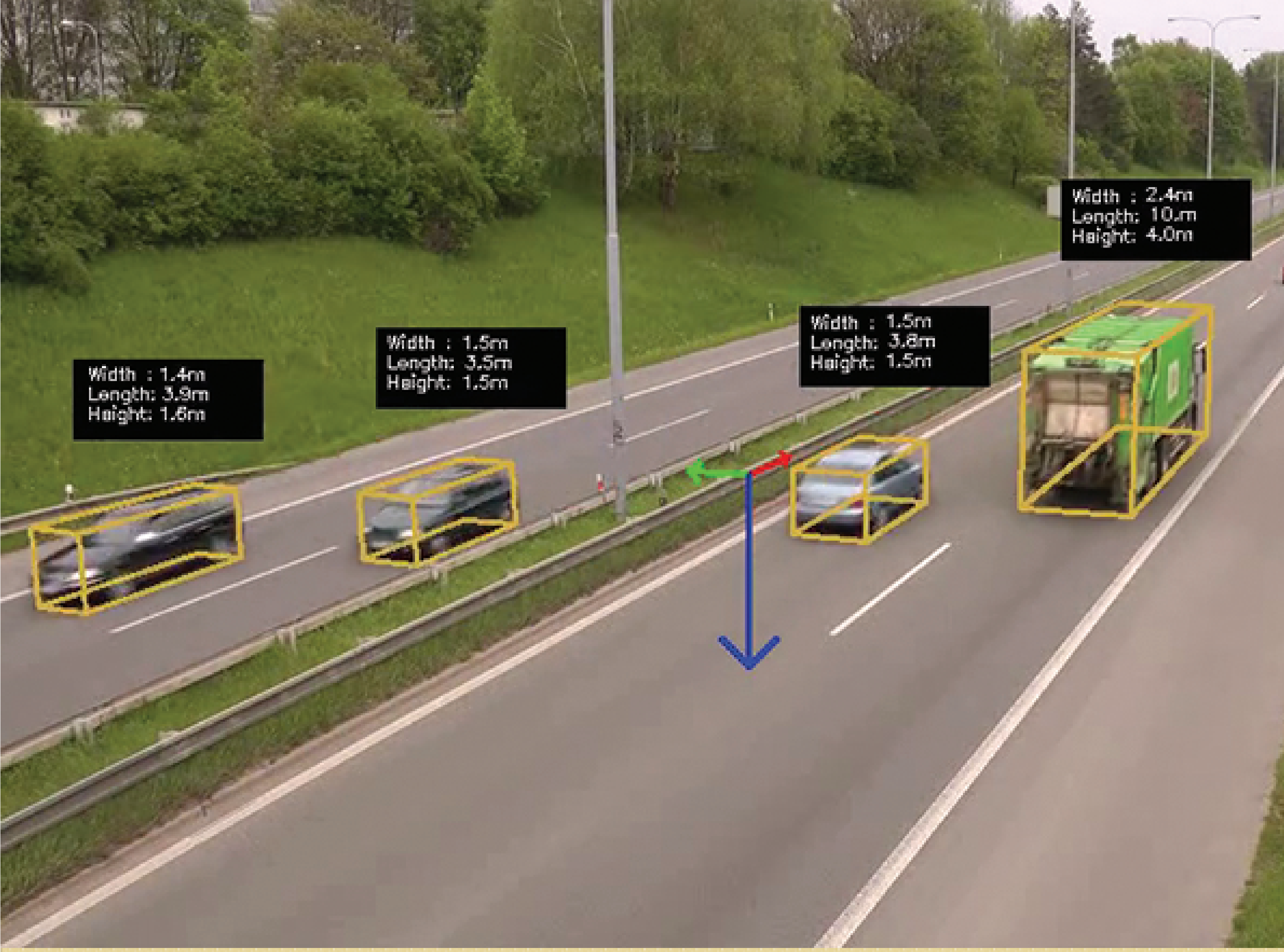


André Miguel Martins Ferreirinha

Fully automatic roadside video camera calibration and traffic analysis

September 2016

· U C ·

Universidade de Coimbra

Department of Electrical and Computer Engineering,
Faculty of Sciences and Technology, University of Coimbra,
3030-290 Coimbra, Portugal.

A Dissertation for Graduate Study in MSc Program
Master of Science in Electrical and Computer Engineering

Fully automatic roadside video camera calibration and traffic analysis

André Miguel Martins Ferreirinha

Research Developed Under Supervision of:
Prof. Jorge Manuel Moreira de Campos Pereira Batista

Jury:
Prof. Doutor Hélder de Jesus Araújo
Prof. Doutor João Pedro de Almeida Barreto
Prof. Doutor Jorge Manuel Moreira de Campos Pereira Batista

September 2016

Work developed in the Institute of Systems and Robotics of the University of Coimbra.

INSTITUTO DE SISTEMAS E ROBÓTICA
UNIVERSIDADE DE COIMBRA

Abstract

Traffic surveillance system development have attracted many interest by the scientific community over the past years. Motivated by improving drivers security, systems capable real-time accident detection, wrong way or others infractions detection are highly desirable. The challenge today is to reduce the number of accidents on the roads or, in case of accident, performing the most quickly possible assistance on the field. Therefore, an Intelligent Transportation System capable of real-time trigger of those events can send automatic notifications to the competent authorities in order to provide quick medical assistance if needed, for example.

Some developed works in the area of traffic surveillance tasks require no camera calibration, however some user inputs are necessary and, in general, no more tasks can be added to system. For a complete designed system, calibration of the traffic surveillance cameras is needed. Some works require some user inputs to perform such calibration. They can be in the form of lane marks, lane width or specific feature points identification on the road, differing from camera location and traffic scenario. Others require the camera to be located in a specific scenario in order to calibrate the camera automatically from pedestrians or even zebra crossings.

The purposed work falls within a Brisa project to design a traffic surveillance system basis consisting on the calibration of the road scene without knowing any prior information about the road itself. The calibration is based on road vanishing point detection. For its detection specific lines are collected in order to retrieve their location both on the image plane and in the world coordinate system. It was concluded that, in almost every tested video stream, both vanishing points can be retrieved in less than 5 minutes of video time. Some previously developed works actually required 2 hours of processed video time to accomplish such calibration.

Such an implementation leads to the ability of performing multiple posterior traffic surveillance tasks in real time on a future complete designed intelligent system. Some traffic analysis tasks were performed to corroborate the implemented calibration algorithm. Vehicles were segmented with a foreground segmentation algorithm and individual bounding box volumetric information was retrieved, up to scale. A lane detection method based on each car bounding box location information was also implemented. Processing speeds were evaluated to confirm the real-time performance of the developed tasks.

Keywords: Camera Calibration, Vanishing Points, Foreground Segmentation, Traffic Analysis, Lane Detection.

Resumo

Nos últimos anos vários trabalhos foram realizados pela comunidade científica sobre sistemas de monitorização de tráfego. Sistemas capazes de detectar acidentes em tempo real, carros em contra mão ou qualquer outro tipo de infracções são bastante procurados, com o objectivo de melhorar a segurança rodoviária dos condutores. O desafio actual incide sobre a diminuição do número de acidentes nas estradas ou, em caso de acidente, providenciar assistência no local o mais rápido possível. Assim, um sistema inteligente de monitorização de tráfego capaz de detectar estas incidências em tempo real pode emitir avisos para as autoridades competentes de forma a providenciar, por exemplo, uma mais rápida assistência médica em caso de necessidade.

Alguns dos trabalhos já desenvolvidos na área do tráfego rodoviário, recorrendo a sistemas de visão, não requerem qualquer calibração de câmaras. Apesar disso estes requerem que o utilizador forneça manualmente dados específicos e, de forma geral, não podem ser acrescentadas mais funcionalidades ao sistema desenvolvido para além da funcionalidade inicial. No entanto, para desenvolver um sistema mais completo e versátil, é necessário efectuar previamente a calibração das câmaras rodoviárias. Alguns trabalhos de calibração destas câmaras já existentes requerem também dados específicos introduzidos pelo utilizador em função do posicionamento da câmara. Geralmente os dados introduzidos são marcas das faixas, largura entre as mesmas ou outros pontos específicos marcados na estrada, que diferem novamente da localização da câmara e do ambiente de tráfego existente. Posteriormente foram também desenvolvidos alguns sistemas capazes de calibrar automaticamente câmaras rodoviárias em cenários específicos, a partir de passadeiras e peões, por exemplo.

O trabalho desenvolvido enquadra-se num projecto da Brisa para desenvolver a base de um sistema completo de monitorização rodoviária. A base consiste no desenvolvimento de um algoritmo de calibração automático de câmeras capaz de calibrar a mesma sem qualquer conhecimento da cena que está a visionar. A calibração baseia-se na detecção de pontos de fuga da estrada. Para efectuar esta detecção são recolhidas algumas linhas específicas com o objectivo de saber a localização dos pontos de fuga tanto no plano da imagem como no plano de coordenadas do mundo. Concluiu-se que, em praticamente todos os vídeos testados, é possível saber a localização exacta dos pontos de fuga nos dois sistemas de coordenadas em menos do que 5 minutos de vídeo. Alguns dos trabalhos desenvolvidos anteriormente requeriam um processamento de cerca de 2 horas de vídeo para conseguir uma calibração semelhante.

Esta implementação permite que sejam desenvolvidas múltiplas tarefas de análise de tráfego em tempo real num sistema inteligente futuro. Algumas dessas possíveis tarefas foram realizadas de forma a corroborar o algoritmo de calibração implementado. Os veículos foram segmentados a partir do plano de fundo com um algoritmo de segmentação do primeiro-plano e foi calculada informação volumétrica individual para cada veículo, com base na caixa que o envolve, com medidas acima da escala. Foi também desenvolvido um método de detecção de faixas baseado na localização de cada uma das caixas envolventes a cada veículo e foram medidos os tempos computacionais de forma a confirmar que as tarefas desenvolvidas conseguem ser realizadas em tempo real.

Palavras-Chave: Calibração de câmaras, Pontos de fuga, Segmentação do primeiro plano, Análise de tráfego, Detecção de faixas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The rapidly increasing variety of digital cameras, computation power, and innovations in video compression standards has led to a recent strong growth of video digital content. In addiction, in the past recent years there has been a fast increase of Internet-connected cameras, not only for personal but also for commercial use. A significant amount of them are being used as part of surveillance camera systems in highways on many countries around the world. The data stored by those systems needs to be analysed in order to assist traffic management and security. According to the World Health Organization [37] 1.25 million people are killed each year on world's roads. A lot of effort has been taken to reduce road traffic deaths, however it is still considered insufficient. Many developed countries have adopted several real-time technology approaches addressing many aspects of road safety. The challenge today is to downward road traffic deaths seen in developed countries and to replicate it in underdeveloped ones.

Consequently, traffic surveillance videos has attracted significant interest over the past years in the computer vision community due to its high variety of possible applications such as vehicle trajectories generation which implies traffic congestion information, accident, stopped vehicle or wrong way detection, speeding vehicles, and so on. On a motorway, alarms can be triggered by Intelligent Transportation Systems (ITS), enabling a fast response and improving drivers security. Also, single vehicle detection and knowing its classification can be useful to take measures to improve safety on specific roads.

Many of the motorway companies hire specific skilled staff in order to deal with acquisition and maintenance of existing traffic monitoring cameras. In general those cameras always require some manual user input parameters [24], [36], [7].

Since highways have already installed several cameras in order to monitor traffic for surveillance and security reasons, the focus of this work is to use video streams from those already installed cameras to provide an automatic roadside calibration solution in a common highway traffic-monitoring scenario. The purposed solution is considered fully automatic since there is no *a priori* informations about the road itself, in the sense that there are no user inputs required. No vehicle appearance model is considered either even they might differ from country to country. The only assumption is the approximately straight shape of the road, being the majority of curves in highways still "straight enough" to meet this assumption. In addiction to this work 3D bounding boxes over passing vehicles with up to scale volumetric information will be constructed.

This work falls within a Brisa project to automatically collect diverse vehicle information in both urban and highway scenarios. Regarding the last ones, the implemented solution could serve as basis of a future traffic monitoring system for real-time classification based on collected vehicles volumetric data.

1.2 Summary of the work developed

The work developed was based upon the usage of a modified scheme of the Cascaded Hough Transform (CHT) where only one Hough space is accumulated - the space of the vanishing points, purposed by Dubská *et al.* [14]. This parameterization of the vanishing points, *via* diamond-space, is based on a Parallel Coordinates Lines (PCLines) parameterization, purposed previously by the same authors [10], capable of mapping the whole real projective plane to a finite space. Since this work goal is to accomplish orthogonal vanishing point detection on real traffic scenarios, specific lines are accumulated in this space, through a voting process, and two points are following backprojected to the two-dimensional plane, corresponding to the first two VP locations on the image plane. Since the three vanishing points are orthogonal, the third VP was computed from the first and second one.

Knowing the 2D vanishing points locations, a traffic analysis algorithm was implemented. The first step of this approach is to isolate moving vehicles from the background. To accomplish it a background subtraction algorithm purposed by Zivkovic [40] was implemented. In order to improve the quality of foreground segmented blobs, a shadow detection algorithm purposed by Horprasert [21] was implemented allowing the majority of car shadows to be removed. After each vehicle segmentation, 3D bounding boxes are constructed around each vehicle and up to scale volumetric information is retrieved, based on [11]. Lastly, a simple lane detector algorithm was implemented based on the constructed car boxes location.

1.3 Outline of the thesis

- Chapter 2: State of the Art. Goes over relevant research related to traffic surveillance in the area as well as purposed computer vision algorithms used for traffic analysis. In addiction, the currently implemented system by Brisa in Portuguese highways is described.

- Chapter 3: Roadside Camera Calibration. It presents the theory behind the implemented camera calibration algorithm. It also presents a brief explanation of its theoretical basis on parallel coordinates.

- Chapter 4: Traffic Analysis. It describes the theory behind the background subtraction algorithm implemented as well as each vehicle 3D box construction with measurements.

- Chapter 5: Development and Results. The implementation basis is described along with example figures as well as the obtained experimental results and respective evaluation of the produced tests.

- Chapter 6: Conclusions and Future Work. It presents the final conclusions from the work and suggested future work in order to design a functional traffic analysis system.

Chapter 2

State of the Art

Vehicle detection with speed measurements and classification has received much attention by the scientific community in past years. Considering that the majority of European motorways have already installed surveillance cameras and since camera costs are constantly decreasing, systems capable of real-time detection and classification are highly desired and a wide variety of approaches have been proposed over the past two decades.

2.1 Traffic Surveillance Systems

Disregarding traffic cameras calibration techniques, one of the first designed traffic systems for traffic analysis purposes was proposed by Beymer *et al.* [3] in 1997. The system was focused on measuring traffic parameters based on corner features detection. It was able to measure vehicle speed, count cars and also to retrieve some dimensions in the frame. However, this system required several user inputs. Before running the system, the user had to specify the region of entrance and exit of the cars as well as four points of correspondence for homography [34] used for parallelization of lane-dividing lines. Dailey *et al.* [9] developed in 2000 an algorithm to estimate traffic speed using a sequence of images from an uncalibrated camera. The algorithm used geometric relationships inherently available in the image, reducing the problem to one-dimensional geometry approaches, leading to a reliable traffic speed estimation from parameters retrieved from the distribution of vehicle lengths. Lately, in 2005, the same authors proposed a new method to compute enough calibration information to establish the necessary scale factor for even most accurate speed measurements [7].

2.1.1 Manual and Automatic Calibration techniques

Several algorithms were introduced for traffic cameras calibration based on road detected vanishing points in order to serve as basis in traffic surveillance systems. The algorithms can be defined as automatic, requiring no user inputs, or manual ones, requiring some specific inputs. In 1997 Bas *et al.* [2] proposed one of the firsts calibration techniques, requiring *a priori* road edges mark as well as camera height and tilt inputs. Based on the detection of one vanishing point (of the traffic flow), the system was capable of computing the focal length and pan of the camera. Later, in 2002, Schoepflin *et al.* [30] proposed an algorithm using motions and edges of passing vehicles to estimate the camera position. Knowing its position, the algorithm then estimates the lane boundaries and the two vanishing points of the lines along the road to calibrate the camera.

In 2006, Lv *et al.* [26] proposed an auto-calibration method for urban traffic scenarios, where from vertical line segments of the same height, detecting the head and feet positions of a walking human in his leg-crossing phases, it was able to calibrate the camera. The algorithm was proven accurate and robust with respect to various viewing angles and subjects. On the same year, Song *et al.* [32] proposed a semi-manual approach to calibrate pan-tilt-zoom cameras in traffic monitoring scenarios. To compute the camera parameters of focal length, tilt angle, and pan angle, the algorithm uses a set of parallel lane markings with input lane width, followed by automatic image processing procedures for finding parallel lane markings.

Zhang *et al.* [38] proposed another method to completely recover camera intrinsic and extrinsic parameters automatically, based on appearance and motion of objects in the video stream, in 2008. However, this method is also considered a manual one since it requires input of the camera height from the ground plane. Later, in 2010, Kanhere *et al.* [25] made a study on the existing calibration methods, both manual and automatic, evaluating the relative strengths of calibration methods, developing novel ones based on the previous algorithms weaknesses. The authors concluded that the generality of methods developed to the date that use a known road length would outperform others in terms of error.

Recently, in 2011, Hodlmoser *et al.* [20] proposed another reliable auto-calibration method based on the observation of pedestrians and zebra crossings on the traffic urban scene. One vanishing point was extracted from zebra crossings whereas another one was extracted from the vertical line of mass of pedestrians on the traffic scene. Since the three vanishing points were orthogonal, the third one was extracted from the other two, allowing to recover the camera intrinsic parameters. For recovering the extrinsic parameters, the camera height was considered as well as the distance between zebra-crossing edges.

2.2 Brisa Portugal

From segmentation processes and stopped vehicles detection [29], [17] to accidents and wrong way detection [18], several implementations were developed over the past years for Brisa Portugal under the Intelligent Traffic Surveillance project [19]. A vehicle velocity estimation [27] was also implemented to automatically estimate vehicle velocity and count the number of vehicles per lane. This method avoided camera calibration not relying on tracking systems. However, this method required the *a priori* inputs of two lengths on the ground plane and required constant highway ground plane width. The calibration and traffic analysis implemented on this thesis could serve as an improve to some already known limitations of the developed systems. The future goal is to develop a full automatic traffic monitoring system where no user inputs were required with the possible lowest number of limitations.

In Portugal, Brisa offers users real-time traffic information with some traffic surveillance cameras location, including a recently captured frame that refreshes every 5-10 minutes in [4]: The majority of those cameras define traffic surveillance scenarios that are suitable for real world implementation of the developed work.

Highway	# Of Cameras	Highway	# Of Cameras
A1	57	**A9**	12
A2	29	**A10**	3
A3	22	**A12**	5
A4	9	**A13**	4
A5	25	**A14**	7
A6	13	**A17**	12
Total # of Cameras	198		

Table 2.1: Table summarizing user available live traffic cameras in Portuguese highways provided by Brisa Portugal.

Chapter 3

Roadside Camera Calibration

In this chapter we present the theory behind the used video camera calibration algorithm. It consists on detecting three orthogonal vanishing points from a traffic scenario and is based on a paper by Dubská *et al.* [15].

In a road scene with a single principal direction of movement it is possible to define the position of the ground plane, relative to the camera, and the vehicle moving direction by three vanishing points [8], [6]. Some examples are illustrated below.

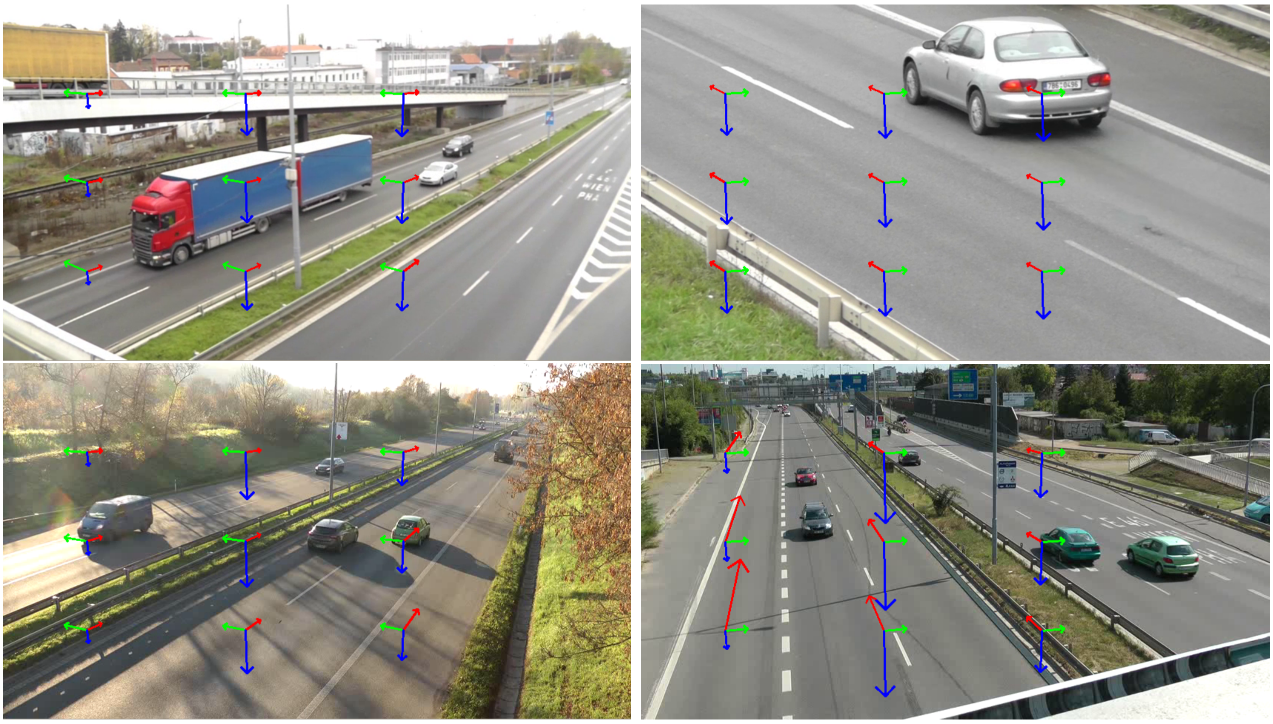


Fig. 3.1: Various scenes in which the algorithm automatically calibrates the camera.

The VP notation is following presented: The VP in the direction of car motion is denoted as the first VP, marked red. The one in the ground plane, perpendicular to the vehicle motion, is the second VP, marked green. Lastly, the third VP is perpendicular to the ground plane and is marked blue.

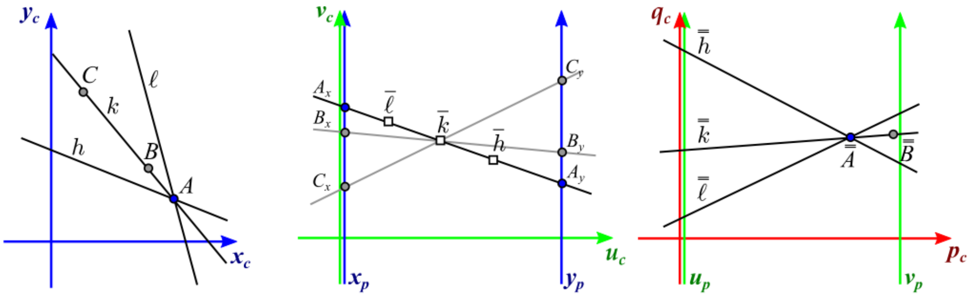
It is important to note that the position of the VP markers on Figure 3.1 are not relevant, being only distributed in a general grid for illustration purposes.

3.1 Vanishing Points Detection

The detection of the vanishing points is done using a recent convenient parameterization of lines for the Hough transform proposed by Dubská [14], using the an accumulation scheme denoted as diamond space. It has the advantage of mapping the whole 2-D projective plane into a finite space by a linear mapping of lines (PCLines) using parallel coordinates.

Parallel coordinates is a coordinate system where the axes are mutually parallel [22]. In this coordinate system a 2-D point [x,y] is represented by a line intersecting the parallel coordinate axis in values x or y.

The parameterization of the vanishing points, *via* diamond space, is based on the PClines line parameterization [10] , purposed previously by Dubská. The method is considered a point-to-line-mapping (PTLM) between two spaces where each point in the Cartesian space is mapped to a line in the space of parallel coordinates. Representation of collinear points intersect at one point in the space of parallel coordinates (Figure 3.2 a) b)).



(a) **(b)** **(c)**

Fig. 3.2: Example of two cascaded PCLines transformations. Figure copyright from [15].

Figure 3.2a represents the original Cartesian space with points and lines while 3.2b represent the same objects of a) defined in parallel coordinates. Simultaneously a new Cartesian coordinate system is also defined (u_c, v_c), in green. Figure 3.2c represents the second transformation to parallel coordinates (u_p, v_p), in green. The new Cartesian coordinate system is also represented (q_c, p_c), in red.

In the above figures subscript p is used for coordinate axes representation in the space of parallel coordinates and subscript c for Cartesian coordinate axes. Both systems of parallel coordinates have their axes oriented toward the same direction. This transformation is denoted $\mathcal{S}$ (straight). Also, accordingly to [14] if the axis in the parallel coordinate system are oriented toward different directions the transformation is denoted $\mathcal{T}$ (twisted). Following this notation, four composite mappings of parallel axis coordinate systems can be constructed, represented in Figure 3.3: $\mathcal{S} \circ \mathcal{S}$, $\mathcal{S} \circ \mathcal{T}$, $\mathcal{T} \circ \mathcal{S}$, $\mathcal{T} \circ \mathcal{T}$.

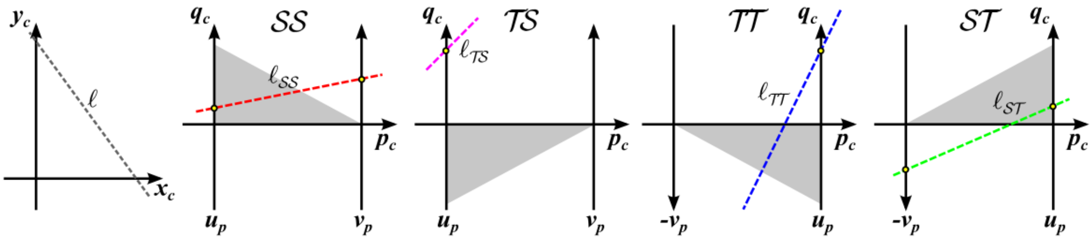


Fig. 3.3: Representation of a line in the different composite parallel coordinate system mappings. Figure copyright from [15]

The dual space (Hough space) covers the whole real projective plane, for a limited original space. Dubská *et al.* attached finite subspaces of two dual spaces one to another where the second space has one of the parallel axes negated. The method is based on the Cascaded Hough Transform (CHT), with modified procedures so that there is only one accumulation space – the diamond space, where vanishing lines are going to be accumulated. Two stages of the CHT are stacked and one intermediate accumulation step is skipped. The CHT first accumulates the edge points in the dual space, detect maxima (corresponding to lines) accumulating these maxima again into the next Hough space. Since in the second Hough space the set of maxima of concurrent lines lie on one line their representations intersect in one point, the vanishing point.

The four finite triangular subspaces represented in Figure 3.3 can be considered as parts that, attached together, form the referred diamond space. Each of the mappings is a transformation of one infinite space to another, but the application of those mappings leads to a finite domain of transformation, each on one quadrant of the real projective plane.

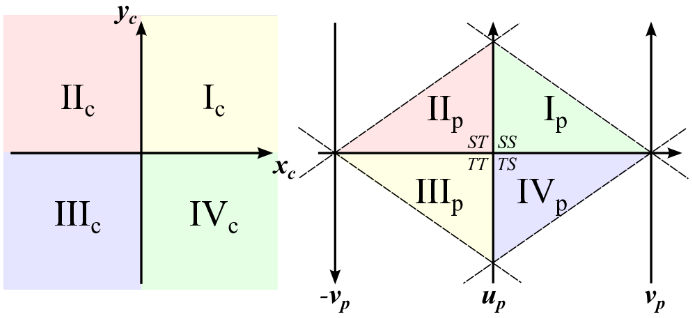


Fig. 3.4: Infinite Cartesian coordinates quadrants and triangle subspaces of the space of parallel coordinates correspondence. Figure copyright from [15]

3.1.1 Accumulation of Lines into Diamond Space

For each VP detection, specific lines are accumulated in the diamond space. A line in the 2-D projective plane is represented by a polyline in the diamond space. Considering that (a, b, c) denotes a line with homogeneous coordinates defined by $ax + by + c = 0$ in the Cartesian space, it can be accumulated in the finite space by a polyline. The polyline is defined by four endpoints which are obtained by Equation 3.1 and Equation 3.2, where sgn denotes non-zero signum function. The parts of the polyline are defined by the consequent points of Equation 3.2 and are segment by segment rasterized into the diamond space. The number of the polyline segments corresponds to the number of quadrants the line passes in the Cartesian space. When a line passes through two quadrants only (horizontal lines, vertical lines, or lines through the origin), one segment always degenerates to a point.

$$\begin{aligned} \alpha &= sgn(ab) \\ \beta &= sgn(bc) \\ \gamma &= sgn(ac) \end{aligned} \tag{3.1}$$

$$(a, b, c) \rightarrow \left[\frac{\alpha a}{c + \gamma a}, \frac{-\alpha c}{c + \gamma a}\right], \left[\frac{b}{c + \beta b}, 0\right], \left[0, \frac{b}{a + \alpha b}\right], \left[\frac{-\alpha a}{c + \gamma a}, \frac{\alpha c}{c + \gamma a}\right] \tag{3.2}$$

Each polyline, defining a line in the image plane, is individually accumulated in the diamond space, incrementing each pixel the line crosses. Such transformation is used as a parameterization in the Hough transform scheme.

Lastly, the diamond space is searched for a global maximum corresponding to the point where the maximum amount of lines pass through. This point is considered the desired vanishing point, with homogeneous coordinates $[p, q, 1]$, is then backprojected to the original 2-D projective plane by Equation 3.3.

$$[p, q, 1] \rightarrow \left[q, sgn(p)p + sgn(q)q - 1, p\right] \tag{3.3}$$

3.1.2 First Vanishing Point Detection

As introduced in the beginning of this chapter, the first VP is the one of direction parallel to the vehicle motion, marked red in Figure 3.1.

Processing of the videostream is done frame by frame. On each frame feature points are collected using Shi and Tomasi's Good Features to Track [31], using the minimum eigenvalue algorithm. Hence, after corner features are detected, a Kanade–Lucas–Tomasi (KLT) feature tracker [35] track those corners in the subsequent frame.

In order for all movement to be stable and detectable by the KLT it should be noted that the processed video is downsampled to 12.5 frames per second (FPS). The amount of skipped frames s can be computed accordingly to Equation 3.4, where FPS denotes the framerate of the original videostream.

$$s = \left[\frac{FPS}{12.5}\right] \tag{3.4}$$

The detected and tracked points exhibiting motion are extended into infinite lines each one defined by an image point (x_t, y_t) and (x_{t+1}, y_{t+1}) corresponding to the positions of features points in each frame (t) and it's subsequent $(t + 1)$. Those fragments of trajectories are all assumed to pass through the first VP.

If the tracked points number are low, only a longer period of observation is needed in order to accumulate a sufficient amount of evidence. Applying equations 3.1 and 3.2, all these lines vote in the diamond space accumulator of the first VP.

Despite the fact that vehicles might be changing lanes more or less randomly, the dominant motion of the vehicles should be very clear. In addiction, the Hough transform is not influenced by the outliers. The diamond space is considered robust to noise providing reliable estimates of the most distinctive point where lines intersect. As the observation and accumulation period increases, the first VP is considered very stable and accurate [15].

Lastly, the space is searched for its global maxima. The most voted point is considered to be the first VP.

3.1.3 Second Vanishing Point Detection

For the second VP detection, marked green on Figure 3.1, another diamond space is used. Starting with the assumption that many vehicle edges coincide with the second VP, they will be filtrated to vote in the accumulation space.

In order to detect edges on moving vehicles a background edge model is used. In each frame the model is updated to deal with shadows and other slow lightning changes. The edge background model stores for each pixel (i, j) the confidence of occurrence of an oriented edge. For an incoming frame I_t of size $w \times h$, the gradient $G(i, j)$ with magnitude m and orientation for each pixel (i, j) is obtained, using responses of horizontal and vertical difference filters K_x and K_y, that will be further described in chapter 5, being the orientation discretized in eight bins. Therefore H_t is created with size $w \times h \times 8$, representing the gradients. Since each plane corresponds to one orientation bin, $H_t(i, j, k)$ is set to magnitude m if the discretized orientation of pixel (i, j) is k, otherwise it is set to 0. The background model is initialized with $B_1 = H_1$ and B_t is then updated with Equation 3.5 where α is a smoothing coefficient closer to 1. For the implementation, a value $\alpha = 0.95$ was used.

$$B_t = \alpha B_{t-1} + (1 - \alpha) H_t \tag{3.5}$$

A background test is performed for every edge pixel with gradient magnitude higher than a predefined threshold τ_1 of an incoming frame. The edge passes the background test if the confidence that the point belongs to the background is lower than a threshold τ_2. Both thresholds will be explained in the implementation section in Chapter 5.

Briefly, for each image pixel $I_t(i, j)$ several conditions are evaluated and if they are satisfied, pixel (i, j) and its orientation $G(i, j)$ defines a line which contains the point (i, j) and is perpendicular to the orientation $G(i, j)$ and it will vote in the accumulator space. The point which could potentially define a line has to meet the following conditions:

- Is has to be detected by a Canny edge detector [5].

- The confidence that the point belongs to the background has to be lower than a predefined threshold (τ_2), meaning it belongs to the foreground (moving vehicles).
- Magnitude of the gradient has to be higher than a predefined threshold τ_1.
- The line must not be directed towards the first VP or must not be vertical, based on the assumption that the scene horizon must be approximately horizontal, with level of tolerance ($\pm 45^o$).

The first vanishing point is used for constraining the position of the second vanishing point. Edges that, extended to infinite lines, are passing close to the first VP, already known from subsection 3.1.2, are excluded from further processing.

Edges passing the previously stage vote in the diamond space and the second vanishing point is again detected by the global maximum of the diamond space. However, the global maximum is searched in a masked diamond space as the location of the second VP is constrained by the first VP. The line which contains the principal point (assumed in the middle of the image) and is perpendicular to the line defined by the principal point and the first VP has to separate the second VP from the first one. If the first and second vanishing points would be on the same side of the line, the expression under square-root of Equation 3.6 would be negative.

3.1.4 Third Vanishing Point and Focal Length Computation

Assuming the principal point is in the center of the image, it is possible to compute the focal length and the third VP point as well as world coordinates of all vanishing points.

Let the first VP be denoted as $U = (u_x, u_y)$, the second VP as $V = (v_x, v_y)$ and the principal point as $PP = (p_x, p_y)$. The focal length f is computed accordingly to Equation 3.6.

$$f = \sqrt{-(U - PP) \cdot (V - PP)} \tag{3.6}$$

The world coordinates of the vanishing points are denoted U', V' and W', for the first, second and third VP respectively. The world coordinates of the first and second VP are computed accordingly to Equation 3.7 and Equation 3.8. Considering $PP' = (p_x, p_y, 0)$ it is possible to compute the world coordinate of the third VP with Equation 3.9.

$$U' = (u_x, u_y, f) \tag{3.7}$$

$$V' = (v_x, v_y, f) \tag{3.8}$$

$$W' = (U' - PP') \times (V' - PP') \tag{3.9}$$

Lastly, the world coordinates of the third VP can be transformed into image coordinates accordingly to Equation 3.10 as $W' = (w'_x, w'_y, w'_z)$.

$$W = (\frac{w'_x}{w'_z} f + p_x, \frac{w'_y}{w'_z} f + p_y) \tag{3.10}$$

It is following presented the accomplished vanishing point detection from a road scene as well as the constructed diamond spaces for first and second vanishing points.

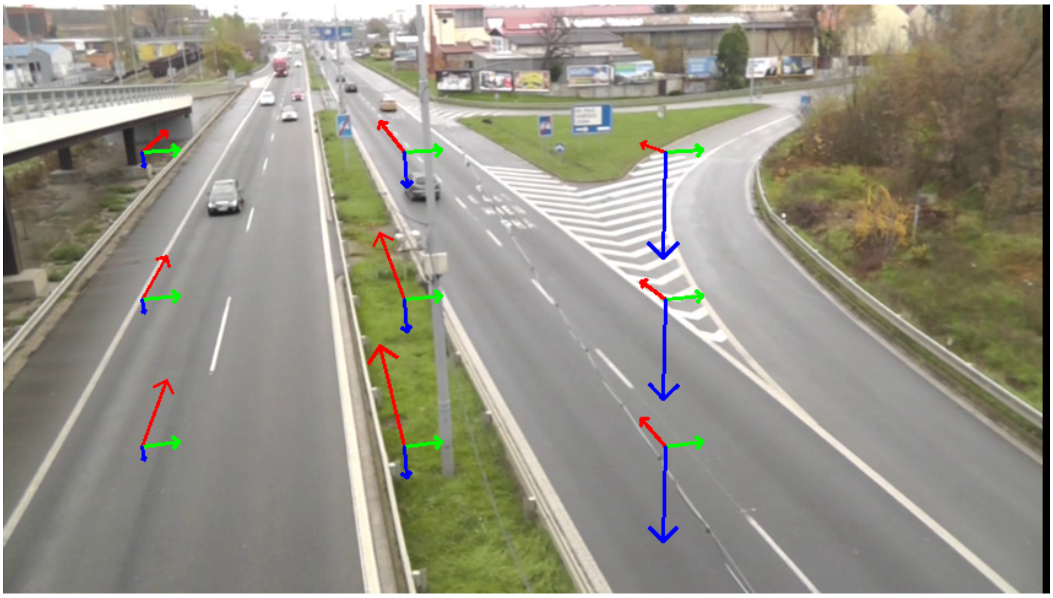
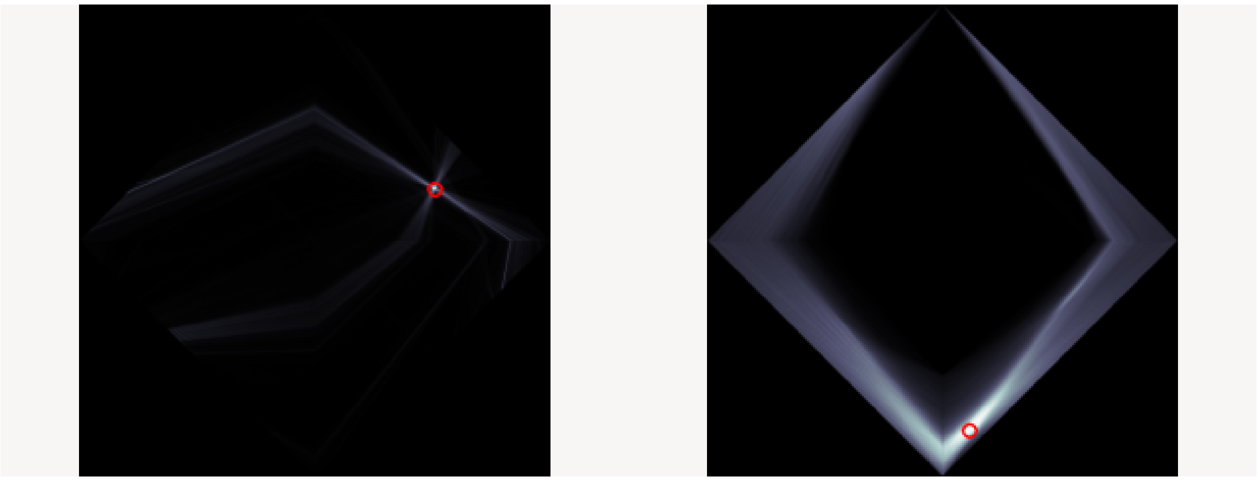


Fig. 3.5: Vanishing vectors grid representation in a real traffic scene.

3.2 Camera Calibration from the Vanishing Points

The camera projection matrix P transforms a world point $[x_w, y_w, z_w, 1]$ into a point in the image plane $[x_p, y_p, 1]$, accordingly to Equation 3.11 .



(a) (b)

Fig. 3.6: Diamond spaces for the Figure 3.5 first (a) and second (b) vanishing points. The red circle signalizes the most voted point in each accumulation space.

$$\lambda_p \left[x_p, y_p, 1\right]' = P \left[x_w, y_w, z_w, 1\right]' \tag{3.11}$$

Camera projection matrix P can be decomposed into the orientation and position of the camera relative to a the world coordinate system, i.e, the extrinsic parameters 3 × 3 rotation matrix R and the 3 × 1 translation vector T.

$$P = K[R\ T] \tag{3.12}$$

K is the camera intrinsic parameter matrix in the form:

$$K = \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

Here, α_u and α_v represent the focal length of the camera in terms of pixel dimensions in the u and v directions respectively, where γ represents the skew parameter and $(u_0, v_0)'$ are the coordinates of the principal point.

It is assumed zero skew ($\gamma = 0$), unit aspect ratio ($\alpha_u = \alpha_v = f$) and , as stated before, the principal point in the image plane centre. That's also a general assumption by many authors for traffic surveillance cameras [39], [24], [28], [1]. Consequently, the intrinsic parameter matrix 3.13 is simplified. The rotation matrix R is fully defined by the positions of the orthogonal vanishing points. Due to the fact that points at infinity correspond to the three orthogonal directions, it is possible to derive simple constraints on the elements of the projection matrix with Equation 3.14, where $U = (u_1, v_1)'$, $V = (u_2, v_2)'$, $W = (u_3, v_3)'$ are coordinates of the three orthogonal VP in the image plane and $\lambda_1, \lambda_2, \lambda_3$ are initially unknown scaling factor parameters.

$$\begin{bmatrix} \lambda_1 u_1 & \lambda_2 u_2 & \lambda_3 u_3 \\ \lambda_1 v_1 & \lambda_2 v_2 & \lambda_3 v_3 \\ \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix} = P \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.14}$$

Accordingly to Equation 3.12, 3.14 can be expressed in terms of K and R:

$$\begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} = KR \tag{3.15}$$

Since the rotation matrix R satisfies $R \cdot R^\top = 1$, Equation 3.15 can be rearranged to derive constrains on K and the unknown scaling parameters λ_i:

$$\begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix} \begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{bmatrix} = KK^\top \tag{3.16}$$

where,

$$KK^T = \begin{bmatrix} \alpha_u^2 + u_0^2 & u_0 v_0 & u_0 \\ u_0 v_0 & \alpha_v^2 + v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{bmatrix} \tag{3.17}$$

Equation 3.16 can be solved to recover the three intrinsic camera parameters (f, u_0, v_0) but also the three unknown scale parameters $\lambda_1, \lambda_2, \lambda_3$:

$$\begin{aligned} \lambda_1^2 &= \frac{(v_0 - v_3)(u_2 - u_3) - (u_0 - u_3)(u_2 - u_3)}{(v_1 - v_3)(u_2 - u_3) - (u_0 - u_3)(v_2 - v_3)} \\ \lambda_2^2 &= \frac{(v_1 - v_3)(u_0 - u_3) - (u_1 - u_3)(v_0 - v_3)}{(v_1 - v_3)(u_2 - u_3) - (u_1 - u_3)(v_2 - v_3)} \\ \lambda_3^2 &= 1 - \lambda_1^2 - \lambda_2^2 \end{aligned} \tag{3.18}$$

With K and $\lambda_1, \lambda_2, \lambda_3$ it is possible to solve R matrix in Equation 3.15.

$$R = \begin{bmatrix} \lambda_1(u_1 - u_0)/\alpha_u & \lambda_2(u_2 - u_0)/\alpha_u & \lambda_3(u_3 - u_0)/\alpha_u \\ \lambda_1(v_1 - v_0)/\alpha_v & \lambda_2(v_1 - v_0)/\alpha_v & \lambda_3(v_3 - v_0)/\alpha_v \\ \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix} \tag{3.19}$$

For the translation vector T additional information is required. Some authors can derive it from the height of the camera to the ground [39] or from a known distance of two projected points, but it was not a goal of this thesis to accomplish it.

Concluding, as the focal length and third vanishing point are computed, the camera is calibrated up to scale, since it is not possible to determine how far the camera is from the road plane.

Chapter 4

Traffic Analysis

In this chapter it is described the theory behind the construction of vehicle three-dimensional (3D) bounding boxes from an incoming videostream. To accomplish it the scene needs to be previously calibrated, i.e, all the three vanishing points already detected following the procedures of the previous chapter.

The first step of this approach is to extract single vehicle silhouettes from incoming video frames. To accomplish it a foreground segmentation is necessary. In the following sections the theory behind the implemented algorithm is described.

4.1 Gaussian Mixture Model for Background Subtraction

The capability of extracting moving vehicles from a video sequence is the backbone of this 3D construction approach. The main idea of background subtraction is to subtract the current frame from a reference created one, ideally acquired from a static background during a period of time, leaving only new objects on the scene. In the majority of the cases it is not possible to have such an image due to lightning changes, wind or any other effect.

Several algorithms were introduced for solving this issue and the implementation is based on Stauffer [33], where the idea of adaptive background modelling with Mixture of Gaussians is described, Zivkovic's [40], [41] where some improvements were made to Stauffer's approach, and Horprasert's [21] that previously developed a shadow elimination procedure that combined with Zivkovic provides higher accuracy results in traffic surveillance scenarios.

Based on the assumption that frames are collected by a stationary video camera, the main idea is to model the background color of each pixel in the scene with a Mixture of Gaussians, used to differentiate the background color of the pixel from the foreground ones. Three dimensional Gaussians were used, necessary to describe three colors: red, green, blue. Since the Gaussians are not in a form of three-dimensional Gaussian probability density function, Equation 4.1, the algorithm uses the covariance matrix Σ in the form $\Sigma = \sigma^2 I$. The standard deviation in all directions is represented by σ and I is a 3×3 identity matrix.

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{3}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right) \tag{4.1}$$

Only the procedure applied to one pixel will be described as the others are processed the same way. Let $x^{(t)}$ be the value of the pixel at time t and α be the learning rate, which specifies how fast a new cluster will be adopted. The parameters which will be held for each Gaussian are the following: π_m, the weights which are non-negative and sums up to one; μ_m the mean value; σ_m^2 the variance. Maximally M gaussians will be held for each pixel. The pixel is processed as follows:

- The Gaussian with the biggest weight π_m and the squared Mahalanobis distance $D_m^2(x^{(t)})$, Equation 4.3, smaller than the threshold value T_g is found. The reasonable value for this threshold is 9, implying that the Gaussian is considered close to the value if the value $x^{(t)}$ is less than 3σ far from the mean. The closest Gaussian has the ownership variable value ($o_m^{(t)}$) set to 1 as the others have this value set to 0.
- Gaussian parameters are updated accordingly to equations 4.4, 4.5 and 4.6.

$$\delta_m = x^{(t)} - \mu_m^{(t)} \tag{4.2}$$

$$D_m^2(x^{(t)}) = \delta_m^T \delta_m / \sigma_m^2 \tag{4.3}$$

$$\pi_m \leftarrow \pi_m + \alpha(o_m^{(t)} - \pi_m) - \alpha c_T \tag{4.4}$$

$$\mu_m \leftarrow \mu_m + o_m^{(t)} (\alpha / \pi_m) \delta_m \tag{4.5}$$

$$o_m^2 \leftarrow o_m^2 + o_m^{(t)}(\alpha/\pi_m)(\delta_m^T\delta_m - \sigma_m^2) \tag{4.6}$$

- If there is no close Gaussian found, it is necessary to create a new one from the current pixel value $x^{(t)}$ with parameters $\pi_{m+1} = \alpha$, $\mu_{m+1} = x^{(t)}$ and $\sigma_{m+1}^2 = \sigma_0^2$.

- If there are more Gaussians than the maximum amount M, the weakest one is removed. The weights are normalized so that the the sum is equal to 1 after the process.

- For a given pixel the squared Mahalanobis is used in order to determine if the pixel should be considered as background or foreground. If a Gaussian with a distance smaller than a predefined threshold c_{thr} is found, the pixel is considered to be part of the background. If no Gaussian is found, the pixel is considered to be foreground.

The term αc_T in Equation 4.4 is responsible for decreasing the weight π_m of the Gaussian if there was no pixel with the color corresponding the Gaussian. Zivkovic [40] suggested an improvement to this term. If after the subtraction of Equation 4.4 the weight of the Gaussian drops below zero, the Gaussian is removed from the pixel's model. This approach brings the ability to select the proper number of Gaussians to each pixel, considering the upper limit M, leading more dynamics to the number of Gaussians used for modelling the background of the pixel. This approach decreases the processing speed time of one image, improving performance of the algorithm.

4.1.1 Shadow Detection

Since none of Zivkovic's described algorithms [40], [41] have the capability of shadow detection, moving vehicles segmentation were improved with Horprasert's algorithm [21] in order to increase the precision of motion detection. The vehicle is detected with a much higher accuracy, as Figure 4.4 shows.

Let only one pixel be described as the other pixels are processed the same way. This algorithm works upon the background modelled by the Mixture of Gaussians (MG), but it will be explained with only one Gaussian, without loss of generality. It will be sufficient if the shadow is detected by one Gaussian from the MG to consider the pixel as part of the shadowed background.

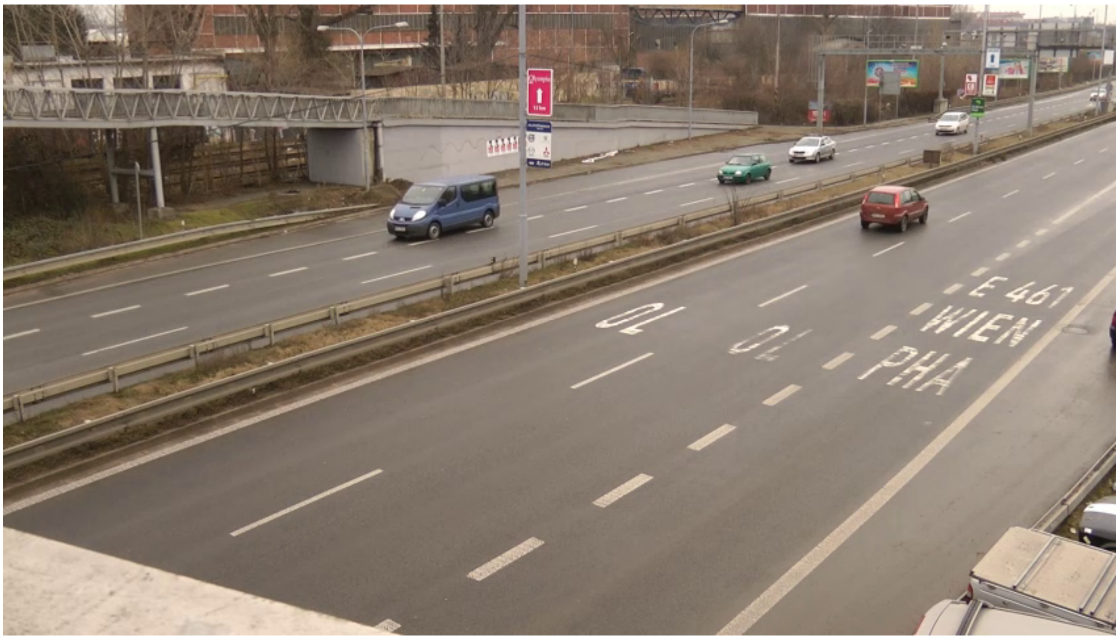


Fig. 4.1: Original n^{th} frame from a incoming videostream.

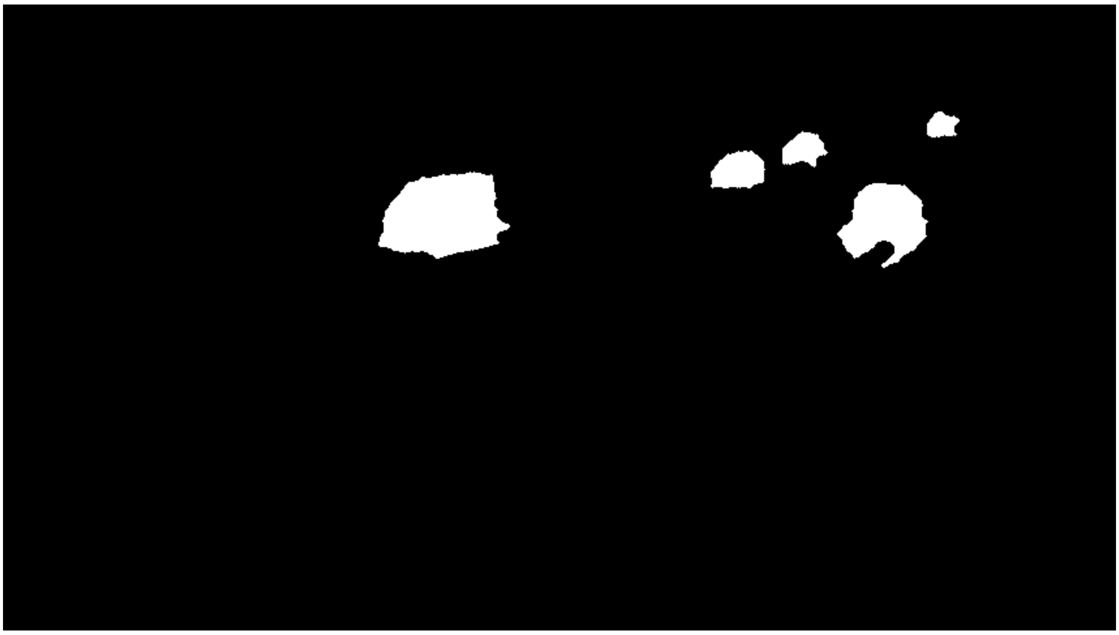


Fig. 4.2: Example of the described background subtraction algorithm applied to Figure 4.1 frame without shadow detection.

The main idea is to decompose mean value distortion μ of the Gaussian from x, the current pixel value. It is done considering two components in the three-dimensional color space (RGB): brightness distortion β and chromaticity distortion CD, as shown in Figure 4.3.

Brightness distortion β is a scalar that brings the observed color close to the expected chromaticity line. It is computed accordingly to Equation 4.7 by minimizing distance of point x from

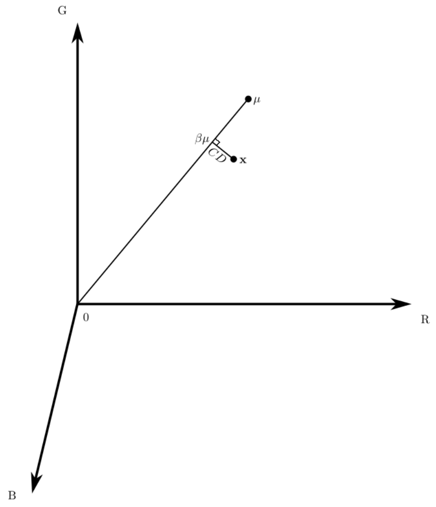


Fig. 4.3: Distance between the pixel color mean value μ and the current pixel value x decomposed into brightness distortion β and chromaticity distortion CD

the line defined by the origin $(0, 0, 0)$ and μ. $\beta\mu$ is a point in the 3D space corresponding to the orthogonal projection of x onto the line defined by the origin and μ.

$$\beta = \arg\min_{\beta \in \mathbb{R}} \|x - \beta\mu\| \tag{4.7}$$

Chromaticity distortion CD is defined as the orthogonal distance between the observed color and the expected chromaticity line. It is also a scalar and can be calculated with Equation 4.8. x, μ and $\beta\mu$ form a right triangle.

$$CD = \|x - \beta\mu\| \tag{4.8}$$

The standard deviation is also taken into account and the values of brightness and chromaticity distortion are calculated with equations 4.9 and 4.10. Since the Mixture of Gaussians described in the previous section uses $\sigma = \sigma_R = \sigma_G = \sigma_B$, they can be simplified.

$$\beta = \frac{\dfrac{x_R\mu_R}{\mu_R^2} + \dfrac{x_G\mu_G}{\mu_G^2} + \dfrac{x_B\mu_B}{\mu_B^2}}{\left(\dfrac{\mu_R}{\sigma_R}\right)^2 + \left(\dfrac{\mu_G}{\sigma_G}\right)^2 + \left(\dfrac{\mu_B}{\sigma_B}\right)^2} \tag{4.9}$$

$$CD = \sqrt{\left(\frac{x_R - \beta\mu_R}{\sigma_R}\right)^2 \left(\frac{x_G - \beta\mu_G}{\sigma_G}\right)^2 \left(\frac{x_B - \beta\mu_B}{\sigma_B}\right)^2} \tag{4.10}$$

Brightness distortion β and chromaticity distortion CD need to be normalized through the computation of the square root mean with equations 4.11 and 4.12. These are root mean square values corresponding to the length of the side in the right triangle in Figure 4.3. Equations 4.11 and 4.12 can be computed continuously, without any history of the β and CD values. After a and b values are computed the normalized brightness and chromaticity distortion, $\widehat{\beta}$ and $\widehat{CD}$ respectively, are calculated accordingly to equations 4.13 and 4.14.

$$a = \sqrt{\frac{\sum_{i=0}^{N} (\beta - 1)^2}{N}} \tag{4.11}$$

$$b = \sqrt{\frac{\sum_{i=0}^{N} CD^2}{N}} \tag{4.12}$$

$$\widehat{\beta} = \frac{\beta - 1}{a} \tag{4.13}$$

$$\widehat{CD} = \frac{CD}{b} \tag{4.14}$$

In order to get a final decision about the pixel class, several thresholds are used. The whole procedure is described in Equation 4.15. The first threshold, τ_{CD}, specifies the maximal chromaticity distortion which one pixel might have in order to not be considered as foreground. If $\widehat{\beta}$ is smaller than $\tau_{\beta lo}$, the pixel is also considered as foreground. If $\widehat{\beta}$ and $\widehat{CD}$ are not within those limits, the pixel is considered as background, which can be normal, shadowed or lighter. There is a pair of thresholds $\tau_{\beta 1}$ and $\tau_{\beta 2}$ specifying values of $\widehat{\beta}$ for which the pixel is considered as background.

$$\text{pixel} = \begin{cases} \text{foreground} & \text{if } \widehat{CD} > \tau_{CD} \quad \vee \quad \widehat{\beta} < \tau_{\beta lo} \\ \text{background} & \text{if } \widehat{\beta} < \tau_{\beta 1} \wedge \widehat{\beta} > \tau_{\beta 2} \\ \text{shadowed} & \text{if } \widehat{\beta} < 0 \\ \text{lighted} & \text{otherwise} \end{cases} \tag{4.15}$$

If 4.15 processes only pixels classified as foreground by the Mixture of Gaussians, the rule for the background can be eliminated. Only the two thresholds $\tau_{\beta lo}$ and τ_{CD} are required.

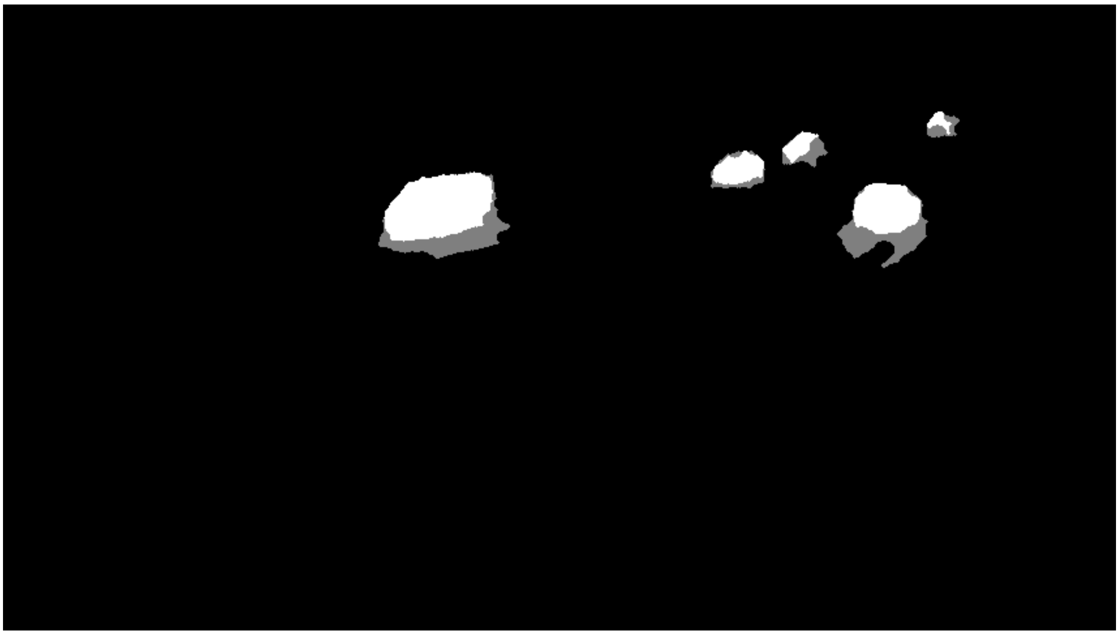


Fig. 4.4: Example of the described background subtraction algorithm with shadow detection applied to Figure 4.1 frame. Shadow pixels in grey, foreground pixels in white, and background pixels in black.

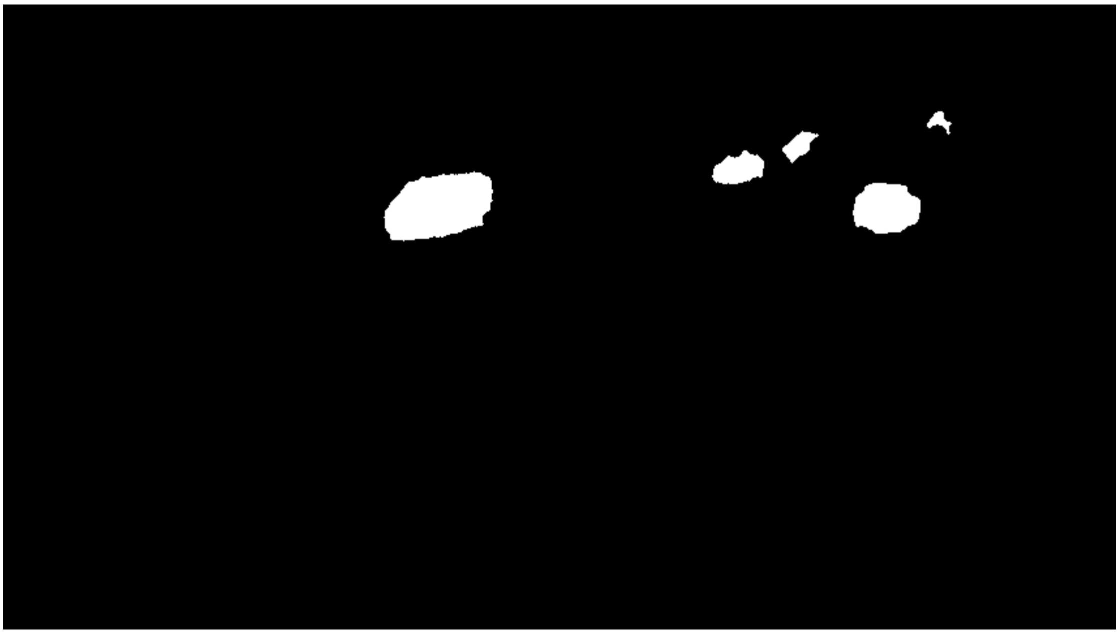


Fig. 4.5: Example of the shadow elimination applied to Figure 4.1 frame. Foreground pixels in white, background pixels in black.

4.2 Three-Dimensional Bounding Boxes

After the three vanishing points detection and moving vehicles segmentation, three-dimensional bounding boxes can be computed. This approach is based on the fact that vehicles tend to have edges very stable and reliable, extracted in the foreground segmentation process. The three-dimensional bounding box is a cuboid that surrounds the detected vehicle. Several perspectives of the bounding box construction can be seen on Figure 4.6. The box vertices will be further explained in this section. The cuboid also defines vehicle dimensions and the base position can be used further for more precise detection of lanes.

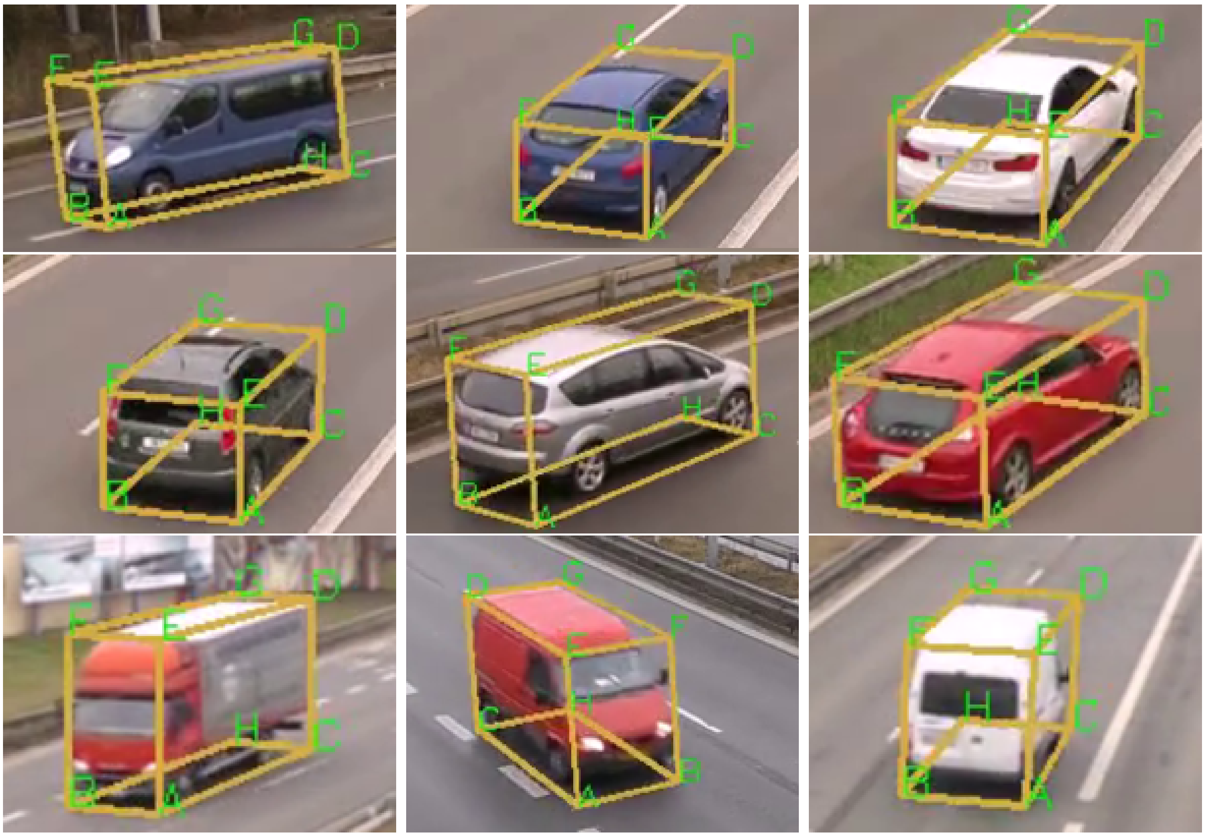


Fig. 4.6: Various close car captions on different scenes where 3D boxes are drawn.

4.2.1 Tangent Computation

In order to construct individual three-dimensional bounding boxes, segmented blobs from the foreground detection need to be processed. The blob contour is extracted and it is necessary to compute tangent lines to it from all the vanishing points of the scene. The tangent computation needs to be effective as several lines are computed for each vehicle in each video frame.

Firstly, it is necessary to distinguish the left tangent line from the right tangent one. Considering a vanishing point P, the left tangent is the line on the left side departing from point P. Similarity, the right tangent line is the right one. The algorithm for tangent computation works as follows:

- Considering a car contour C, the algorithm iterates over all points $I \in C$ and computes the angle between the line defined from point P and point I and the positive half of the x axis of the image.

- The point I_{left} is the point with the minimal angle corresponding to the left tangent point. The point I_{right} is the point with the maximum angle.
- As the angles might be returned in the interval $\{-\pi, \pi\}$ it is necessary to perform some additional transformations to the computed angles if point P from which the lines are tested is on the right side of the contour. Summarizing, if point P has a higher x coordinate and the angle is below zero, 2π must be added to the computed angle, transforming the interval to $\{0, 2\pi\}$.

4.2.2 Three-Dimensional Bounding Box Computation

After the tangent lines are detected, the individual three-dimensional bounding boxes can be computed. Considering that U,V and W denote the first, second and third vanishing point respectively, let us denote the left and right tangent line from those vanishing points as t^U_{left}, t^U_{right}, t^V_{left}, t^V_{right}, t^W_{left} and t^W_{right}.

The process of finding the bounding box vertices is described below. It should be noted that there are slight differences on these vertices calculation when the configuration of the VPs is different (see vertices difference between Figure 4.6 bottom centre and the other ones). As there are only minor changes, both processes will be described. There are other possible scenarios, but since the videos of the used dataset have two scenarios only, only those will be described.

Starting by Figure 4.6 bottom centre vehicle scenario, the first step it to determine locations of points A,B and C. Those are points of the vehicle base witch are not hidden behind the vehicle. They are computed by equations 4.16, 4.17 and 4.18:

$$A = t^U_{right} \bigcap t^V_{left} \tag{4.16}$$

$$B = t^V_{left} \bigcap t^W_{right} \tag{4.17}$$

$$C = t^U_{right} \bigcap t^W_{left} \tag{4.18}$$

Secondly, points D, E and F are located. Points D and F can be computed accordingly to equations 4.19 and 4.20:

$$D = t^{V}_{right} \bigcap t^{W}_{left} \tag{4.19}$$

$$F = t^{U}_{left} \bigcap t^{W}_{right} \tag{4.20}$$

In order to ensure that the whole contour of the vehicle will be enclosed within the drawn drawing box, the position of point E derive from point D or F. Two $|AE|$ distances are computed with equations 4.21 and 4.22. The larger distance is the one that defines point E (Equation 4.23).

$$E_D = \overleftrightarrow{DU} \bigcap \overleftrightarrow{AW} \tag{4.21}$$

$$E_F = \overleftrightarrow{FV} \bigcap \overleftrightarrow{AW} \tag{4.22}$$

$$E = \begin{cases} E_D & \text{if } |AE_D| \geqslant |AE_F| \\ E_F & \text{if } |AE_D| < |AE_F| \end{cases} \tag{4.23}$$

Lastly, points G and H are computed with equations 4.24 and 4.25.

$$G = \overleftrightarrow{DV} \bigcap \overleftrightarrow{FU} \tag{4.24}$$

$$H = \overleftrightarrow{CV} \bigcap \overleftrightarrow{BU} \tag{4.25}$$

There might be four possible VP scenarios, however, only two different scenarios were found in the dataset used. For the other scenario, visualized on the Figure 4.6 top row, middle row and left and right bottom row figures, equations 4.26 through 4.30 are applied for finding the respective points.

$$A = t^{U}_{left} \bigcap t^{V}_{right} \tag{4.26}$$

$$B = t^{V}_{right} \bigcap t^{W}_{left} \tag{4.27}$$

$$C = t^{U}_{left} \bigcap t^{W}_{right} \tag{4.28}$$

$$D = t^{V}_{left} \bigcap t^{W}_{right} \tag{4.29}$$

$$F = t^{U}_{right} \bigcap t^{W}_{left} \tag{4.30}$$

Points E, G and H are computed the same way with equations 4.23, 4.24 and 4.25. Since it is important that the bounding boxes surround the vehicle tightly for dimensions extraction this method is considered very reliable.

4.2.3 Vehicle Dimensions Extraction

In order to extract each vehicle real dimensions (width, length and height) it is necessary to compute real world points positions in the world coordinate system (WCS). Computing image plane points A, B, C, D in the WCS leads us to the refereed distance measurements corresponding to the box dimensions. However, as the camera is calibrated from the vanishing points only up to scale, dimensions are not retrieved in SI units, but in relative units. Those units need to be converted to meters with an unknown scale factor λ that has to be determined.

Firstly, points need to be projected to the road ground plane. Since the ground plane $\mathcal{G}$ parameters are unknown it is necessary to compute them first. Let us consider a three-dimensional coordinate system with principal point $P_w = [p_x, p_y, f]$, where $[p_x, p_y]$ are the image plane position of the principal point and f is the focal length previously computed. The normal vector $\vec{n}$ of the ground plane $\mathcal{G}$ is computed accordingly to Equation 4.31, where W' is the world coordinate of the third vanishing point. Considering $\vec{n} = [a, b, c]$, the general form of the ground plane can be written as in Equation 4.32, with a missing d parameter.

$$\vec{n} = W' - P_w \tag{4.31}$$

$$\mathcal{G} \rightarrow ax + by + cz = d \tag{4.32}$$

Therefore, an arbitrary value is chosen for d. When the equation of the ground plane is fully known it is possible to project a 2D point $X = [x, y]$ on the ground plane with Equation 4.33, where $X_{3D} = [x, y, f]$ and $\mathcal{O} = [p_x, p_y, 0]$, corresponding to the camera location on the three-dimensional coordinate system (Figure 4.7). Therefore, the relative distance between two points X_1 and X_2 can be computed accordingly to Equation 4.34. It should be noted that only world coordinates on the ground plane can be computed this way, i.e. only box points A, B and C corresponding to the box interest vertices to calculate relative width and length.

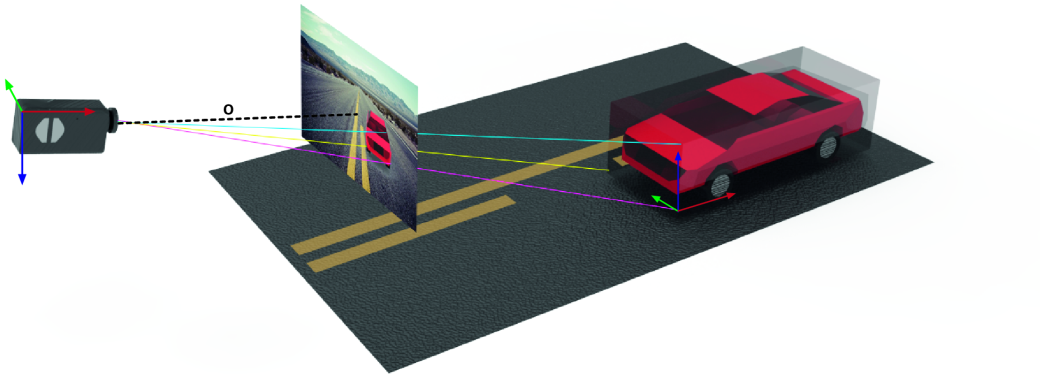


Fig. 4.7: Example of a traffic scene with a camera observing a vehicle and its coordinate system representation. The line in magenta intersects to point A_w, the yellow one to B_w and the cyan to E_w.

$$\rho(X) == \mathcal{G} \bigcap \overleftrightarrow{\mathcal{O}X_{3D}} \tag{4.33}$$

$$d_{relative}(X_1, X_2) == \left|\rho(X_1)\rho(X_2)\right| \tag{4.34}$$

Applying Equation 4.33 to points A, B and C:

$$A_w = \mathcal{G} \bigcap \overleftrightarrow{\mathcal{O}A} \tag{4.35}$$

$$B_w = \mathcal{G} \bigcap \overleftrightarrow{\mathcal{O}B} \tag{4.36}$$

$$C_w = \mathcal{G} \bigcap \overleftrightarrow{\mathcal{O}C} \tag{4.37}$$

To compute the box relative height it is necessary to know the world coordinates of point E, which is not on the ground plane. To accomplish it a new plane $\mathcal{J}$ perpendicular to plane $\mathcal{G}$ containing the computed point A_w has to be defined ($\mathcal{J} \perp \mathcal{G} and A_w \in \mathcal{J}$). Therefore, the calculation of point E_w is done accordingly to Equation 4.38. Lastly, the computation of relative box length (l_r), width (w_r) and height (h_r) is done with Equation 4.39. Efficient line-plane and plane-plane intersection computation was based on theory described by [16].

$$E_w = \mathcal{J} \bigcap \overleftrightarrow{\mathcal{O}E} \tag{4.38}$$

$$(l_r, w_r, h_r) = (|A_w C_w|, |A_w B_w|, |A_w E_w|) \tag{4.39}$$

For retrieving the car box dimensions in SI units an histogram of relative values is accumulated for each dimension on the first 1500 video frames, corresponding to one minute of video in 25 FPS streams (Figure 4.8). Then, median car dimensions are computed (l_M, w_m, h_m) and meter box dimension is calculated with Equation 4.40, where 4.27, 1.74 and 1.51 are the considered mean vehicle length, width and height, respectively , in meters.

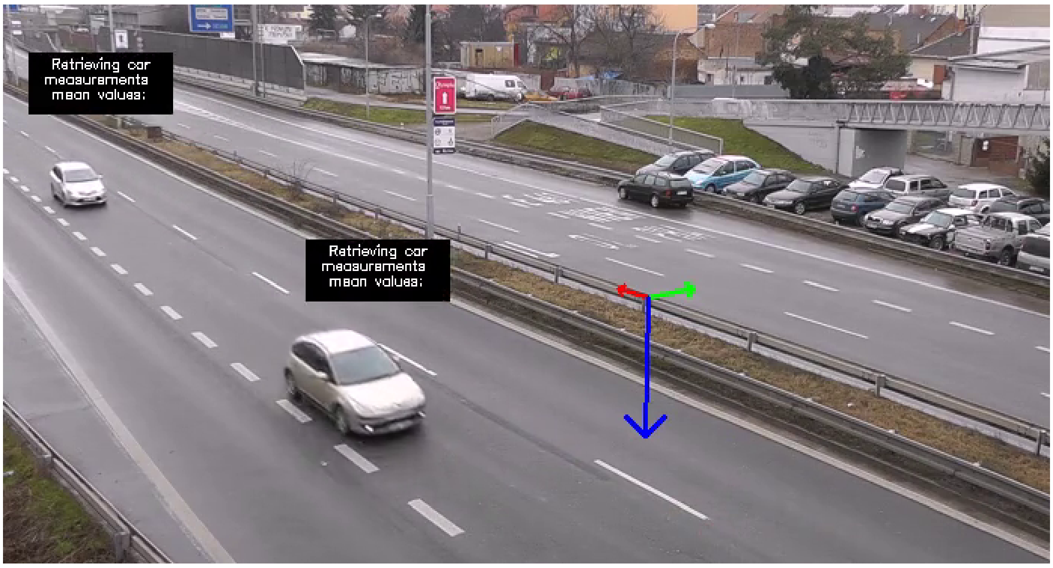

Retrieving car measurements mean values:
Retrieving car measurements mean values:

Fig. 4.8: Relative box dimensions collecting process.

$$l_{real} = \frac{4.27 l_r}{l_M} \qquad w_{real} = \frac{1.74 w_r}{w_M} \qquad h_{real} = \frac{1.51 h_r}{h_M} \tag{4.40}$$

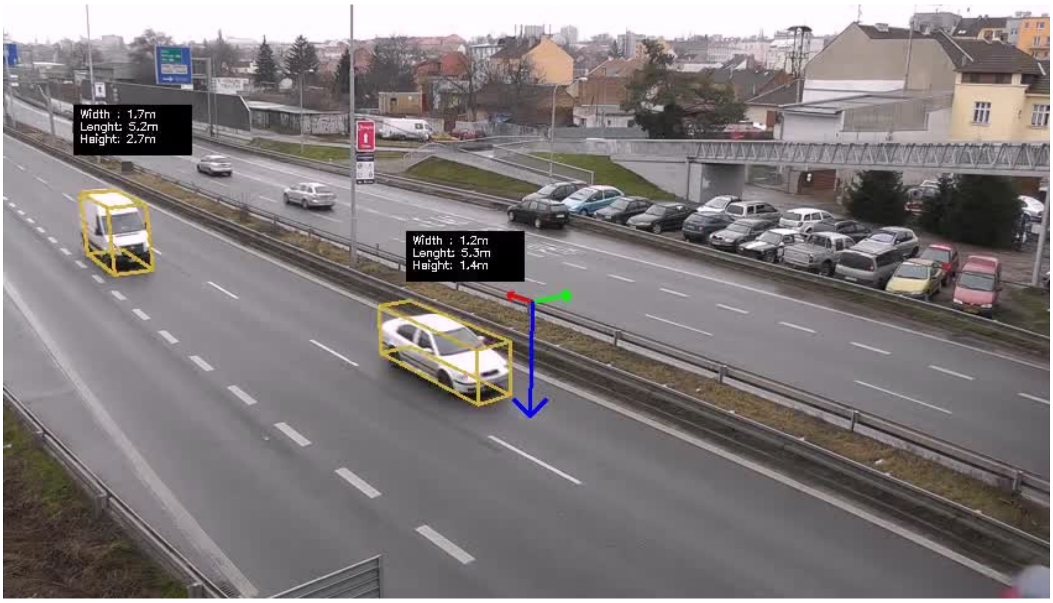


Fig. 4.9: Real box dimensions after relative box dimensions collecting process.

4.2.4 Lane Detection

Knowing the position of each bounding box base on each frame it is possible to construct a simple lane analysis algorithm. It consists on a histogram with the number of bins equal to the number of pixels on a video frame. All the pixels of the base of the box vote +1 on the respective pixel. It can also give the idea about the traffic on each lane of the road (Figure 4.10).

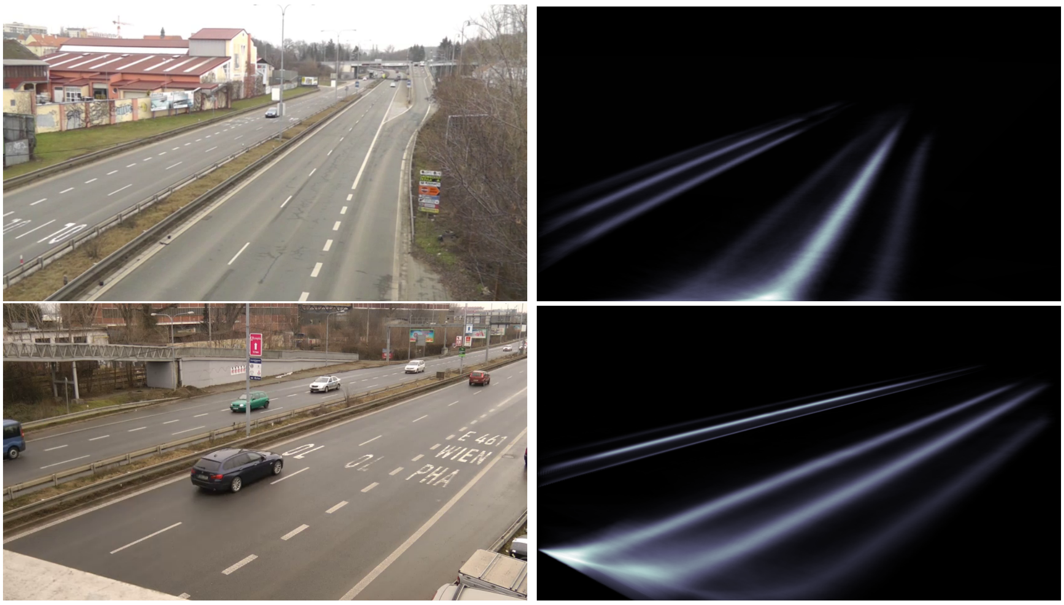


Fig. 4.10: Original traffic frame (left) and corresponding lane detector histogram (right).

A peak finder was applied to the histogram middle row in order to get the number of presented lanes and their location. Experiments and results are presented on the next chapter (5).

Chapter 5

Development and Results

5.1 Development of the work

In order to improve the processing speed, the implemented work was developed in C++. Two separate programs were developed as a command line utility using the OpenCV C++ library [23]. The first program is responsible for the calibration of the video stream retrieving the vanishing points location both on the image plane and in 3D coordinates. The second one, based on the first one collected data, processes the video stream in order to acquire mean vehicle dimensions and to analyse vehicle traffic dimensions with the optional ability of also performing a lane analysis. The architecture of both programs is modular and different data output, or even a video output, can be easily retrieved. Both programs processes the full video stream.

5.1.1 Dataset

Regarding calibration evaluation, 5 groups of videos were provided by the Computer Graphics Research Group of the Faculty of Information Technology of Brno, in Czech Republic [13]. Each group corresponds to a traffic scenario recorded from different perspectives in similar weather conditions and videos have approximately 3 minutes each. Table 5.1 resumes each group scenario and weather conditions.

For traffic analysis evaluation, 5 extra videos were provided by the same research group. This dataset [12] contains three videos of approximately 9 minutes each and two videos of approximately 24 minutes each with good weather conditions. The reason why all provided videos have

Group	# of Videos	Observations
g1	6	Cloudy with an open road view
g2	5	Cloudy with a close road view
g3	8	Sunny, with several lightning changes and shadows, in an open road view
g4	5	Rainy, open road view with view occlusion is some videos
g5	7	Sunny, with no general lightning changes. Open road view
Total # of Videos	31	

Table 5.1: Table summarizing each group number of videos as well as observations *per* group.

this weather conditions is because of the difficulty of accurate background subtraction algorithms in bad weather scenarios.

Thumbnails of both provided video streams are presented in Figure 5.1 and Figure 5.2:

Fig. 5.1: Thumbnails of the provided calibration dataset [13].

5.1.2 Roadside Calibration

The processing of each video stream is made frame by frame considering pairs of frames. As already stated before, video frames are skipped in order to analyse the video as a 12.5 FPS one.

To filter the lines to vote on the first diamond space for first VP detection Shi *et al.* is applied with a maximum detection of 500 corner feature points on the previous frame. If there are more corners than the ones found, the strongest 500 of them are returned. However, generally, this

Fig. 5.2: Thumbnails of the provided traffic analysis dataset [12].

is not the case. The quality level, characterizing the minimal accepted quality of image corners, is set to 0.01. This parameter value is multiplied by the best corner quality measure, which is the minimal eigenvalue. This means that corners with quality measure less than the product are rejected. Also, the minimum possible Euclidean distance between the returned points is set to 10.

After collecting feature corners on the previous frame, a KLT tracker is applied in the next frame to the previously collected points. Moving corner points, i.e., with a difference of its position higher that 4 pixels, are considered, and the lines that define them vote in the accumulator space.

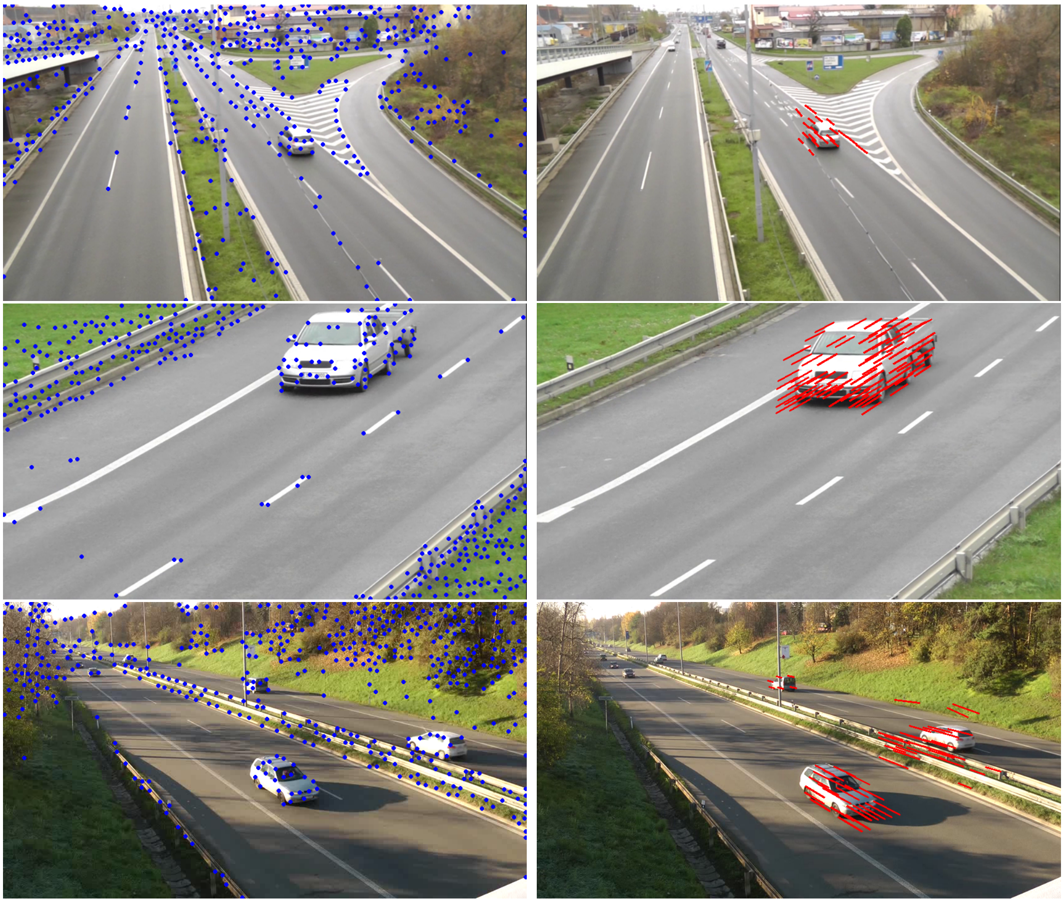


Fig. 5.3: Various traffic scenarios with Good Features to Track [31] applied on the previous frame (left) and KLT tracker [35] of moving points on the subsequent frame (right)).

For the second VP line filtering a edge background model was implemented, based on the oriented values of the gradient of each frame I. The used sobel masks to compute the gradient are presented on equations 5.1 and 5.2.

$$K_y = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{bmatrix} \tag{5.1}$$

$$K_x = K_y^T = \begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix} \tag{5.2}$$

Hence, gradient and magnitude $G(i, j)$ and $Mag(i, j)$ are computed with:

$$G_x = I * K_x \tag{5.3}$$

$$G_y = I * K_y \tag{5.4}$$

$$G(i, j) = atan2(G_y(i, j), G_x(i, j)) \tag{5.5}$$

$$Mag(i, j) = \sqrt{G_x^2 + G_y^2} \tag{5.6}$$

The edge background model stores for each pixel (i, j) the confidence score of occurrence of an oriented edge and is updated at correct bin for edge orientation (i, j).

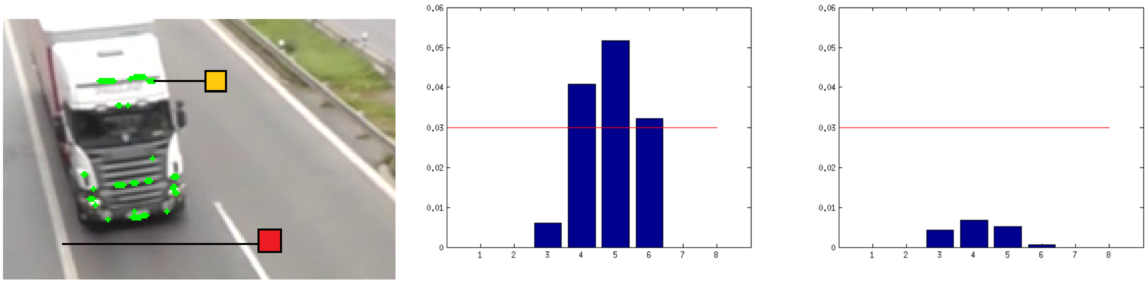
Background mask is initialized with:

$$W(G(i, j), bin) = max\left(\frac{cos(2 \times (G(i, j) - bin * pi))}{8}, 0\right) \tag{5.7}$$

Hence, background edge model B of size $(I_{width} \times I_{height} \times 8)$ is:

$$B_t(bin) = W(G(i, j), bin) \times Mag(i, j) \tag{5.8}$$

Following, a matrix G_{bins} is created of size $I_{width} \times I_{height}$ containing the corresponding bin number for each pixel of G. Only values of $B_t(G_{bins}) < 0.03$ are considered as foreground (τ_2 threshold). If pixel (i, j), considered as foreground, is detected by a canny detector to the frame, its magnitude ($Mag(i, j)$) is higher than 75% of the magnitude matrix values (dynamic τ_1 threshold), its orientation ($G(i, j)$) is not close to the detected first VP and two neighbourhood pixels are also considered as foreground, it votes in the accumulator space of the second VP. Two edge pixel histograms were shown in Figure 5.4 and all edges voting on the complete frame in Figure 5.5.



(a) **(b)** **(c)**

Fig. 5.4: Histograms of two pixels in a single frame a). b) represents the real histogram of the background edge model of the pixel represented in red in a). c) represents the real histogram of the background edge model of the pixel represented in yellow in a). As the histogram of the red pixel passes above the τ_2 threshold it is considered as background. The contrary is applied to the yellow one.

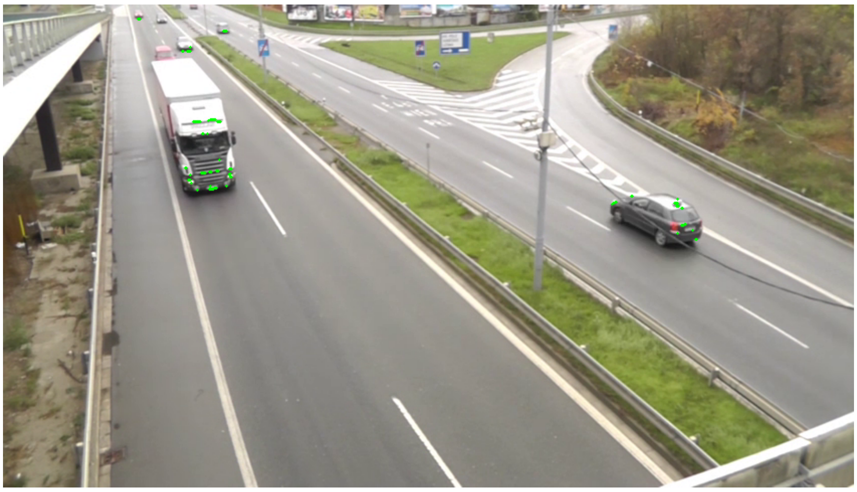


Fig. 5.5: Edges voting on VP2 space accumulator in a single frame.

The model B_t is then updated and the same procedure applies to every video frame.

5.1.3 3D Bounding Boxes

After vanishing point detection, a background subtraction algorithm is applied and 3D bounding boxes are constructed with the implementation procedures described in Chapter 4. Examples of this construction are shown in Figure 5.6.

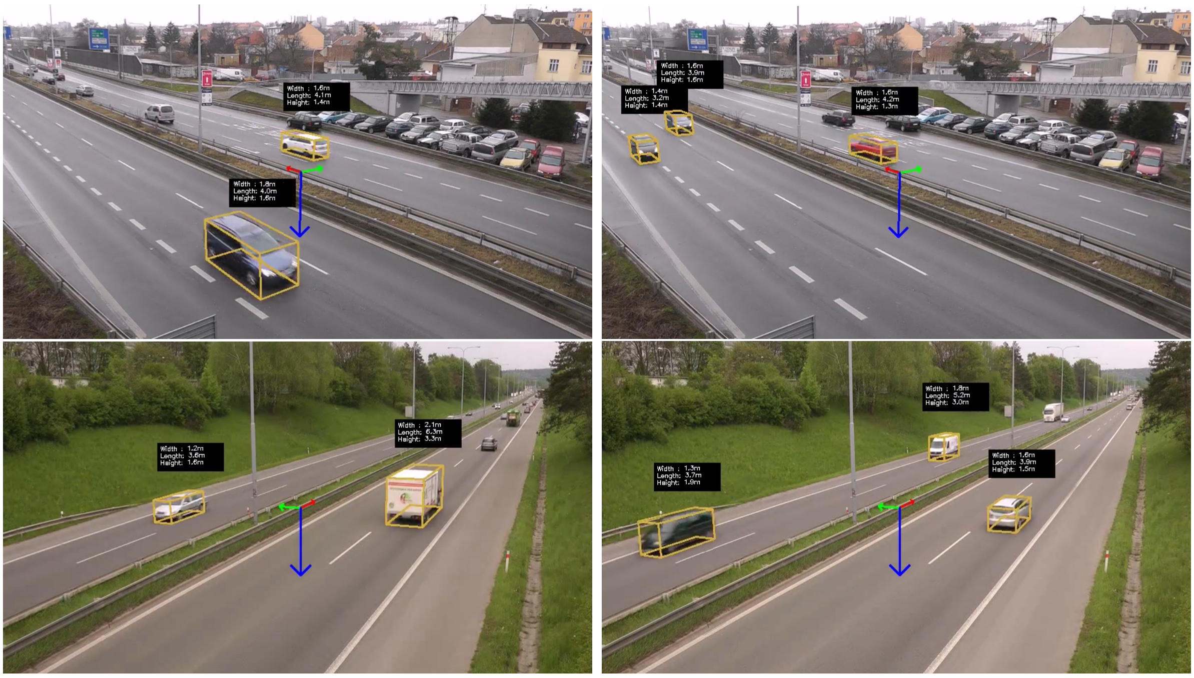


Fig. 5.6: Examples of constructed 3D vehicle bounding boxes in two different traffic scenarios.

5.1.4 Lane Detection

For correct lane number detection and lane location identification, a peakfinder is applied to the middle row of the constructed lane voting histogram. In order to reduce peak outliers, voting values are sampled every 10 values and a low threshold is applied. Each value in the row histogram lower than 60 is set to 0. An example of the result of the described procedure is shown in Figure 5.7.

5.2 Results and Evaluation

The results were done on a machine with Intel(R) Core(TM) i7 CPU 950 3.07GHz and 8-GB of RAM. Processing speed results include both reading and video decompression and computation time.

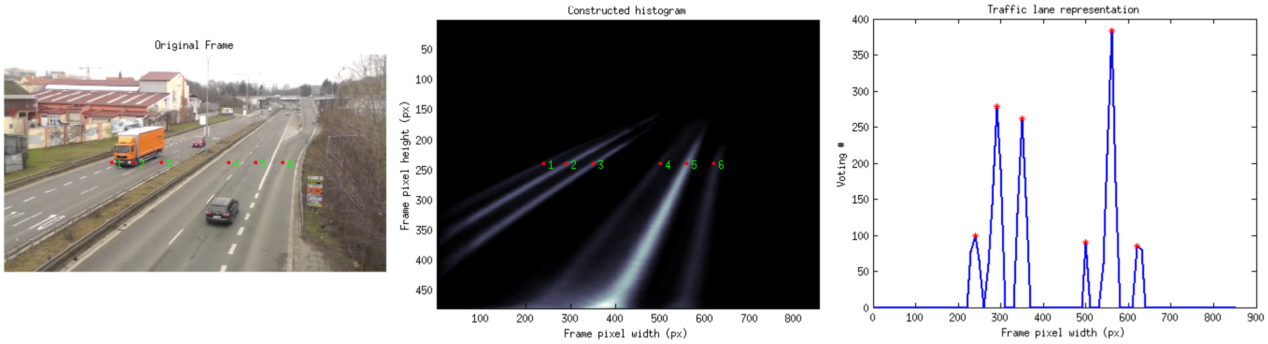


Fig. 5.7: Result of the implemented lane analysis procedure to Video 1.

5.2.1 Calibration

In order to evaluate calibration implemented method, convergence times are evaluated for first and second vanishing points. Some convergence graphs are following presented where the pixel distance between the detected VP at a given time and the final detected VP is evaluated.

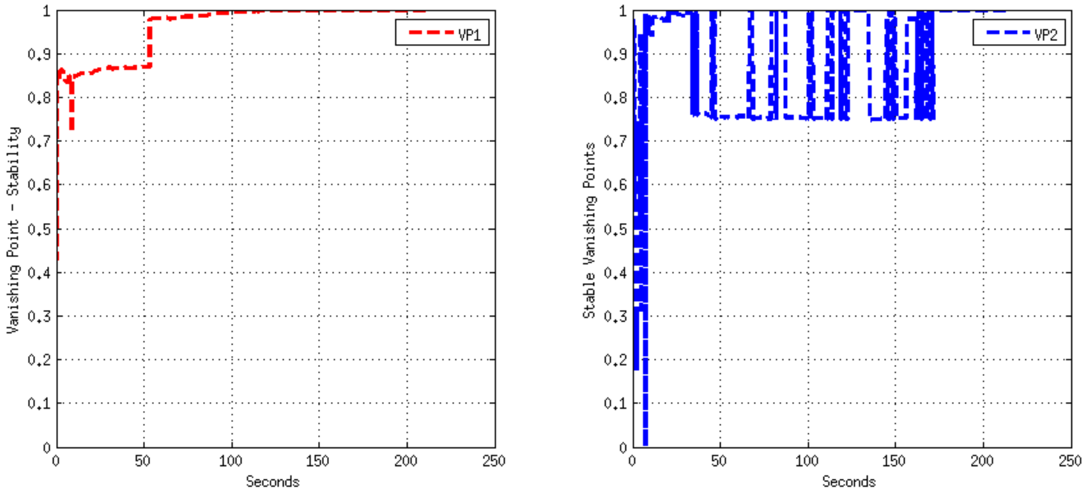


Fig. 5.8: Convergence of VP1 and VP2 on the 5th video of group 5.

Some videos from the provided calibration dataset shown instability on the second VP evaluation (Figure 5.8). That is due to the fact that since the videos have approximately 3 minutes of recording time, is not sufficient to collect sufficient amount of evidence about the second VP. However, it is possible for the 2nd VP to converge in a 3 minute video (Figure 5.9). Experiments with the second dataset, provided for traffic analysis, shown mean convergence times of the first VP to be around 120 seconds of video and convergence of the second VP in about 140 seconds of video time (Figure 5.10). From all the analysed videos, the maximum time to compute the second VP shown is actually 450 seconds of video Figure 5.11. The authors concluded similar

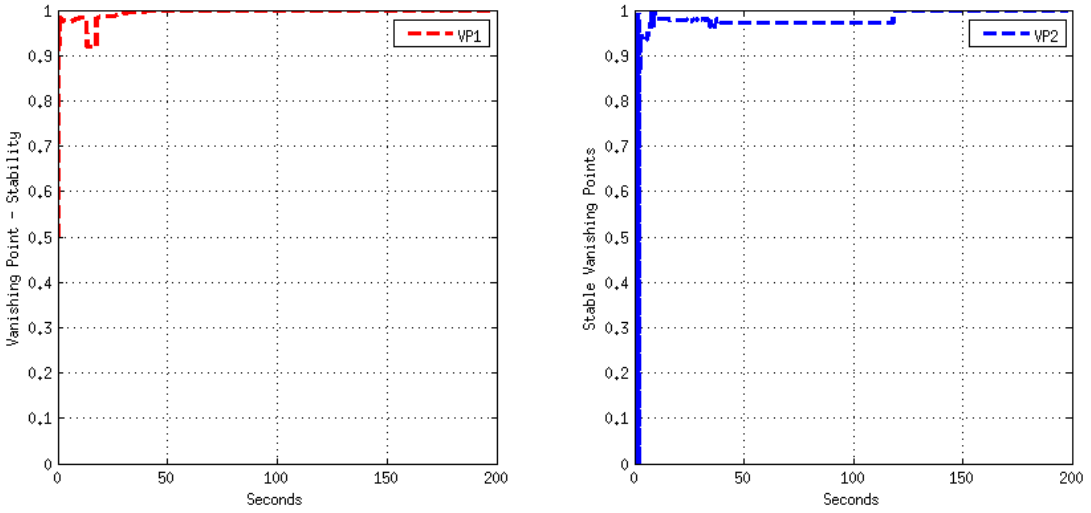


Fig. 5.9: Convergence of VP1 and VP2 on the 3rd video of group 4.

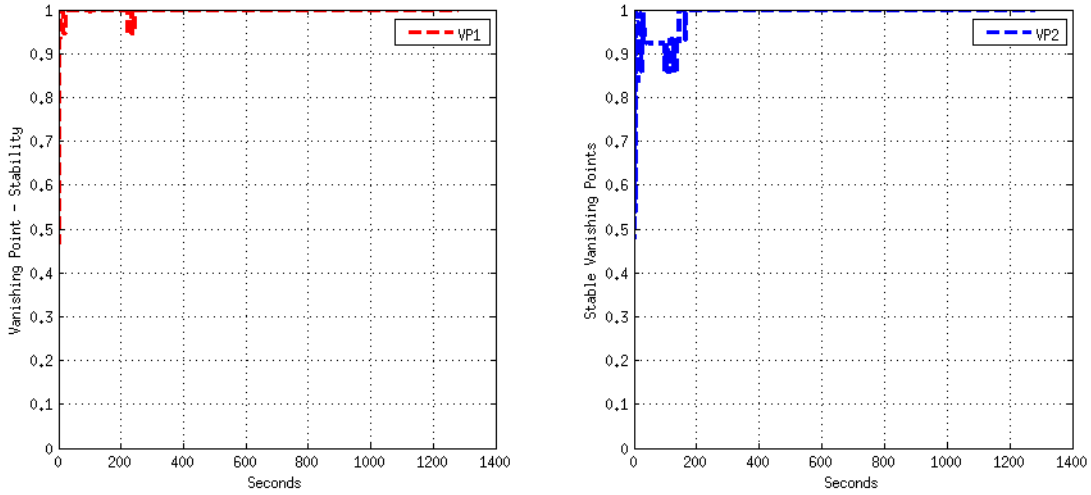


Fig. 5.10: Convergence of VP1 and VP2 on 5th video of the second dataset.

mean computation times [15] of about 160 seconds for the first VP and 250 seconds for the second VP.

5.2.2 Lane detection

The reason why there's no correct number of lane identification in the third video is because the sixth lane corresponds to an exit road lane and a lot of cars are constantly changing to and from this specific lane, leading to inaccurate lane voting and consequently inaccurate lane analysis results. Evaluation results of lane detection in the five videos is presented on Appendix A.

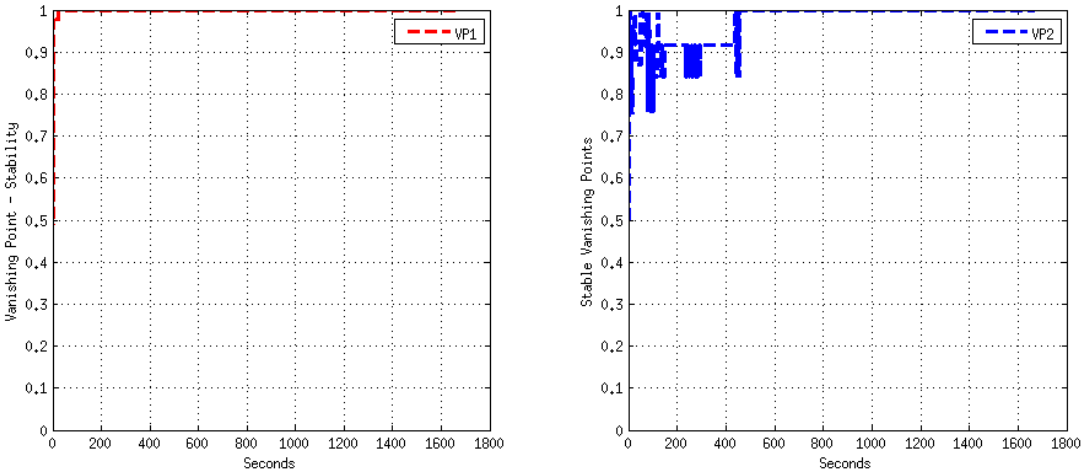


Fig. 5.11: Convergence of VP1 and VP2 on 4th video of the second dataset.

Video	# of Lanes	# of Correctly Detected Lanes	Accuracy (%)
Video 1	6	6	100
Video 2	6	6	100
Video 3	6	5	83.33
Video 4	4	4	100
Video 5	4	4	100
Total % of Accuracy	96.66%		

Table 5.2: Table summarizing accuracy of the developed algorithm for lane detection.

5.2.3 Processing speed

This section presented processing times are averaged from all dataset videos.

Frame Size	Processed Frames per Second
480 × 854	∼ 2.44 FPS
322 × 688	∼ 4.22 FPS

Table 5.3: Table summarizing mean processing speed of the developed calibration method with different frame sizes for all tested calibration streams.

Frame Size	**Traffic Intensity**	**With Lane Detection**	**Without Lane Detection**
480 × 854	High	∼ 32.21 FPS	∼ 68.53 FPS
	Medium	∼ 42.53 FPS	∼ 71.32 FPS
322 × 688	High	∼ 52.51 FPS	∼ 98.44 FPS
	Medium	∼ 63.81 FPS	∼ 110.93 FPS

Table 5.4: Table summarizing mean processing speed of the developed traffic analysis method with different frame sizes in high and low traffic scenarios for all tested analysis streams. High traffic: ∼ 40 vehicles per minute. Medium traffic: ∼ 20 vehicles per minute

Chapter 6

Conclusions and Future Work

6.1 Conclusions

With the development of the work it is possible to conclude that the implemented calibration algorithm is reliable for multiple traffic surveillance tasks. In fact, experiments shown that the algorithm runs above real time ($\geqslant$ 25 FPS) and many applications can be developed based on the retrieved calibration data.

6.2 Future Work

As future work, there are several tasks that can be done. Some of them are listed bellow.

- Foreground segmentation improvements, in order to design a robust system for worst weather scenarios than the ones of the provided dataset.
- Real-time speed measurements, based on the collected vanishing points location. A tracking algorithm can be implemented to the constructed vehicle bounding boxes to accomplish it.
- Real-time classification, based on retrieved vehicle bounding box dimensions in SI units.
- Real-time wrong-way detection, based on traffic flow learning on each detected lane and using the information retrieved from the implemented KLT tracker.

References

[1] S. Alvarez, D. F. Llorca, and M. A. Sotelo. Camera auto-calibration using zooming and zebra-crossing for traffic monitoring applications. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 608–613, 2013.

[2] E.K. Bas and J.D. Crisman. An easy to install camera calibration for traffic monitoring. *Proceedings of Conference on Intelligent Transportation Systems*, pages 362–366, 1997.

[3] D Beymer, P McLauchlan, B Coifman, and J Malik. A real-time computer vision system for measuring traffic parameters. *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 495–501, 1997.

[4] Brisa Portugal. Ferramentas - informação de trânsito — Via Verde. `https://www.viaverde.pt/Ferramentas/informacao-de-transito` Accessed on: 15-08-2016.

[5] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[6] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.

[7] F. W. Cathey and D. J. Dailey. A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2005:777–782, 2005.

[8] R Cipolla, T Drummond, and D Robertson. Camera calibration from vanishing points in images of architectural scenes. *Review Literature And Arts Of The Americas*, 2:382–391, 1999.

[9] Daniel J. Dailey, F. W. Cathey, and Suree Pumrin. An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):98–107, 2000.

[10] M. Dubská, A. Herout, and J. Havel. PClines; Line detection using parallel coordinates. *Cvpr 2011*, pages 1489–1494, 2011.

[11] M Dubská, J Sochor, and a Herout. Automatic Camera Calibration for Traffic Understanding. *Proceedings of BMVC 2014*, page 2014, 2014.

[12] Marketa Dubská. Parallel Coordinates in Computer Vision - 2014-BMVC-VehicleBoxes, 2014. `http://www.fit.vutbr.cz/research/groups/graph/pclines/pub{_}page.php?id=2014-BMVC-VehicleBoxes` Acessed on: 02-03-2016.

[13] Marketa Dubska. Parallel Coordinates in Computer Vision - 2014-ITS-MotionVanPoints, 2014. `http://www.fit.vutbr.cz/research/groups/graph/pclines/pub{_}page.php?id=2014-ITS-MotionVanPoints` Acessed on: 02-03-2016.

[14] Markéta Dubská and Adam Herout. Real Projective Plane Mapping for Detection of Orthogonal Vanishing Points. *Bmvc2013*, pages 1–10, 2013.

[15] Marketa Dubska, Adam Herout, Roman Juranek, and Jakub Sochor. Fully automatic roadside camera calibration for traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1162–1171, 2015.

[16] Christer Ericson. *Real-time collision detection.* 2005.

[17] J. Marcos G. Monteiro and J. Batista. Stopped Vehicle Detection System for Outdoor Traffic Surveillance. *RECPAD 2008*, 2008.

[18] J. Marcos G. Monteiro, M. Ribeiro and J. Batista. A Framework for Wrong Way Driver Detection Using Optical Flow. *Lecture Notes in Computer Science*, 4633(Springer Berlin), 2007.

[19] G.Monteiro. Intelligent Traffic Surveillance - University of Coimbra - ISR, 2008. `http://its.isr.uc.pt/` Accessed on: 01-09-2016.

[20] Michael Hodlmoser, Branislav Micusik, and Martin Kampel. Camera auto-calibration using pedestrians and zebra-crossings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1697–1704, 2011.

[21] T Horprasert, D Harwood, and L S Davis. A statistical approach for real-time robust background subtraction and shadow detection. *Ieee Iccv*, 99:1–19, 1999.

[22] Alfred Inselberg and Bernard Dimsdale. *Parallel Coordinates for Visualizing Multi-Dimensional Geometry*, pages 25–44. Springer Japan, Tokyo, 1987.

[23] Itseez Intel Corporation, Willow Garage. OpenCV User Site, 2000. `http://opencv.org/` Acessed on: 02-03-2016.

[24] N K Kanhere and S T Birchfield. A Taxonomy and Analysis of Camera Calibration Methods for Traffic Monitoring Applications. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):441–452, 2010.

[25] Neeraj K. Kanhere and Stanley T. Birchfield. A taxonomy and analysis of camera calibration methods for traffic monitoring applications. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):441–452, 2010.

[26] Fengjun Lv, Tao Zhao, and Ramakant Nevatia. Camera calibration from video of a walking human. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1513–1518, 2006.

[27] Cristina Maduro, Katherine Batista, Paulo Peixoto, and Jorge Batista. Estimation of vehicle velocity and traffic intensity using rectified images. In *Proceedings - International Conference on Image Processing, ICIP*, pages 777–780, 2008.

[28] Osama Masoud and Nikolaos P. Papanikolopoulos. Using geometric primitives to calibrate traffic scenes. *Transportation Research Part C: Emerging Technologies*, 15(6):361–379, 2007.

[29] Gonçalo Monteiro, João Marcos, Miguel Ribeiro, and Jorge Batista. Robust segmentation for outdoor traffic surveillance. In *Proceedings - International Conference on Image Processing, ICIP*, pages 2652–2655, 2008.

[30] T. N. Schoepflin and Daniel Dailey. Dynamic camera calibration of roadside traffic management cameras. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, volume 2002-Janua, pages 25–30, 2002.

[31] Carlo Shi, Jianbo and Tomasi. Good features to track. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593—-600, 1994.

[32] Kai Tai Song and Jen Chao Tai. Dynamic calibration of pan-tilt-zoom cameras for traffic monitoring. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(5):1091–1103, 2006.

[33] Chris Stauffer and W E L Grimson. Adaptive background mixture models for real-time tracking. *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149*, 2(c):246–252, 1999.

[34] Richard Szeliski. Computer Vision : Algorithms and Applications. *Computer*, 5:832, 2010.

[35] Carlo Tomasi. Detection and Tracking of Point Features. *School of Computer Science, Carnegie Mellon Univ.*, 91(April):1–22, 1991.

[36] Kunfeng Wang, Hua Huang, Yuantao Li, and Fei Yue Wang. Research on lane-marking line based camera calibration. *2007 IEEE International Conference on Vehicular Electronics and Safety, ICVES*, pages 1–6, 2007.

[37] World Health Organization. Global Status Report on Road Safety 2015. 2015.

[38] Zhaoxiang Zhang, Min Li, Kaiqi Huang, and Tieniu Tan. Practical camera auto-calibration based on object appearance and motion for traffic scene visual surveillance. In *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.

[39] Zhaoxiang Zhang, Tieniu Tan, Kaiqi Huang, and Yunhong Wang. Practical camera calibration from moving objects for traffic scene surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(3):518–533, 2013.

[40] Zoran Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 2(2):28–31, 2004.

[41] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006.

Appendix A

Lane Analysis Results

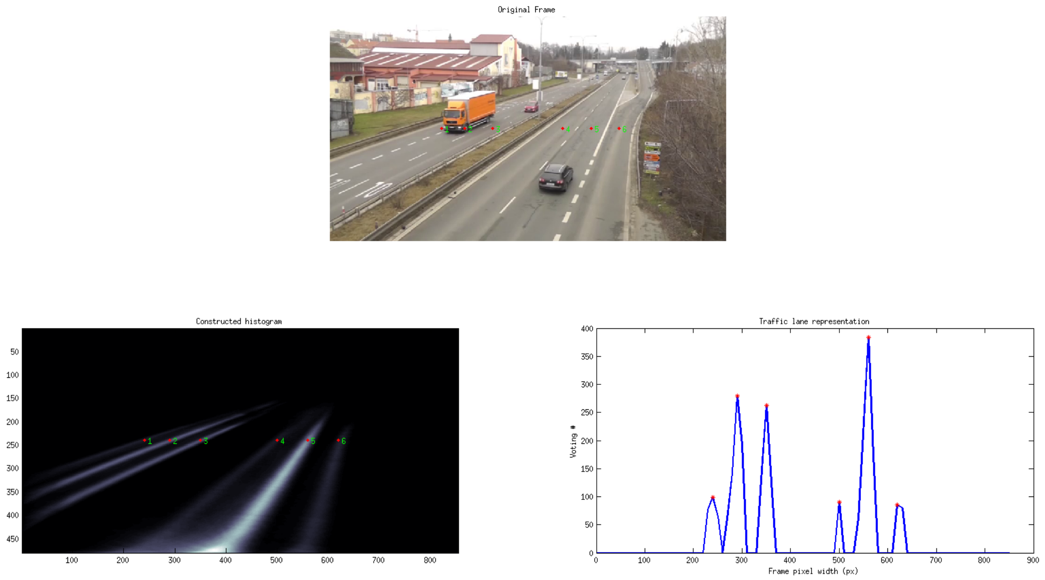


Fig. A.1: Result of the implemented lane analysis procedure to Video 1.

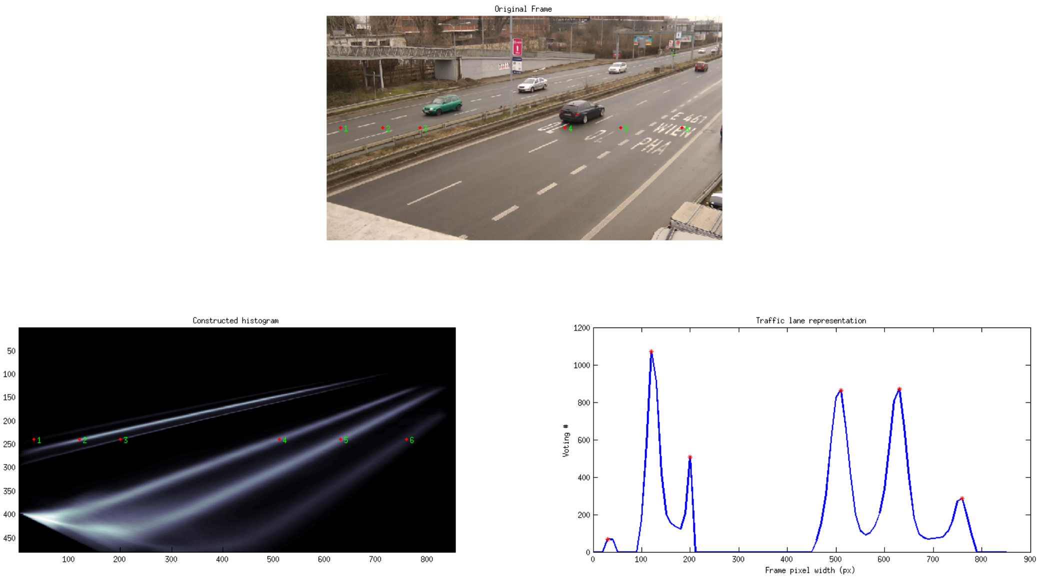


Fig. A.2: Result of the implemented lane analysis procedure to Video 2.

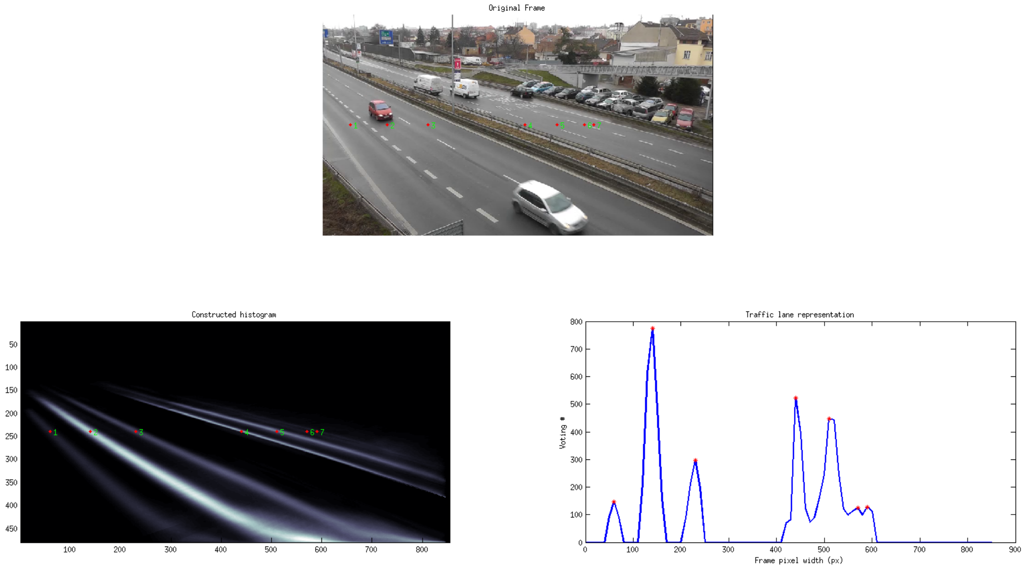


Fig. A.3: Result of the implemented lane analysis procedure to Video 3.

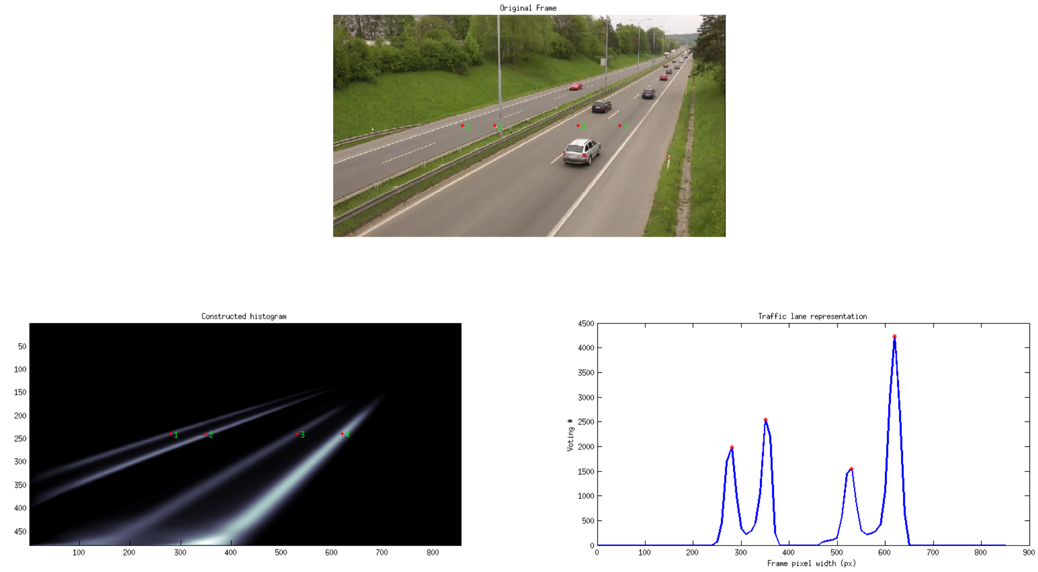


Fig. A.4: Result of the implemented lane analysis procedure to Video 4.

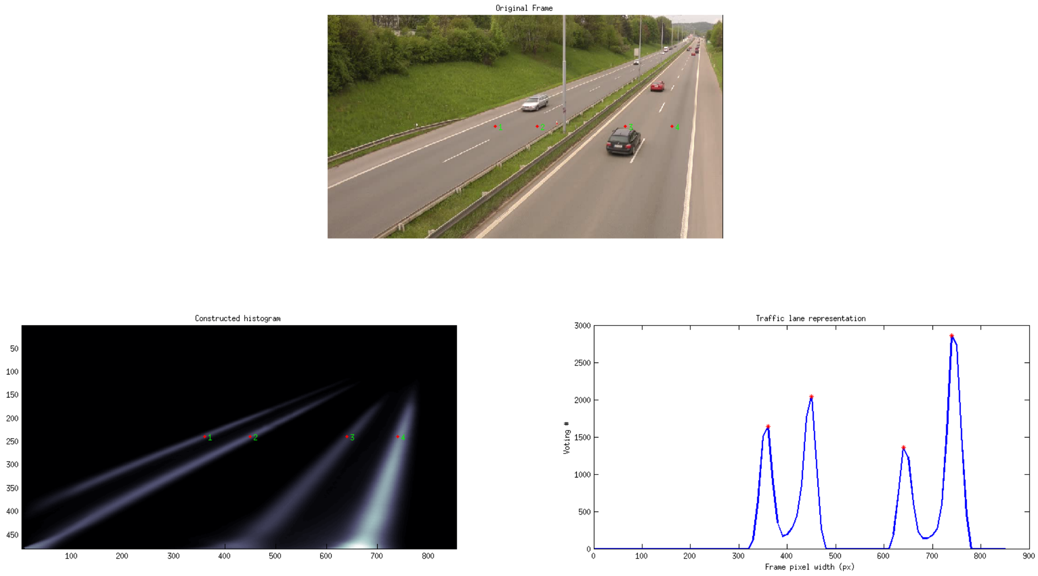


Fig. A.5: Result of the implemented lane analysis procedure to Video 5.

Acronyms and symbols

Abbreviation	Meaning
ISR	Institute of Systems and Robotics
ITS	Intelligent Transportation Systems
VP	Vanishing Point
PC	Parallel Coordinates
PCLines	Parallel Coordinates Line
PTLM	Point-to-Line-Mapping
CHT	Cascaded Hough Transform
KLT	Kanade–Lucas–Tomasi
FPS	Frames per Second
PP	Principal point
3D	Three-dimensional space
2D	Two-dimensional space
MG	Mixture of Gaussians
WCS	World Coordinate System
SI	International System of Units
GB	Giga-bytes
GHz	Giga-Hertz
RAM	Random Access Memory