

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Color-based Texture Image Segmentation for Vehicle Detection

Ricardo Mejía-Iñigo, María E. Barilla-Pérez and Héctor A. Montes-Venegas
Universidad Autónoma del Estado de México
México

1. Introduction

Traffic flow is the study of interactions between vehicles, drivers and related infrastructure with the goal of understanding and developing an efficient road network. A number of devices have been implemented for monitoring traffic flow such as magnetic strips and radar detectors (Haidarian-Shahri et al. (2007)). All these devices have served for a very specific purpose. However, they do not provide as much information of the scene as cameras may do. The resulting information acquired from cameras should be properly processed to monitor vehicles' behavior. Vision-based techniques have been increasingly used to extract useful traffic information from image sequences such as vehicle count, vehicle flow, vehicle speed, vehicle classification, and lane changes. All these tasks require an accurate vehicle tracking (Zhu et al. (2009)).

Vehicle tracking is a technique that generates the trajectory of the vehicle over time by locating its position in every frame of an image sequence (Yilmaz (2006)). Usually, vehicle tracking requires three initial steps, namely, (a) choosing a suitable object representation, (b) selection of image features used as an input for the tracker and (c) vehicle detection. Vehicle detection is considered a difficult task, as an appropriate feature selection is needed to deal with the visual complexity of the road scene that may include congestions, shadows, weather incidences, headlights, road conditions and other traffic participants (Coifman et al. (1998)).

Different approaches can be used to carry out the vehicle detection task, such as point detectors, background subtraction, segmentation or supervised learning methods. The selection of discriminative features plays a critical role because they should be so unique as to be easily distinguished in a feature space (Yilmaz (2006)).

Texture is considered to be a rich source of information about the nature and the three-dimensional shape of objects. It is defined by Petrou (2006) and Materka & Strzelecki (1998) as complex visual patterns and subpatterns with singular properties such as brightness, color and size. There exist different approaches for texture modeling, such as statistical models, structural models and models based on transforms. Nevertheless, texture by itself is not enough to differentiate among different objects because contrasting surfaces cause false positives detections (Sun et al. (2006)).

Color is considered a relevant feature when dealing with the perception of static and moving images. Visual contrast is useful to filter information present in each color component (Trémeau et al. (2008)) and to distinguish among similar gray-scale intensities (Barilla-Pérez & Spann (2008)).

The combination of color and texture have been proved to achieve better results and could be exploited more effectively when integrated than when treated in isolation (Barilla-Pérez & Spann (2008)). In this chapter we propose two segmentation methods based on color and texture to solve the problem of differentiating vehicles from the background. Both features were used under the following assumptions:

- Road color is considered homogeneous because the road surface is made of the same material.
- An abrupt difference between the color of the road and the color of the vehicles is useful to discriminate them from each other.
- Even if vehicles are similar in color to the road, car and road textures are different. Thus, texture may still be used to effectively discriminate between road and cars.

In this chapter two methods are described for vehicle detection by segmenting the regions they occupy, both methods use color and texture as main features. The first method uses $L^*u^*v^*$ color space and the Dual-Tree Complex Wavelet Transform (DTCWT) for texture modeling and a simple background identification to carry out the vehicle detection. The second method is based on a change detection technique that integrates the intensity and texture differences between the current frame and a previously modeled background. $L^*u^*v^*$ color space is also used in this method and texture differencing measure is based on a function between gradient vectors. In both methods, an additional process is also applied using morphological operations to improve the detection of the regions that describe the vehicles.

The chapter also includes an overview of related work and a detailed description of how color and texture features can be properly combined with both approaches to achieve accurate vehicle detection results. An experimental comparison of the two vehicle detection methods is performed both qualitatively and quantitatively. The Outex framework have been used in order to compare the accuracy of both methods (Ojala et al. (2002)). Finally, the work is summarized and results are discussed.

2. Overview of previous work

This section presents an overview of relevant vision-based vehicle detection systems. Vehicle detection using optical sensors is a very challenging task due to the various conditions found in any road scene. The variability of such conditions may be caused by the weather, by illumination changes, by the presence of shadows and by the numerous objects that are part of the scene. Additionally, vehicle features such as shape, size and color are not always sufficient for a reliable detection as the appearance of a vehicle also depends on its pose and its interaction with other traffic participants. In this section we identify several methods for vehicle detection, namely, background extraction methods, edges detection methods and statistical-based approaches. We also review previous works that deal with the occlusion problem as well as with the shadow removal problem.

2.1 Background subtraction

Background subtraction for vehicle detection methods compare each video frame against a reference background model. Assuming that all moving objects are vehicles, the elements in the current frame that are significantly different from the background model are considered the objects of interest. Methods for building a background model include estimates based on median filters (Cucchiara et al. (2003); Lo & Velastin (2001); Zhou & Aggarwal (2001)), Kalman filters (Halevy & Weinshall (1999); Wren et al. (1997)), Gaussian models (Cheng et al. (2007);

Lee et al. (2003); Stauffer & Grimson (2000)) or simple histograms and blob formation models (Elgammal et al. (1999); Sen-Ching et al. (2005)).

2.2 Edge detection

Edge detection methods have been used under the assumption that they are robust to illumination changes. In order to increase robustness of vehicle detection, edges have been used in conjunction with color information.

Tsai et al. (2007) proposed a color model to find candidate vehicles under the assumption that vehicles colors change with lightning conditions. Possible vehicles are then actually found by using local features such as corners, edge maps and wavelet coefficients. Eng et al. (2008) also use color and edge information to detect vehicles. They developed a vehicle detection method by subtracting objects from a background which is previously modeled by color and edges. Techawatcharapaikul et al. (2008) proposed a method for vehicles and shadows segmentation using temporal edge density information from adjacent frames. Mo & Zhang (2010) have detected vehicles by finding their voting center which is then used to implement a traffic analysis system. The vehicles voting centers are calculated by matching the results of a segmentation process with a previously learned spatial distribution of the objects of interest.

2.3 Statistical-based approaches

Wang & Lien (2008) proposed an statistical approach for vehicle detection based on the vector projection of three main local features of vehicles, namely the roof and the two headlights. Both *Principal Component Analysis* (PCA) and *Independent Component Analysis* (ICA) techniques are combined in order to improve the tolerance of the detection process under illumination variations and for vehicle position. Although the authors show very promising results, it is unclear if the method is robust under significant rotations of the main local features of the car. An additional work using a graph-based approach that considers the identity of neighboring vehicles has been proposed by Shahri et al. (2007). Their results showed that vehicle tracking is feasible, even for low quality and low frame rate traffic cameras.

2.4 Handling occlusions

Occlusion is one of the most difficult problems associated with vehicle detection. In order to address this problem Kamijo et al. (2000) proposed a method based on a *Spatio-Temporal Markov Random Field* model that considers the texture correlation between consecutive images and the correlation among neighbors within an image. This algorithm requires only gray scale images and does not assume any physical models of the vehicles. Even though occlusions in the scene happened in very complicated manners (vehicles appear in various shapes and they move in unpredictable ways), the method achieved a success rate as high as 95%.

Also to detect and handle vehicle occlusions, Zhang et al. (2008) presented a framework composed of three levels named *intraframe*, *interframe*, and *tracking*. On the *intraframe* level, occlusions are handled by evaluating the size of vehicles. On the *interframe* level, occlusions are detected by using the motion vectors of vehicles. On the *tracking* level, occlusions are handled while tracking the vehicle by performing a bidirectional occlusion reasoning algorithm. Most partial occlusions are handled on the *intraframe* and *interframe* levels, and full occlusions are handled on the *tracking* level. In addition to this work, Zhang & Yuan (2007) used low-resolution images for detecting and tracking vehicles as most of these images were captured far from the vehicles. The authors used borders to build a model to find outliers that can differentiate the vehicle from another object in the scene. Kalman Filters in combination with the body's radius allow handling total and partial occlusions when obstacles blocked the view of the vehicle. This method requires

very low-resolution images to work correctly. Earlier research efforts for solving vehicle occlusions on monocular traffic-image sequences (Pang et al. (2004)) and for detecting and tracking vehicles by segmenting scenes with severe occlusions (Gentile et al. (2004)) have been published. However, based on their experimental results, Zhang et al. (2008) stated that “quantitative evaluation and comparison demonstrate that the proposed method outperforms state-of-the-art methods”.

2.5 Shadows removal

Shadows interfere with moving vehicle detection because they cause erroneous segmentation of objects. For this reason, there is a recent interest on developing algorithms to deal with shadows of the vehicles present in the scene. Chen et al. (2010) have removed shadows from the luminance component of an image while keeping the chrominance components intact. Fang et al. (2008) have used spectral and geometrical properties of shadows to create a model to eliminate them from the scene. This method is considered a previous step before the actual vehicle detection.

2.6 Driver assistance

For traffic monitoring tasks it is crucial to detect the lane and other features of the road and then identify the presence of other vehicles. Kastrinaki et al. (2003) present a survey on video-based systems considering both areas of road traffic monitoring and automatic vehicle guidance. In their paper, the authors survey methods for the two main subtasks involved in traffic applications, i.e. the automatic lane finding and vehicle detection. Two particular cases are considered in this paper. The first one is the case in which a static camera (usually located in a high position) observes a road scene for the purpose of traffic surveillance. In the second case at least one camera is mounted on a vehicle that moves in a dynamic road scene. In a later work, Sun et al. (2006) present an extended review of systems for vehicle detection for on-board automotive driver assistance systems. For the methods summarized in this paper, the camera is mounted on the vehicle. Diaz-Alonso et al. (2008) have proposed an aid system to help the driver in overtaking and lane changing decisions. These tasks are addressed with an optic-flow-driven scheme, focusing on the visual field in the side mirror by placing a camera on top of it.

3. Color-based texture segmentation

Color-based texture segmentation is a process that divides an image into homogeneous regions using both color and textures features to distinguish from non-similar regions (Barilla-Pérez (2008)). Using more than one descriptor helps to improve the segmentation process when the descriptors are visually relevant and they can be integrated to obtain a single similarity measure.

In this section, a color-based texture segmentation algorithm originally proposed by Mejia-Iñigo et al. (2009) is described. The algorithm uses a feature vector based on color and texture to differentiate vehicles from the background. This selection of features is based on the assumptions listed in Section 1.

3.1 Texture and color

Texture is regarded as a rich source of information about the nature and three-dimensional shape of objects (Petrou (2006)). It consists of complex visual patterns and subpatterns with singular properties such as brightness, color and size (Materka & Strzelecki (1998)). There exist different approaches for texture modeling, such as statistical models, structural models

and models based on transforms. Materka & Strzelecki (1998) suggested that transform methods permit the variation of spatial resolution to represent textures at different scales. In this sense, Wavelet Transform is a viable method to obtain time-frequency localization analysis and to identify stationary and non-stationary textures. As a framework, the Dual-Tree Complex Wavelet Transform (DTCWT) overcomes several difficulties present in other Wavelet Transform methods (Selesnick et al. (2005)) and offers the following advantages:

- Nearly shift invariance. When the transform is applied, the results seem to be mildly affected by translations in the original input.
- Directional selectivity. A resulting set of 6 different positive and negative directions: 15° , 45° , 75° , -15° , -45° and -75° .
- Moderate redundancy. The redundancy introduced by Complex Wavelets is $2^d:1$ for d dimensions (Kingsbury (1999)).

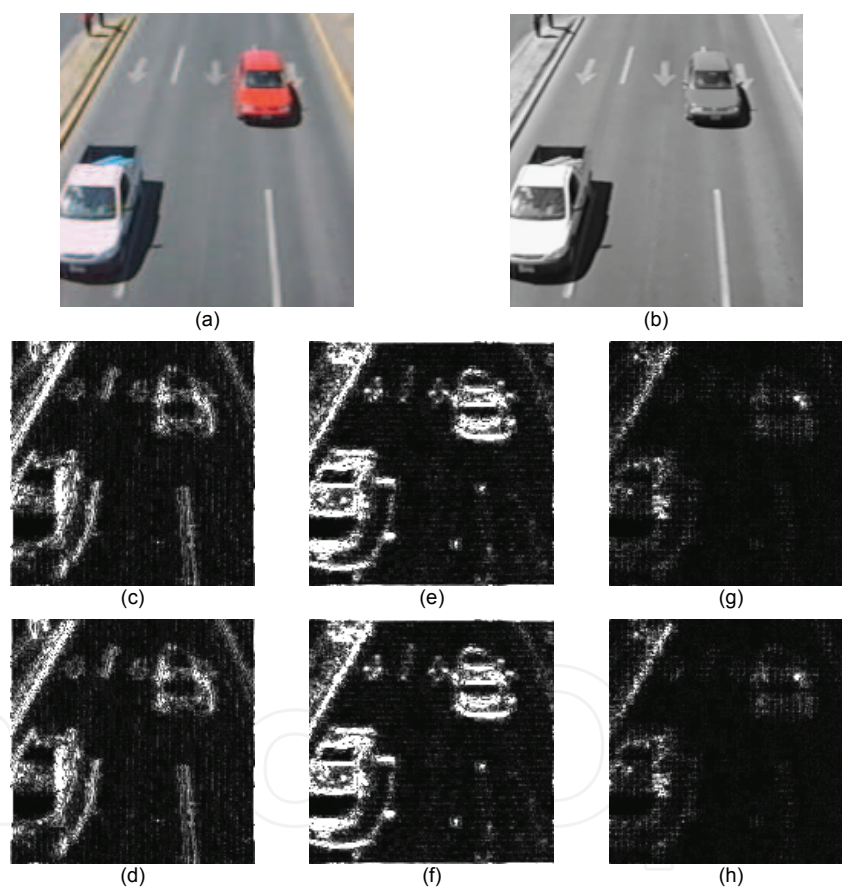


Fig. 1. Directional results of applying the Dual-Tree Complex Wavelet Transform to L^* component (b) of a car image (a) for (c) 75° , (d) -75° , (e) 15° , (f) -15° , (g) 45° and (h) -45° directions

Color is perceived as a non-linear psycho-visual system too complex to be modeled by a simple function (Bovik & Bovik (2000)). The $L^*u^*v^*$ color space is considered one of the most suitable color spaces to represent human perception, as it is perceptually uniform (Malacara (2001)). Its L^* component provides lightness information useful to describe texture by illumination changes on the surface of the object, whereas u^* and v^* provide chrominance

information about the pixel values. Compared to other color spaces, $\mathbf{L}^*\mathbf{u}^*\mathbf{v}^*$ has been proved to achieve high success rates for image segmentation when using texture and color features (Barilla-Pérez (2008)).

3.2 Feature extraction

Let be $I_{L^*u^*v^*}(x, y)$ a color image in the $\mathbf{L}^*\mathbf{u}^*\mathbf{v}^*$ color space of size $M \times N$ where $\{x = 0, \dots, M - 1\}$ and $\{y = 0, \dots, N - 1\}$. For the color plane $I_{L^*}(x, y)$ of the image $I_{L^*u^*v^*}(x, y)$, we compute

$$W_{I_{L^*}}^{(s,b)}(i, j) = \text{DTCWT}\{I_{L^*}(x, y)\}. \quad (1)$$

where $W_{I_{L^*}}^{(s,b)}(i, j)$ are the wavelet coefficients as complex numbers, $\{s = 1, \dots, S\}$ represents the levels of resolution from the highest to the lowest and $\{b = 1, \dots, 6\}$ indicates the number of subbands produced by the wavelet transform at scale s . The size of the subbands b at scale s is $\frac{M}{2^s} \times \frac{N}{2^s}$ where $\{i = 0, \dots, \frac{M}{2^s} - 1\}$ and $\{j = 0, \dots, \frac{N}{2^s} - 1\}$. Computing the modulus of the wavelet coefficients yields its energy. The feature vectors are constructed from this energy combined with the chrominance components \mathbf{u}^* and \mathbf{v}^* .

Vehicles traveling on a one-direction road have texture qualities in different directions, but if the vehicles are moving towards the capture device we only consider -75° and 75° given that these two components provide more information about the vertical displacement of the vehicles (see Fig. 1). Also, DTCWT can be applied successively until scale $s = S$ is achieved. Considering the resolution of the images, a $S = 3$ feature vector limit is employed.

$\mathbf{L}^*\mathbf{u}^*\mathbf{v}^*$ color space components are separated in chrominance and lightness information (Sangwine et al. (1998)). DTCWT is applied on the \mathbf{L}^* component at different coarseness levels because texture features are considered to be more sensitive to lightness and scale changes. In contrast, color remains almost invariant to different textures variations. By reducing DTCWT components, the dimensional space of the feature vector is also reduced to decrease computational cost. The feature vector extraction process is depicted in Fig. 2.

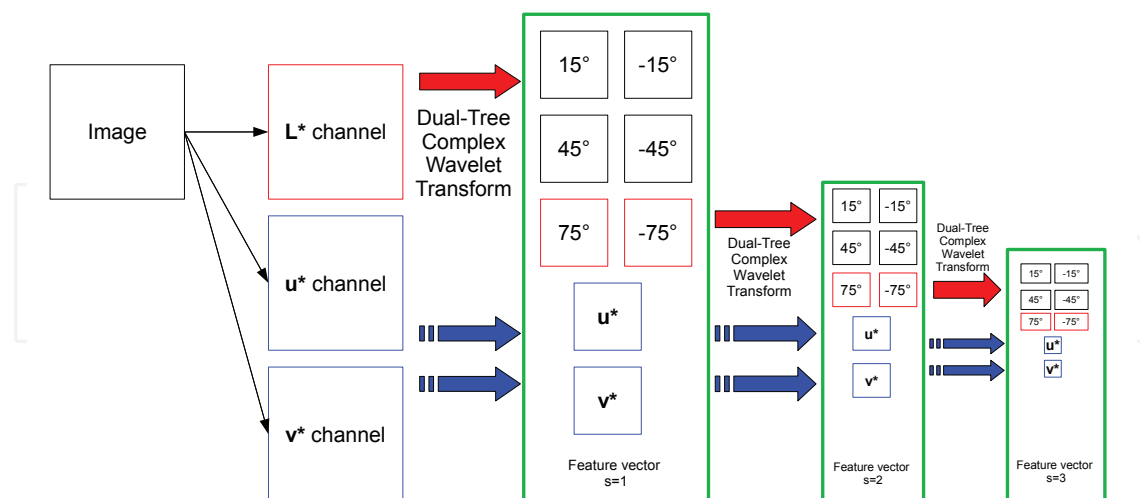


Fig. 2. Feature vector extraction and wavelet components.

3.2.1 Segmentation

Once the feature vectors have been calculated, the coarsest level at scale S is processed to obtain a global initialization that will be used to segment higher-resolution vectors. K-means clustering was used to provide this initial segmentation.

The reason for using K-means clustering is that can be easily implemented and provides a good reference to identify the most representative k macro-textures for the coarsest level $s = S$ (Barilla-Pérez (2008)). There are other segmentation methods that could be used to provide this initialization, such as statistical or graph-based methods (Felzenszwalb & Huttenlocher (2004)), but they require several parameters usually defined heuristically. In contrast, K-means clustering only requires a k parameter to indicate the most significant regions to be identified within the image.

Feature vectors are thought to have 2 representative regions: cars and road, so $k = 2$ seems to be a reasonable parameter value. The resulting classification at scale $s = S$ is used as an initialization for the Expectation-Maximization method at scale $s = S - 1$, and the result is used to process the feature vector at scale $s = S - 2$. This process is repeated until the same action is performed to the feature vector at scale $s = 1$ (See Fig. 3).

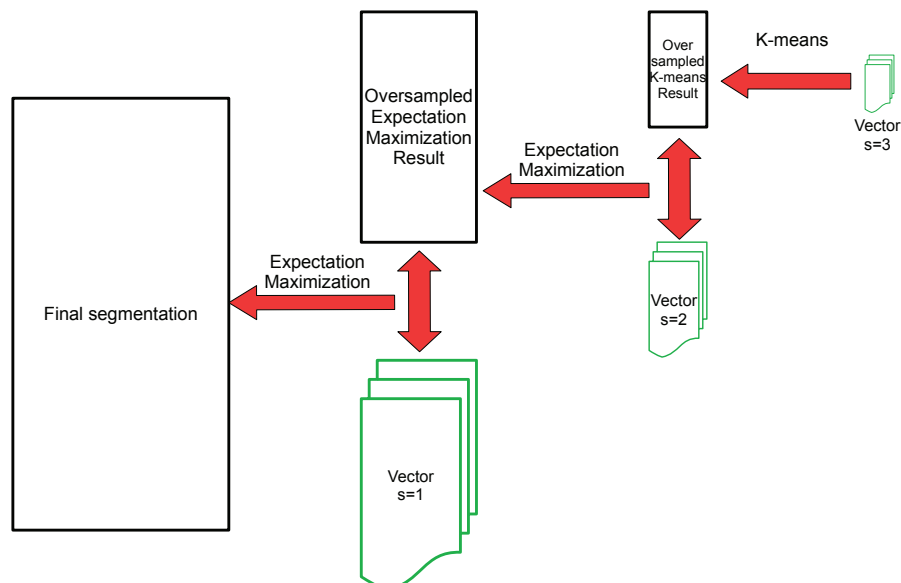


Fig. 3. Segmentation process of feature vectors at different scales.

Classification combines Multi Resolution Analysis (MRA) with a Gaussian Mixture Model to segment images at scale s based on the segmentation provided by scale $(s + 1)$. Segmentation results are in the form of a binary image as depicted in Fig. 4.

3.3 Vehicle motion detection

The analyzed scene is considered static, where only the vehicles are in constant change and the background always preserves the same texture and color. Modeling of the background is needed to separate vehicles from the road.

3.3.1 Background modeling

Considering a scene where images are taken at a time t and an interval between frames Δt , it is possible to build the background from the scene using binary images. Let $Frame(t)$ be a frame taken at a time t ; a pixel is part of the background if it keeps the same value from



Fig. 4. Binary image obtained by a 2-cluster segmentation. The white blob is the vehicle in motion and the white vertical stripes are divisions of the lanes.

$(t - 1)$ to $(t - m)$ at interval Δt , where m can be as big as needed to classify the frame. For the experiments discussed in this chapter, the size of m is set to 9. Equation (2) represents the function that classifies pixels on an image at a given time.

$$Background(t, m) = \begin{cases} 0 & Frame(t - i) = Frame(t). \\ 1 & \text{otherwise.} \end{cases} \quad (2)$$

$$\forall i \in \{1, \dots, m\},$$

An example of a classified background is shown in Fig.5.d.

3.3.2 Using logic operations for vehicle detection

The background model is represented by a binary value of 1 for all possible objects that can be described as a moving vehicle, and 0 for all objects that do not change in the images from $Frame(t - m)$ to $Frame(t)$. A possible vehicle, $PosVehicle$, is extracted by applying the logical AND operation between the modeled background $Background(t, m)$ and frame $Frame(t)$. The resultant image contains the pixels that represent the vehicles (see Fig.5.e).

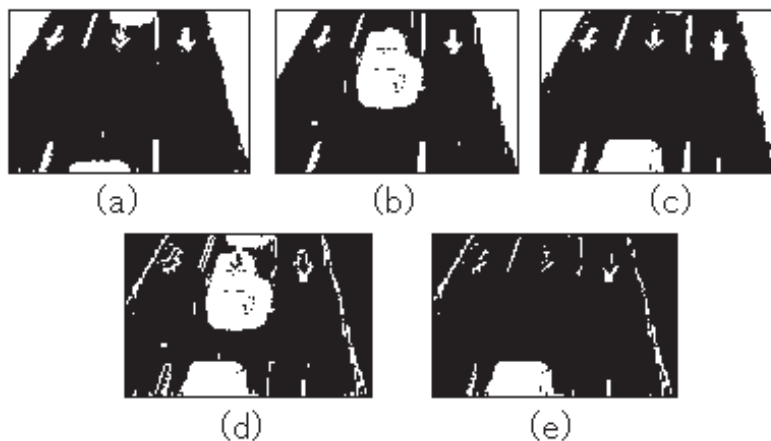


Fig. 5. Possible vehicle detection process. (a) $Frame(t-2)$, (b) $Frame(t-1)$, (c) current scene $Frame(t)$, (d) modeled background $Background(t)$ and (e) possible vehicle.

A number of morphological operations are then used to reduce undesired elements present in the image where the possible vehicles have been detected.

3.3.3 Morphological processing

The resultant binary image contains the area of the vehicles along with several undesired elements as well as small holes inside those vehicles caused by illumination changes, background irregularities and vibrations of the capture device (Fig. 6 (a)). In order to remove these unwanted elements while distorting the shape of the objects as little as possible, erosion, opening, dilation and closing operations are applied (Fig. 6 (b)). These morphological operations are useful as describe above, but there are still elements that do not belong to the vehicle, seen as holes inside the objects of interest. Using region filling (González & Woods (2002)), these holes can be removed to produce a more precise shape of the vehicles detected (Fig. 6.c).

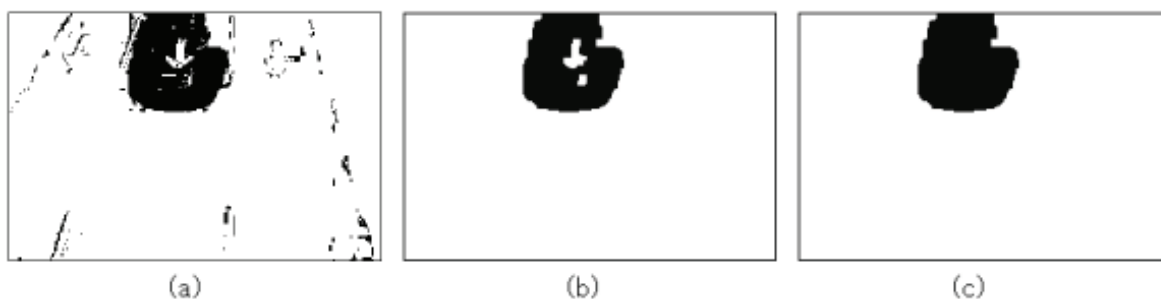


Fig. 6. Morphological processing. (a) Resultant binary image, (b) unwanted elements removal and (c) region filling.

4. Segmentation based on lightness and texture differences

Vehicle detection can be understood as a segmentation process, which allows us to separate the two most significant homogeneous regions in a traffic scene: (a) vehicles and (b) road. The method described in this Section attempts to deal with time-variant objects in the scene by differencing those regions in the current frame with a previously modeled background. The background and current frame difference is computed using features of texture and L^* component intensities of the $L^*u^*v^*$ color space.

4.1 Background modeling

Vehicle detection can be achieved by building a background model and then identifying the pixels of every frame of the image sequence that vary significantly from that model (Yilmaz (2006)). Those pixels are assumed to be a moving vehicle.

Several techniques have been proposed to model background as they overcome different drawbacks usually related to the scene being examined (Cheung & Kamath (2004)). These techniques can be classified as follows:

- Recursive techniques: a single background model is updated on each input image. These techniques required less storage, but errors in the background model can persist over the time.
- Non-recursive methods: require an image buffer to estimate a background and variation is estimated for each pixel. Adaptability level is very high as they do not depend on previous background processing, but a large buffer is required to detect movement in congested roads.

In order to deal with noise and other objects that may appear in the scene, it is preferable to use non-recursive techniques to model the background, for instance, a *Median Filter*.

Let us consider an image buffer $B(x, y, t)$ that consist on a set of β images, where x and y are defined as spatial image coordinates at a given time t . Median filter background is defined as

$$\text{Background}(x, y, t) = \text{Median}(B(x, y, t - \beta), \dots, B(x, y, t - 1)) \forall x, \forall y . \quad (3)$$

The median filter is very consistent even in the presence of outliers caused by noise or sudden object appearance. Correct background model is achieved only when the background is visible at least on 50% of the image buffer for each pixel (Gutchess et al. (2001)).

The background is modeled with a buffer using the L^* component of the $L^*u^*v^*$ color space. This background model is updated every β frames and it is used as a reference to identify the most persistent pixels within an image. Vehicles are formed by the pixels of the current frame that have a significant deviation from the background.

4.2 Image difference

Image difference commonly use absolute value of two-frames intensities subtraction to identify moving objects. However, intensity subtraction is prone to errors because two different objects can have similar intensity. Adding an additional feature, such as texture, may help to effectively discriminate between different objects.

In line with Li and Leung's work (Li & Leung (2002)), texture and intensity information are used to detect changes between two images by using a gradient vector relationship that supports noise and illumination changes. Integrating intensity and texture differences for change detection works under the assumption that background texture remains relatively stable with respect to noise and illumination variations unless it is covered by moving objects or when abrupt lightness changes occur. When the background is covered by a moving object, even if intensities of foreground and background are similar, textures in both images are different. If texture difference is not sufficient to identify homogeneous regions, then intensities are considered more relevant to discriminate regions.

4.2.1 Texture difference

Li & Leung (2002) suggested gradients as a texture measure because neighborhood information is included and it is not easily affected by lightness conditions. Let I be an image, there exists an image $I' = (I^x, I^y)$, where I^x and I^y are results of Sobel gradient operators for its corresponding direction.

Image cross-correlation $C_{a,b}$ of gradient vectors for two frames a and b is defined as:

$$C_{a,b} = \|I'_a\| \|I'_b\| \cos\theta \quad (4)$$

where θ is the angle between both vectors. Added to this measure, auto-correlation for a gradient vector frame is

$$C_{a,a} = \|I'_a\|^2 \quad (5)$$

Vectorial correlation components satisfies triangle inequality

$$C_{a,a} + C_{b,b} \geq 2C_{a,b} \quad (6)$$

$$\|I'_a\|^2 + \|I'_b\|^2 \geq 2\|I'_a\| \|I'_b\| \cos\theta \quad (7)$$

From equation 7, we can see that when $\theta \approx 0$ and when $\|I'_a\|^2 \approx \|I'_b\|^2$, it means that a significative change inside the image does not exist. If there is a change in gradient image

vectors directions of images \mathbf{a} and \mathbf{b} , the difference of both values will grow according to their magnitude and direction.

As mentioned above, auto-correlation and cross-correlation is applied to differentiate vector direction changes between the current frame \mathbf{L}^* component and the background model of the same component. Let $M_{x,y}$ be a 5x5 neighborhood centered at pixel (x, y) , a gradient texture difference measure to discriminate objects between the background and the input frame is defined as follows:

$$R(x, y) = 1 - \frac{\sum_{u,v \in M_{x,y}} 2C_{B,F}(u, v)}{\sum_{u,v \in M_{x,y}} (C_{B,B}(u, v) + C_{F,F}(u, v))} \quad (8)$$

where B is the modeled background and F is the \mathbf{L}^* component of the current frame. The region $M_{x,y}$ is used to integrate gradient vector intensities of surrounding pixels that represent local texture features. A change is deemed *significant* when variations in direction are bigger than 90° in their orientation. Values of $R(x, y)$ are in the $[0,1]$ interval; where lower values represent non significant changes in texture and if values approximate to 1 indicates a higher difference in gradient texture directions. This measure is valid as long as there exist noise and intensity variations in the pixels of the $M_{x,y}$ region; otherwise, texture becomes less relevant as there is not significant gradient change.

$R(x, y)$ is then adjusted through a series of validity weights $w_{texture}(x, y)$ using the definition given by Li & Leung (2002). To compute $w_{texture}(x, y)$, Li & Leung (2002) produced an image g (equation 9) that combines autocorrelations of two images to validate the pixel values of R . Information of autocorrelation is considered relevant as it contains significative intensity changes in both R and g images. Equation 9 uses a maximum function because texture information may exist in any images that have been compared.

$$g(x, y) = \max_{i \in \{B, F\}} \sqrt{\frac{1}{|M_{x,y}|} \sum_{u,v \in M_{x,y}} C_{i,i}(u, v)} \quad (9)$$

The $g(x, y)$ values from equation 9 are coefficients used to detect all pixels with relevant gradient values whether those values belong to the background image or to the current frame. These coefficients are also used to estimate the validity weights $w_{texture}(x, y)$ in the $[0, 1]$ interval. For the results reported in this chapter, it is not necessary to detect every change in texture because the images involved can have undesired textured regions not relevant in the current frame; i.e., when background is not modeled properly due to the existence of cluttered regions inside the image buffer, it then causes that the moving objects are considered part of the background. If so, texture detection is not precise because cluttered regions will cause high autocorrelation values, and consequently, high g values that promote a vehicle detection for as long as the same background model is used.

In order to address this issue, equation 9 has been modified as follows:

$$g(x, y) = \sqrt{\frac{1}{|M_{x,y}|} \sum_{u,v \in M_{x,y}} C_{F,F}(u, v)} \quad \forall x, y \in F \quad (10)$$

$g(x, y)$ is used to estimate the validity weights $w_{texture}(x, y)$ as follows:

$$w_{texture}(x, y) = \begin{cases} 1, & \text{if } g(x, y) > 2T_w \\ g(x, y)/(2T_w) & \text{otherwise} \end{cases} \quad \forall x, y \quad (11)$$

From equation 11, T_w is a noise distribution parameter calculated from the image intensity difference. It is required that $T_w > 2\sqrt{2}\sigma_d$, however, a very large T_w would deem texture information less significant in R . For the experiments reported in this chapter, T_w has been set to a value of 50% larger than $2\sqrt{2}\sigma_d$; in turn, σ_d is defined in equation 16. Finally, the texture difference measure is computed as follows:

$$d_{texture}(x, y) = w_{texture}(x, y) \cdot R(x, y), \forall x, \forall y \quad (12)$$

4.2.2 Lightness difference

To compute lightness difference, first we perform an initial subtraction $d(x, y)$ defined as follows:

$$d(x, y) = L_F^*(x, y) - L_B^*(x, y) \quad (13)$$

where B is the modeled background and F is the current frame. $d(x, y)$ measures lightness variations that might be caused by the moving vehicles and other elements in the scene. These elements are removed by smoothing $d(x, y)$ using the following procedure:

$$\bar{d}(x, y) = \frac{1}{|M_{x,y}|} \sum_{u,v \in M_{x,y}} d(u, v) \quad (14)$$

Now we compute a difference image d' that contains only the lightness variations:

$$d'(x, y) = d(x, y) - \bar{d}(x, y) \quad (15)$$

It is assumed that the noise contained in image $d(x, y)$ (equation 13) follows a Gaussian distribution $N(0, \sigma_d)$. Consequently, σ_d is estimated from $d'(x, y)$ as it contains all the small lightness variations.

Outlier pixels with high lightness variation can cause a poor noise estimation when deviation needs to be calculated. To tackle this problem, the Least Median of Squares method (*LMedS*) (Rousseeuw & Leroy (1986)) is used on $d'(x, y)$ as it supports corruption of up to 50% of the input data. Following the definition given by Rosin (1988), the standard deviation of the noise is calculated as follows:

$$\sigma_d = \frac{LMedS}{0.33724} \quad (16)$$

Finally, $d(x, y)$ is used to estimate the lightness difference $d_{lightness}(x, y)$ as follows:

$$d_{lightness}(x, y) = \begin{cases} 1, & \text{if } |d(x, y)| > 2T \\ |d(x, y)|/(2T) & \text{otherwise} \end{cases} \quad (17)$$

The T parameter needs to be properly calculated to deal with Gaussian noise. Li & Leung (2002) sets the T parameter as:

$$T = |\bar{d}_s| + 3\sigma_d. \quad (18)$$

\bar{d}_s in equation 18 describes lightness changes. It grows when global illumination changes occur, but shadows and local changes diminish this value. \bar{d}_s is calculated as:

$$\bar{d}_s = \frac{1}{|N_s|} \sum_{u,v \in N_s} \bar{d}(u,v) \quad (19)$$

N_s is the set of pixels $\bar{d}(x,y)$ that satisfy equation 20.

$$N_s = \{x,y : (|d'(x,y)| < 2\sigma_d) \wedge (|\bar{d}(x,y)| < T_{50\%})\} \quad (20)$$

where $T_{50\%}$ is the median value of $|\bar{d}(x,y)|$.

4.2.3 Texture and lightness differences integration

Assuming that texture and lightness differences complement each other in a [0,1] interval, $w_{lightness}(x,y) = 1 - w_{texture}(x,y)$, the integration of texture and lightness intensity differences with their corresponding validation weights is carried out by equation 21.

$$d_{texture,lightness}(x,y) = w_{lightness}(x,y) \cdot d_{lightness}(x,y) + w_{texture}(x,y) \cdot d_{texture}(x,y) \quad \forall x, \forall y \quad (21)$$

The final segmentation for vehicle detection is produced by thresholding the $d_{texture,lightness}(x,y)$ image with the median value of $[0,1] = 0.5$.

4.2.4 Morphological processing

The resultant binary image contains a number of the same unwanted elements that are present in the images produced by the *Color-based texture segmentation* method (Section 3). As a result, the morphological operations used in that method are also applied here in the same fashion.

5. Experimental results

In this section we present the results of applying both segmentation methods using natural scenes images. The image database used in our experiments is publicly available and it is described in Martel-Brisson & Zaccarin (2007). We have proceeded according to the supervised texture segmentation testing protocol of the Outex database (Ojala et al. (2002)). This framework involves the following components:

- **Image segmentation test suite.** The test sequence is called *Highway III* and it includes traffic flow scenes and ground truth images. Ground truth images show individual pixel group of vehicles and shadows. However, we have modified these images so both the vehicle and its shadow are within the same object.
- **Input data in an individual problem.** I is the image to be segmented with N_I pixels. R_I is the number of regions in I , each one with a distinct class label L_R .
- **Required output in an individual problem.** O is the output labeled image equal in size to I .
- **Performance metric in an individual problem.** Let I_{ij} and O_{ij} denote the class labels of pixel (i,j) in I and O , respectively. The score of an individual problem S_p is the proportion of all pixels that are correctly labelled:

$$S_p = \frac{100}{N_I} \sum_{ij} \delta(I_{ij}, O_{ij})\% \quad (22)$$

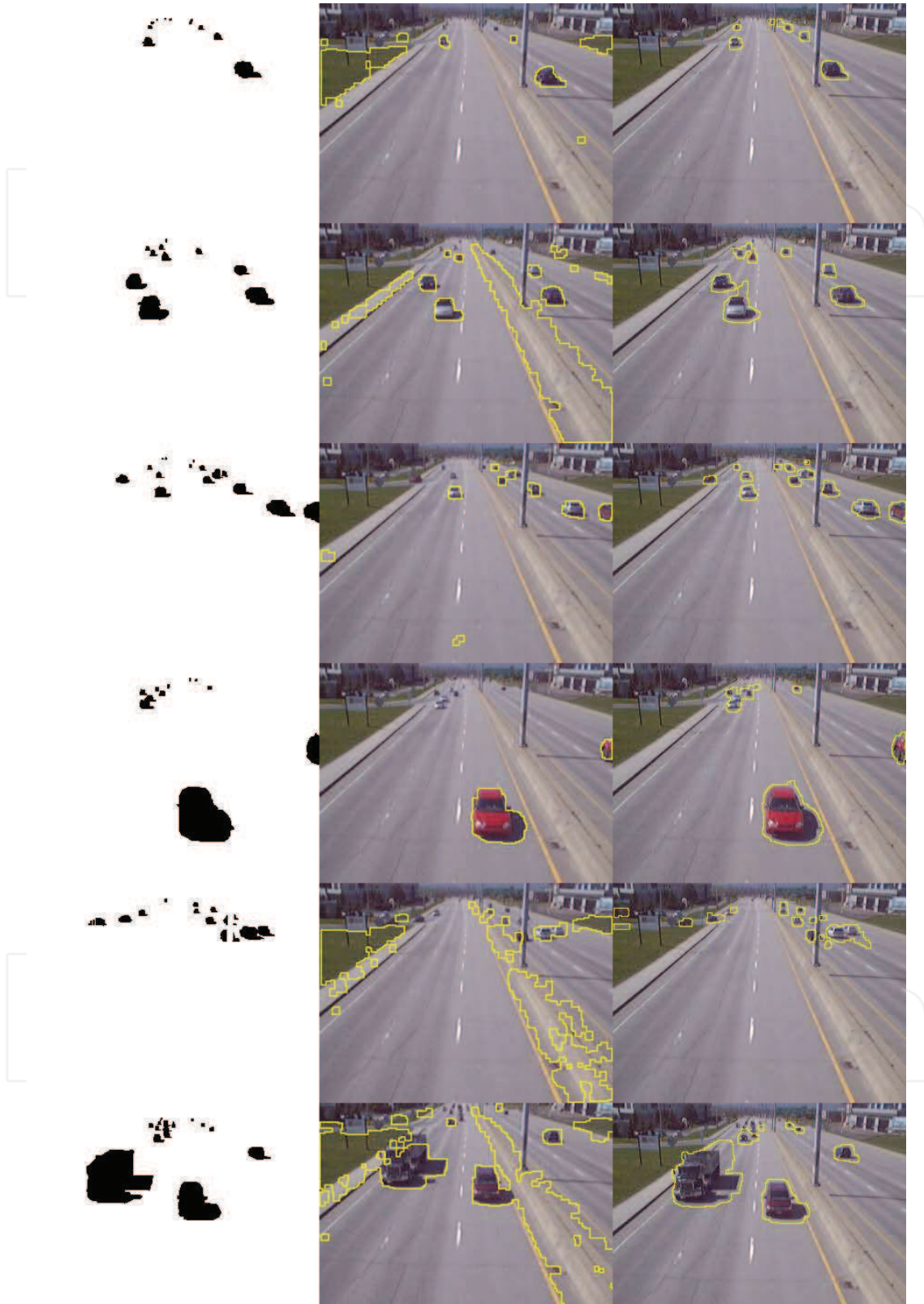


Fig. 7. First column shows the ground truth images. Second and third columns show the results of the methods described in Section 3 and Section 4, respectively.

where δ denotes the Kronecker delta function.

- **Test suite score.** Let P be the number of problems to be considered. If test suite contains $P > 1$ problems, then the scores from equations 23, 24, 25 and 26 are computed.

$$S_{mean} = \frac{1}{P} \sum_p^P S_p \quad (23)$$

$$S_{min} = \min(S_p) \quad \forall p \in \{1, \dots, P\} \quad (24)$$

$$S_{max} = \max(S_p) \quad \forall p \in \{1, \dots, P\} \quad (25)$$

$$S_{dev} = \sqrt{\frac{1}{P} \sum_p^P (S_p - S_{mean})^2} \quad (26)$$

We have measured the performance of the above values in both segmentation methods. Qualitative image results are shown in Figure 7 and quantitative measures over the image test suite are listed in Table 1.

Image ground truth	Color-based texture segmentation (Section 3)	Segmentation based on lightness and texture differences (Section 4)
S_{min}	78.5091%	97.2174%
S_{max}	99.1536%	99.4817%
S_{mean}	89.7853%	98.5516%
S_{dev}	8.0121%	0.6786%

Table 1. Performance scores of the two segmentation methods described in this chapter over the image test suite.

6. Discussion

Given the estimators shown in table 1, particularly the mean and standard deviation, it is evident, without any further analysis or evidence beyond themselves, that the *Segmentation based on lightness and texture differences* method (Section 4) has achieved a better performance than the *Color-based texture segmentation* method (Section 3).

The S_{mean} score measures how successful the segmentation rate is. This score is the average of the performance values of all the individual problems considered in our experiments expressed as a percentage. The *Segmentation based on lightness and texture differences* method achieved a $S_{mean} = 98.5516$, whereas the *Color-based texture segmentation* method achieved a $S_{mean} = 89.7853$.

Also, the S_{dev} score shows that the performance of the method is consistent under similar experimental conditions, as its value tends to be smaller when the dispersion from the mean (successful segmentation rate) is also small. For this score, the *Segmentation based on lightness and texture differences* method achieved a $S_{dev} = 0.6786$, whereas the *Color-based texture segmentation* method achieved a $S_{dev} = 8.0121$.

In the *Segmentation based on lightness and texture differences* method, the computed gradient vector's directions have a high tolerance to noisy images and illumination changes. This

method builds a background model to detect the changes of the objects of interest with respect to that reference model. The method achieves a higher accuracy rate but the borders slightly exceed the region of interest because a number of filtering processes are used to make homogeneous the gradient textures.

On the other hand, the *Color-based texture segmentation* method is very discriminative when abrupt changes in color are present in the input images. Even though only two components are used in this method, namely 75° and -75° , calculating the DTCWT coefficients still involves multiple filtering of images at different coarseness levels, thus increasing the computational time cost.

The results derived from the experiments of the methods presented in this chapter can be useful in the preliminary phases of a vehicle tracking system. We will include these results in our future research efforts.

7. Acknowledgements

This work has been partially supported by the *Secretaría de Comunicaciones y Transportes del Gobierno del Estado de Tabasco* within the project "UAEM 110171: Plan Integrado de Movilidad de Villahermosa y su zona metropolitana Fase II" and the research grant PROMEP/103.5/08/3016.

8. References

- Barilla-Pérez, M. E. (2008). *Colour-based Texture Image Segmentation*, PhD thesis, University of Birmingham.
- Barilla-Pérez, M. E. & Spann, M. (2008). Colour-based texture image classification using the complex wavelet transform, *5th International Conference on Electrical Engineering, Computing Science and Automatic Control*.
- Bovik, A. & Bovik, A. (2000). *Handbook of Image and Video Processing*, Academic Press.
- Chen, C.-T., Su, C.-Y. & Kao, W.-C. (2010). An enhanced segmentation on vision-based shadow removal for vehicle detection, pp. 679–682.
- Cheng, S., Luo, X. & Bhandarkar, S. M. (2007). A multiscale parametric background model for stationary foreground object detection, *IEEE Workshop on Motion and Video Computing*.
- Cheung, S.-C. S. & Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video, *Visual Communications and Image Processing* 5308(1): 881–892.
- Coifman, B., Beymer, D., McLauchlan, P., Malik, J. & Malik, H. (1998). A real-time computer vision system for vehicle tracking and traffic surveillance.
- Cucchiara, R., Grana, C., Piccardi, M. & Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(10): 1337–1342.
- Diaz-Alonso, J., Ros-Vidal, E., Rotter, A. & Muhlenberg, M. (2008). Lane-change decision aid system based on motion-driven vehicle tracking, *IEEE Transactions on Vehicular Technology* 57(5): 2736–2746.
- Elgammal, A., Harwood, D. & Davis, L. (1999). Non-parametric model for background subtraction, *Proc. IEEE ICCV '99 Frame-rate workshop*, Corfu, Greece.
- Eng, H.-L., Thida, M., Chew, B.-F., Leman, K. & Anggrelly, S. (2008). Model-based detection and segmentation of vehicles for intelligent transportation system, pp. 2127–2132.
- Fang, L. Z., Qiong, W. Y. & Sheng, Y. Z. (2008). A method to segment moving vehicle cast shadow based on wavelet transform, *Pattern Recognition Letters* 29(16): 2182–2188.
- Felzenszwalb, P. F. & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation, *International Journal of Computer Vision* 59(2): 167–181.

- Gentile, C., Camps, O. & Sznaiar, M. (2004). Segmentation for robust tracking in the presence of severe occlusion, *IEEE Transactions on Image Processing* 13(2): 166–178.
- González, R. C. & Woods, R. E. (2002). *Digital Image Processing*, 2nd edition edn, Prentice Hall.
- Gutchess, D., Trajkovic, M., Cohen-Solal, E., Lyons, D. & Jain, A. K. (2001). A background model initialization algorithm for video surveillance, in *proc. International Conference on Computer Vision*, pp. 733–740.
- Haidarian-Shahri, H., Namata, G., Navlakha, S., Deshpande, A. & Roussopoulos, N. (2007). A graph-based approach to vehicle tracking in traffic camera video streams, *Proceedings of the 4th International Workshop on Data Management for Sensor Networks*.
- Halevy, G. & Weinshall, D. (1999). Motion of disturbances: detection and tracking of multi-body non-rigid motion, *Machine Vision and Applications* 11(3): 122–137.
- Kamijo, S., Matsuchita, Y., Ikeuchi, K. & Sakauchi, M. (2000). Occlusion robust vehicle tracking utilizing spatio-temporal markov random field model, *Proc. IEEE Intelligent Transportation Systems*, Dearborn, MI.
- Kastrinaki, V., Zervakis, M. & Kalaitzakis, K. (2003). A survey of video processing techniques for traffic applications, *Image and Vision Computing* 21: 359–381.
- Kingsbury, N. G. (1999). Shift invariant properties of the dual-tree complex wavelet transform, *International Conference on Acoustics, Speech and Signal Processing ICASSP99*, Phoenix, AZ. Paper SPTM 3.6.
URL: citeseer.ist.psu.edu/kingsbury99shift.html
- Lee, D. S., Hull, J. & Erol, B. (2003). A bayesian framework for gaussian mixture background modeling, *Proc. IEEE International Conference on Image Processing (ICIP 03)*.
- Li, L. & Leung, M. (2002). Integrating intensity and texture differences for robust change detection, *IEEE Transactions on Image Processing* 11: 105–112.
- Lo, B. & Velastin, S. (2001). Automatic congestion detection system for underground platform, *Proc. International Symposium on Intelligent Multimedia*.
- Malacara, D. (2001). *Color Vision and Colorimetry, Theory and Applications*, SPIE International Society for Optical Engineering, Bellingham, Washington USA.
- Martel-Brisson, N. & Zaccarin, A. (2007). Learning and removing cast shadows through a multidistribution approach, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29(7): 1133–1146.
- Materka, A. & Strzelecki, M. (1998). Texture analysis methods - a review, *Technical report*, University of Lodz.
- Mejia-Iñigo, R., Barilla-Perez, M. E. & Montes-Venegas, H. (2009). Color-based texture image segmentation for vehicle detection, *6th International Conference on Electrical Engineering, Computing Science and Automatic Control*, pp. 1–6.
- Mo, G. & Zhang, S. (2010). Vehicles detection in traffic flow, Vol. 2, pp. 751–754.
- Ojala, T., Mäenpää, T., Pietikäinen, M., Viertola, J., Kyllönen, J. & Huovinen, S. (2002). Outex - new framework for empirical evaluation of texture analysis algorithms. *Proc. 16th International Conference on Pattern Recognition*, Quebec, Canada, 1:701 - 706.
- Pang, C. C. C., Lam, W. W. L. & Yung, N. H. C. (2004). A novel method for resolving vehicle occlusion in a monocular traffic-image sequence, *IEEE Transactions on Intelligent Transportation Systems* 5(33): 129–141.
- Petrou, M. (2006). *Image Processing, Dealing With Texture*, John Wiley & Sons Ltd.
- Rosin, P. (1988). Thresholding for Change Detection, *Brunel University*.
- Rousseeuw, P. & Leroy, A. (1986). Least median of squares: A robust method for outlier and model error detection in regression and calibration, *Analytica Chimica Acta* 187: 171–179.

- Sangwine, S. J. & Horne, R. E. N. (1998). *The Colour Image Processing Handbook*, Chapman and Hall.
- Selesnick, I. W., Baraniuk, R. G. & Kingsbury, N. G. (2005). The dual-tree complex wavelet transform, *IEEE Signal Processing Magazine* pp. 123–151.
- Sen-Ching, Cheung, S. & Kamath, C. (2005). Robust background subtraction with foreground validation for urban traffic video, *EURASIP Journal on Applied Signal Processing* 14: 2330–2340.
- Shahri, H. H., Namata, G., Navlakha, S., Deshpande, A. & Roussopoulos, N. (2007). A graph-based approach to vehicle tracking in traffic camera video streams, *Proceedings of the 4th International Workshop on Data Management for Sensor Networks*.
- Stauffer, C. & Grimson, W. (2000). Learning patterns of activity using real-time tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8): 747–757.
- Sun, Z., Bebis, G. & Miller, R. (2006). On-road vehicle detection: a review, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28(5): 694–711.
- Techawatcharapaikul, C., Kaewtrakulpong, P. & Siddhichai, S. (2008). Outdoor vehicle and shadow segmentation by temporal edge density information of adjacent frames, Vol. 1, pp. 433–436.
- Trémeau, A., Tominaga, S. & Plataniotis, K. N. (2008). Color in image and video processing: most recent trends and future research directions, *J. Image Video Process.* 2008: 1–26.
- Tsai, L.-W., Hsieh, J.-W. & Fan, K.-C. (2007). Vehicle detection using normalized color and edge map, *Image Processing, IEEE Transactions on* 16(3): 850–864.
- Wang, C.-C. R. & Lien, J.-J. (2008). Automatic vehicle detection using local features - a statistical approach, *Intelligent Transportation Systems, IEEE Transactions on* 9(1): 83–96.
- Wren, C., Azarbayejani, A., Darrell, T., & Pentland, A. (1997). Pfister: real-time tracking of the human body, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7): 780–785.
- Yilmaz, A. (2006). Object tracking: A survey, *ACM Computing Surveys* 38: 1–45.
- Zhang, H. & Yuan, F. (2007). Vehicle tracking based on image alignment in aerial videos, *Energy Minimization Methods in Computer Vision and Pattern Recognition*.
- Zhang, W., Wu, Q. M. J., Yang, X. & Fang, X. (2008). Multilevel framework to detect and handle vehicle occlusion, *IEEE Transactions On Intelligent Transportation Systems* 9(1): 161–174.
- Zhou, Q. & Aggarwal, J. (2001). Tracking and classifying moving objects from videos, *Proc. 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS 01)*.
- Zhu, H., Li, M., Zhu, Y. & Ni, L. M. (2009). Hero: Online real-time vehicle tracking, *IEEE Transactions on Parallel and Distributed Systems* 20: 740–752.

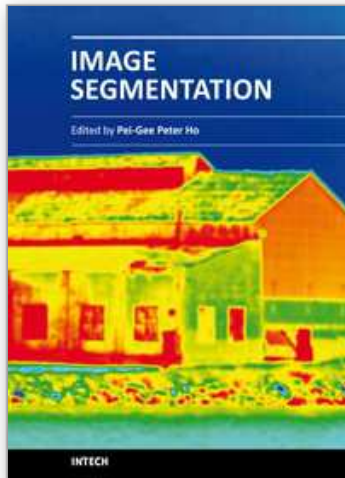


Image Segmentation

Edited by Dr. Pei-Gee Ho

ISBN 978-953-307-228-9

Hard cover, 538 pages

Publisher InTech

Published online 19, April, 2011

Published in print edition April, 2011

It was estimated that 80% of the information received by human is visual. Image processing is evolving fast and continually. During the past 10 years, there has been a significant research increase in image segmentation. To study a specific object in an image, its boundary can be highlighted by an image segmentation procedure. The objective of the image segmentation is to simplify the representation of pictures into meaningful information by partitioning into image regions. Image segmentation is a technique to locate certain objects or boundaries within an image. There are many algorithms and techniques have been developed to solve image segmentation problems, the research topics in this book such as level set, active contour, AR time series image modeling, Support Vector Machines, Pixion based image segmentations, region similarity metric based technique, statistical ANN and JSEG algorithm were written in details. This book brings together many different aspects of the current research on several fields associated to digital image segmentation. Four parts allowed gathering the 27 chapters around the following topics: Survey of Image Segmentation Algorithms, Image Segmentation methods, Image Segmentation Applications and Hardware Implementation. The readers will find the contents in this book enjoyable and get many helpful ideas and overviews on their own study.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ricardo Mejía-Iñigo, María E. Barilla-Pérez and Héctor A. Montes-Venegas (2011). Color-based Texture Image Segmentation for Vehicle Detection, Image Segmentation, Dr. Pei-Gee Ho (Ed.), ISBN: 978-953-307-228-9, InTech, Available from: <http://www.intechopen.com/books/image-segmentation/color-based-texture-image-segmentation-for-vehicle-detection>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen