Expert Systems with Applications 38 (2011) 1619–1631

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A background subtraction algorithm for detecting and tracking vehicles

Nicholas A. Mandellos [a,*], Iphigenia Keramitsoglou [b], Chris T. Kiranoudis [a]

[a] Department of Process Analysis and Systems Design, National Technical University of Athens, GR-15780 Athens, Greece
[b] Institute for Space Applications and Remote Sensing, National Observatory of Athens, Metaxa & Vas. Pavlou, GR-15236 Athens, Greece

ARTICLE INFO

Keywords:
Computer vision
Background subtraction
Background reconstruction
Background maintenance
Background update
Vehicle detection
Traffic surveillance
Tracking

ABSTRACT

An innovative system for detecting and extracting vehicles in traffic surveillance scenes is presented. This system involves locating moving objects present in complex road scenes by implementing an advanced background subtraction methodology. The innovation concerns a histogram-based filtering procedure, which collects scatter background information carried in a series of frames, at pixel level, generating reliable instances of the actual background. The proposed algorithm reconstructs a background instance on demand under any traffic conditions. The background reconstruction algorithm demonstrated a rather robust performance in various operating conditions including unstable lighting, different view-angles and congestion.

1. Introduction

The escalating increase of contemporary urban and national road networks over the last three decades emerged the need of efficient monitoring and management of road traffic. The surface transportation system of United States for instance, consists of approximately 3.7 million miles of roads, estimated to increase by 30% over the next decade. Environmental pressures as well as socioeconomic problems are associated with this increase due to prolonged congestions and slowing down of the average highway speed. To deal with this problem, one option is to increase network capacity and the other one is to increase efficiency by investing in Intelligent Transportation Systems (ITS) technology (Gutchess, Trajkovic, Kohen-Solal, Lyons, & Jain, 2001).

Conventional technology for traffic measurements, such as inductive loops, sonar or microwave detectors, suffer from serious drawbacks: they are expensive to install, they demand traffic disruption during installation or maintenance, they are not portable and they are unable to detect slow or stationary vehicles. On the contrary, video based systems are easy to install, can be a part of ramp meters and may use the existing traffic surveillance infrastructure. Furthermore, they can be easily upgraded and they offer the flexibility to redesign the system and its functionality by simply changing the system algorithms. Those systems allow vehicle counting, classification, measurement of vehicle's speed and the identification of traffic incidents (such as accidents or heavy congestion).

There is a wide variety of systems based on video and image processing employing different methodologies to detect vehicles and objects. A review of such image processing methodologies, presented in Kastrinaki, Zervakis, and Kalaitzakis (2003), comprises thresholding, multi-resolution processing, edge detection, background subtraction and inter-frame differencing. Thresholding is the simplest process of the above and it was part of the very first automatic surveillance systems in the decades of 1970s and 1980s when such systems were loop detector simulators (Mahmassani, Haas, Zhou, & Peterman, 2001). Those systems had low accuracy and they are not used nowadays. Multi-resolution processing lies on scale space theory (Lindeberg, 1996), that uses coarse and fine level color pixel information to cluster the image and to separate objects from the background. However, it is not accurate enough for traffic problems since it breaks parts of the image (i.e. road lines, glares and shadows) and merges them with parts of vehicles having the same chromatic range. Another main drawback is that the system cannot effectively deal with image perspective, therefore, vehicles standing away from the camera are under-segmented and vehicles standing near the camera are over-segmented. Moreover, it cannot distinguish vehicles in congestion. Edge-based methodologies have the main advantage that the extracted features are scale and lighting invariant (Koller, Weber, & Makik, 1994), but it is quite difficult to derive vehicle shapes especially in congested scenes that vehicles stop frequently. The inter-frame differencing methodology is accurate enough to detect parts of moving objects by comparing two consecutive frames. However, it can identify only differences in the background and, as a result, it detects only parts of a vehicle covering the background in the previous frame. Despite some enhancing techniques (Cucchiaraa & Piccardi, 1999) this methodology cannot satisfactory deal with

* Corresponding author. Tel.: +30 210 772 3128; fax: +30 210 772 3155.
E-mail addresses: nmand@central.ntua.gr (N.A. Mandellos), ikeram@space.noa.gr (I. Keramitsoglou), kyr@chemeng.ntua.gr (C.T. Kiranoudis).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.07.083

realistic traffic circumstances where vehicles might remain still for a long time. Finally, background subtraction detects the actual background and extracts objects that do not belong to it. The concept of this method is described below.

In a typical background model a prototype of the image background (an initialization of the background) is considered first and then each pixel of the prototype is compared with the actual image color map. If the color difference exceeds a predefined threshold it is assumed that this pixel belongs to the foreground. Consequently, raw foreground information is derived. This information is grouped to compact pixel sets (blobs). In case of outdoor scenes, when the background is not completely static, lighting fluctuations, shadows or slight movements (i.e. leaves and branches waving) can degrade the effectiveness of the foreground extraction. To overcome this, a number of algorithms have modeled the aforementioned nuisances. More specifically, mixture models use statistical filters to eliminate continuous slight movements on the background by grouping time evolving pixel characteristics in clusters or color prototypes and characterizing as background the more populated one (Kim, Chalidabhongse, Harwood, & Davis, 2005; Stauffer & Grimson, 1999; Zivkovic & Van der Hijden, 2006), while parametric models such as the ones proposed by Haritaoglu, Harwood, and Davis (2000), Horprasert, Harwood, and Davis (1999), Pless (2005) simulate the background by taking into account color characteristics. In the work (Horprasert et al., 1999) each pixel is classified in one of four classes namely: 'Foreground', 'Shaded background', 'Highlighted background' and 'Background'. Thus, the system can 'recognize' background discontinuities due to lightings and shadows and consequently register them as background.

This methodology has the great advantage of separating objects by using background information even in images that comprise shadows or glares (Senior, Tian, Brown, Pankanti, & Bolle, 2001). The main drawback of the background subtraction algorithm is the complexity to define the background. A common practice is to initialize the algorithm by employing an 'empty scene'. Another important issue in this methodology is the difficulty to maintain the background instance through time in outdoor captures.

The creation of a reliable initial instance is a critical issue for the quality of the overall process. A general solution for this problem does not exist, and the common practice is to average a sequence of frames presenting a scene without moving objects, which in fact is too difficult to acquire in a crowded highway. Despite the importance of this issue, there has only been limited research published focusing on the reconstruction of a starting background instance. (Colombari, Cristani, Murino, & Fusiello, 2005; Gutchess et al., 2001), for instance, are significantly complicated to implement. On top of that, they are based on several restrictive assumptions. The latter work in particular refers to an inpainting technique, (Criminisi, Perez, & Toyama, 2004) where background parts are reconstructed exploiting color and texture information.

In outdoor captures, the background prototype often fails to reflect the actual background due to lighting condition changes, shadow casting with respect to the sun position and background alterations with permanent effect. Moreover, the insertion of new objects in the road scene can induce permanent or temporary changes of the background (e.g. a vehicle that has been pulled over for a long time or an object in the road deck). Common practice in such cases is to use adaptive update models, such as those of Toyama, Krumm, Brumitt, and Meyers (1999), Gupte, Masoud, Martin, and Papanikolopoulos (2002), Wren, Azarbayejani, Darrell, and Pentland, (1997), that keep the background template recursively updated, so that, the background template is adapted in forthcoming image changes. Nevertheless, in most cases, after some time the noise pollution of the background results into the degradation of the overall process quality.

In this study we present an innovative algorithm, the background reconstruction algorithm, as part of a system for locating and tracking vehicles through traffic video captures. The purpose of the present work is to overcome the two main weaknesses of the background subtraction algorithm, namely initialization and background update and to build a robust methodology, capable of detecting vehicles under realistic traffic circumstances.

The background reconstruction algorithm is a heuristic that provides a periodically updated background and enhances the efficiency of the well known background subtraction methodology in case of outdoor captures. Indeed, it is a key process for a typical background subtraction algorithm, because it supports the weakest part of it, which is the initialization step. This methodology guarantees a fresh instance of the actual background periodically, which is achieved by collecting scatter color information through a series of sequential images and assembling them to reconstruct the actual background. This process is applied to each pixel separately and the result is a color map of the actual image background.

Our algorithm is presented as a part of an integrated surveillance system that can be set up in existing traffic surveillance infrastructure. This system locates, counts and tracks vehicles in a variety of lighting conditions such as cloudiness and glares. Moreover, it adapts quickly to any changes of the background, as transition between different lighting conditions (i.e. from cloudiness to direct sunlight and vice versa), various traffic conditions including stop-and-go traffic flow as well as permanent changes to the background (for instance, when a vehicle has pulled over). This overcomes the weaknesses of previous systems described above.

A typical surveillance system consists of a traffic camera network, which processes captured traffic video on-site and transmits the extracted parameters in real time. In this study we focus on the algorithmic part of such a system.

The innovation of this study lies on the ability of the proposed algorithm to reconstruct the actual background color map without the need of any human intervention even in harsh traffic conditions, such as stop-and-go traffic flow, stopped vehicles (i.e. accident) and rain or snow. In our approach a new background prototype is constructed every 1 or 2 min, restricting the problem of background pollution to the interval between two consecutive updates. Each newly recreated background instance is assumed to be steady within the update period. Thus, the background instance is used as a prototype in order to separate the foreground from the image for each image frame within the update period.

This paper is structured as follows: In Section 2 a description of the system together with its specifications and the testing arrangement are given. Section 3 presents the Vehicle Detection Unit. Emphasis is given to the background reconstruction algorithm which is analyzed in detail. In Section 4 the Tracking Unit is presented. In Section 5 we present our experiments, which aim to support the basic assumptions of this work and to evaluate the developed background reconstruction algorithm. Finally, in Section 6 we summarize our results and we present our conclusions.

2. System conception

The innovative algorithm of background reconstruction is part of a contemporary and realistic surveillance system. The integrated system locates, tracks and extracts traffic parameters in real time. Furthermore, the system can utilize any existing traffic surveillance infrastructure without further modification or tuning (except for the camera calibration that calculates image metrics).

A typical road traffic surveillance infrastructure consists of a camera network that has the ability to transmit images in real time to a central operational center. The processing of the images can be

carried out on-site saving valuable network bandwidth as it transmits only the outcome of the calculations. Else, the whole process can be performed either in real time video streaming from an operational center or in already stored video material.

In such network installation, the cameras must be sited approximately 10–15 m or more above road level to minimize the effect of occlusion. The system must be adaptive to a series of perturbations that may affect the clarity of the captured video, such as vibrations of the camera and slow changes in the background due to lighting conditions, (Mimbela & Klein, 2000).

In order to simulate the algorithmic part of an integrated road traffic surveillance system, we used the following arrangement: A commercial CSS DV camcorder was installed about 10 m above road level and was sited above the central lane of the road, facing the traffic at an angle of 65°. The characteristics of this camera are as follows:

- 48 mm focal length equivalent to 35 mm camera, which defines a 27° vertical and 40° horizontal angle of view;
- 25 fps of 720 × 576 pixels in PAL video format.

Some predefined spots, on each test scene, were chosen in order to calibrate the camera according to DLT (direct linear transformation), a method originally reported in Abdel-Aziz and Karara (1971). The calibration of the camera defines the relationship between the 'real-world' and the pixel matrix of the digital image.

The architecture of the proposed system is described in Fig. 1. The system consists of two units namely the Vehicle Detection Unit and the Tracking Unit, the latter being indicated in gray color. Fig. 1 shows that first, a series of frames (raw traffic capture) enters the Vehicle Detection Unit (presuming that an initial background template has been created). Subsequently, the stream of frames feeds the background reconstruction algorithm in order to create the next background template that replaces the current after a predefined number of frames. While a new background template is created, the background in use is maintained using the simple adaptive filter that is applied in pixel level and proposed by Toyama et al. (1999):

$$B_t = (1-\alpha)B_{t-1} + \alpha I_t \quad (1)$$

where B_t is the color vector of the background model in the t frame, I is the actual color vector of the same pixel in the frame t and α is the coefficient that declares the rate of adaptation with values in the range 0–1.

In the main flow of the detection unit, the raw foreground information is derived by a background subtraction procedure, Fig. 1. The result of this step is a set of partly connected pixels, which must be further processed in order to form compact objects (clustering/convex hull, Fig. 1). If those pixels lie along the "Entrance Zone" (Fig. 2) a region growing algorithm (Davies, 2005) merges all those pixels that potentially belong to a common vehicle domain. Else, if the pixels from the background subtraction procedure lie within the Main Area, then they are further processed to form blobs (connected pixels that form a shape) using a convex hull procedure (Section 3.3). The detected blobs from this process are merged to form objects. Merging occurs to blobs that lie partly or totally within the frontiers of a recognized vehicle shape from the previous frame $t-1$, whose position has been appropriately corrected for frame t (vehicle matching – diagram of Fig. 1). Candidate vehicles are recognized by a cognitive clustering procedure (classifier – diagram of Fig. 1). This cognitive clustering process has the following concept: candidate object is considered to be a vehicle only if its location is consistent with its prior calculated trajectory and detected object dimensions remain unchanged through frames. A vehicle that does not match the previous frame, or seems to have an irregular trajectory must be rejected (i.e. a backward vehicle movement that cannot be explained by its trajectory).

Occlusion can be handled by simple rules of merging and splitting vehicle domains regarding their trajectory. Each detected vehicle belongs to one of the following classes: 'vehicle', 'large vehicle' or 'non-vehicle object'. 'non-vehicle' objects are not further tracked and are ignored by the system. Finally, if a previously de-

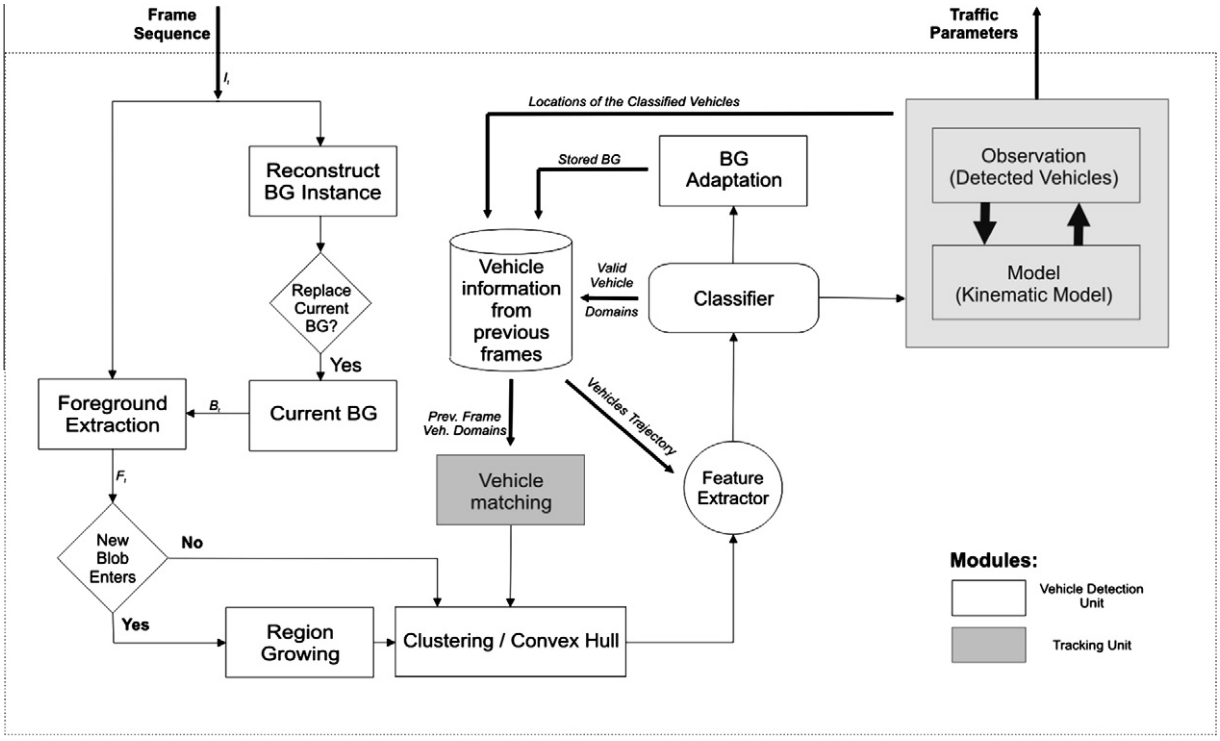

Fig. 1. Flow diagram of the algorithm.

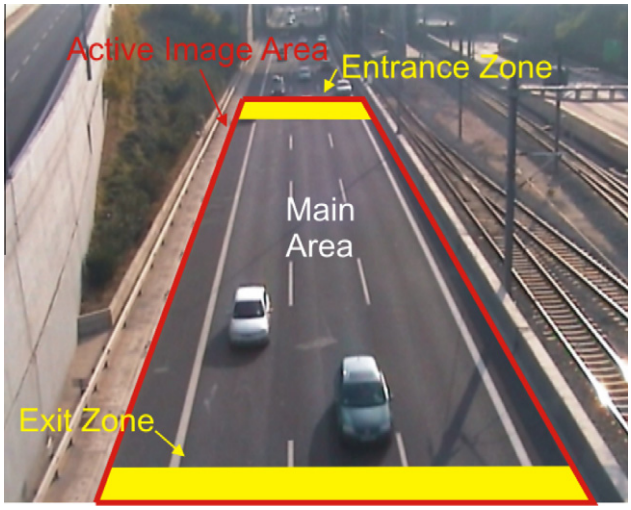

Fig. 2. The calculations take place in the pixels included in the active image area to save calculation time and increase efficiency. The active image area is divided to Main Area and to the entrance and exit zones.

tected vehicle touches the exit zone it is totally omitted from any further processing.

Regarding the Tracking Unit, the vehicles are identified through sequential frames (vehicle matching, Fig. 1) by maximizing the overlapping surface of the shape corresponding to frame t and that of $t-1$ (Criminisi et al., 2004). An array of the detected locations for each vehicle per frame is derived, which is then used to derive the vehicles trajectory. However, the calculated trajectories are expected to be distorted due to image noise. For this reason, a set of 2D-motion Kalman filter equations is applied to the extracted trajectories to make them smooth and coherent. The results from the tracking procedure feed the next step of the detection unit by providing the necessary information for the clustering and classification procedures.

3. Vehicle detection

The system is based on the well known algorithm of background subtraction, as mentioned in previous paragraphs. Typically a background subtraction algorithm is carried out in three steps:

- Initialization of the background.
- Foreground extraction.
- Background maintenance.

The most popular algorithms for background extraction found in the literature and used for comparison purposes in the current work are the Mixture Of Gaussians Model (MOG, Stauffer & Grimson, 1999, (Zivkovic & Van der Hijden, 2006) and the Codebook model (Kim et al., 2005). The MOG methodology models each pixel history as a cluster of Gaussian type distributions and uses an on-line approximation to update its parameters. According to this, the background is found as the expected value of the distribution corresponding to the most populated cluster (Stauffer & Grimson, 1999). This methodology is greatly improved on grounds of performance by considering recursive equations to adaptively update the parameters of the Gaussian model (Zivkovic & Van der Hijden, 2006). According to the Codebook model (Kim et al., 2005), sample background values at each pixel are quantized into codebooks that represent a compressed form of background model for a long image sequence. The codebook is enriched in new codewords in the presence of a new color that cannot be assigned to the existing groups.

Our approach is different from previously published works, in terms of the background handling during the initialization and maintenance step. The proposed method focuses on the calculation and reconstruction of the background template, based on pixel-level information obtained scatterly from a series of consecutive image frames. On one hand, this mechanism allows the background subtraction process to periodically obtain an updated background instance regardless the probable presence of foreground objects, and on the other hand, it guarantees the initialization of the background subtraction algorithm by providing an initial background instance under any circumstances. Hence, our approach unifies the first and last step of a typical background subtraction procedure. The following subsections present in detail the main algorithmic procedures proposed in this article for vehicle detection.

3.1. Background reconstruction

We propose a probabilistic algorithm to reconstruct the background of a traffic scene by eliminating the moving object information. According to this, the background color information of crowded scenes is dynamically retrieved by assessing color variation per pixel through a series of frames. The overall idea is based on the notion that a specific location is occupied by moving objects for a time period shorter than that for which it remains unoccupied.

The implementation of the algorithm has been done applying the $L^*u^*v^*$ color system (the first uniform color space adapted by the International Commission on Illumination in 1976), whose coordinates are related to the RGB values by non linear transformations. The L^* parameter is the lightness coordinate and the chromatic information is carried in the u^*, v^* parameters. This color system has been chosen because it defines a uniform color space with the perceived color differences measured by Euclidian distances (Comaniciu & Meer, 1997).

Loosely speaking, if a specific pixel in a series of frames would 'vote' for its color property, the majority of the 'votes' is expected to be concentrated in the chromatic neighborhood of the actual background. In mathematical terms, this 'voting' schema is described in detail in the following paragraphs.

Let $\overline{\Phi}$ denote the discrete color space, $\overline{\Phi} \equiv \{l = [L^*/h], u = [u^*/h], v = [v^*/h]\} \in Z^3$ (where '[]' is the floor operator and h is the chromatic distance defined by the bin dimensions), which is generated from the continuous $L^*u^*v^*$ color space $\Phi \equiv \{L^*, u^*, v^* \in \Re^3\}$, by considering cubic bins $b_{l,u,v}$, $\{l,u,v\} \in N^3$, all edges of which have length equal to h. Each discretization element $b_{l,u,v}$ is responsible for a continuous chromatic range of colors, where $l \leqslant L^*/h < l+1$, $u \leqslant u^*/h < u+1$ and $v \leqslant v^*/h < v+1$, with the value that corresponds to the discrete color parameters $l = [L^*/h]$, $u = [u^*/h]$, $v = [v^*/h]$.

Given a video capture that consists of F sequential frames of resolution $n \times m$, let $I_{ij}(t) = (I_{ij}^{L^*}(t), I_{ij}^{u^*}(t), I_{ij}^{v^*}(t))$, denote the color vector at the (i,j) pixel of the frame at time t, $I_{ij}^{L^*}(t), I_{ij}^{u^*}(t)$ and $I_{ij}^{v^*}(t)$ denote the L^*, u^*, v^* elements of $I_{ij}(t)$ respectively, and $B = \{B_{ij}\}$ the background color map. The pixel (i,j) color variation with respect to time is estimated by a sampling procedure, where the color values $I_{ij}(t)$ obtained by T consecutive frames, starting from t_0, are collected. Thus, the temporal sample $S_{ij}(t_0) = (I_{ij}(t_0), I_{ij}(t_0+1), \ldots, I_{ij}(t_0+T-1))$ of pixel (i,j) defines the frequency $\hat{f}_{ij}(l,u,v)$ of the examined pixel having color value belonging into the $b_{l,u,v}$ bin:

$$\hat{f}_{ij}(l,u,v) = \sum_{t=t_0}^{t_0+T-1} \delta\left(l - \left[\frac{I_{ij}^{L^*}(t)}{h}\right]\right)\delta\left(u - \left[\frac{I_{ij}^{u^*}(t)}{h}\right]\right)\delta\left(v - \left[\frac{I_{ij}^{v^*}(t)}{h}\right]\right) \quad (2)$$

where $l,u,v \in N$, $\delta()$ is the kronecher delta function.

The frequency $\hat{f}_{ij}(l_m,u_m,v_m)$ within the mode bin b_{l_m,u_m,v_m} corresponds to the most persistent color $I_m = (l_m,u_m,v_m)$ in a sequence of T frames for pixel (i,j). For this reason, our

approach assumes that this color represents the actual background B_{ij} of point (i, j). Thus, the reconstruction of the background is the problem of extracting the mode for each of the $n \times mS_{ij}$ samples:

$$B_{ij} = \arg\max[\hat{f}_{ij}(l,u,v)\rfloor = (l_m, u_m, v_m) \tag{3}$$

The methodology described is of $O(n^3)$ complexity in terms of memory and $O(sn^3)$ in terms of calculations involved (where s denotes the total number of frames in the sample and n represents the magnitude of discretization for each color parameter). In normal traffic conditions a 100–200-frames sample corresponding to 4–8 s of traffic observation is adequate for the identification of the actual background. However, in general conditions where the vehicle flow is dense having low speed and/or involving stop-and-go behavior, the demanded sample size is expected to be higher. Our tests in such conditions showed that an average of 1250 frames, corresponding to 1 min capture length may be required to reliably reconstruct the actual background. In these cases the sample size includes a vast volume of information and therefore demands an increased memory capacity, which may be prohibitive for the design and operation of the system.

The limitations posed by the hardware motivated us to seek for a more efficient way to solve the problem while keeping the memory usage and the required amount of calculations within acceptable limits. Towards this goal, our research focused on a different approach of managing the discrete temporal chromatic information l, u, v of the chromatic space $\overline{\Phi}$. Thus, we calculated the frequencies $\hat{f}_{ij}^{(l)}(l), \hat{f}_{ij}^{(u)}(u), \hat{f}_{ij}^{(v)}(v)$ for each l, u, v parameter separately, through the following summations:

$$\begin{aligned}
\hat{f}_{ij}^{(l)}(l) &= \sum_{u=u_{\min}}^{u_{\max}} \sum_{v=v_{\min}}^{v_{\max}} \hat{f}_{ij}(l,u,v), \\
\hat{f}_{ij}^{(u)}(u) &= \sum_{l=l_{\min}}^{l_{\max}} \sum_{u=u_{\min}}^{u_{\max}} \hat{f}_{ij}(l,u,v), \\
\hat{f}_{ij}^{(v)}(v) &= \sum_{l=l_{\min}}^{l_{\max}} \sum_{v=v_{\min}}^{v_{\max}} \hat{f}_{ij}(l,u,v),
\end{aligned} \tag{4}$$

where $l_{\max}$, $l_{\min}$, $u_{\max}$, $u_{\min}$, $v_{\max}$ and $v_{\min}$ are the maximum and minimum values of the discrete l, u, v parameters respectively.

The calculation of the frequencies above Eq. (4) provides information on the reconstruction of the background. According to the proposed methodology, the most persistent color value in a sequence of frames, for a specific pixel, is the one that is most likely to represent the actual background, and can also be calculated by maximizing Eq. (2). Alternatively, it can be approximated by composing an artificial color, which is composed by each one of the frequency modes (l_{mode}, u_{mode}, v_{mode}) maximizing Eq. (4):

$$\begin{aligned}
B_{ij} &= \left(\arg\max\left[\hat{f}_{ij}^{(l)}(l)\right], \arg\max\left[\hat{f}_{ij}^{(u)}(u)\right], \arg\max\left[\hat{f}_{ij}^{(v)}(v)\right]\right) \\
&= (l_{\text{mode}}, u_{\text{mode}}, v_{\text{mode}})
\end{aligned} \tag{5}$$

The whole idea is implemented on the reconstruction of the actual background based on the clustering of pixel temporal color values into two basic classes: 'Background' and 'non-background'. One of the most efficient methodologies for clustering color information is the popular methodology of mean-shift, introduced by Abdel-Aziz (1971). However, this methodology involves a vast amount of calculations for the set of color values that correspond

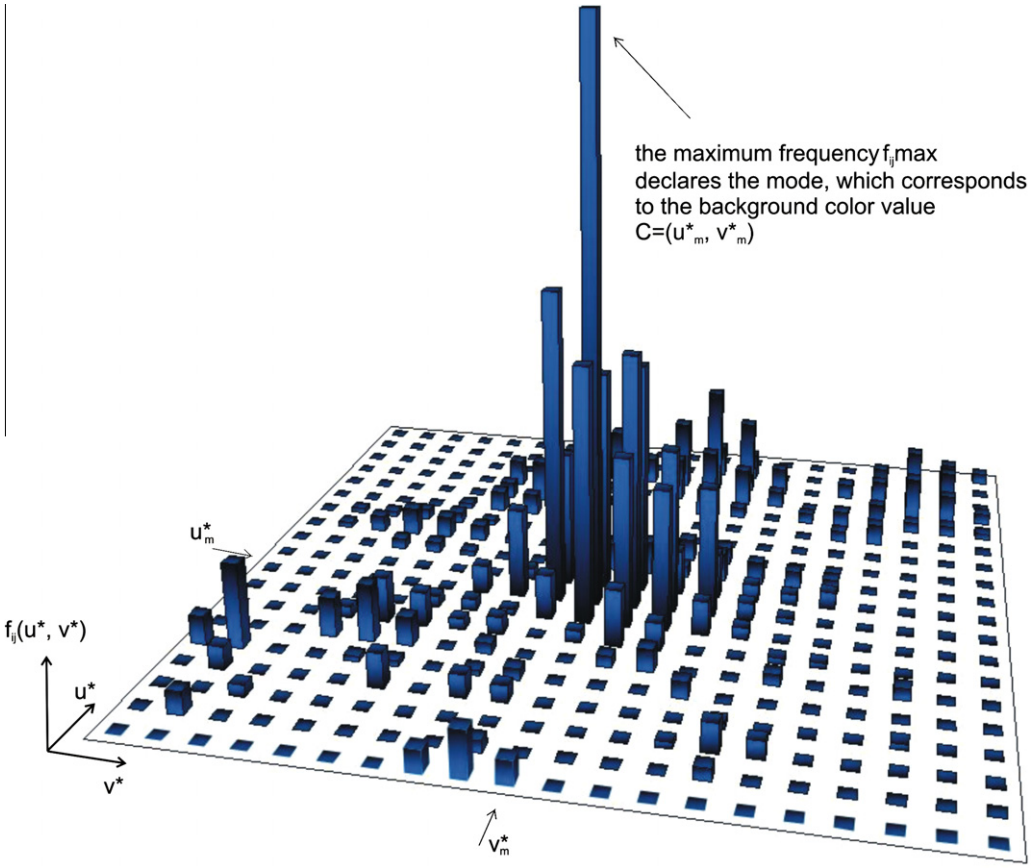

Fig. 3. Illustrative example of the principles of the background reconstruction methodology: in a 2D distribution (v^*u^* plane) of image color values, the majority is concentrated around a value (mode) that represents the background color.

to a single pixel, making a solution for the whole image not realistic. Additionally, the memory required is of the same complexity as the one required for Eq. (2).

To overcome this obstacle we propose to use Eq. (4) to carry out this clustering in a more flexible manner that is alleviated by the main characteristics of the problem: (i) the distribution of sampling in color space $\overline{\Phi}$ is extremely sparse as each sample involves, say, in a very extreme case, 20,000 color values compared to a total of 414,720 color values involved in a common PAL 720×576 pixels image, (ii) the majority of these values is concentrated in the vicinity of the background color cluster (see Fig. 3), while color values representing other objects are scattered throughout the color space.

The main goal is to detect the background color cluster and to locate its mode (the local maximum of a distribution of values). In this case the concentration of values in the background cluster can be roughly estimated by the integration of frequencies Eq. (4) because the overall contribution of other color clusters to the calculations is negligible.

To better illustrate the mode location based on our methodology we shall employ the example of Fig. 3. The background color cluster is the dominant in the distribution and this becomes obvious in this example: The color values are accumulated around a core, where the density of color values forms a steep peak. On the contrary, the foreground colors tend to be distributed following the universal distribution and for that reason they do not form remarkable value concentration around a color (attractors).

The calculation of frequencies $\hat{f}_{ij}^{(l)}(l), \hat{f}_{ij}^{(u)}(u), \hat{f}_{ij}^{(v)}(v)$ of Eq. (4) requires first the calculation and storage of the overall frequency function $\hat{f}_{ij}(l,u,v)$ of Eq. (2), which is of $O(n^3)$ complexity in terms of memory and $O(sn^3)$ in terms of calculations involved. For that reason the frequencies $\hat{f}_{ij}^{(l)}(l), \hat{f}_{ij}^{(u)}(u), \hat{f}_{ij}^{(v)}(v)$ can be equally derived from the histograms $H_l^{L^*}, H_u^{u^*}, H_v^{v^*}$:

$$
\begin{aligned}
H_l^{L^*} &\equiv \hat{f}_{ij}^{(l)}(l) = \sum_{t=t_0}^{t_0+T-1} \delta\left(l - \left[\frac{I_{ij}^{L^*}}{h}\right]\right) \\
H_u^{u^*} &\equiv \hat{f}_{ij}^{(u)}(u) = \sum_{t=t_0}^{t_0+T-1} \delta\left(u - \left[\frac{I_{ij}^{u^*}}{h}\right]\right) \\
H_v^{v^*} &\equiv \hat{f}_{ij}^{(v)}(v) = \sum_{t=t_0}^{t_0+T-1} \delta\left(v - \left[\frac{I_{ij}^{v^*}}{h}\right]\right)
\end{aligned}
\tag{6}
$$

where $l, u, v \in N$ and $\delta()$ is the kronecher delta function.

The calculations involved in Eq. (6) are of $O(n)$ complexity in terms of memory and $O(tn)$ in terms of calculations involved (t denotes the time size of the sample and n represents the magnitude of discretization for each color parameter). Hence, the proposed methodology is realistic in design and operational level because it does not depend on a predefined empty scene (initialization step), but it dynamically calculates the background template. Furthermore, the complexity reduction of problem Eq. (2) to problem Eq. (6) makes possible the real time operation of the system under typical hardware infrastructure.

3.2. Foreground extraction

The foreground extraction is one of the standard procedures of a typical background subtraction algorithm. In this stage, the foreground is being extracted by comparing each frame with the instance of the background. The simplest way to perform this operation is to calculate the chromatic difference for each pixel between the current frame and the background template. Thus, each pixel for which the chromatic difference is greater than a predefined threshold is classified as the foreground mask M_{ij}.

In our work the chromatic difference between the current frame and the background model B_{ij} is defined by a norm that combines the difference in lightness L^* with the chromatic difference of the u^*, v^* parameters in $L^*u^*v^*$ color space. The foreground mask M_{ij} is calculated then by the following relation:

$$
M_{ij} = \begin{cases} 1, & \left|I_{ij}^{L^*} - B_{ij}^{L^*}\right| > \text{threshold} \wedge ||I_{ij}^{u^*,v^*} - B_{ij}^{u^*,v^*}|| > \text{threshold} \\ 0, & \text{elsewhere} \end{cases}
\tag{7}
$$

where $||I_{ij}^{u^*,v^*} - B_{ij}^{u^*,v^*}||$ is the Euclidean Norm in terms of chromatic parameters u^*, v^* of the current frame and the background model.

The pixels belonging to the foreground mask are grouped together to form connected components. Usually the connected components are further processed in order to remove holes or other irregularities. Although, the most common practice is application of a morphological filter (Davies, 2005), it demands valuable calculation time. Thus, we have utilized a convex hull algorithm for shaping and forming objects (see next Section 3.3).

3.3. Shaping and clustering

At this step, the extracted foreground segments belonging to a common object are grouped and shaped. The grouping process is a complex procedure, especially for vehicles that have just passed the entrance zone (see Fig. 2) for which there is no prior information on their trajectory. In this case, the segments are grouped based on their spatial characteristics via a region growing algorithm. The vehicles that have already passed the entrance zone prior information is available and can be appropriately utilized in order to group segments that belong to the same vehicle. The grouped segments are further processed to form compact and convex vehicle shapes via an appropriate convex hull algorithm, as described below.

3.3.1. Convex hull

The extracted objects (connected components) usually are not compact and their shapes are likely to be non-convex with a broken surface having holes and cavities or/and broken into two or more pieces. In many cases the extracted objects are just artifacts of noise. This can lead to miscalculation of the amount of vehicles existing in the image frame and inconsistency with previous images. For instance an artifact of noise can be perceived as a vehicle that only appears in one or more frames and suddenly vanishes in the next one ('ghosts'). A popular technique to repair this kind of problem is morphological filters (erosion, dilution and combinations of them). We have chosen to use a convex hull technique to deal with doughnut-like object that are commonly encountered in scenes where vehicles participate. In addition the $O(n\log n)$ complexity of the convex hull algorithm allows very fast computations.

To avoid such undesirable events a filtering procedure is repeatedly applied that is based on a convex hull algorithm. In our approach a Graham Scan convex hull algorithm (Graham & Yao, 1983) is employed in order to plot the convex hull for each set of pixels and finally to form a connected component. This algorithm is applied repetitively until there are no more sets of pixels to be merged. Furthermore, after clustering has grouped all convex segments a hull algorithm is applied to form convex compact objects. The outcome of this process is a mask of convex polygons (first and second row of Fig. 4). These polygons represent compact vehicles, but in many cases these polygons correspond to vehicle segments that can be merged through the following clustering procedure.

3.3.2. Clustering

The outcome of the foreground extraction procedure is more likely to be vehicle segments than compact vehicle shapes. This

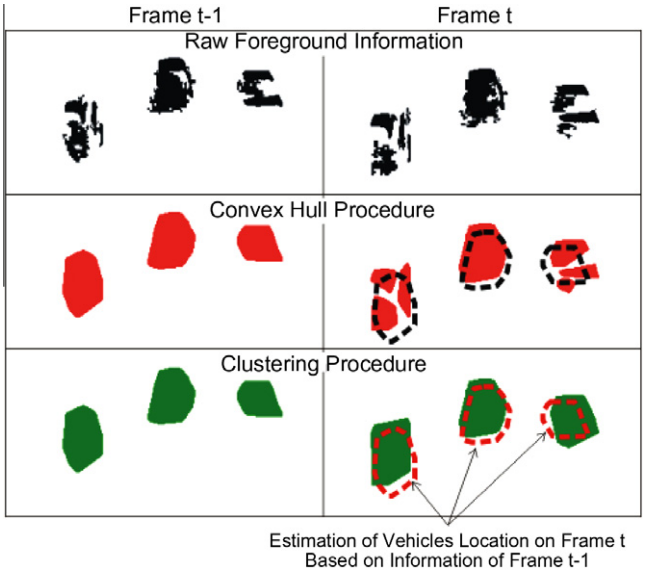

Fig. 4. Clustering procedure: first row: group of pixels extracted from the background subtraction procedure; second row: the convex hull of the first row; and third row: clustering using the expected positions of the vehicles regarding their trajectory and application of the convex hull algorithm to form compact objects.

is a result of shadows, glares and vehicle colors that are quite similar to the background. Thus, the clustering procedure aims to group such segments to form a unique and compact vehicle. To achieve this goal prior information is utilized. The trajectories of already detected vehicles in previous frames are used to calculate vehicles motion and consequently to estimate their locations in the current frame. Then, we merge the segments of this frame that contact the traces of the estimated vehicle locations (see Fig. 4).

When a vehicle is entering the image (entrance zone – Fig. 2) no prior information is available (e.g. dimensions and trajectory). For that reason, the entering segments are simply clustered by a popular clustering algorithm namely, region growing. Region growing algorithm is a heuristic for clustering segments based on their locations. It lies on the simple assumption that two segments are merged if the eventually formed shape can belong to a vehicle according to its dimensions.

3.3.3. Shadows and glares

We have used the approach of Horprasert et al. (1999) implemented in *LUV* space. According to this we compare the chromaticity and brightness of a pixel with the corresponding pixel of the background model. The chromaticity of a pixel is defined by the *UV* component, while the brightness is determined by the lightness component *L*. In case that the chromaticity is similar but the brightness differs, we distinguish two cases: the brightness differs significantly from the background, in that case this pixel cannot be classified as a shadow or glare, and otherwise if the brightness is less than the background it is classified as a shadow else if the brightness is higher than the background it is classified as a glare.

3.4. Classification and occlusion handling

The system classifier distinguishes the detected vehicles into two classes, namely 'vehicle' and 'large vehicle', by assessing their dimension and trajectory. The classifier rules are based on the following simple assumptions: Firstly, it is certified that the detected object was present in the previous frame and the two shapes (previous and current) match (see Section 4). Then, it is examined whether the vehicle position is consistent with vehicle motion and recorded trajectory. The new position of the object should satisfy the motion model of the vehicle, which is derived from the recorded trajectory. Same as before, its dimensions should match. Otherwise, the detected object is rejected.

In case of considerable discrepancies of dimensions it is examined whether the vehicle under consideration occludes another one. The rules for the occlusion are adopted from Criminisi et al. (2004), where a graph is constructed that associates the nodes $C_{i,t-1}$ (vehicle i at frame $t-1$) with the detected objects $P_{i,t-1}$ (object i at frame t). Subsequently, the objects can be merged or splitted. A merging is the occlusion of two (or more vehicles) in the current frame, whereas a split is when two previously occluded vehicles are separated. If the detected objects are consistent with the association graph, then the detected object is approved to be a vehicle.

However, the classification of the vehicle in one of the aforementioned classes is performed only after its first contact with the exit zone. It is then that the length of the detected vehicle is compared with the length of a prototype; if their ratio is much larger than one then it is classified as a 'large vehicle', else as a 'vehicle'.

4. Tracking

Tracking is a very important issue in computer vision. Recently, there is a profound interest in surveillance applications. The aim of tracking in computer vision is to recognize and locate a prototype in a series of sequential frames. A lot of applications are based on tracking such as video processing, security, surveillance and automatic procedures. In our case, we need to track multiple vehicles to record their trajectory and derive relevant information such as vehicle speed, direction and driver behavior. Such tracking methodologies are the mean-shift algorithm and template matching.

Mean-shift algorithm is originally introduced by Comaniciu as a segmentation methodology (Abdel-Aziz, 1971) before it was appropriately modified to a robust tracking system (Comaniciu, Ramesh, & Meer, 2000). The main idea for a mean-shift tracking algorithm is to plot a 2D probability space where an object template can be located. Similarly, the template matching algorithm aims to locate the maximum in a 2D probability space in order to specify the location of a predefined template. Although, the two methodologies have the same principles: the mean-shift accelerates the procedure by comparing histograms instead of template pixel by pixel comparison. Both template matching and mean-shift algorithms are robust in tracking a predefined template. The main weakness of these algorithms is the lack of flexibility when tracking is influenced by image perspective.

More specifically, the template of a vehicle changes both in size and resolution while passing through the image active area. Moreover, a template does not consist of the vehicle figure only, but usually part of the background is also present in the template. The template becomes less accurate when the vehicle is located in the depth of the image. This deteriorates the efficiency and the accuracy of this process. Empirically, we found that our simple matching algorithm is more efficient than such a complex tracking algorithm.

We tested template matching and mean-shift in order to enhance the vehicle matching procedure but found that both algorithms presented problems. The first drawback was that they demanded a large amount of calculations and they suffered instability problems. These result in loss of the tracking object. The main cause for instability is the template update: while the vehicle moves towards the camera its shape becomes larger, and as a consequence its resolution becomes better than the template's. Hence,

a simple matching methodology was found to meet better the needs of this problem.

The matching procedure adopted in this study is similar to (Criminisi et al., 2004) and is based on the assumption that the next position for a vehicle can be estimated by its motion. According to this, we estimate the positions of previous frame vehicles and we draw their traces in the current frame. Then a vehicle V_1 having mask M_1 matches with vehicle V'_1 having mask M'_1 from previous frame only if $M_1 \cap M'_1 \neq \varnothing$. If there is a conflict between two vehicles then the matching vehicle is the one that maximizes the common surface.

Even with the most accurate algorithm for locating templates in an image, small drifts and miscalculations due to conversion of distances in the image discrete space into real conditions result in

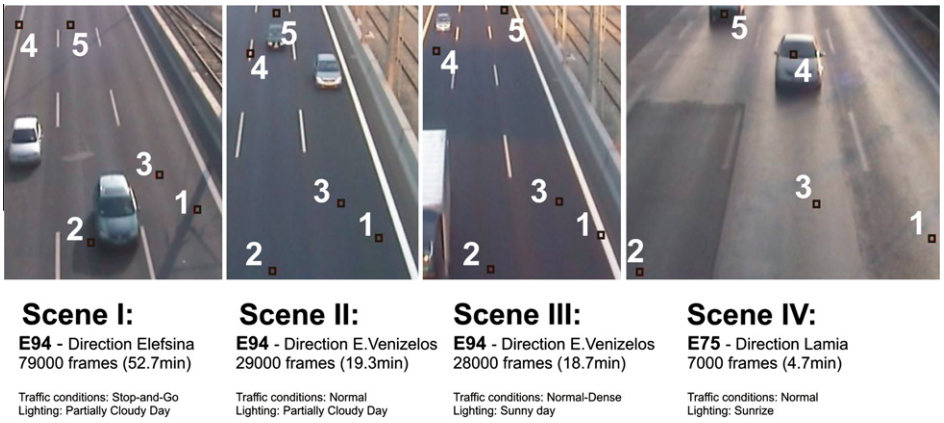

Fig. 5. The snapshots above were taken from the four captures used for evaluation. In each snapshot, five pixel positions (1–5) have been chosen (three pixels at the front of the image and two at the back). Note that some of the pixels are sited over the white stripe of the road in order to study the behavior of the system in a color different from the asphalt. Evaluation results are presented in Tables 1 and 2 as well as at Fig. 6.

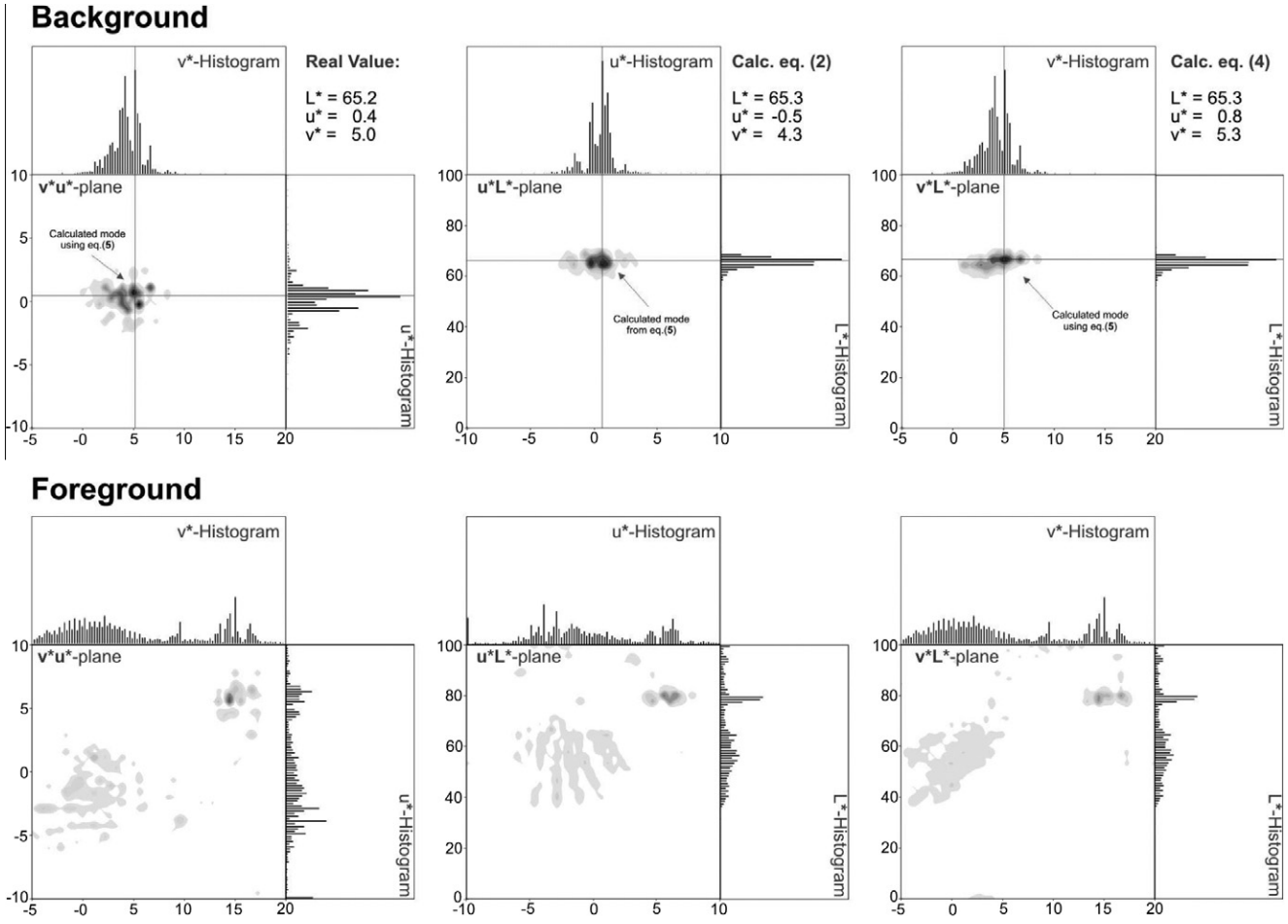

Fig. 6. The 2D-topology of the pixel series (PC_k, k = 0, 1, ...) of Scene I at pixel 1 of Fig. 5: top, middle and bottom rows present "background", "foreground" and all elements respectively. The side bar graphs in each topology correspond to the color parameter histogram.

small errors in measurements. In order to form a more accurate and smooth trajectory for each vehicle a Kalman filter algorithm is employed.

Kalman filter employs a procedure that a state variable is repetitively predicted according to a theoretical model and is subsequently corrected by an actual measurement. The state variable of the described system is a vector of vehicle location, speed and length. In our approach we assumed simple constant straight motion along the direction of the road, a rational approach for traffic in avenues and national roads. In addition, we assume constant vehicle length for our kinematic model along its trajectory.

5. Experimental results – evaluation

In order to validate the effectiveness of the background reconstruction algorithm we created an evaluation process which aims to provide evidence supporting the color that is more frequent at a specific pixel in a series of frames is more likely to belong to background rather than foreground.

For that reason, we created a testing group of four video captures (Scene I-79,000 frames, Scene II-29,000 frames, Scene III-28,000 frames and Scene IV-7000, see Fig. 5). In each scene five pixel positions have been chosen, two at the back of the image and three at the front. For each pixel position of a specific scene two arrays were constructed, as described below:

The first array is a collection of color values $PC^{p}_{k,s}$ (PC = [pixel color, class], p = testing pixel, k = array index = 1, 2, ..., s = scene) appropriately classified as one of the following classes: 'Foreground' or 'Background'. This array comprises a sampling of the color values collected at the pre-selected pixels for each testing scene taken in a 500 frames interval. According to this, the 1st, 500th, 1000th, ... frames (of each testing scene and each pre-selected pixel position) were manually extracted and classified to construct the $PC^{p}_{k,s}$ array.

The second array consists of the background color values at the pre-selected pixels of the testing scenes $BG^{p}_{k,s}$ (BG = background color, p = testing pixel, k = array index = 1, 2, ..., s = scene). As in the first array, the background color values of the 1st, 500th, 1000th, ... frame (of each testing scene and each pre-selected pixel) were recorded in order to form the $BG^{p}_{k,s}$ array whenever this was possible (when the testing pixel was not obstructed by foreground objects). If the testing pixel was obstructed by a foreground object we sought for the nearest frame where the testing pixel could clearly be defined.

The graphical representation of the first array $PC^{p}_{k,s}$ (testing scene I, pixel 1) is presented in Fig. 6. The topology is analyzed into the three combinations of planes: $v^{*}u^{*}$, $u^{*}L^{*}$ and $v^{*}L^{*}$ and for each

Table 1
Comparison with other models.

Scenes	MOG (Stauffer & Grimson, 1999; Zivkovic & Van der Hijden, 2006)		Codebook Kim et al., 2005		This work	
	FG (%)	BG (%)	FG (%)	BG (%)	FG (%)	BG (%)
Scene I E94 – direction Elefsina stop-and-go traffic conditions (79,000 frames capture)	74.3	99.6	92.5	93.6	97.1	98.3
Scene II E94 – direction El.Venizelos normal traffic conditions (29,000 frames capture)	82.4	98.7	94.9	92.2	95.0	99.0
Scene III E94 – direction El.Venizelos normal and dense flow traffic conditions (28,000 frames capture)	80.7	98.7	93.4	91.1	94.2	98.7
Scene IV E75 – direction lamia dense normal and dense flow traffic conditions (7000 frames capture)	77.0	96.6	88.1	93.8	91.2	97.1

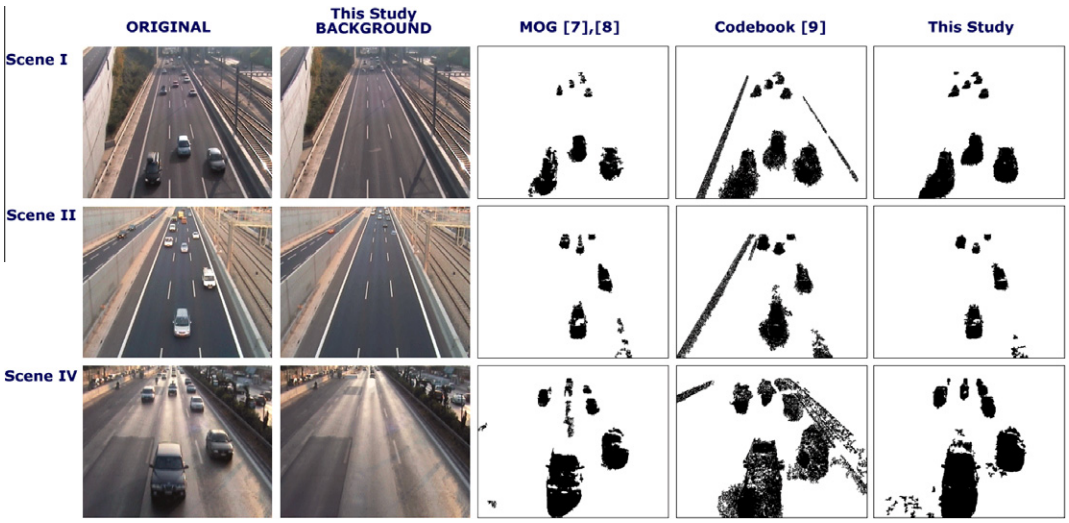

Fig. 7. Visual presentation of compared methodologies (see Table 1).

Table 2
Background Reconstruction process outcome.

Scenes	Test pixel 1 of Fig. 5 Mean color value (first row), standard deviation (second row)						Test pixel 5 of Fig. 5 Mean color value (first row), standard deviation (second row)					
	L^*		u^*		v^*		L^*		u^*		v^*	
	Experimental	Our work	Experimental	Our work	Experimental	Our work	Experimental	Our work	Experimental	Our work	Experimental	Our work
500-frames sample background reconstruction												
Scene I												
E94 – direction Elefsina stop-and-go traffic conditions (79,000 frames capture)	65.7 2.1	66.6 8.7	0.5 1.3	0.4 1.8	4.3 2.0	4.5 2.8	73.9 2.1	75.2 2.0	1.1 12.8	0.5 3.7	3.1 8.7	3.5 2.8
Scene II												
E94 – direction El.Venizelos normal traffic conditions (29,000 frames capture)	60.0 1.7	60.4 0.7	−3.9 2.1	−4.0 1.2	−2.5 2.3	−2.9 1.1	71.0 1.7	71.5 2.3	−3.1 0.7	−3.2 1.1	−0.1 0.7	−0.2 1.1
Scene III												
E94 – direction El.Venizelos normal and dense flow traffic conditions (28,000 frames capture)	59.7 1.7	60.1 1.2	0.8 1.4	1.6 1.2	−2.0 2.2	−1.7 2.0	74.2 1.7	74.4 2.2	3.8 1.2	3.7 2.0	6.2 1.2	6.2 2.0
Scene IV												
E75 – direction lamia dense normal and dense flow traffic conditions (7000 frames capture)	63.7 0.6	63.8 0.5	−0.3 1.3	−0.3 0.3	3.5 0.8	3.9 0.3	81.0 0.6	87.8 0.8	1.9 0.5	2.7 0.3	12.0 0.5	14.3 0.3
2500-frames sample background reconstruction												
Scene I												
E94 – direction Elefsina stop-and-go traffic conditions (79,000 frames capture)	66.8 1.2	66.0 1.2	0.7 1.1	0.5 0.6	4.8 1.7	4.5 1.4	75.6 1.2	75.1 1.7	1.4 15.2	0.6 2.0	4.2 1.2	3.4 1.4
Scene II												
E94 – direction El.Venizelos normal traffic conditions (29,000 frames capture)	59.5 2.0	60.3 0.8	−3.0 1.3	−4.1 1.3	−1.9 2.2	−2.9 0.9	71.7 2.0	71.4 2.2	−2.9 0.9	−3.2 0.7	0.7 0.8	−0.3 0.9
Scene III												
E94 – direction El.Venizelos normal and dense flow traffic conditions (28,000 frames capture)	59.5 0.7	60.0 1.4	0.7 0.5	1.6 1.1	−3.0 1.7	−1.9 2.0	73.3 0.7	74.2 1.7	3.9 1.3	3.6 1.9	5.7 1.4	6.2 2.0
Scene IV												
E75 – direction lamia dense normal and dense flow traffic conditions (7000 frames capture)	63.0 0.4	64.3 0.7	−1.4 0.4	−0.3 0.4	3.0 0.6	4.0 0.0	85.7 0.4	87.6 0.6	1.8 0.7	2.7 0.4	14.3 0.7	14.3 0.0

parameter L^*,u^*,v^* the corresponding histogram H^{L*}, H^{u*}, H^{v*} is generated. In each diagram the darkest areas represent high concentration of values, which also corresponds to high values at the side histograms.

The distribution of $PC^p_{k,s}$ array elements that have been classified as 'Background' and 'Foreground' are presented in the first and second row respectively. In the last row the $PC^p_{k,s}$ series are illustrated independent of their classification.

It can be clearly seen that in the first row the distribution of the background color values is densely populated around a central value, which is the mode of this distribution. The mode can also be located from the side bar graphs, where the bars are steeply

increased around a narrow interval. On the contrary, in the second row (foreground values distribution), the side bar graphs tend to be flat with multiple modes dispersed uniformly in the 2D parameter space. When the two distributions are mixed, the color that is

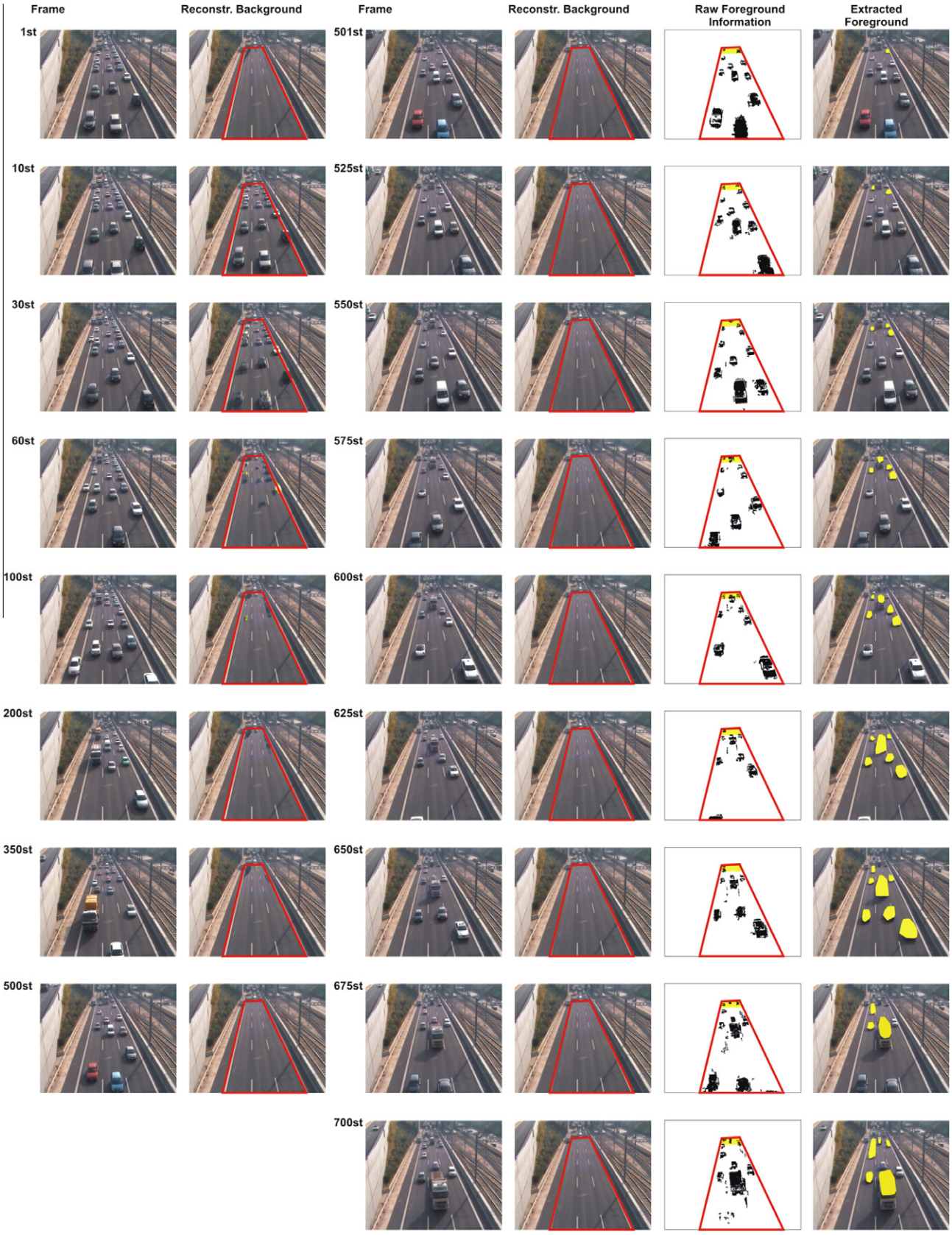

Fig. 8. Background reconstruction process in a 500 frames interval.

present in the majority of frames (so its color value is distributed in a narrow zone of a central value) is the background color. Moreover, the side 1D histograms H^{L^*}, H^{u^*}, H^{v^*} can precisely locate the background color (using Eq. (5) providing almost the same results as in the 2D distributions.

In order to test further our methodology we carried out the following test: We processed the data of Scenes I–IV by applying the background reconstruction algorithm to implement the 3D and 1D problem – Eqs. (3) and (5), respectively for a specific test frame, chosen for each scene (I–IV, Fig. 5). Moreover, the proposed background subtraction algorithm was tested against two of the most popular algorithms found in literature, that is to say MOG and codebook. For the case of MOG model the Mahalanobis distance was used to account for problems where the standard deviation of the Gaussian distribution is high. The results are presented in Table 1. For all scenes, the percentage of successfully detecting the foreground and background pixels is given for all methods tested. The methodology proposed outperforms all previous algorithms. Visual presentation of the results is given in Fig. 7 (Scene III was left out since the actual scene data were similar to scene II – same location). In addition, the performance of the suggested algorithm was faster since it did not involve the computational burden of adopting the MOG cluster parameters or the enrichment of codebook codewords.

Background reconstruction is a statistical methodology, therefore the identification of the sample size is important. In general, the sample size should be large enough to carry enough information for extracting the background color in each image pixel. To achieve this goal the sample size, in terms of time, should exceed the average time that a passing vehicle occupies any pixel in the image. In our test we chose a 500 frames and a 2500 frames sample, translated in terms of time to a 20 and 100 s exposure correspondingly.

The 500 frames sample is quite satisfactory for a highway where vehicles' average speed is about 80 km h^{-1} (22.22 ms^{-1}) and the occupation of a specific pixel close to the camera is expected to be less than a second. We chose the 2500 frames sample in order to examine if overexposure can improve the reconstruction procedure. We observe that the choice of a larger sample size tends to decrease the accuracy (Tables 1 and 2) for two reasons: first, as the sample size increases, the background color changes following the diminutive change of lighting (this also explains the differences on the measured values in different sample sizes for the same pixel) and second, the larger the sample size the more the inserted noise and the subsequent degradation of the background reconstruction performance.

The proposed system was implemented and tested as it is shown in Fig. 8, where the result of the Scene I experiment is presented. This result is also published on the internet corresponding author's personal page http://www.users.ntua.gr/nmand/BGReconstruction.htm. Moreover, the background reconstruction procedure for the test scenes I–IV is also included in the same web page.

Overall, the system was found to work satisfactorily and the background reconstruction algorithm added robustness to the process. In normal traffic conditions the system responded well and the outcome results regarding vehicle speed and trajectory were accurate enough. The maximum number of vehicles detected and tracked simultaneously for the heavy traffic instances of scene 1, was 10.

6. Conclusion

In this study we presented a system that implements a classical computer vision algorithm, the background subtraction, appropriately modified for the purposes of a traffic surveillance system. The innovation of this study lies on a new algorithm for reconstruction of the actual background. This algorithm is based on statistical color sampling per pixel over time. This algorithm is robust in reconstructing the actual background, even in real time. This was achieved due to algorithm's low complexity: $O(n)$ complexity in terms of memory and $O(tn)$ in terms of calculations involved (t denotes the time size of the sample and n represents the magnitude of discretization for each color parameter). The experiments carried out showed that the proposed algorithm is capable of real time operational working due to its low complexity.

The reconstruction of a new background instance, wherever this is required, enhances the typical background subtraction algorithm. Thus, in our approach the implementation of the background subtraction does not depend on an initial background instance and for that it has broadened its applicability. One of the main advantages of the proposed system is that it can be applied in an existing traffic surveillance system without substantial modifications and the background reconstruction algorithm allows the unobstructed operation of the system without human intervention. The system works well either in real time mode or in already stored video.

The testing arrangement used, which simulates the operation of a traffic surveillance system, was found to work satisfactory in outdoor diverse lighting conditions. In all cases background reconstruction algorithm managed to accurately reconstruct the actual background in various harsh conditions including heavy congestion and changes in the lighting. This methodology added robustness to the traditional background subtraction algorithm and overcame known instability issues.

In future work, we aim to focus on night surveillance, where some primary tests leave space for improvement on the existing algorithms reported in literature. However, the other modules of our proposed system should be improved, focusing on the occlusion handling and vehicle matching procedure. Moreover, it remains a challenge to utilize the capabilities of the proposed algorithm to other kind of machine vision problems, such as security, remote sensing, ship surveillance and a plethora of surveillance applications.

References

Abdel-Aziz, Y.I., Karara, H.M. (1971). Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proceedings of the symposium on close-range photogrammetry* (pp. 1–18). VA: American Society of Photogrammetry.

Colombari, A., Cristani, M., Murino, V., Fusiello, A. (2005). Exemplar-based background model initialization. *ACM workshop on video surveillance and sensor networks VSSN*, pp. 29–36.

Comaniciu, D., Ramesh, V., Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *Proceedings of the IEEE conference on computer vision and pattern recognition, Hilton Head, SC. Vol. 1*. pp. 142–149.

Comaniciu, D., & Meer, P. (1997). Robust analysis of feature spaces: Color image segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 750–755.

Criminisi, A., Perez, P., & Toyama, K. (2004). Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing, 13*, 1200–1212.

Cucchiara, R., Piccardi, M. (1999). Vehicle detection under day and night illumination. In *Proceedings of 3rd international ICSC symposium on intelligent industrial automation*.

Davies, E. R. (2005). *Machine vision* (3rd ed.). San Francisco, US: Elsevier Inc.. p. 104.

Davies, E. R. (2005). Mathematical morphology. In *Machine vision* (3rd ed., pp. 233–261). San Francisco, US: Elsevier Inc.

Graham, R., & Yao, F. (1983). Finding the convex hull of a simple polygon. *Journal of Algorithms, 4*, 324–331.

Gupte, S., Masoud, O., Martin, R., & Papanikolopoulos, N. (2002). Detection and classification of vehicles. *IEEE Transactions on Intelligent Transportation Systems, 3*, 37–47.

Gutchess, D., Trajkovic, M., Kohen-Solal, E., Lyons, D., Jain, A. 2001. A background model initialization algorithm for video surveillance. In *Proceedings of the eighth international conference on computer vision* (*Vol. 12*, pp. 733–740). Vancouvier, Canada.

Haritaoglu, I., Harwood, D., & Davis, L. (2000). W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 809–830.

Horprasert, T., Harwood, D., Davis, L. (1999). A statistical approach for real-time robust background subtraction and shadow detection. *IEEE ICCV FRAME-RATE workshop*.

Kastrinaki, V., Zervakis, M., & Kalaitzakis, K. (2003). A survey of video processing techniques for traffic applications. *Image and Vision Computing, 21*, 359–381.

Kim, K., Chalidabhongse, T., Harwood, D., & Davis, L. (2005). Real-time foreground–background segmentation using codebook model. *Real-Time Imaging, 11*(3), 172–185.

Koller, D., Weber, J., Makik, J. (1994). Robust multiple car tracking with occlusion reasoning. In *Proceedings of the third European conference on computer vision* (pp. 189–196). Stockholm, Sweden, May 2–6.

Lindeberg, T. (1996). Scale-space: A framework for handling image structures at multiple scales. *CERN School of Computing*, 695–702.

Mahmassani, H., Haas, C., Zhou, S., Peterman, J. (2001). Evaluation of incident detection methodologies. CTR report: http://www.utexas.edu/research/ctr/pdf_reports.

Mimbela, L., Klein, (2000). Non-intrusive technologies. In A summary of vehicle detection and surveillance technologies used in intelligent transportation systems (1st ed., pp. 5.1–5.27). Federal Highway Administrations (FHWA) Intelligent Transportation Systems Joint Program Office: Washington, DC, US.

Pless, R. (2005). Spatio-temporal background models for outdoor surveillance. *EURASIP Journal on Applied Signal Processing, 14*, 2281–2291.

Senior, A., Tian, H.A.Y., Brown, L., Pankanti, S., Bolle, R. (2001). Appearance models for occlusion handling. In *2nd IEEE workshop on performance evaluation of tracking and surveillance PETS*.

Stauffer, C., & Grimson, W. (1999). Adaptive background mixture models for real-time tracking. *Computer Vision Pattern Recognition*, 246–252.

Toyama, K., Krumm, J., Brumitt, B., & Meyers, B. (1999). Wallflower: Principles and practice of background maintenance. *ICCV99*, 255–261.

Wren, C., Azarbayejani, A., Darrell, T., & Pentland, A. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*, 780–785.

Zivkovic, Z., & Van der Hijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters, 55*(5), 773–780.