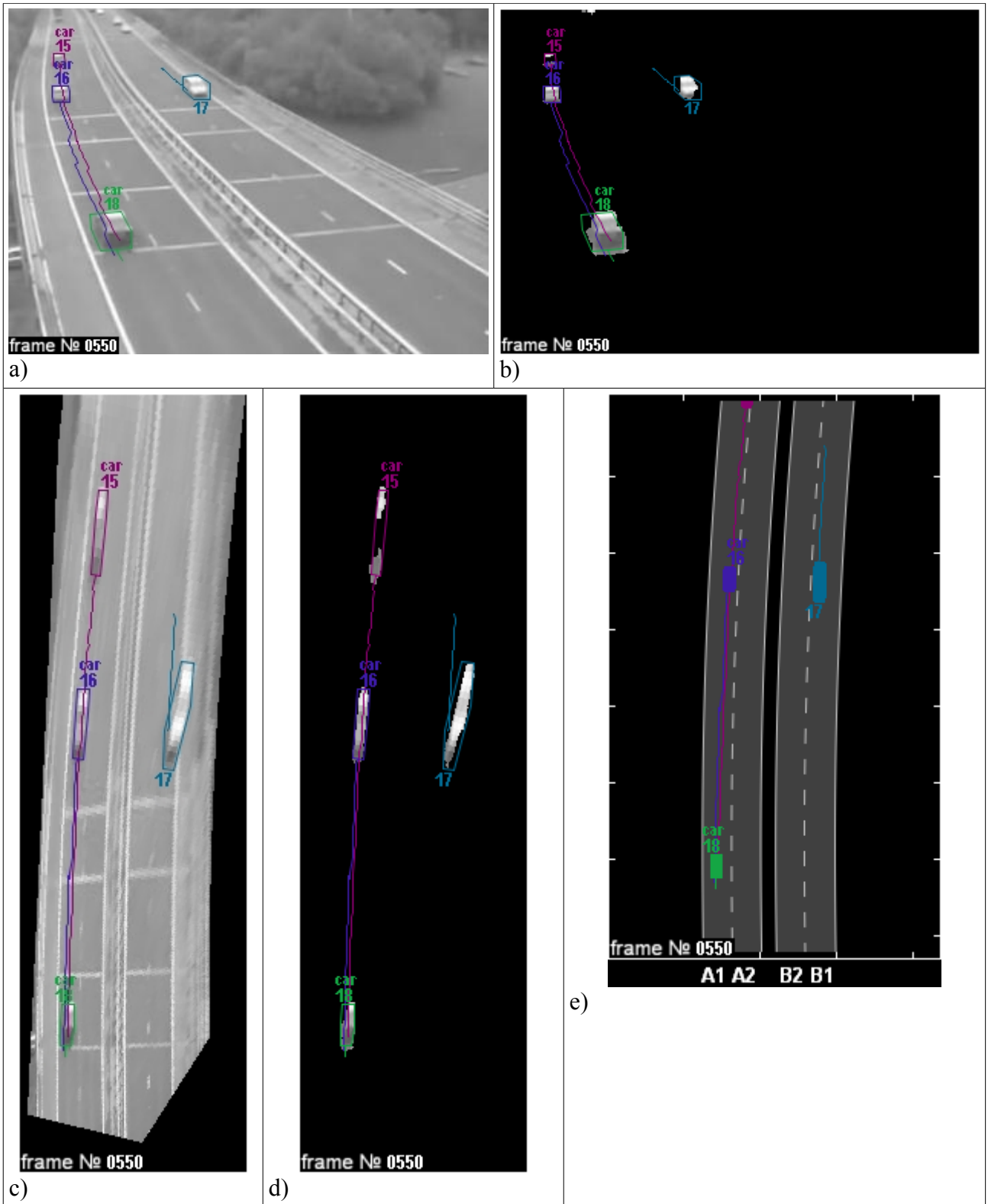


Fig. 5.3. Results for scenario 1.
a) Video frame No.300 from [8] with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.



a) b) c) d) e)

Fig. 5.4 Results for scenario 1.
a) Video frame No.550 from [8] with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.

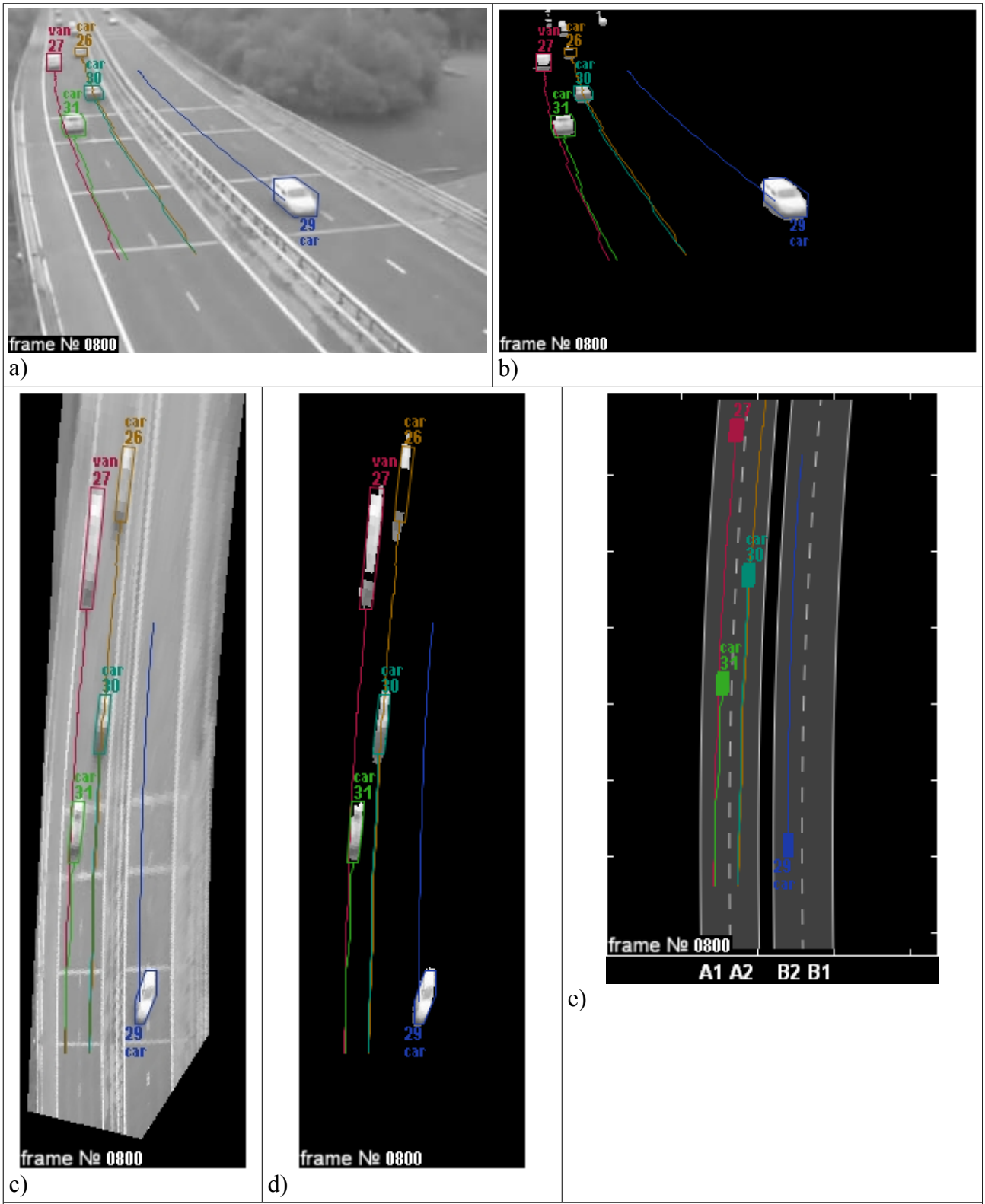


Fig. 5.5. Results for scenario 1.
a) Video frame No.800 from [8] with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.

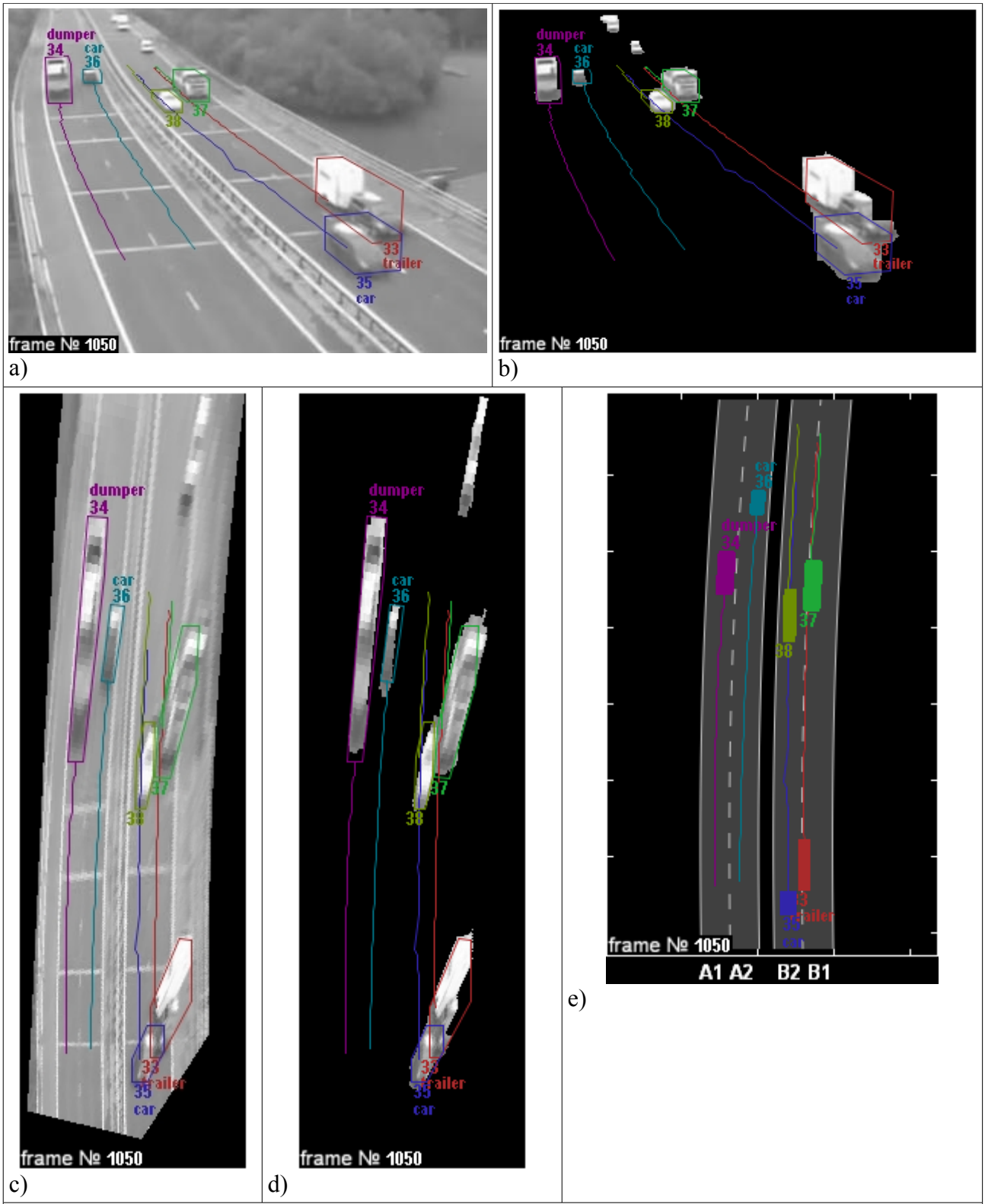
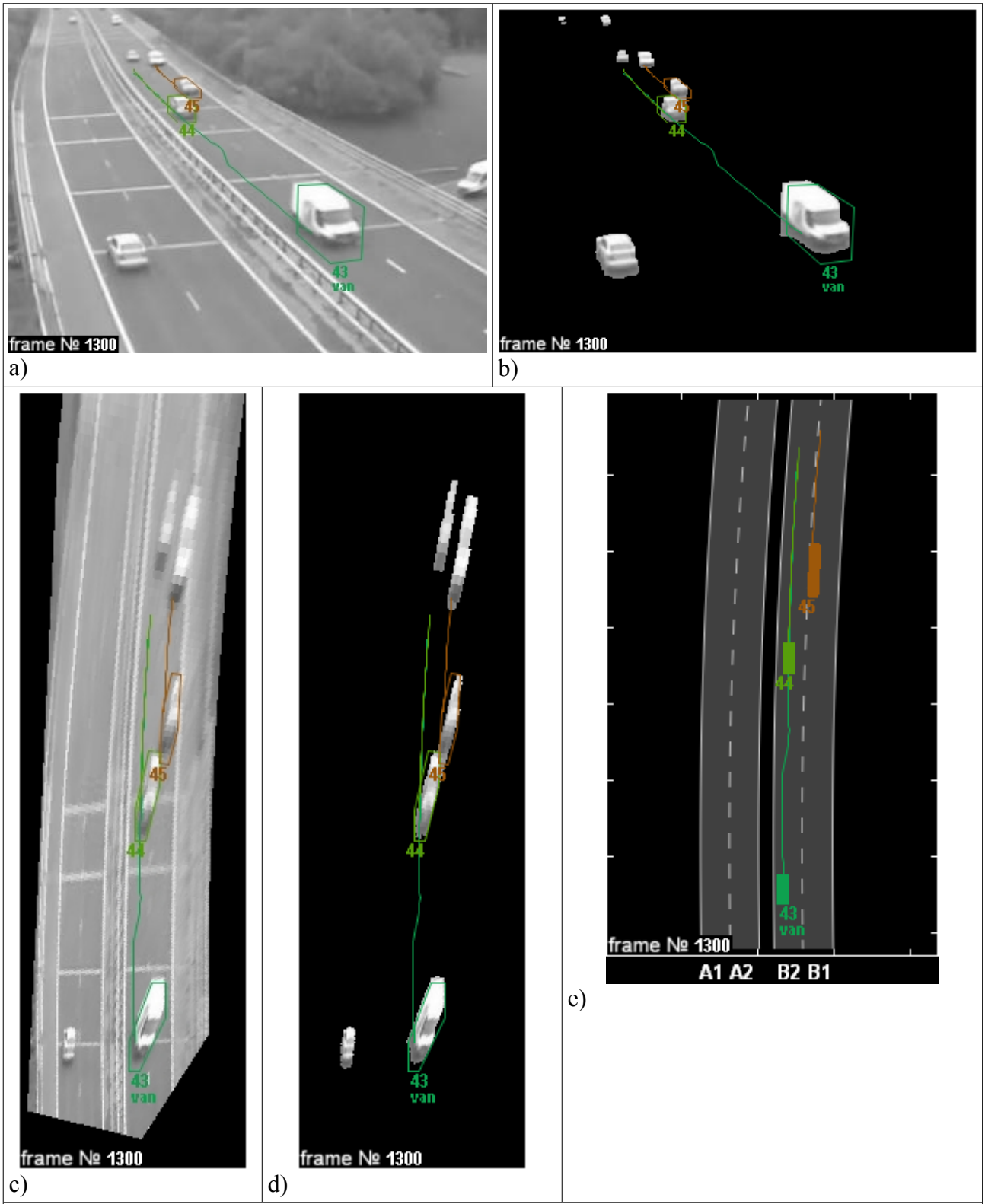


Fig. 5.6. Results for scenario 1.
a) Video frame No.1050 from [8] with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.



a) b) c) d) e)

Fig. 5.7. Results for scenario 1.
a) Video frame No.1300 from [8] with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.

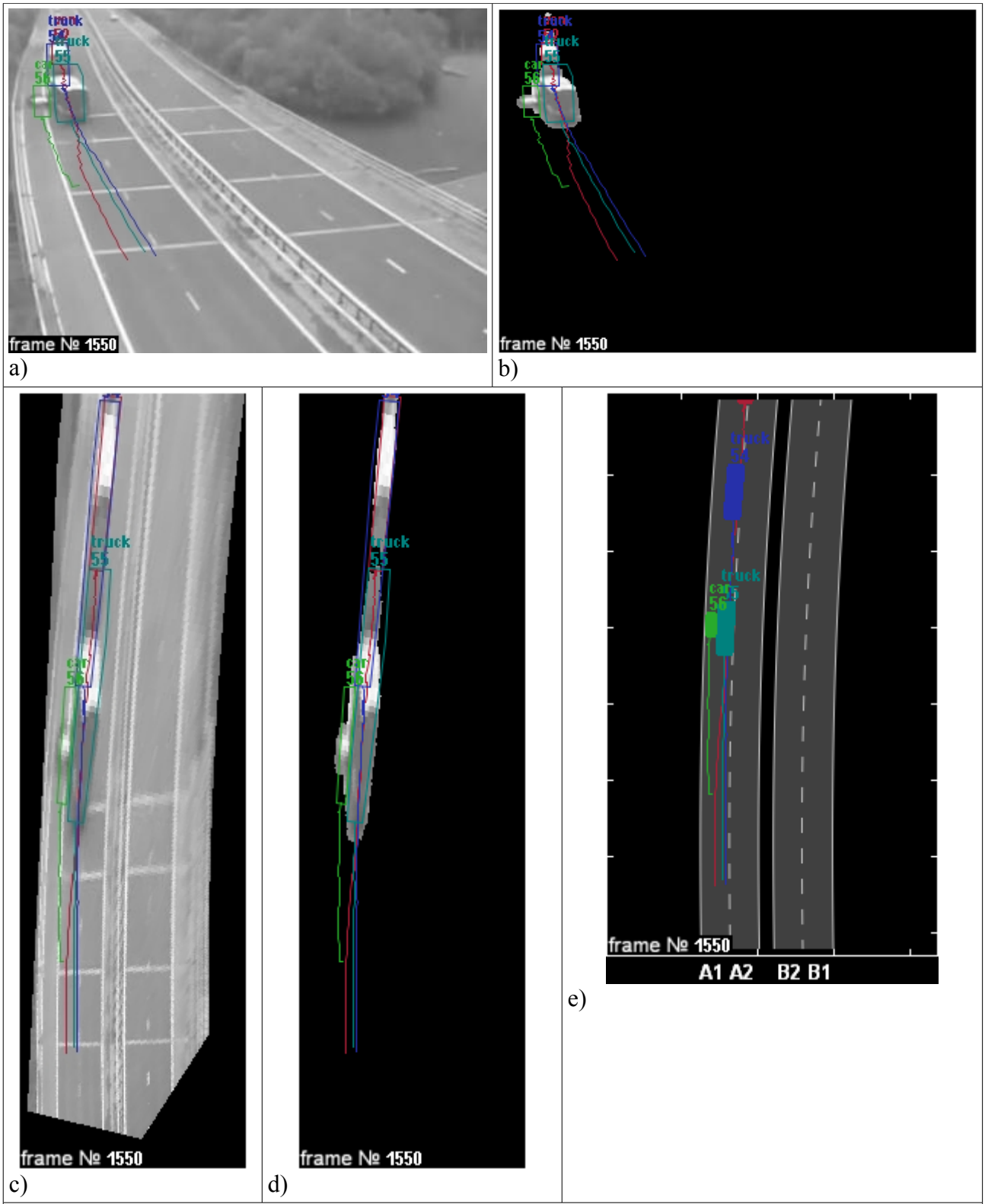


Fig. 5.8. Results for scenario 1.
a) Video frame No.1550 from [8] with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.

After the detection of each object, its classification into one of the vehicle types (car, van, truck, dumper, trailer) was made by comparing its visual appearance with a sample of a vehicle of each type. Statistics on the vehicle speed was gathered for each lane and each vehicle type. Different stages of image processing for this scenario are shown at Fig. 5.2 – 5.8. The resulting traffic statistics is presented in Table 5.1.

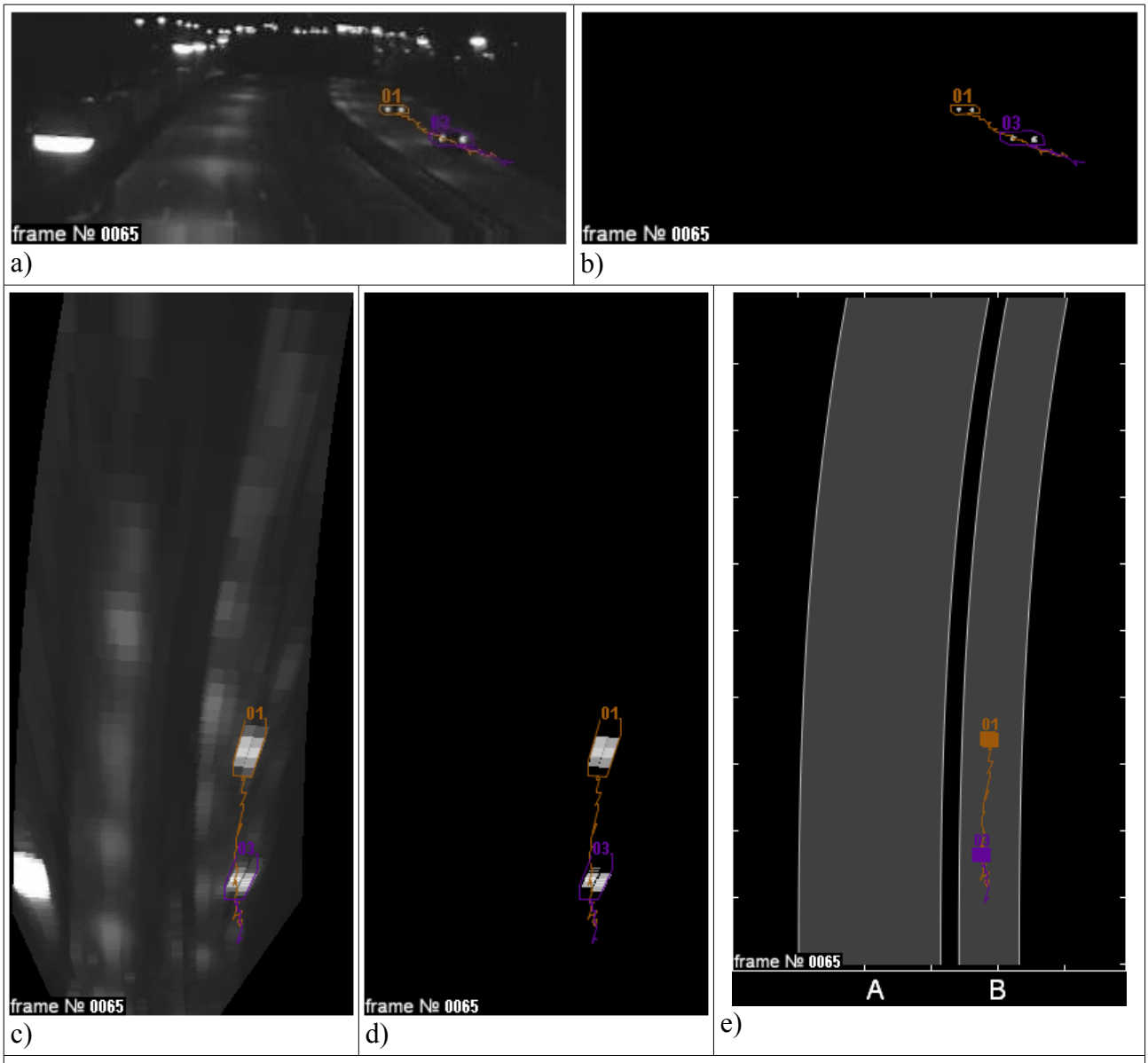
		No. of vehicles per lane				%	speed, km/h		
		A1	A2	B2	B1		min.	aver.	max.
No. of vehicles per type	cars	10	7	7	12	69	51	69	102
	vans	2	0	2	4	15	53	66	85
	dumpers	2	0	0	1	6	44	49	58
	trucks	1	0	0	2	6	45	60	70
	trailers	0	0	0	1	2	60	60	60
unclassified		1	0	0	0	2	59	59	59
%		31	13	17	38				
speed, km/h	min.	44	64	78	56		44		
	aver.	54	72	86	65			66	
	max.	63	81	102	75				102

Table 5.1. Statistics for the vehicle speed per lane and per vehicle type for scenario 1.

5.2. Scenario 2: Video from an unstable static camera, taken at nighttime

A video sequence provided by Prof. Irene Gu was used for this scenario. The camera calibration was done in assumption that the true distance between the points A and B (see Fig. 5.9) satisfies the Swedish national standards on lane width and is equal to 3.75 m, and, since the point from where the video had been taken was known, the camera height was estimated to be h = 7.8 m. The other parameters of the camera appeared to be the following:
camera focal length f = 61.6 m;
tilt angle $\varphi = -3.7°$,
pan angle $\theta = 5.3°$.

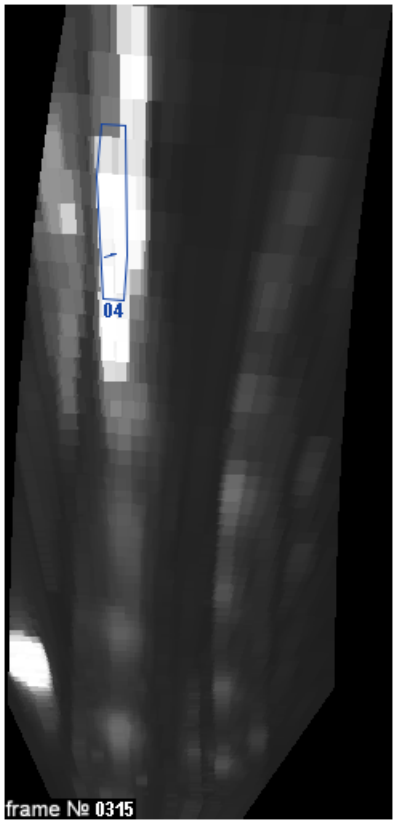
Fig. 5.9. The background image used for camera calibration in scenario 2, and the points whose coordinates in the world coordinate system were assumed to be known.



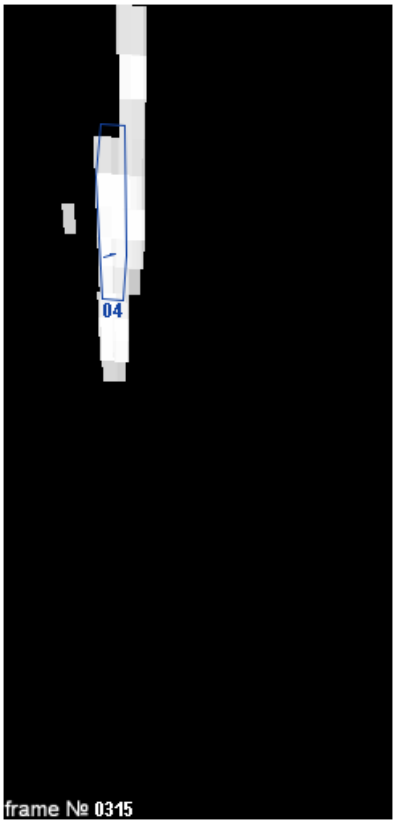
a) b) c) d) e)

Fig. 5.10. Results for scenario 2.
a) Video frame No.65 for scenario 2 with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.

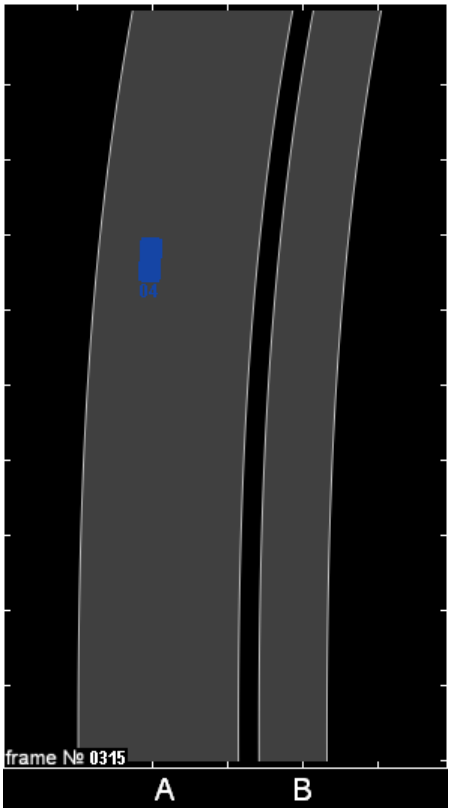
a)



c)



d)



e)

Fig. 5.11. Results for scenario 2.
a) Video frame No.315 for scenario 2 with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.

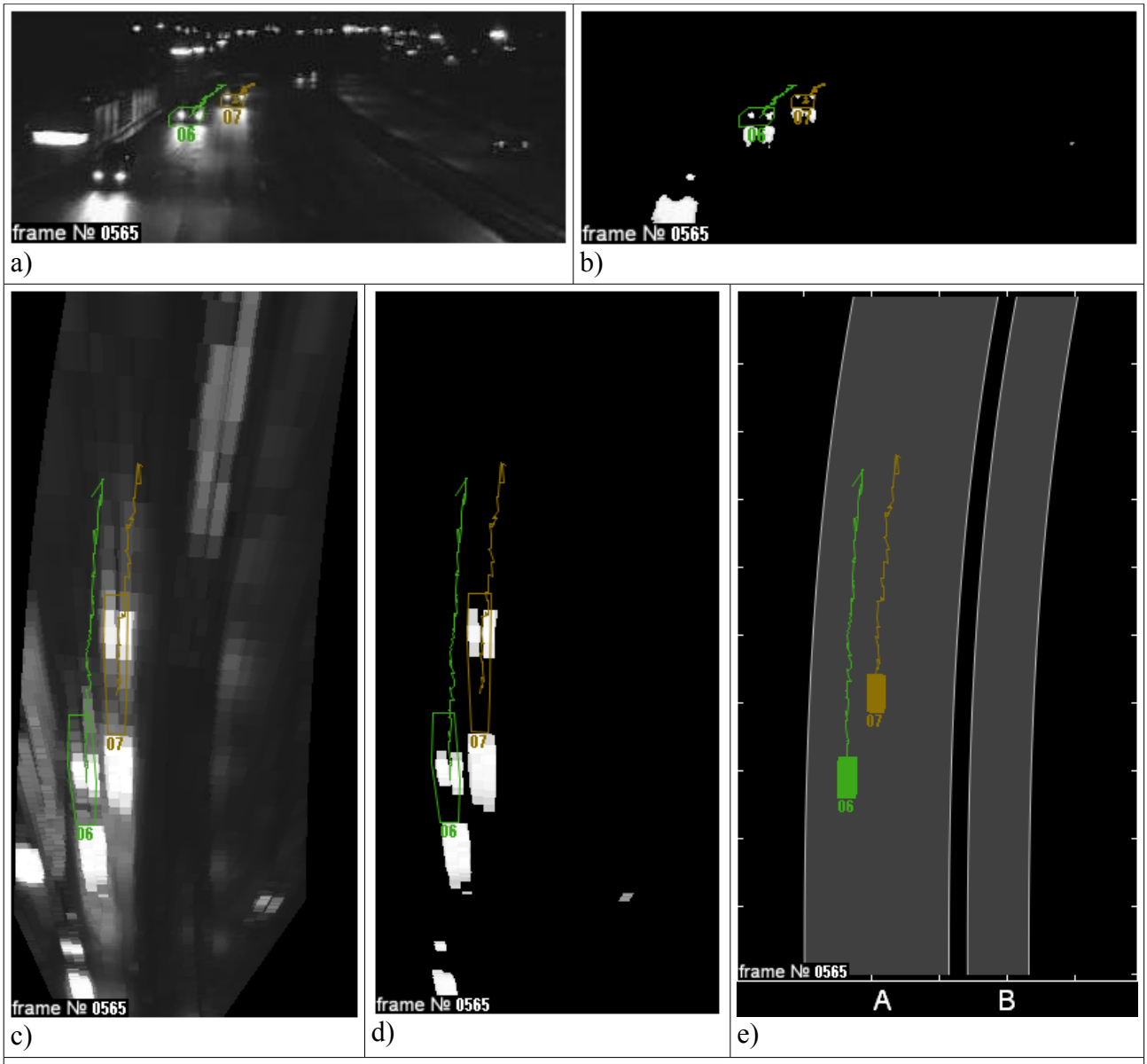


Fig. 5.12. Results for scenario 2.
a) Video frame No.565 for scenario 2 with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.

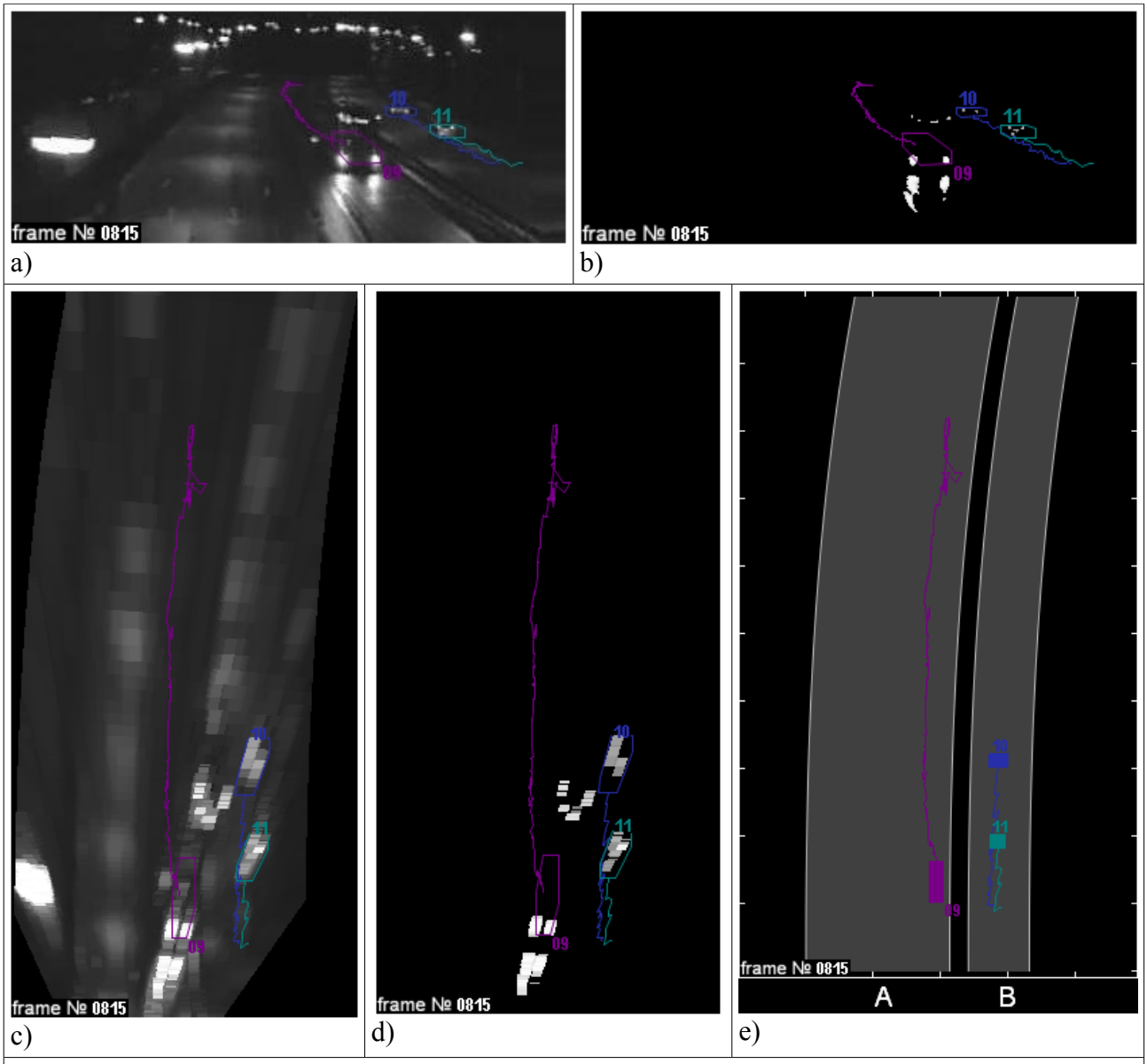


Fig. 5.13. Results for scenario 2.
a) Video frame No.815 for scenario 2 with the tracked vehicles and their trajectories marked.
b) The foreground objects from (a).
c) The road plane projection of the frame from (a).
d) The road plane projection of (b).
e) Synthetic "road map" image showing the traffic situation.

Using these parameters, a sequence of 835 frames was proceeded. 11 vehicles passed through the tracking area in this video sequence, all of them were successfully detected, no misdetection of irrelevant objects was observed.

Due to the poor visibility at the nighttime and high instability of the camera pose, noise in object coordinates was significant; therefore, statistics on the vehicle speed was gathered only per traffic direction. Different stages of image processing for this scenario are shown at Fig. 5.10 – 5.13. The resulting traffic statistics is presented in Table 5.2.

No. of vehicles per traffic direction		speed, km/h		
A	B	min.	aver.	max.
5	6	24	33	53

Table 5.2. Statistics for the traffic intensity per direction and the vehicle speed for scenario 2.

5.3. Scenario 3: Video from a highly unstable aerial camera, taken at daytime

The video sequence from [9] was used for this scenario. In order to apply the algorithms for the static camera case to this scenario, a preliminary stabilization of the video sequence was necessary. This was made by selecting 7 stable feature points visible in all frames (see Fig. 5.14), searching for them at each frame, and using them to build a planar homography between the desired fixed coordinates of these points at the target image and their found coordinates at the source image, as described in [20, p.2–6]. Then each frame was transformed to the same projection using inverse mapping.

Since the video sequence represents an aerial view, it was suitable for being treated as a road plane projection; therefore, no special camera calibration was necessary in this case. The background image was combined from parts of different frames containing no vehicles; it is shown at Fig. 5.15.

2138 frames were proceeded, extracting statistical information on the traffic direction between different streets in the crossing shown in the video sequence. 88 vehicles (including 1 motorcycle) passed through the tracking area in this video sequence, all of them were successfully detected, no misdetection of irrelevant objects was observed.

Different stages of image processing for this scenario are shown at Fig. 5.16 – 5.24. The resulting traffic statistics is presented in Table 5.3.

		traffic to			
		Street 1	Street 2	Street 3	Street 4
traffic from	Street 1	7	2	7	7
	Street 2	0	8	2	19
	Street 3	0	5	9	0
	Street 4	4	14	0	4

Table 5.3. Statistics on the traffic directions for scenario 3 (If the source and destination of the traffic are the same, this means that the vehicles either were standing or the video sequence does not show their whole way).

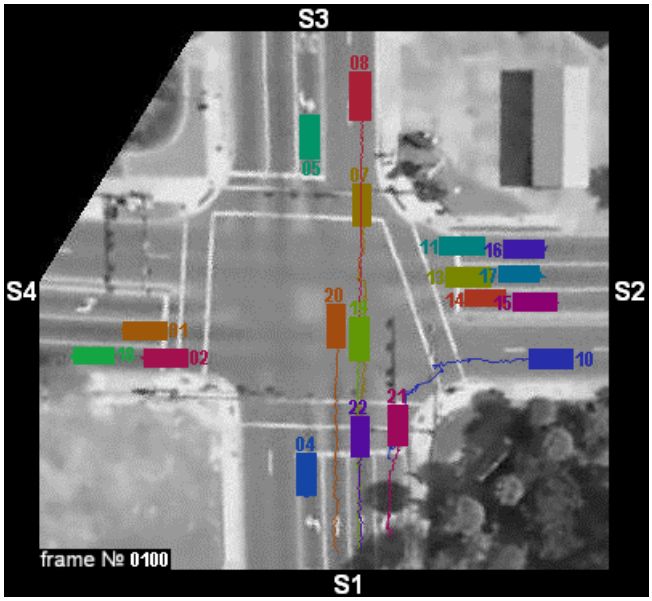
Fig. 5.14. A frame from the video used for scenario 3, with the points used for stabilization marked.

Fig. 5.15. The background image for scenario 3.

a)

b)

c)



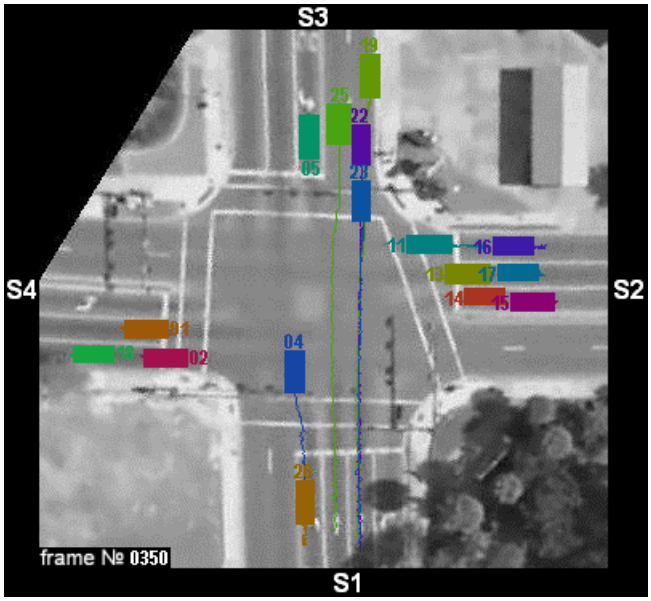
d)

Fig. 5.16. Results for scenario 3.
a) Video frame No.100 from [9].
b) Stabilized frame from (a) with the tracked vehicles and their trajectories marked.
c) The foreground objects from (b).
d) Synthetic "road map" image showing the traffic situation.

a)

b)

c)

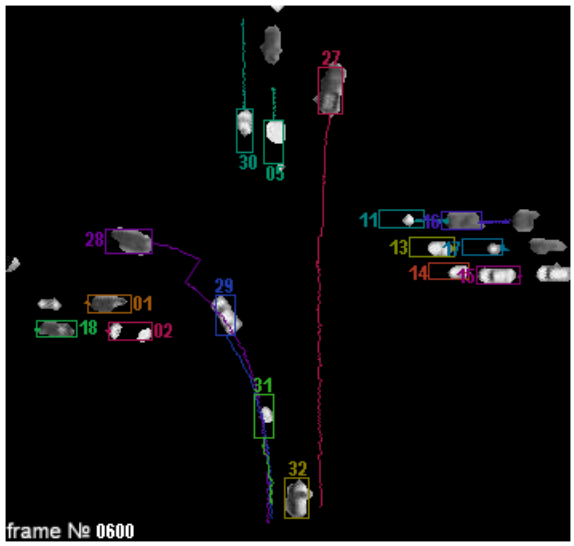


d)

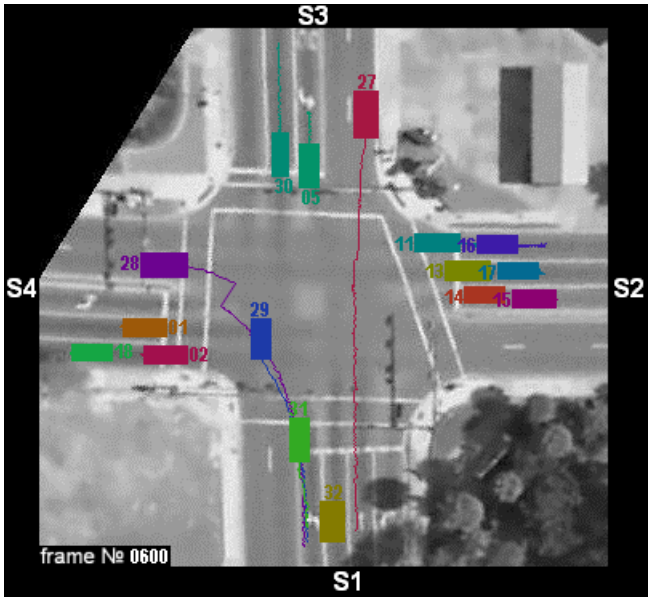
Fig. 5.17. Results for scenario 3.
a) Video frame No.350 from [9].
b) Stabilized frame from (a) with the tracked vehicles and their trajectories marked.
c) The foreground objects from (b).
d) Synthetic "road map" image showing the traffic situation.

a)

b)



c)

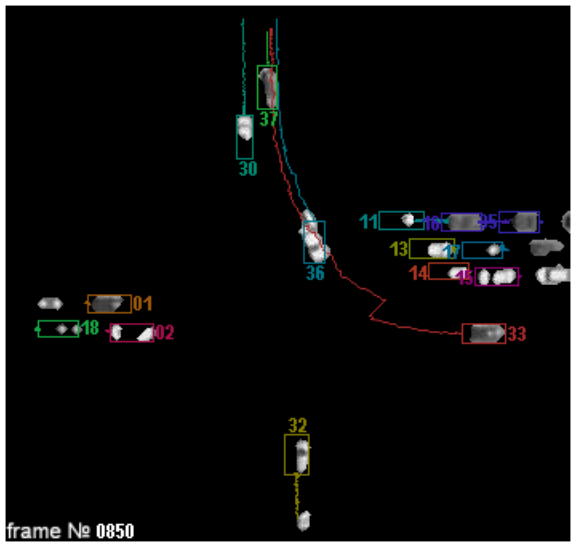


d)

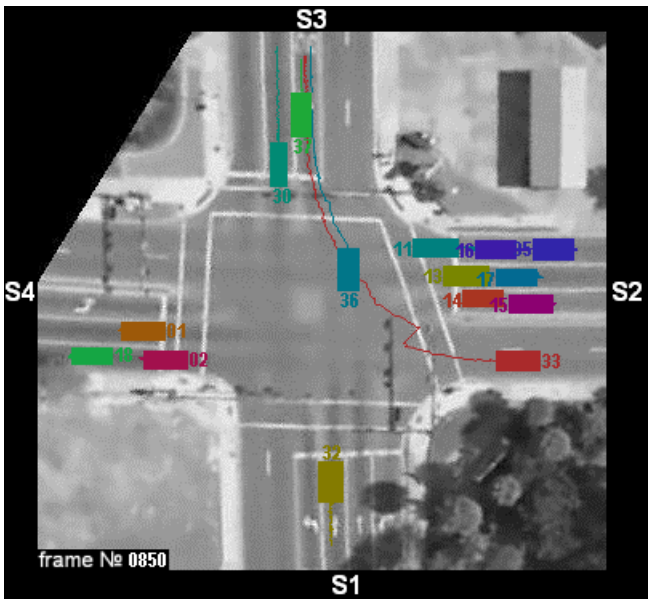
Fig. 5.18. Results for scenario 3.
a) Video frame No.600 from [9].
b) Stabilized frame from (a) with the tracked vehicles and their trajectories marked.
c) The foreground objects from (b).
d) Synthetic "road map" image showing the traffic situation.

a)

b)



c)

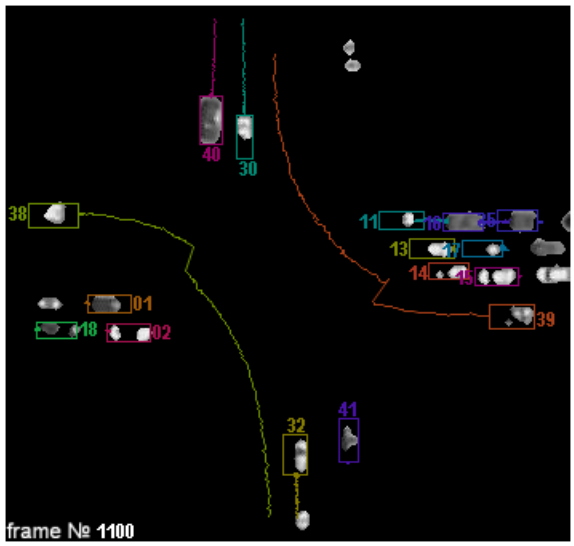


d)

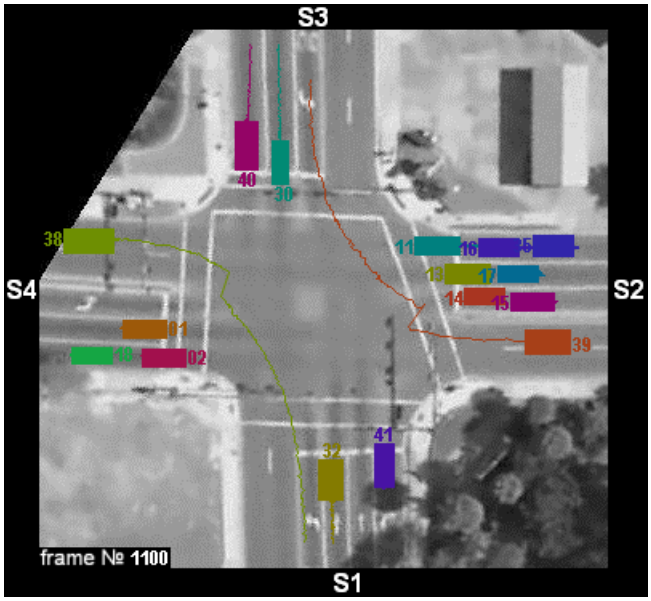
Fig. 5.19. Results for scenario 3.
a) Video frame No.850 from [9].
b) Stabilized frame from (a) with the tracked vehicles and their trajectories marked.
c) The foreground objects from (b).
d) Synthetic "road map" image showing the traffic situation.

a)

b)



c)

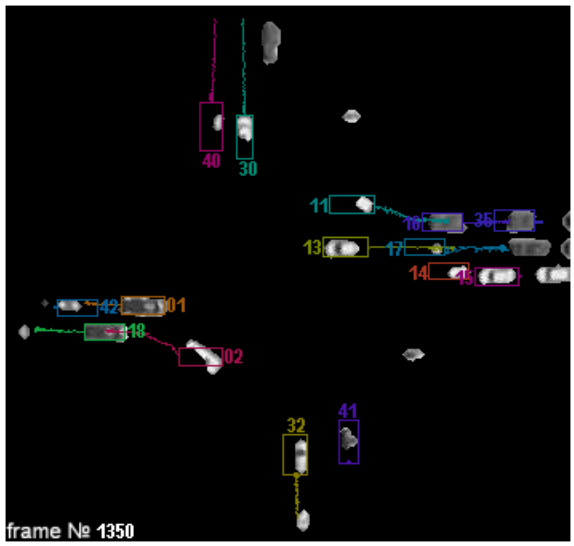


d)

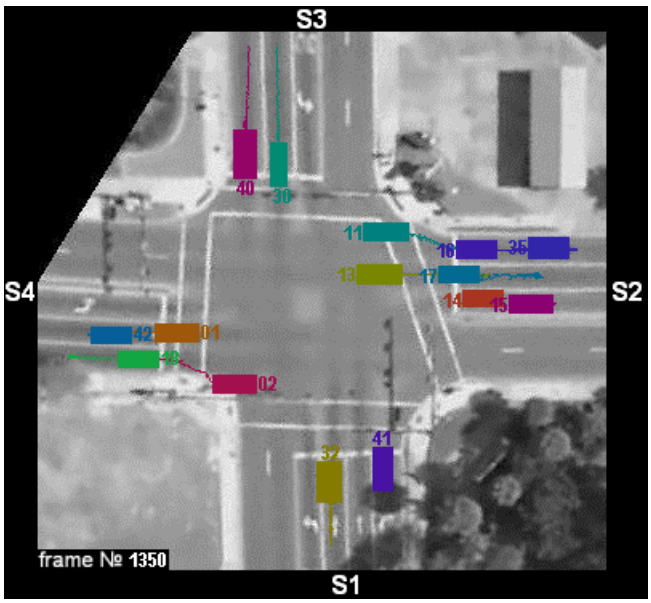
Fig. 5.20. Results for scenario 3.
a) Video frame No.1100 from [9].
b) Stabilized frame from (a) with the tracked vehicles and their trajectories marked.
c) The foreground objects from (b).
d) Synthetic "road map" image showing the traffic situation.

a)

b)



c)



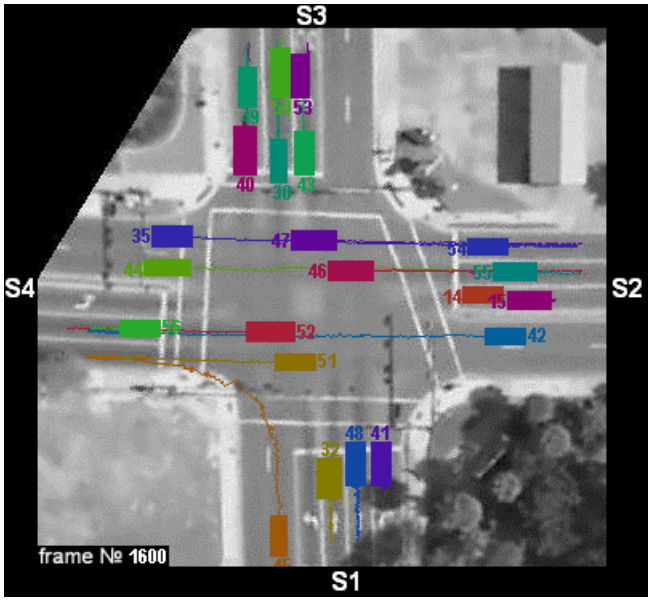
d)

Fig. 5.21. Results for scenario 3.
a) Video frame No.1350 from [9].
b) Stabilized frame from (a) with the tracked vehicles and their trajectories marked.
c) The foreground objects from (b).
d) Synthetic "road map" image showing the traffic situation.

a)

b)

c)

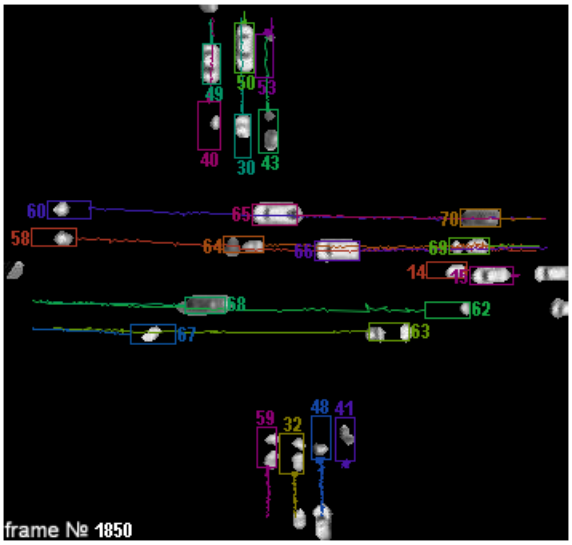


d)

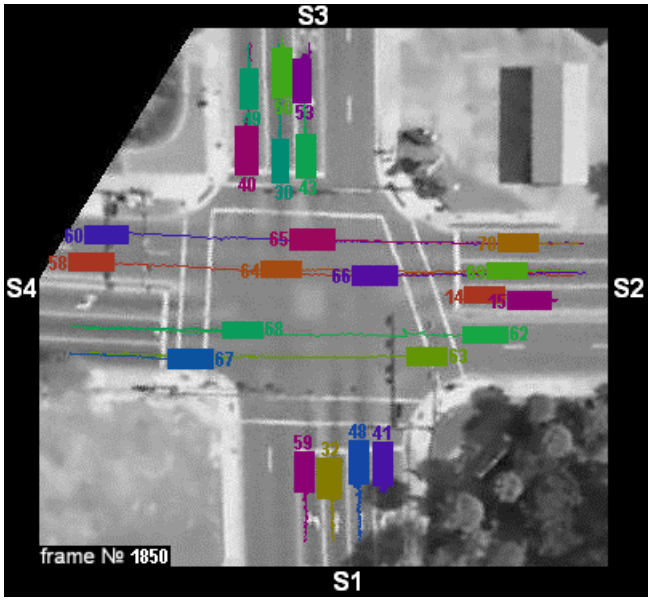
Fig. 5.22. Results for scenario 3.
a) Video frame No.1600 from [9].
b) Stabilized frame from (a) with the tracked vehicles and their trajectories marked.
c) The foreground objects from (b).
d) Synthetic "road map" image showing the traffic situation.

a)

b)



c)

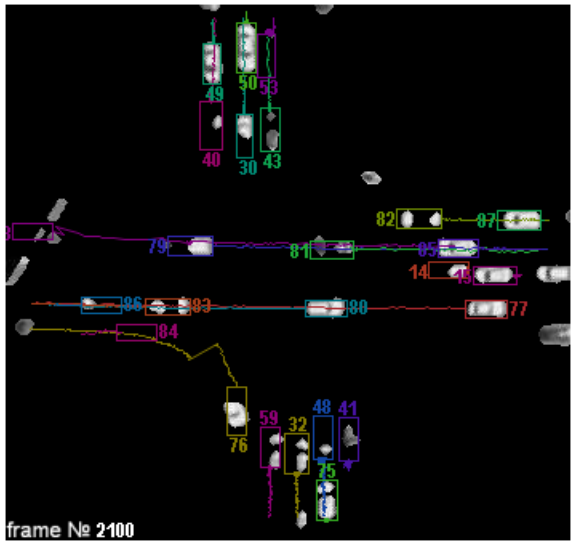


d)

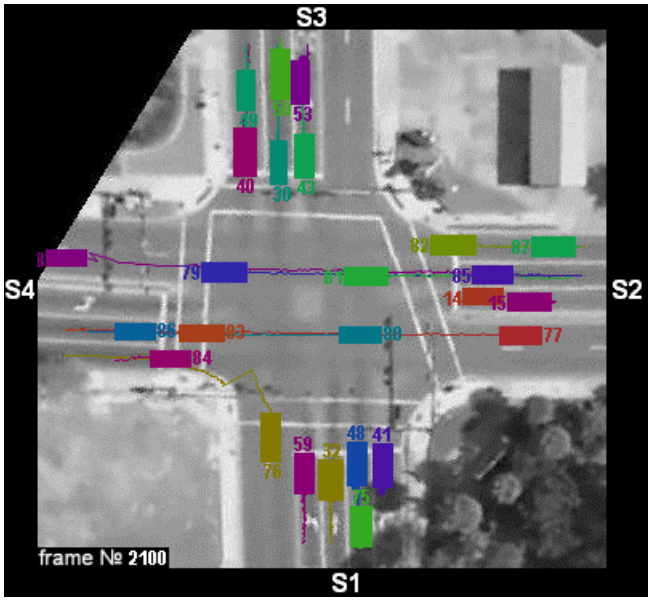
Fig. 5.23. Results for scenario 3.
a) Video frame No.1850 from [9].
b) Stabilized frame from (a) with the tracked vehicles and their trajectories marked.
c) The foreground objects from (b).
d) Synthetic "road map" image showing the traffic situation.

a)

b)



c)



d)

Fig. 5.24. Results for scenario 3.
a) Video frame No.2100 from [9].
b) Stabilized frame from (a) with the tracked vehicles and their trajectories marked.
c) The foreground objects from (b).
d) Synthetic "road map" image showing the traffic situation.

5.4. Comments on the static camera scenarios

The amount of data gathered from the 3 static camera scenarios allows to make some evaluations of the performance of the methods used. First, the effectiveness of background subtraction for detection of moving objects can be assessed, which is done Section 5.4.1. Second, the sensitivity of the vehicle detection procedure (in the sense of the time needed to react on a new object appeared in the tracking area) can be also estimated; the result of this is presented in Section 5.4.2.

5.4.1. Background subtraction performance assessments

In Table 5.4., data on the quality of the resulting image after the background subtraction for the proceeded 3 static camera scenarios is presented. For the sum of the number of vehicles observed in each frame throughout each scenario, the occurrences of the following situations were counted:
– intact objects, i.e. cases when one segment resulting from the background subtraction corresponds to one vehicle;
– fragmented objects, i.e. cases when the image of a vehicle was split into two or more segments by the background subtraction procedure;
– merged objects, i.e. cases when two or more nearly located vehicles were put into a single segment by the background subtraction procedure;
– objects both fragmented and merged, i.e. cases when one fragment of a fragmented object was merged with another object;
– invisible objects, i.e. cases when the background subtraction procedure failed to distinguish a vehicle from the background.

Scenario		Objects in all frames	Intact objects	Fragmented objects	Merged objects	Objects both fragmented and merged	Invisible objects
1	total	4925	3671	686	492	72	4
	%%	*100*	*74.5*	*13.9*	*10.0*	*1.5*	*0.1*
2	total	1576	570	686	103	211	6
	%%	*100*	*36.2*	*43.5*	*6.5*	*13.4*	*0.4*
3	total	38769	29561	8768	2	2	436
	%%	*100*	*76.2*	*22.6*	*0.05*	*0.05*	*1.1*

Table 5.4. Assessment of the background subtraction performance.

As it can be seen, in the daytime videos the background subtraction produces perfect segment to object correspondence in 3/4 of cases. The share of mergers seems to depend on the viewing angle; in a top view they almost never occur. The high percentage of fragmentation in the nighttime scenario reflects the fact that the front or rear lights of a vehicle, which produce two foreground segments in good visibility conditions, were the actual object to track. The occurrence of invisible objects was low, though it tends to grow with increase of the distance from the camera to the object.

5.4.2. Detection delay assessment

In a perfect case, the vehicle identification procedure described in Section 4.2.4. should notice a newly appeared vehicle as soon as it enters the tracking area. However, in some cases detection

does not occur immediately due to some factor that prevents a reliable classification of the object as a vehicle. This can be, for example, the following:
– low quality of background subtraction in this particular area, which produces a fragment smaller than the necessary minimum size, specified in (4.39);
– occlusion of the newly appeared object.
In order to avoid errors, the detection procedure implemented in this thesis work postpones detection to the next frame if it cannot decide certainly whether a newly appeared object is a vehicle. Such situations lead to a delay in object detection. In Table 5.5, data on detection delays in the proceeded 3 static camera scenarios is presented.

Scenario	No. of frames	No. of vehicles	Vehicles detected with a delay		Delay length, frames			
			No.	*%%*	min.	average	median	max.
1	1551	56	18	*32.1*	0	1.2	0	17
2	835	11	8	*72.7*	0	7.8	7	20
3	2138	88	42	*47.7*	0	4.0	0	64

Table 5.5. Assessment of the reaction time of the vehicle identification procedure.

Here it can be seen that in the daytime scenarios most vehicles were detected immediately as they appeared. In the nighttime scenario, delayed detections were more common because of the difficulty in distinguishing between the front lights of a vehicle and the reflection of these lights off the road surface when the distance from the camera to the object is long. In the non-top view scenarios 1 and 2, the longest delays in detection occurred due to partial occlusions. In scenario 3, the longest delay happened when a vehicle was standing in a badly visible place at the edge of the tracking area for long time.

5.5. Scenario 4: Video from a moving camera mounted on a vehicle, taken at daytime

The video sequence from [10] was used for this scenario. The initial camera calibration was done in assumption that the true distance between the points A and B (see Fig. 5.25) is equal to 3 m, and the distance between B and C is 7.5 m.

The initial camera parameters for the first frame appeared to be the following:
camera focal length f = 21.0 m;
camera height over the road surface h = 0.8 m;
tilt angle φ = –9.1°,
pan angle θ = 2.3°.

Since detection of road marking elements was beyond the scope of this thesis work, adjustment of camera parameters for further frames was done manually. After this, 1612 frames were proceeded, extracting statistical information on the vehicle speed for each lane. 12 vehicles passed through the tracking area in this video sequence, all of them were successfully detected, no misdetection of irrelevant objects was observed.

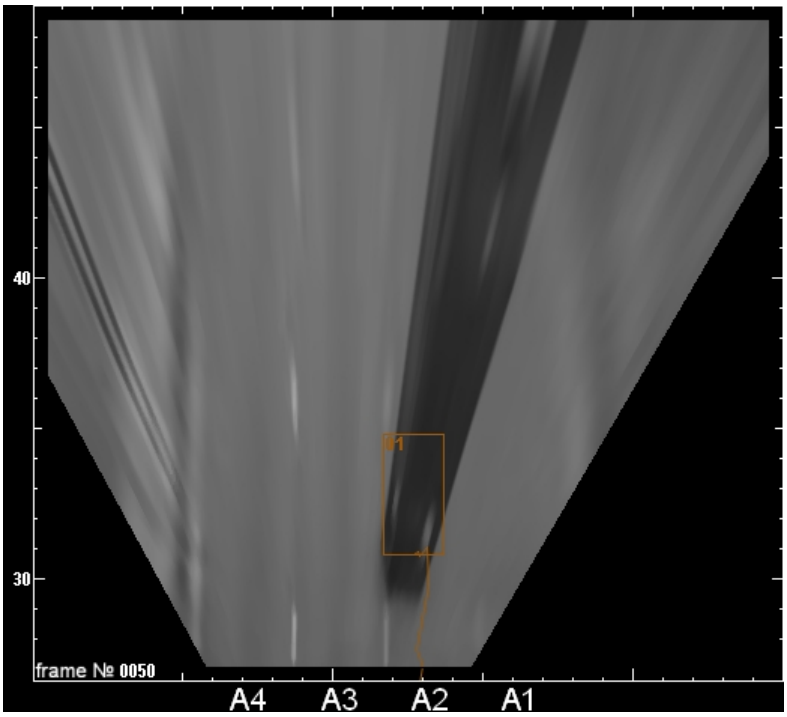
The resulting traffic statistics is presented in Table 5.6. Different stages of image processing for this scenario are shown at Fig. 5.26 – 5.30.

Fig. 5.25. The image used for the initial camera calibration in scenario 4; the distances between A and B and between C and D were assumed to be known.

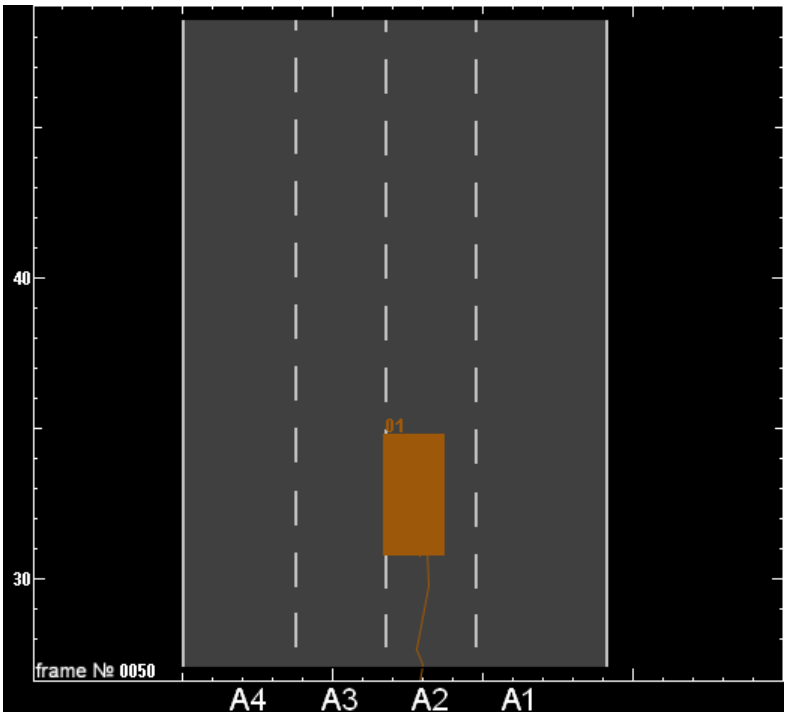
No. of vehicles per lane				speed, km/h		
A4	A3	A2	A1	min.	aver.	max.
0	0	6	6	54	67	86

Table 5.6. Statistics for the traffic intensity per lane and the vehicle speed for scenario 4.

a)



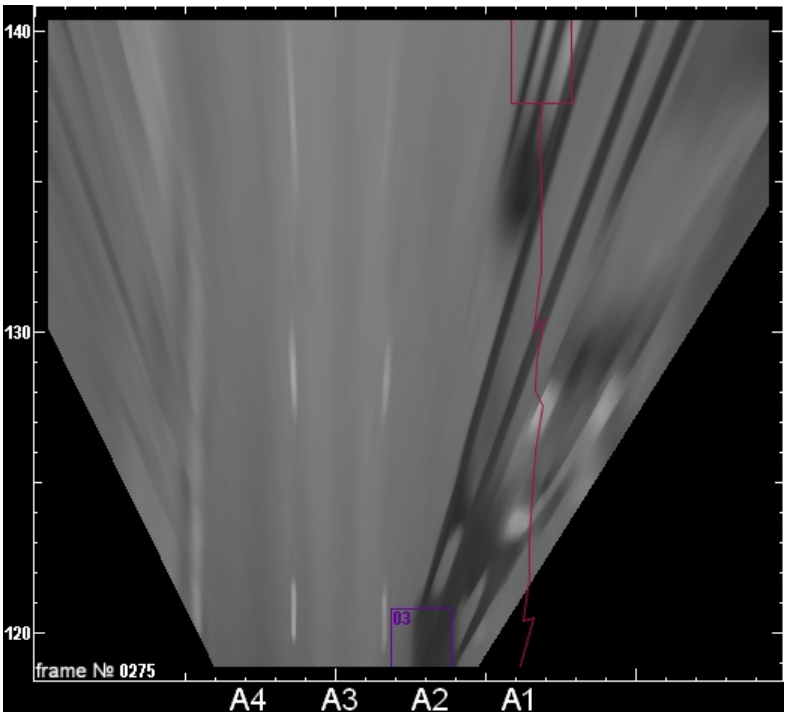
b)



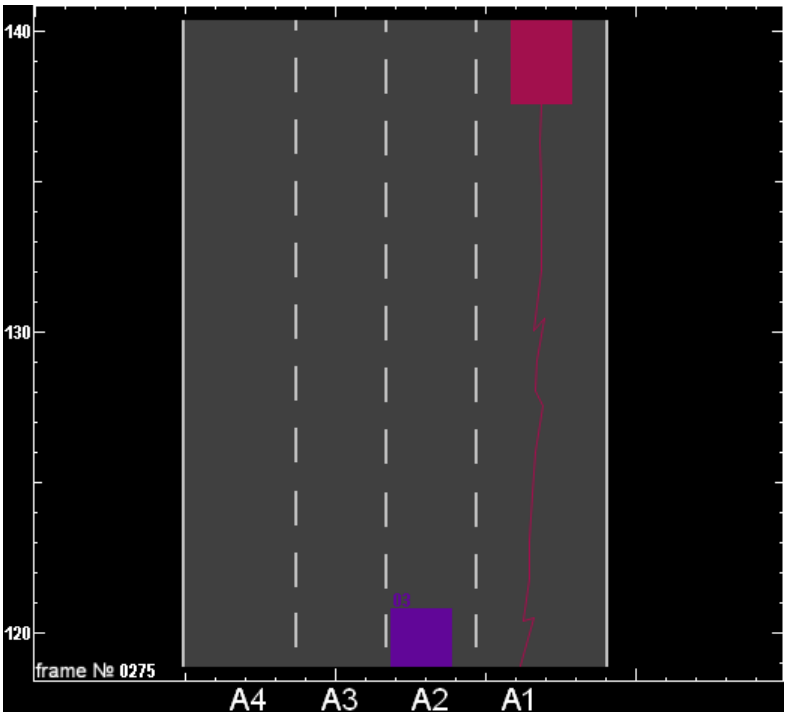
c)

Fig. 5.26. Results for scenario 4.
a) Video frame No.50 with the tracked vehicles and their trajectories marked.
b) Road plane projection of the frame from (a).
c) Synthetic "road map" image based on the frame from (a).

a)



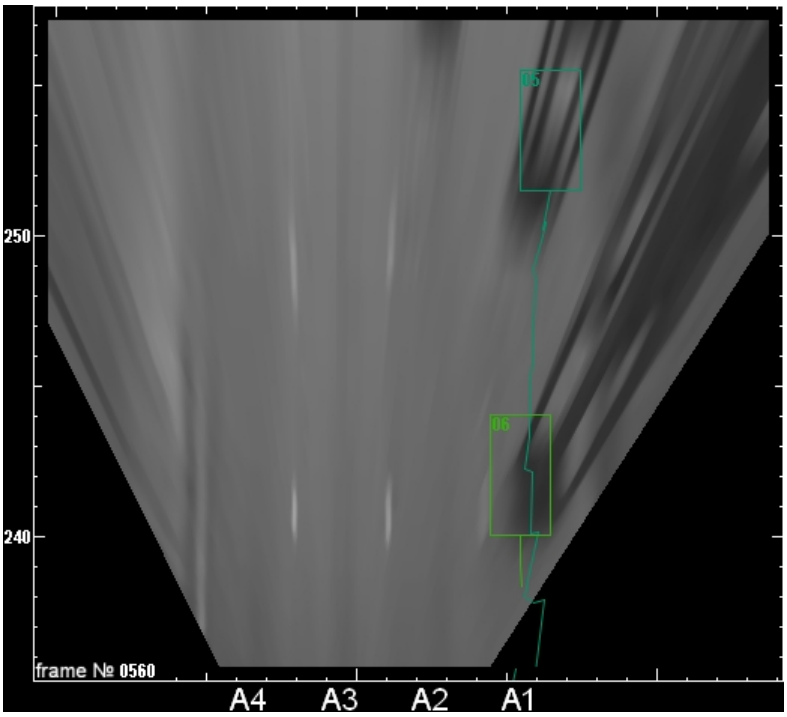
b)



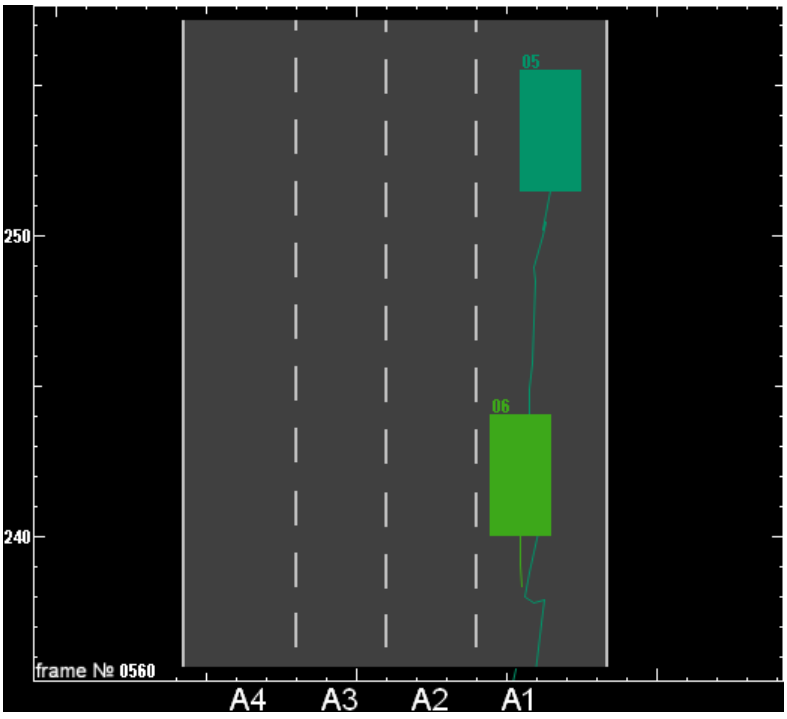
c)

Fig. 5.27. Results for scenario 4.
a) Video frame No.275 with the tracked vehicles and their trajectories marked.
b) Road plane projection of the frame from (a).
c) Synthetic "road map" image based on the frame from (a).

a)



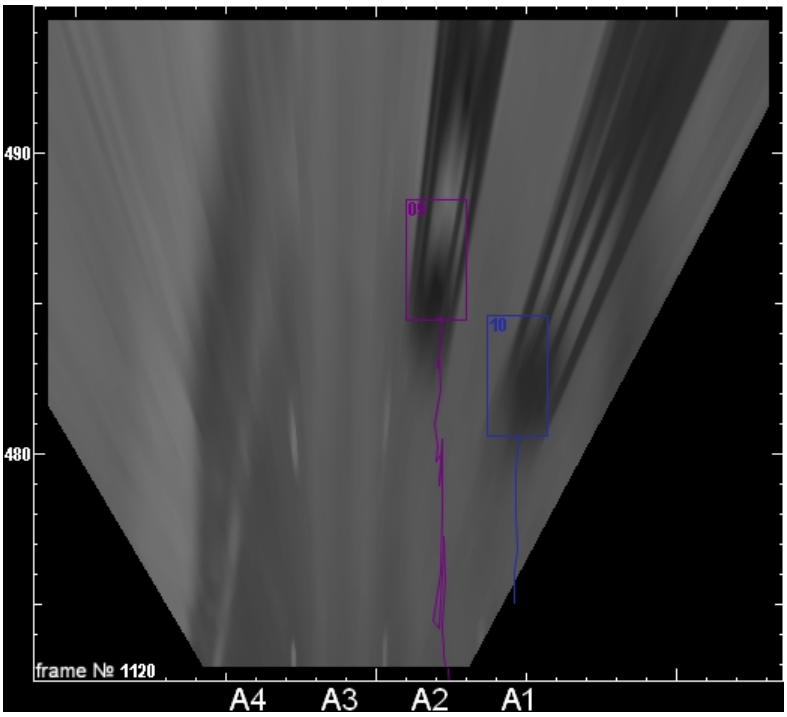
b)



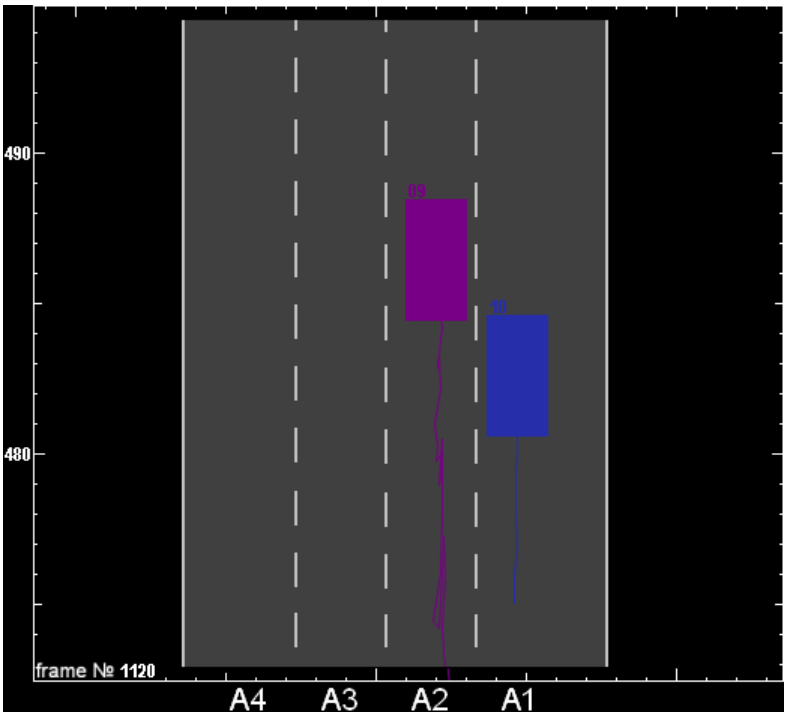
c)

Fig. 5.28. Results for scenario 4.
a) Video frame No.560 with the tracked vehicles and their trajectories marked.
b) Road plane projection of the frame from (a).
c) Synthetic "road map" image based on the frame from (a).

a)



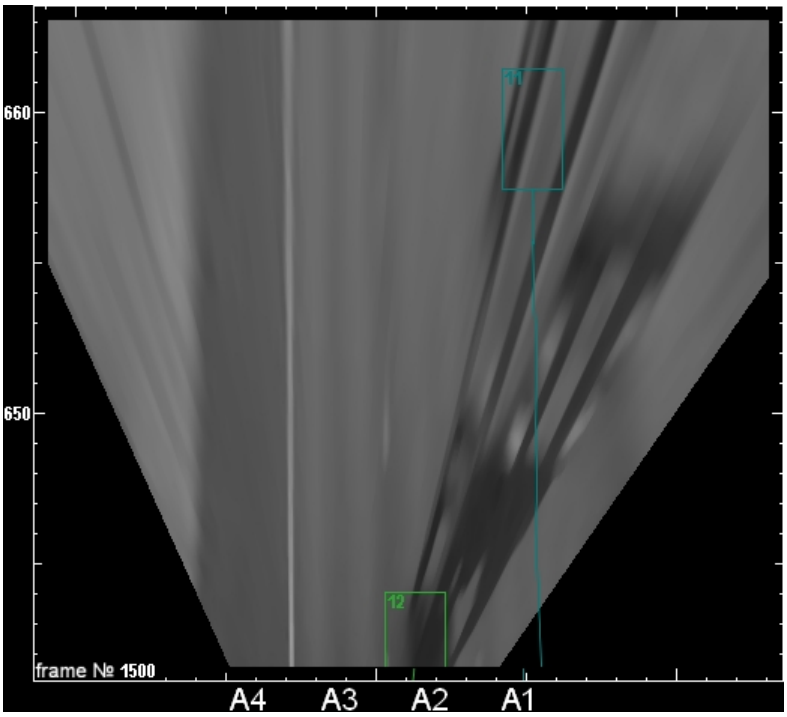
b)



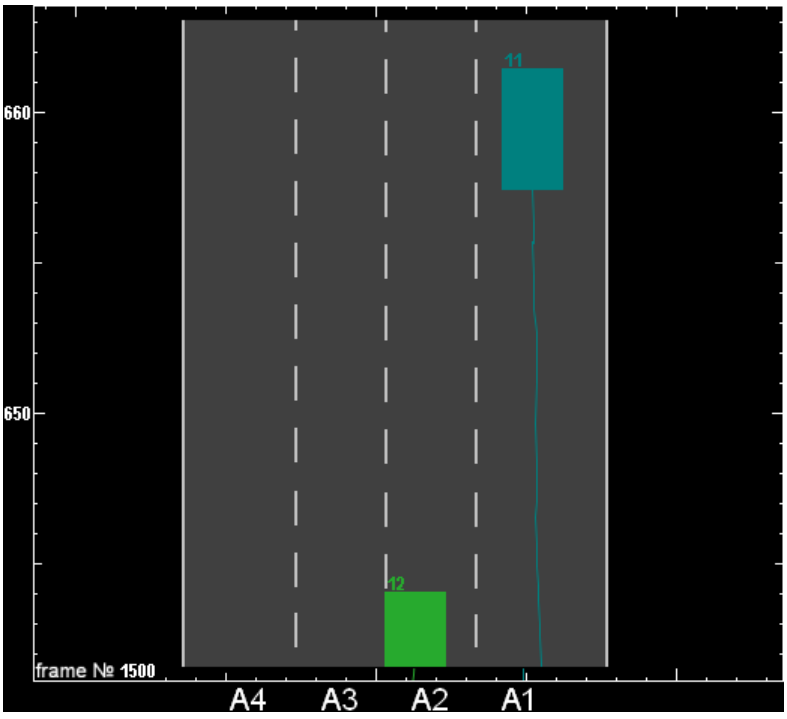
c)

Fig. 5.29. Results for scenario 4.
a) Video frame No.1120 with the tracked vehicles and their trajectories marked.
b) Road plane projection of the frame from (a).
c) Synthetic "road map" image based on the frame from (a).

a)



b)



c)

Fig. 5.30. Results for scenario 4.
a) Video frame No.1500 with the tracked vehicles and their trajectories marked.
b) Road plane projection of the frame from (a).
c) Synthetic "road map" image based on the frame from (a).

5.6. Comments on the moving camera case

One scenario is not sufficient to make any evaluations on the used methods. However, some interesting data on the behavior of the camera parameters during the camera's own movement can be collected.

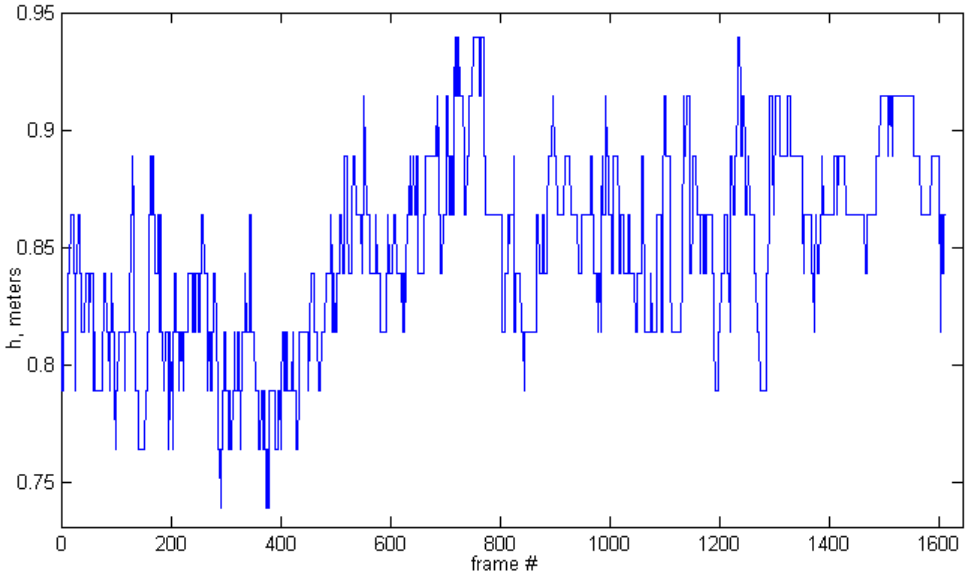
The statistics of the camera parameters change for the whole video sequence is shown at Fig. 5.31 – 5.34. As it can be seen, the camera height h, tilt angle φ, pan angle θ, and roll angle ψ fluctuate around their mean value, but only for the θ the statistical distribution is symmetric. The asymmetry of the statistics for h and φ can be explained by the observation that the up and down movement of the camera mounted on a moving vehicle is not a symmetric process. However, the ψ angle is symmetric with respect to the traffic direction the same way as the θ is. Seemingly, in this case the non-flatness of the road surface (i.e. the fact that real roads are built in a shape of cylindrical surface instead of a perfect plane in order to provide drainage for rainfall water) can matter.

An interesting subject for study can be the correspondences between the camera parameters. As the graphs at Fig.5.35 show, their alterations during the camera movement are not random; they have some correlation with each other. The values of the correlation between different camera parameters for Scenario 4 are presented in Table 5.7.

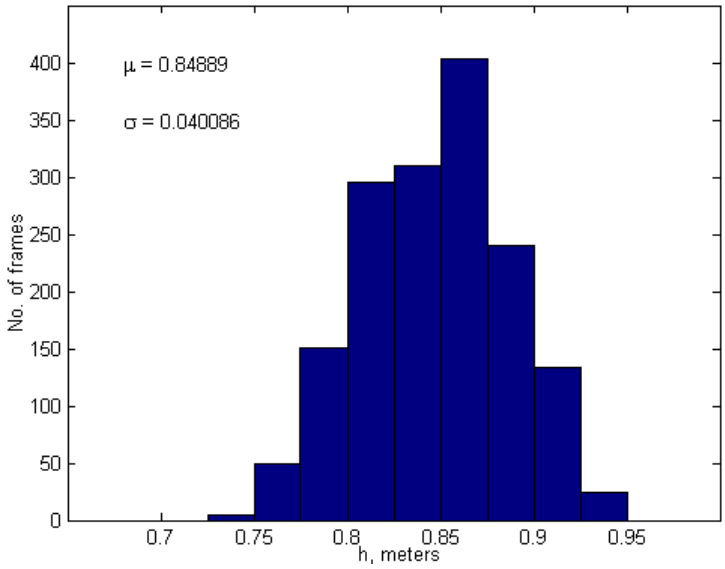
Camera parameters	camera height h	pan angle θ	tilt angle φ
roll angle ψ	0.22	0.01	0.32
tilt angle φ	0.52	0.21	
pan angle θ	0.21		Correlation

Table 5.7. Correlation between different camera parameters in Scenario 4.

As it can be seen, the greatest correlation is observed between the camera height h and its tilt angle φ. This can reflect a specific pattern in the shaking movement of the camera mounted on a moving vehicle. The correlation between the tilt angle φ and is roll angle ψ is also worth mentioning. This can be a promising area for further research.

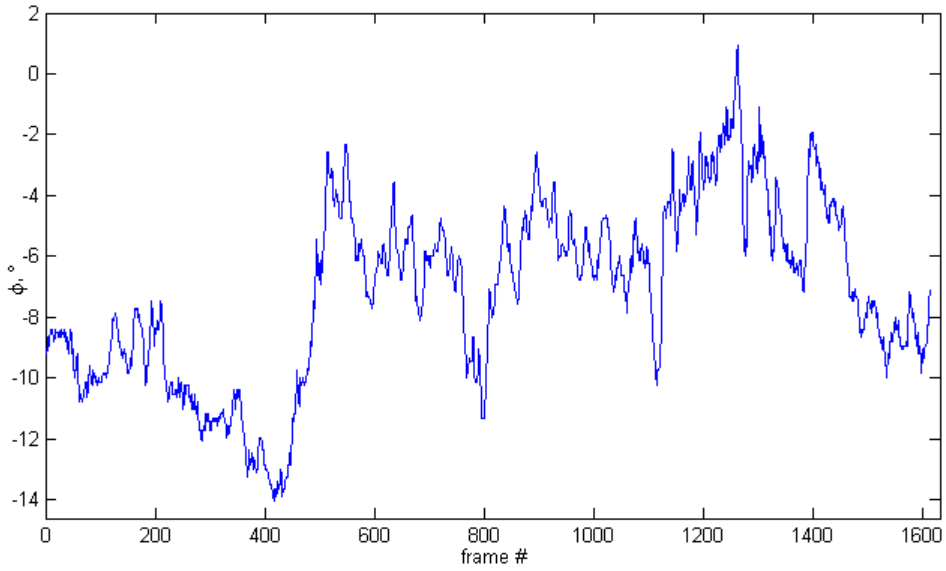


a)

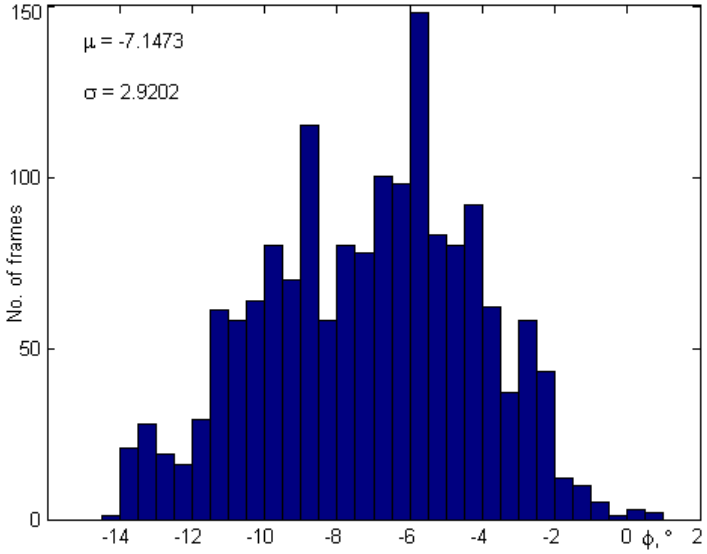


b)

Fig. 5.31. Fluctuation of the camera height *h* for the moving camera scenario 4:
a) time diagram;
b) statistics.

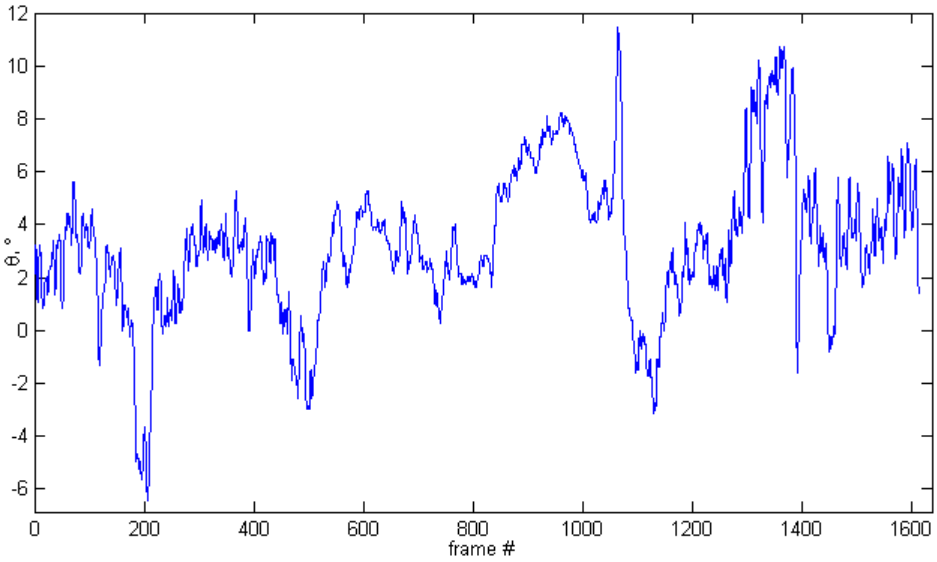


a)

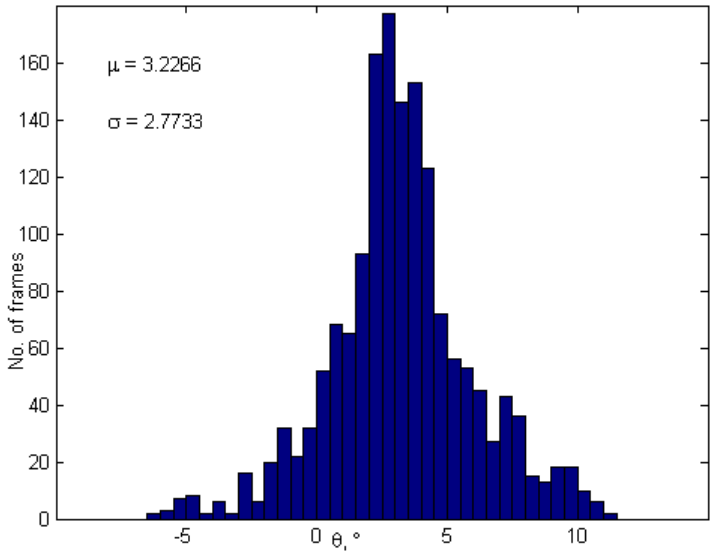


b)

Fig. 5.32. Fluctuation of the tilt angle φ for the moving camera scenario 4:
a) time diagram;
b) statistics.

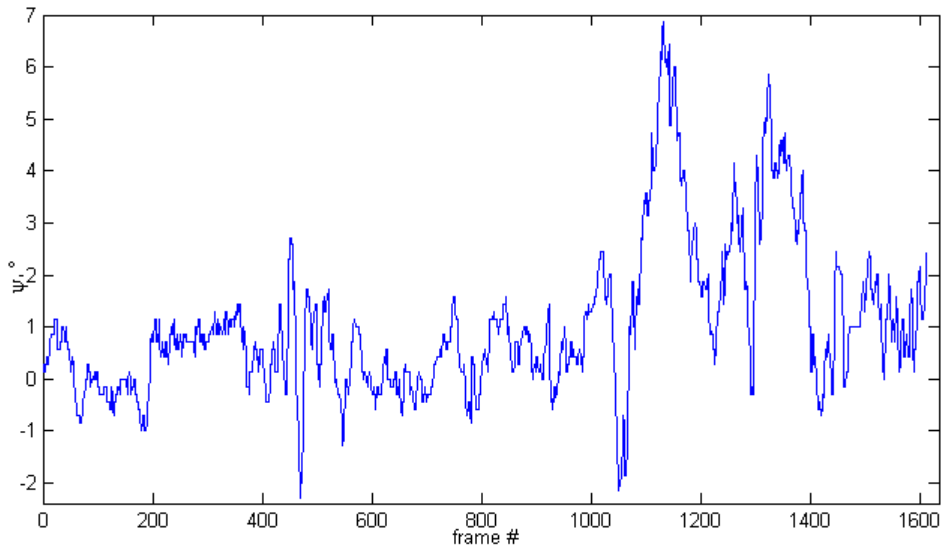


a)

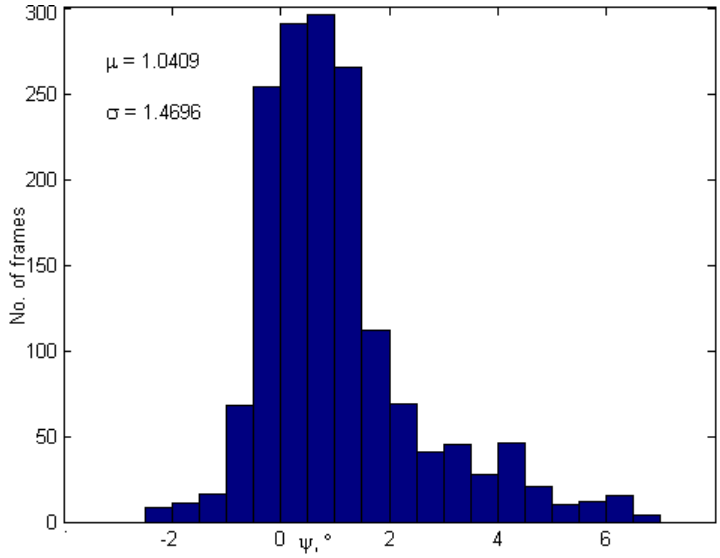


b)

Fig. 5.33. Fluctuation of the pan angle θ for the moving camera scenario 4:
a) time diagram;
b) statistics.



a)



b)

Fig. 5.34. Fluctuation of the roll angle ψ for the moving camera scenario 4:
a) time diagram;
b) statistics.

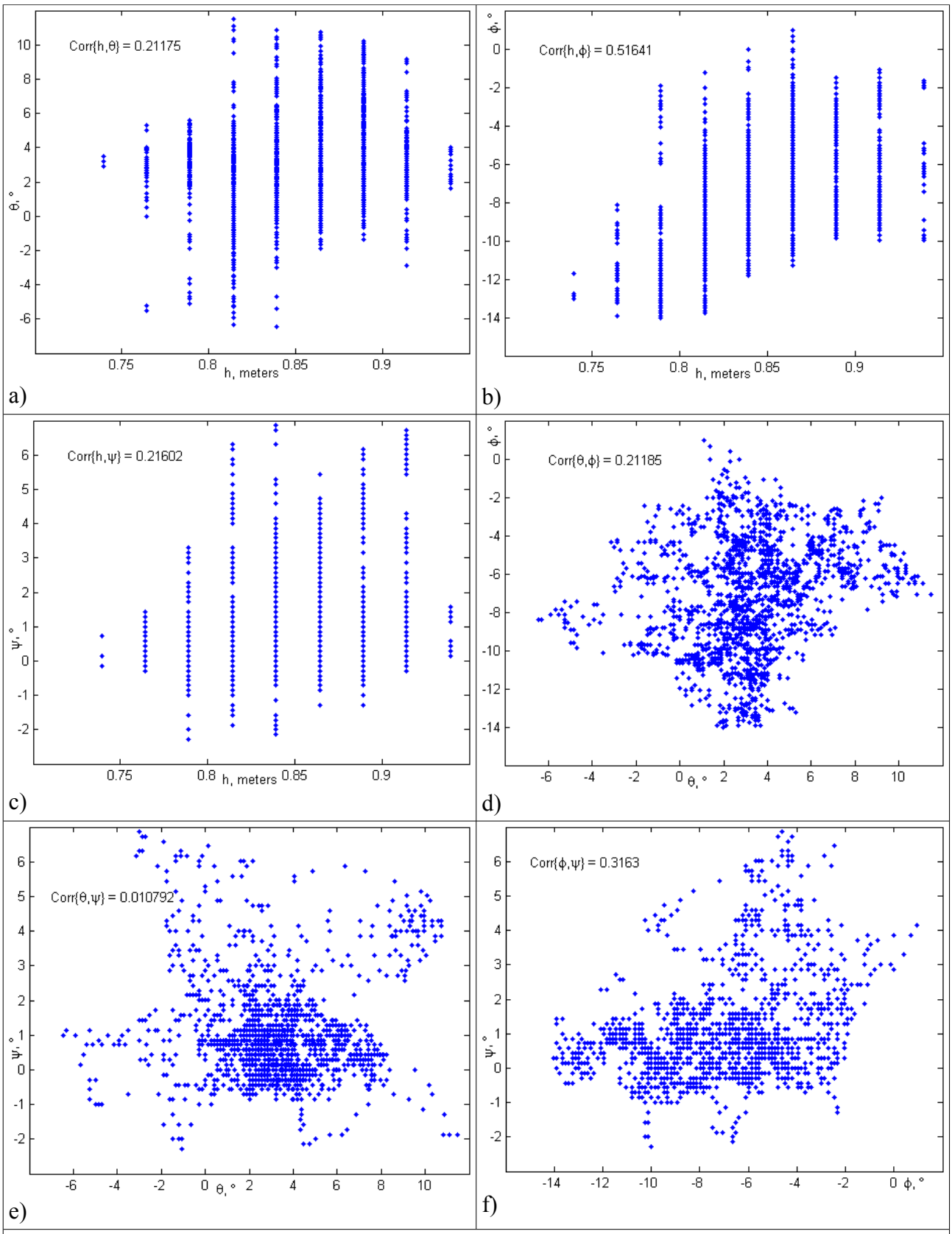


Fig. 5.35. Correspondences between different parameters of the moving camera:
a) camera height h and pan angle θ; b) camera height h and tilt angle φ;
c) camera height h and roll angle ψ; d) pan angle θ and tilt angle φ;
e) pan angle θ and roll angle ψ; f) tilt angle φ and roll angle ψ.

6. Conclusion and future work

This thesis work shows the applicability of different methods of image analysis for the aims of road traffic analysis based on grayscale video sequences.

For the static camera case, object detection and tracking in a world coordinate system was made, instead of the usual solution in the image coordinates. The possibility to identify vehicles in the road plane projection of a video frame based on the typical size and shape of various kinds of vehicles was demonstrated.

For the moving camera case, the applicability of the mean shift tracking method, originally developed for proceeding color video sequences [19], to grayscale video was shown. The problem of false positives produced by the HOG-based detection was overcome by implementing overlapping detections and a voting mechanism.

This thesis work shows the importance of proper detection of road marking elements for the camera calibration. Improving the accuracy of line detection methods in application to the vanishing point detection and precise detection of the ends of dashes in dashed road marking lines can be the task for a future work aiming to increase the performance of the road traffic analysis methods in the case of a moving camera.

In addition, correspondences between the changing camera parameters during the motion of the camera appeared to be another interesting area for research. This can lead to developing some kind of models predicting the camera motion and the corresponding alteration of its parameters, which also can be helpful for the task of road traffic analysis based on videos from a moving camera.

7. References

[1] E.Bas. Road traffic analysis from video. Koc University, Istanbul, Turkey, 2007.

[2] S.Brutzer, B.Höferlin, G.Heidemann. Evaluation of Background Subtraction Techniques for Video Surveillance. Stuttgart University, Germany, 2011.

[3] R.Dahyot. Unsupervised Camera Motion Estimation and Moving Object Detection in Videos. Proceedings of the Irish Machine Vision and Image Processing conference. Trinity College, Dublin, Ireland, 2006.

[4] K.Daniilidis, J.Ernst. Active Intrinsic Calibration Using Vanishing Points. Pattern Recognition Letters, #17, 1996. (Kiel University, Germany)

[5] P.-M. Jodoin, J. Konrad, V. Saligrama, V. Veilleux-Gaboury. Motion detection with an unstable camera. IEEE International Conference on Image Processing, 2008.

[6] N.K.Kanhere. Vision-Based Detection, Tracking and Classification of Vehicles using Stable Features with Automatic Camera Calibration. Clemson University, 2008.

[7] D.Leite, A.Bernardino, J.Gaspar. Auto-Calibration of Pan-Tilt-Zoom Cameras: Estimating Intrinsic and Radial Distortion Parameters. 17th Portuguese Conference on Pattern Recognition. Porto, Portugal, 2011..

[8] http://youtube.com/watch?v=l_eiP8sk_KQ

[9] http://youtube.com/watch?v=gyNdqetEfLQ

[10] http://youtube.com/watch?v=a2Fsg-w5kwg

[11] Chi-Chen Raxle Wang, Jenn-Jier James Lien. AdaBoost Learning for Human Detection Based on Histograms of Oriented Gradients. ACCV 2007, Part I, LNCS 4843, pp. 885–895, 2007.

[12] N.Dalal, B.Triggs. Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.

[13] T.Kobayashi, A.Hidaka, T.Kurita. Selection of Histograms of Oriented Gradients Features for Pedestrian Detection. ICONIP 2007, Part II, LNCS 4985, pp. 598–607, 2008.

[14] R.Hua, J.Collomossea. A Performance Evaluation of Gradient Field HOG Descriptor for Sketch Based Image Retrieval. Preprint accepted to Computer Vision and Image Understanding, February 2013.

[15] F.Suard, A.Rakotomamonjy, A.Bensrhair, A.Broggi. Pedestrian Detection using Infrared images and Histograms of Oriented Gradients. IEEE Intelligent Vehicles Symposium, Tokyo 2006.

[16] T.Furey, N.Cristianini, N.Duffy, D.Bednarski, M.Schummer, D.Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, Vol.16, No.10, Oxford University Press, 2000.

[17] W.Huang, Y.Nakamori, S.Y.Wang. Forecasting stock market movement direction with support vector machine. Computers & Operations Research, #32, 2005, p.2513–2522.

[18] C.W.Hsu, C.C.Chang, C.J.Lin. A Practical Guide to Support Vector Classification. National Taiwan University, Taipei, Taiwan, 2003.

[19] D.Comaniciu, V.Ramesh, P.Meer. Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.25, No.5, May 2003.

[20] E.Dubrofsky, R.J.Woodham. Combining Line and Point Correspondences for Homography Estimation. ISVC 2008, Part II, LNCS 5359, pp. 202–213, 2008.

[21] A.Cavallaro, O.Steiger, T.Ebrahimi. Tracking video objects in cluttered background. IEEE transactions on circuits and systems for video technology. 15(4):575–584, 2005.

[22] P.-M.Jodoin, J.Konrad, V.Saligrama. Modeling background activity for behavior subtraction. ACM/IEEE Int. Conf. Distributed Smart Cameras, Sept. 2008.

[23] A.Ladikos, S.Benhimane, N.Navab. A real-time tracking system combining template-based and feature-based approaches. International Conference on Computer Vision Theory and Applications, Barcelona, Spain, March 2007.

[24] S.Upadhyay, S.K.Singh, M.Gupta, A.K.Nagawat. Linear and non-linear camera calibration techniques. Journal of Global Research in Computer Science. Volume 2, No. 5, April 2011

[25] M.Munderloh, H.Meuel, J.Ostermann. Mesh-based Global Motion Compensation for Robust Mosaicking and Detection of Moving Objects in Aerial Surveillance. IEEE CVPR 2011, 1st Workshop of Aerial Video Processing (WAVP), June 2011

[26] A.Torii, M.Havlena, T.Pajdla, B.Leibe. Measuring camera translation by the dominant apical angle. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08). Anchorage, USA, June 2008.

[27] Y.Sheikh, O.Javed, T.Kanade. Background Subtraction for Freely Moving Cameras. IEEE International Conference on Computer Vision (ICCV), 2009.

[28] R.Dahyott. Unsupervised Camera Motion Estimation and Moving Object Detection in Videos. Proceedings of the Irish Machine Vision and Image Processing conference, pp. 102-109, 2006.

[29] M.Piccardi. Background subtraction techniques: a review. 2004 IEEE International Conference on Systems, Man and Cybernetics.

[30] D.Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), p.91-110, 2004.

[31] T.Kanade, H.Schneiderman. Object Detection Using the Statistics of Parts. International Journal of Computer Vision 56(3), 151–177, 2004.

[32] Hongming Zhang, Wen Gao, Xilin Chen, Debin Zhao. Object detection using spatial histogram features. Image and Vision Computing 24 (2006) 327–341.

[33] H.Bay, A.Ess, T.Tuytelaars, L.Van Gool. SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008.

[34] M.Murshed, A.Ramirez, J.Kim, O.Chae. Statistical binary edge frequency accumulation model for moving object detection. ICIC International, Vol.8, No.7(B), July 2012.

[35] T.Mitsui, H.Fujiyoshi. Object Detection by Joint Features based on Two-Stage Boosting. International Workshop on Visual Surveillance, pp. 1169-1176 (2009).

[36] B.Bose, X.Wang, E.Grimson. Multi-class object tracking algorithm that handles fragmentation and grouping. IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[37] D.J.Salmond, H.Birch. A particle filter for track-before-detect. Proceedings of the American Control Conference, Arlington, VA June 25-27, 2001.

[38] C.M.Buarque Fredrich, R.Q.Feitosa, M.A.Meggiolaro. A Parallel Method For Object Tracking. 17th International Conference on Systems, Signals and Image Processing (IWSSIP) 2010.

[39] Hu Shuoa, Wu Nab, Song Huajunc. Object Tracking Method Based on SURF. Applied Mechanics and Materials, April, 2012.

[40] F.Jurie, M.Dhome. Real Time Robust Template Matching. British Machine Vision Conference 2002, pp. 123–131.

[41] D.Mohr, G.Zachmann. Silhouette area based similarity measure for template matching in constant time. Proceedings of the 6th international conference on Articulated motion and deformable objects, 2010, pp.43-54.

[42] A.Bhattacharyya, M.Gupta, S.Indu. Vehicle Tracking and Speed Estimation using Optical Flow Method. International Journal of Engineering Science and Technology (IJEST), Vol.3, No.1, Jan 2011.

[43] I.Gordon, D.Lowe. What and Where: 3D Object Recognition with Accurate Pose. International Symposium on Mixed and Augmented Reality, 2004.

[44] Ch.Chang, R.Ansari, A.Khokhar. Multiple Object Tracking with Kernel Particle Filter. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).

[45] Yong Rui, Yunqiang Chen. Better Proposal Distributions: Object Tracking Using Unscented Particle Filter. Proceedings of CVPR (2). 2001, 786-793.

[46] Ch.Yang, R.Duraiswami and L.Davis. Fast Multiple Object Tracking via a Hierarchical Particle Filter. IEEE International Conference on Computer Vision, vol. 1, 2005.

[47] B.Z. de Villiers, W.A. Clarke, P.E. Robinson. Mean Shift Object Tracking with Occlusion Handling. Proceedings of the 23rd Annual Symposium of the Pattern Recognition Association of South Africa. Pretoria, 2012.

[48] A. Yilmaz, X. Li, M. Shah. Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence, November 2004, pp. 1531-1536.

[49] A.Yilmaz, X. Li, M.Shah. Object Contour Tracking Using Level Sets. Proceedings of ACCV, Jaju, South Korea, 2004.

[50] P.Melnyk, R.Messner. Log-polar based framework for mobile vehicle tracking with road follower. Defense and Security Symposium, SPIE, Orlando, FL, USA, Apr. 2007.