

Research Article

An Efficient and Robust Moving Shadow Removal Algorithm and Its Applications in ITS

Chin-Teng Lin,¹ Chien-Ting Yang,¹ Yu-Wen Shou,² and Tzu-Kuei Shen¹

¹Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan

²Department of Computer and Communication Engineering, China University of Technology, Hsinchu 303, Taiwan

Correspondence should be addressed to Yu-Wen Shou, owen@cute.edu.tw

Received 1 December 2009; Accepted 10 May 2010

Academic Editor: Alan van Nevel

Copyright © 2010 Chin-Teng Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an efficient algorithm for removing shadows of moving vehicles caused by non-uniform distributions of light reflections in the daytime. This paper presents a brand-new and complete structure in feature combination as well as analysis for orientating and labeling moving shadows so as to extract the defined objects in foregrounds more easily in each snapshot of the original files of videos which are acquired in the real traffic situations. Moreover, we make use of Gaussian Mixture Model (GMM) for background removal and detection of moving shadows in our tested images, and define two indices for characterizing non-shadowed regions where one indicates the characteristics of lines and the other index can be characterized by the information in gray scales of images which helps us to build a newly defined set of darkening ratios (modified darkening factors) based on Gaussian models. To prove the effectiveness of our moving shadow algorithm, we carry it out with a practical application of traffic flow detection in ITS (Intelligent Transportation System)—vehicle counting. Our algorithm shows the faster processing speed, 13.84 ms/frame, and can improve the accuracy rate in 4% ~ 10% for our three tested videos in the experimental results of vehicle counting.

1. Introduction

In recent years, the researches about intelligent video surveillances have increased noticeably. Foreground object detection could be considered as one of the fundamental and critical techniques in this field. In the conventional methods, background subtraction and temporal difference have been widely used for foreground extraction in case of using stationary cameras. The continuing improvements in background modeling techniques have led to many new and fascinating applications like event detection, object behavior analysis, suspicious object detection, traffic monitoring, and so forth. However, some factors like dynamic backgrounds and moving shadows might affect the results of foreground detection and make the problems more complicated. Dynamic backgrounds, one of the factors, might detect and treat the escalators and swaying trees as the foreground regions. Another factor, moving shadows, occurred when light was blocked by moving objects, usually misclassified the foreground regions. This paper would focus on the studies

of moving shadows and aim at developing an efficient and robust algorithm of moving shadow removal as well as the related applications.

About the studies on influences of moving shadows, Zhang et al. [1] classified these techniques into four categories, including color model, statistical model, textural model, and geometric model. Color model used the difference of colors between the shaded and nonshaded pixels. Cucchiara et al. [2] removed moving shadows by using the concept on HSV color space that the hue component of shaded pixels would vary in a smaller range and the saturation component would decrease more obviously. Some researchers proposed shadow detection methods based on RGB color space and normalized-RGB color space. Yang et al. [3] described the ratio of intensities between a shaded pixel and its neighboring shaded pixel in the current image, and this intensity ratio was found to be close to that in the background image. They also made use of the slight change of intensities on the normalized R and G channels between the current and background image. Cavallaro et al. [4]

found that the color components would not change their orders and the photometric invariant features would have a small variation while shadows occurred. Beside of color model, statistical model used the probabilistic functions to determine whether or not a pixel belonged to the shadows. Zhang et al. [1] introduced an illumination invariance feature and then analyzed and modeled shadows as a Chi-square distribution. They classified each moving pixel into the shadow or foreground object by performing a significance test. Song and Tai [5] applied Gaussian model to representing the constant RGB-color ratios, and determined whether a moving pixel belonged to the shadow or foreground object by setting ± 1.5 standard deviation as a threshold. Martel-Brisson and Zaccarin [6] proposed GSM (Gaussian Mixture Shadow Model) for shadow detection which was integrated into a background detection algorithm based on GMM. They tested if the mean of a distribution could describe a shaded region, and if so, they would select this distribution to update the corresponding Gaussian mixture shadow model.

The texture model assumed that the texture of the foreground object would be totally different from that of the background, and that the textures would be distributed uniformly inside the shaded region. Joshi and Papanikolopoulos [7, 8] proposed an algorithm that could learn and detect the shadows by using support vector machine (SVM). They defined four features of images, including intensity ratio, color distortion, edge magnitude distortion and edge gradient distortion. They introduced a cotraining architecture which could make two SVM classifiers help each other in the training process, and they should need a small set of labeled samples on shadows before training SVM classifiers for different video sequences. Leone et al. [9] presented a shadow detection method by using Gabor features. Mohammed Ibrahim and Anupama [10] proposed their method by using division image analysis and projection histogram analysis. Image division operation was performed on the current and reference frames to highlight the homogeneous property of shadows. They afterwards eliminated the left pixels on the boundaries of shadows by using both column- and row-projection histogram analyses. Geometric model attempted to remove shadowed regions or the shadowing effect by observing the geometric information of objects. Hsieh et al. [11] used the histograms of vehicles and the calculated center of lane to detect the lane markings, and also developed a horizontal and vertical line-based method to remove shadows by characteristics of those lane markings. This method might become ineffective in case of no lane markings.

As for some other approaches by combination of the mentioned models, Benedek and Szirányi [12] proposed a method based upon LUV color model. They used “darkening factor”, the distortion of U and V channel, and microstructural responses to be the determinative features, where microstructural responses represented a local textured feature. Their proposed algorithm integrated all those features by using Gaussian model and segmented foreground objects, backgrounds and shadows by calculating their probabilities. Xiao et al. [13] proposed a shadow removal method based

on edge information for traffic scenes which in sequence consisted of an edge extraction technique, the morphological operations on removing the edges of shadows, and the analysis of spatial properties for separating the occluded vehicles which resulted from the influences of shadows. They could reconstruct the size of each object and decide the practical regions of shadows. However, this method intrinsically could not deal with the textured shadowed regions like the regions with lane markings, and the more complicated cases of occlusions like the concave shapes of vehicles would make this method fail to separate vehicles by use of spatial properties.

The color information from color cameras might give fine results for shadow removal, but B/W (Black & White) cameras could provide the better resolution and more sensitive quality under lowly illuminating conditions. That was the reason why B/W cameras rather than color cameras would be much more popular for outdoor applications. The shadow removal method based on color model might not work in such situations. Similarly, texture model could have better results under the unstably illuminating conditions without the color information. However, texture model might give the poorest performances for the textureless objects. Geometric model could be more adaptive to the specific scenes due to dependency upon the geometric relations between objects and scenes, and would be prevalingly applied in simulated environments so far. Its biggest problem in the heavily computational loading would obviously restrict the related uses in real-time cases. By considering all these methods, we proposed a fast moving shadow removal scheme by combining texture and statistical models. Our proposed method was experimentally proved to be stable and used the texture model instead of color model to simplify our systematic procedures efficiently. Furthermore, we made use of statistical methods to improve the performances of systems by successfully dealing with the textureless objects. This paper would be organized as follows, including the overall architecture, foreground object extraction, feature combination, experimental results on both our proposed moving shadow removal algorithm and the additional application (vehicle-counting), and conclusions.

2. Our Systematic Architecture for Moving Shadow Removal

In this section, we would introduce our entire architecture for algorithms of moving shadow removal which consisted of five blocks, including foreground object extraction, foreground-pixel extraction by edge-based shadow removal, foreground-pixel extraction by gray level-based shadow removal, feature combination, and the practical applications. The architecture diagram of our proposed algorithm was shown in Figure 1.

3. Foreground-Object Extraction

As Figure 1 showed, the sequence of images in gray-level should be taken as the input of the foreground object

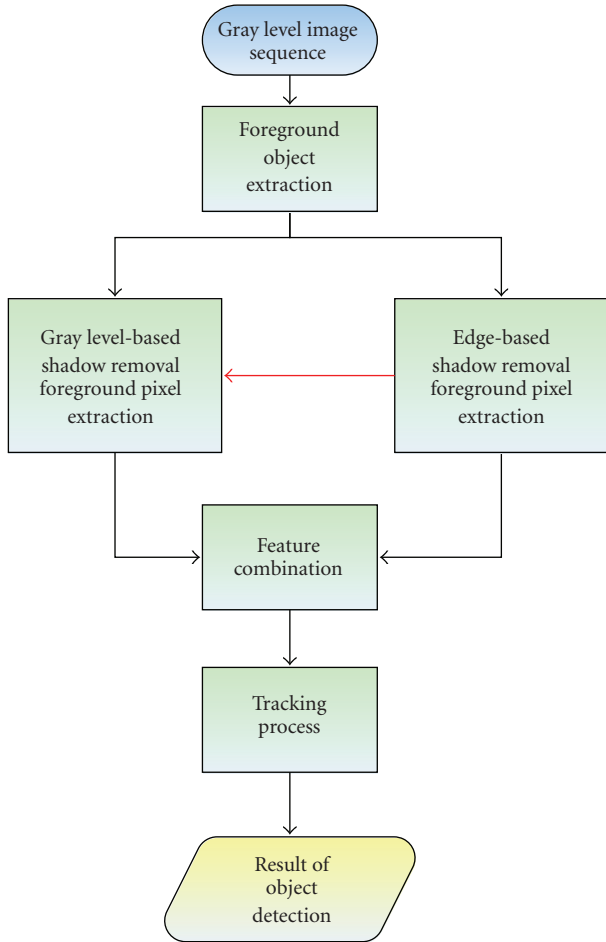


FIGURE 1: Architecture of the proposed shadow removal algorithm.

extraction processes, and the moving object with its minimum bounding rectangle as the output of the algorithm. We would incorporate Gaussian Mixture Model (GMM) into our mechanism functioning as the development of background image which is a representative approach to background subtraction. It would be more appropriate to choose the way by background subtraction for extracting foreground objects rather than that by temporal difference under the consideration of all the pros and cons of these two typical approaches. Furthermore, the latter could do a better job in extracting all relevant pixels, and this paper aimed at tackling the problems generated from the traffic monitoring systems where cameras would be usually set fixedly. Some previous studies [14, 15] have proposed a standard process of background construction, hence we would put a higher premium on the following two parts, inclusive of foreground-pixel extraction by edge-based and gray-level based shadow removal.

3.1. Foreground-Pixel Extraction by Edge-Based Shadow Removal. The main ideas for extracting foreground-pixels by the information of edges in the detected object were inspired by that the edges of object in interest would be identified

and removed much easier if the homogeneity in the shadow region could be within a small range of variance. Relatively, we could also obtain the features of edges for nonshadow regions. The flowchart of foreground-pixel extraction by edge-based shadow removal was shown in Figure 2. More clearly, we firstly used Sobel operations to extract the edges for both GMM-based background images and foreground objects. Figures 3 and 4 showed the results of edge extraction by Sobel operations from BI_edge and FO_edge_{MBR} where BI_edge and FO_edge_{MBR} represent the edges extracted from background images and foreground objects, respectively. The subscript “MBR” means the minimum bounding rectangle, the only region which we have to process inside.

In order to avoid extracting the undesired edges, for example, the edges of lane marking or textures on the ground surface, we technically took advantage of pixel-by-pixel max-operations on the extracted edges of background images and foreground objects, and the results would be expressed as MI_edge_{MBR} in (1):

$$MI_edge_{MBR}(x, y) = \max(FO_edge_{MBR}(x, y), BI_edge(x, y)), \quad (1)$$

where (x, y) represents the coordinate of the pixel, and one example of MI_edge_{MBR} would be illustrated in Figure 5.

Then, we subtracted BI_edge from MI_edge_{MBR} to obtain St_edge_{MBR} . Figure 6 showed the result of St_edge_{MBR} . Similarly, Figure 7 showed the same result without using our proposed procedures. It could be easily observed that the extracted edges of lane marking indeed reduced if our proposed procedure could be applied appropriately. Figure 8 indicated that our proposed procedure would also work well on the images with textured roads.

To demonstrate the necessity of our proposed procedure, we circled the edges of lane markings and textures on ground surface by red ellipses in both Figures 7 and 8(f). By using the max-operation, apparently, we could reduce the effect caused either by lane markings or textures on ground surfaces and also keep the homogeneous property inside the shadows.

After edge extracting, we used an adaptive binarization method to obtain the binary image from St_edge_{MBR} . Here, we took Sauvola’s method [16, 17], one kind of local binarization methods, to provide good results even in the condition of nonuniform luminance. Sauvola used an $n \times n$ mask, covered on the image in each scanning-iteration, to calculate the local mean $m(x, y)$ and standard deviation $s(x, y)$ of pixel-intensities in the mask as to determine a proper threshold according to the contrast in the local neighborhood of a pixel. If there is a high contrast in some region of the image, the equation $s(x, y) \approx R$ may result in the condition $t_{final}(x, y) \approx m(x, y)$. To reduce the influences by unimportant edges, we added a suppression term to Sauvola’s equation. Equation (2) shows the revised equation.

$$t_{final}(x, y) = m(x, y) \left[1 + k \left(\frac{s(x, y)}{R} - 1 \right) \right] + Th_{suppress}, \quad (2)$$

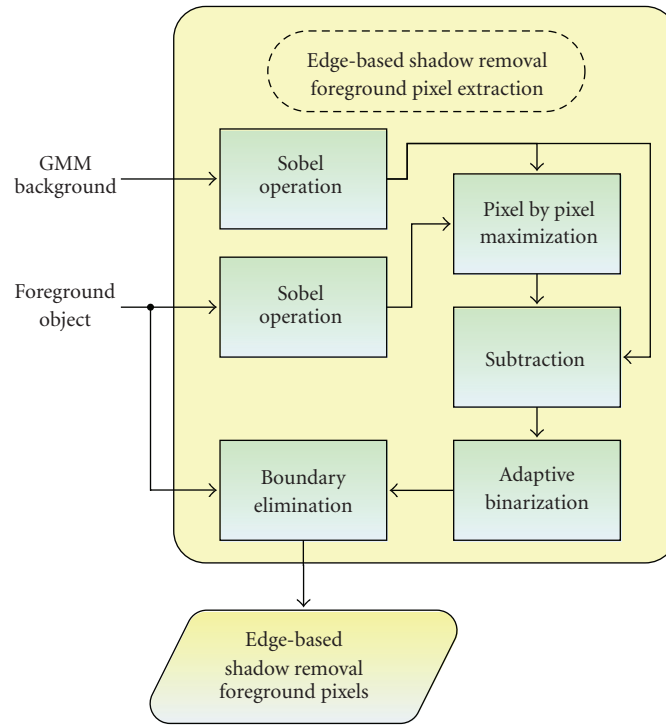
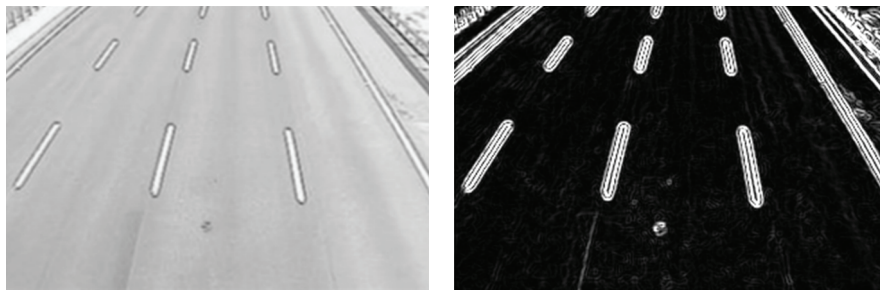


FIGURE 2: The flowchart of foreground-pixel extraction by edge-based shadow removal.



(a) Background image

(b) BL_{edge}

FIGURE 3: Sobel edge extraction from background images.



(a) Foreground Object

(b) FO_{edge}_{MBR}

FIGURE 4: Sobel edge extraction from moving objects.

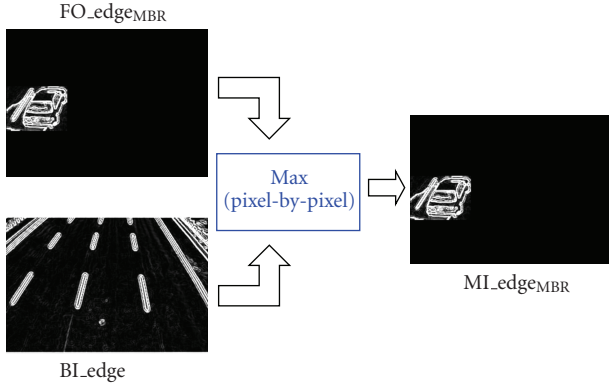


FIGURE 5: Max-operations and MI_edg_{MBR} .

where $m(x, y)$ and $s(x, y)$ are the mean and standard deviation of the mask centered at the pixel (x, y) , respectively, R is the maximum value of the standard deviation (in gray level image, $R = 128$), k is the parameter with positive values in the range $[0.2, 0.5]$, and $Th_{suppress}$ is a suppression term and its value is set to be 50 empirically in this paper.

We then applied $t_{final}(x, y)$ to the following binarization step at location (x, y) according to (3) once $t_{final}(x, y)$ had been obtained.

$$BinI_{MBR}(x, y) = \begin{cases} 0 & \text{if } St_edg_{MBR}(x, y) \leq t_{final}(x, y), \\ 255, & \text{otherwise,} \end{cases} \quad (3)$$

where $BinI_{MBR}$ represented the result after binarization. The result of using the binarization method on St_edg_{MBR} would be given in Figure 9. Another advantage of using adaptive binarization methods instead of taking a fixed threshold was that users would not have to manually set a proper threshold for each video scene. And that should be a significant factor for automatic monitoring systems.

We also had to remove the outer borders of $BinI_{MBR}$ because of the following problems. In Figure 9, the shadow region and real foreground object with the same motion vectors would make them always adjacent to each other. Also, the interior region of shadowed/foreground objects should be homogeneous/nonhomogeneous (nontexture or edgeless/dominant edged), which implied that the edges from shadows would appear at the outer borders of foreground objects. Considering these two properties, the objective of removing shadows could be treated as eliminating the outer borders and preserving the remaining edges which belongs to real foreground objects. Also, the latter property mentioned above might not be always satisfied. The interior region of shadows would be sometimes little textured (e.g., lane markings) like the example shown in Figure 10. We could solve this kind of problem by the procedures that we have mentioned earlier. From Figure 11(b), although the interior region of shadows had few edge-points after binarization, we could easily cope with these noise-like points only by using a specific filter after our subsequent processing procedures.

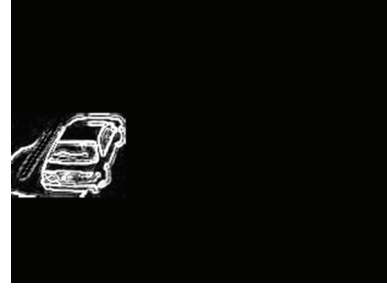


FIGURE 6: The result of subtracting BI_edg from MI_edg_{MBR} .

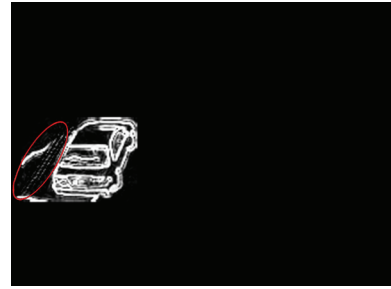


FIGURE 7: The result of subtracting BI_edg from MI_edg_{MBR} .

As mentioned above, we used a 7×7 mask to achieve boundary elimination. According to what have been observed, the widths of edge detected in shadows are in fact very approaching regardless of the edges that are extracted from far or near perspectives in the foreground images. And the width of edge is almost less than the length of 3 pixels in most conditions. As Figure 12 illustrated, we put the green mask on the binarized edge points (marked as yellow color) of $BinI_{MBR}$, and then scanned every point in $BinI_{MBR}$. If the region covered by the mask completely belongs to foreground objects (marked as white point), we reserve this point (marked as red color); otherwise, we eliminate the outer boundary elimination, we could obtain the features for non-shadow pixels, notated as $Ft_Edgebased_{MBR}$. In Figure 13, we showed an actual example with $Ft_Edgebased_{MBR}$ expressed as red points.

3.2. Foreground-Pixel Extraction by Gray Level-Based Shadow Removal. We tried to integrate the foreground-pixel extraction by gray level-based approach into our shadow removal algorithm, and this novel arrangement could enhance and stabilize the performance of that only by edge-based scheme. Figure 14 showed the flowchart of gray level-based shadow removal foreground pixel extraction. Worthily speaking, we developed a modified “constant ratio” rule by Gaussian model. We selected some pixels which belong to shadow-potential regions from foreground objects, calculated the darkening factors as our training data, and then built a Gaussian model for each gray level. Once the Gaussian model was trained, we could use this model to determine if

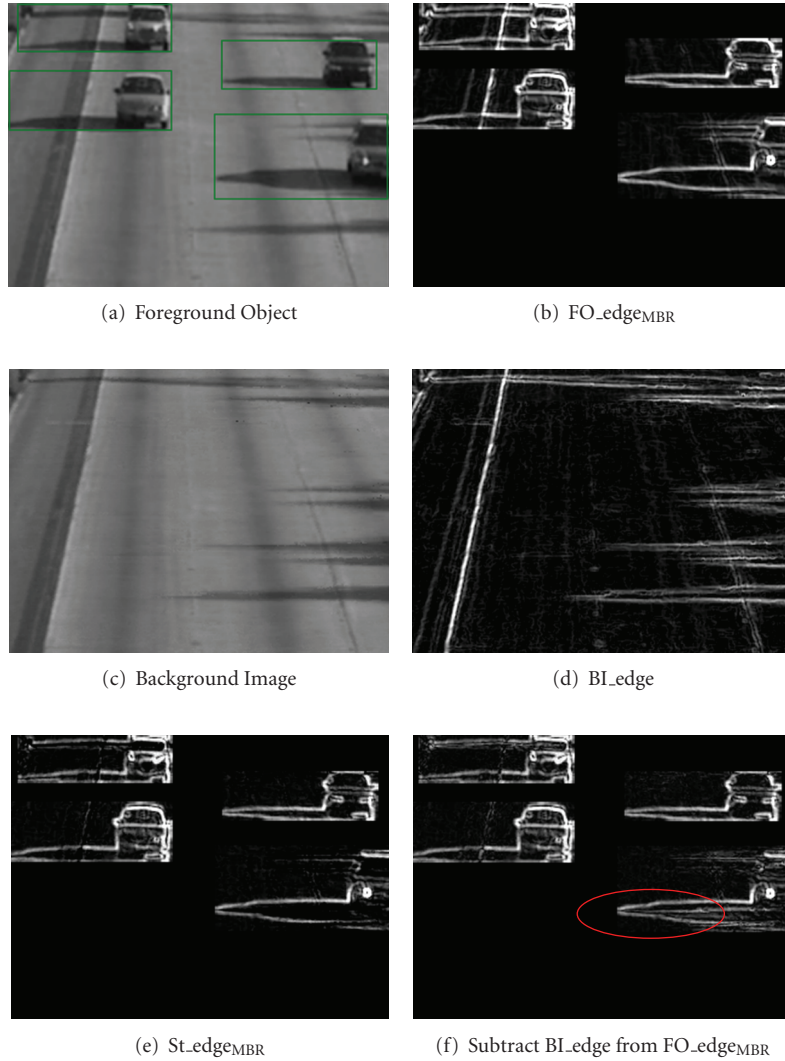
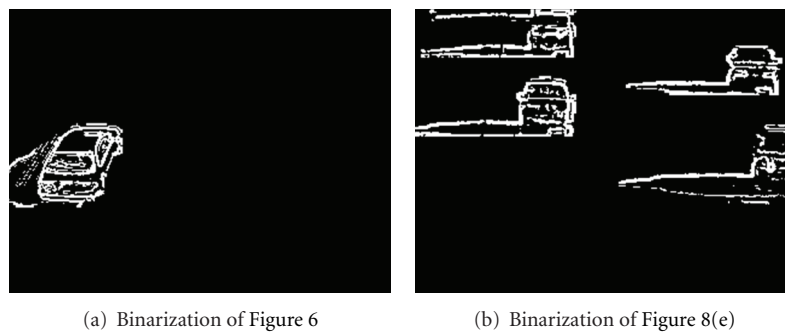


FIGURE 8: Examples of the ground surface with textures.

FIGURE 9: Examples of BinI_{MBR}.

each of the pixels inside foreground objects belonged to the shadowed region or not.

Here we would simply introduce the original “constant ratio” rule so as to illustrate our modifications. Some studies [12, 18–20], using the property of “constant ratio” for

shadow detection, expressed the intensity of each pixel on the coordinate (x, y) in terms of $I(x, y)$ with (4):

$$I(x, y) = \int e(\lambda, x, y) \rho(\lambda, x, y) \sigma(\lambda) d\lambda, \quad (4)$$

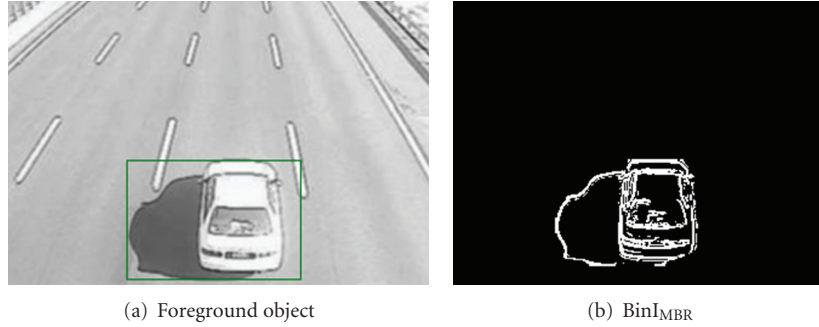


FIGURE 10: Homogeneous property of shadows, example 1.

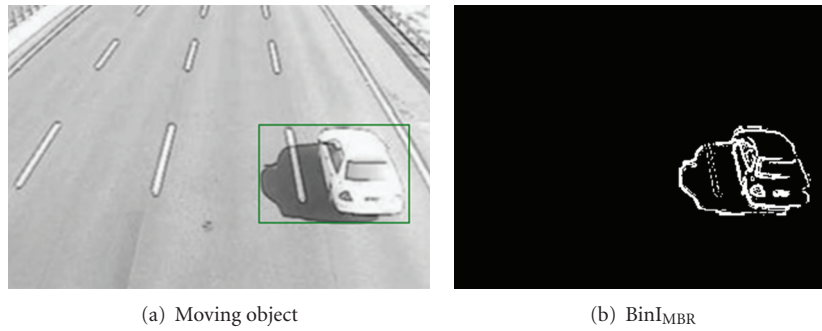


FIGURE 11: Homogeneous property of shadow, example 2.

where λ is the wavelength parameter, $e(\lambda, x, y)$ is the illumination function, $\rho(\lambda, x, y)$ is the spectral reflectance, and $\sigma(\lambda)$ is the sensitivity of camera sensor. The term $e(\lambda, x, y)$ indicated the difference between nonshadowed and shadowed regions. For backgrounds, the term $e(\lambda, x, y)$ is composed of the direct and diffused-reflected light components, but in the shadowed area, $e(\lambda, x, y)$ only contains the diffused-reflected light components. This difference implies the constant ratio property. Equation (5) shows the ratio of $I^{sh}(x, y)$ and $I^{bg}(x, y)$ where $I^{sh}(x, y)$ and $I^{bg}(x, y)$ represent the intensities of a shadowed pixel and a nonshadowed background pixel, respectively, and α is called the darkening factor which will be a constant over the whole image.

$$\frac{I^{sh}(x, y)}{I^{bg}(x, y)} = \alpha. \quad (5)$$

3.2.1. *Gaussian Darkening Factor Model Updating.* As Figure 15 showed, we would rather stimulate the darkening factor with respect to each gray level by one Gaussian Model than that with respect to each pixel. In the beginning, we would select the shadow-potential pixels as the updating data of Gaussian models by our three predefined conditions introduced in the followings.

- (1) Pixels must belong to the foreground objects, for the shadowed pixels must be part of the foregrounds ideally.

- (2) The intensity of a pixel (x, y) in the current frame should be smaller than that in the frame of backgrounds, for the shadowed pixels must be darker than background-pixels.
- (3) The pixels obtained from the foreground-pixel extraction by edge-based shadow removal should be excluded to reduce the number of pixels which might be classified as nonshadowed pixels.

For the practical case shown in Figure 16, the red pixels were the selected points for Gaussian model updating.

After the pixels for updating were selected, we would update the mean and standard deviation of Gaussian model. Figure 17 displayed the flowchart of the updating process of Gaussian darkening factor model. The darkening factor α_k could be calculated as in (6):

$$\frac{I^{selected}(x, y)}{I^{bg}(x, y)} = \alpha_k, \quad (6)$$

where $I^{selected}(x, y)$ is the intensity of the selected pixel at (x, y) , and $I^{bg}(x, y)$ is the intensity of the background-pixel at (x, y) . After calculating the darkening factor, we would update the k th Gaussian model.

We set a threshold as a minimum number of updating times, and the updating times of each Gaussian model must exceed this threshold to ensure the stability of each

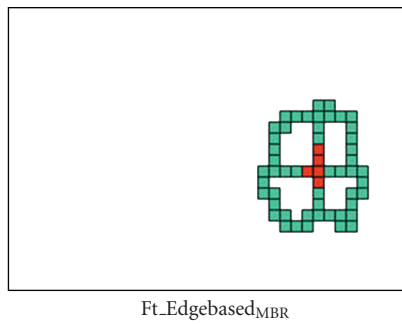
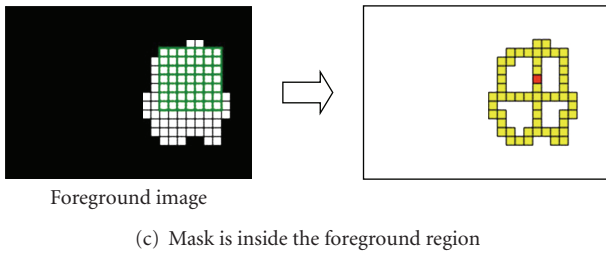
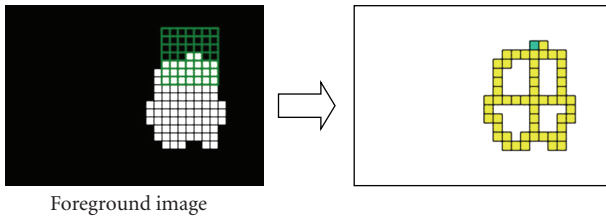
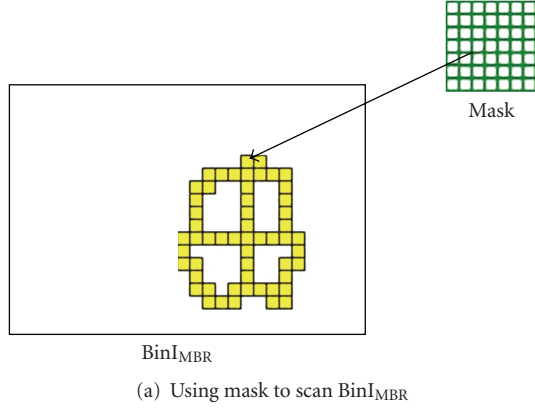


FIGURE 12: Illustrations of boundary elimination.

model. Besides, in order to reduce the computation loading of updating procedure, we gave a limit that each Gaussian model could only be updated at most 200 times for one frame.

3.2.2. Determination of Non-Shadowed Pixels. Here, we introduced how to extract the nonshadowed pixels by using

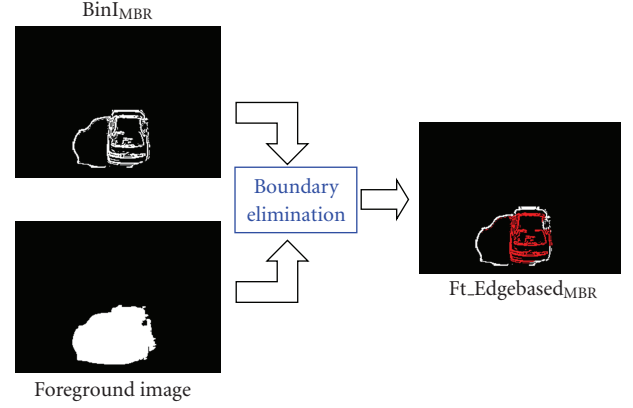


FIGURE 13: An example of boundary elimination.

the trained Gaussian darkening factor model. Figure 18 gave us the rules to calculate the difference between the mean of Gaussian model and the darkening factor, and check if the difference was smaller than 3 times of standard deviation. If yes, the pixel would be classified as shadowed. Otherwise, it would be considered as the nonshadowed pixel and could be reserved as a feature point.

Figure 19 described our tasks to determine the nonshadowed pixels. If the k th Gaussian model was not trained, we would go checking if the nearby Gaussian models were marked as trained or not. In our programs, we selected the nearby 6 Gaussian models for checking, and we chose the nearest one if there existed any trained Gaussian model. Figure 20 gave an example where the pixels labeled by red color were the extracted feature pixels after our determination task, and we denoted the set of these pixels as Ft_DarkeningFactor_{MBR}.

4. Feature Combination

We combined two kinds of features which have been introduced in the Section 3 in our algorithm to extract foreground objects in a more accurate way. Figure 21 exhibited the flowchart of our feature combination. We integrated these two features by applying “OR” operations. Figure 22 demonstrated a real example of processed images by our combined features, called the feature-integration images in this paper.

After acquiring the feature-integration images, we would locate the real foreground objects, namely, the foreground objects excluding the shadowed regions. Hence, we used the connected-component-labeling approach with the minimum bounding rectangle to orientate the real foreground objects. What is different from the common applications, we would make some necessary preprocessing procedures before applying the connected-component-labeling procedure. The preprocesses consisted of filtering and dilation operations. Both the median filter operation and morphological dilation operation was conducted just once. In Figure 23, we could see that there existed some odd points in the left part

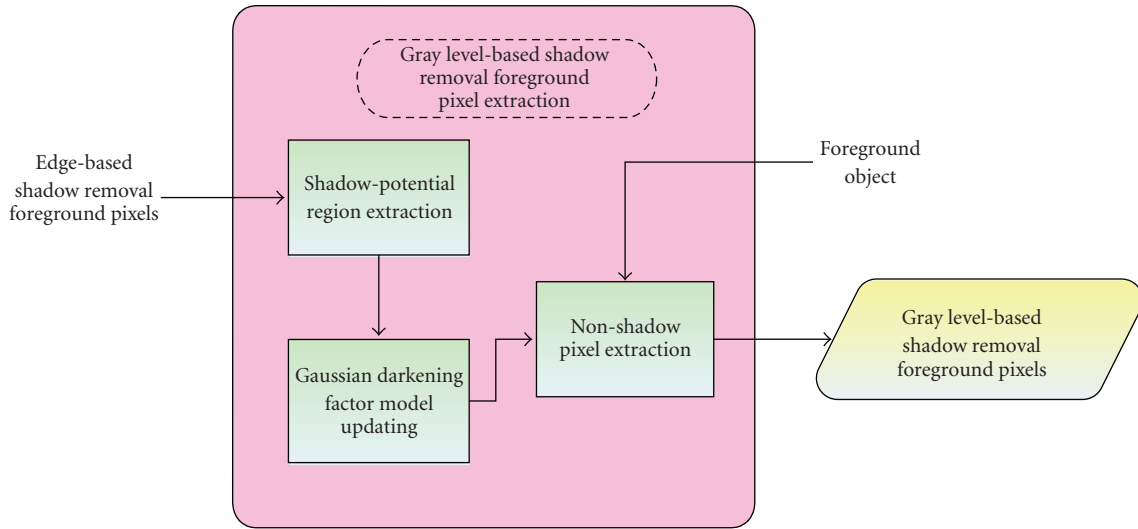


FIGURE 14: Flowchart of gray level-based shadow removal foreground pixel extraction.

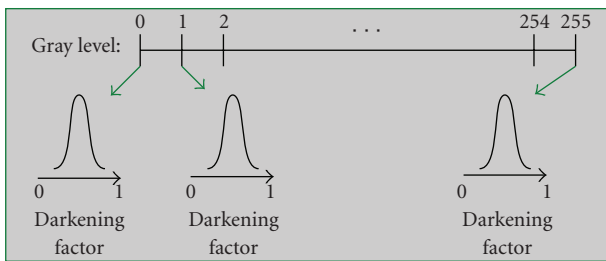


FIGURE 15: An illustrated figure, each gray level has a Gaussian model.

```

if {WidthMBR,k < Thmin_MBR.Width} AND
   {HeightMBR,k < Thmin_MBR.Height}
    Eliminate kth Minimum Bounding Rectangle;
end
    
```

ALGORITHM 1

of the feature-integration images. It was obviously due to the influences by land markings (see Figure 23(a)). In Figure 23(b), these pixels could be eliminated after the filtering procedure. We then applied dilation operations to the results after filtering in order to concentrate the left feature pixels, as shown in Figure 24.

After that, the connected-component-labeling approach could be applied on the dilated images. As for our defined rules for this procedure, we used the minimum bounding rectangle for each independent region. Then, if any two minimum bounding rectangles are close to each other, we will merge these two rectangles. Finally, iteratively check and merge the previous results till no rectangle can be merged with.

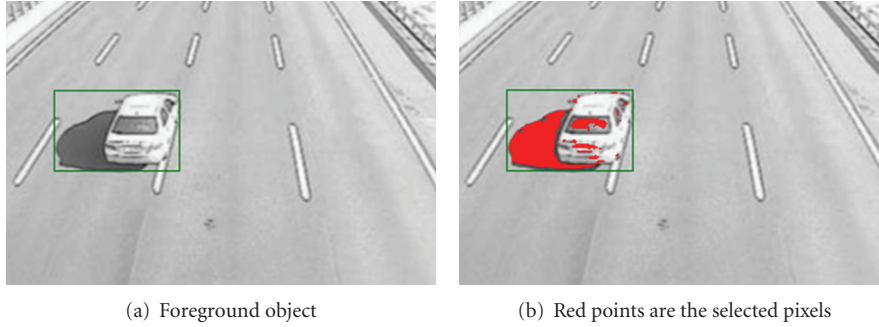
After labeling and grouping, we would use the size filter to eliminate the minimum bounding rectangle of which the width and height were both smaller than a threshold, as depicted in Algorithm 1. The subscript “*k*” indicated the *k*th minimum bounding rectangle. Figure 25 showed some examples of the final located real objects; the green and light blue rectangles revealed the foreground objects and final located real objects, respectively.

5. Experimental Results

In this section, we would demonstrate the results of our proposed shadow removal algorithm. We implemented our algorithm on the platform of PC with P4 3.0 GHz and 1 GB RAM. The software we used is Borland C++ Builder on Windows XP OS. All of the testing inputs are uncompressed AVI video files. The resolution of each video frame is 320 × 240. The average processing time is 13.84 milliseconds for each frame.

5.1. Experimental Results of Our Shadow Removal Algorithm.

In the followings, we showed our experimental results under no occlusion situation in different scenes. In comparison with the results without using our algorithm, we used “red” and “green” rectangles to indicate the detected objects after processing with and without applying our shadow removal algorithm, respectively. In Figure 26, we could see that the proposed algorithm indeed successfully detected the real objects and neutralized the negative influences by shadows, since the intensity of shadowed regions was low enough to be distinguished from that of backgrounds. Besides, our proposed algorithm could also cope with the larger-scale shadows and provide the satisfactory results for different sizes of vehicles like trucks which would be shown in Figures 26(d) and 26(f). In Figure 27, we demonstrated the processed results for different kinds of shadows such as



(a) Foreground object

(b) Red points are the selected pixels

FIGURE 16: The selected pixels for Gaussian model updating.

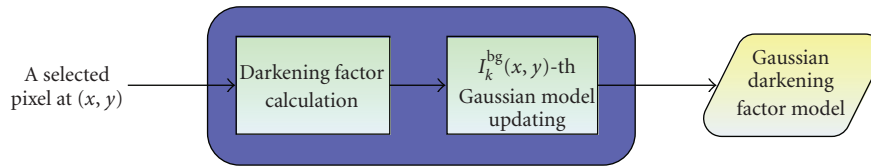


FIGURE 17: Gaussian darkening factor model updating procedure.

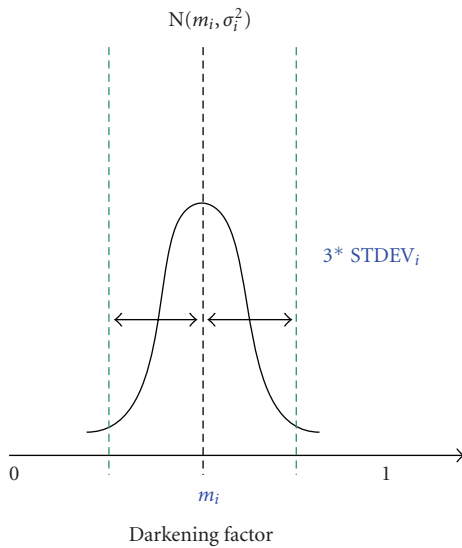


FIGURE 18: Illustration of determination.

the smaller-scale shadows (shown in Figure 27(a)), the low contrast between intensities of shadows and backgrounds (shown in Figure 27(c)), and both the smaller-scale and lowly contrastive shadows (shown in Figure 27(e)). Clearly, our proposed method could work robustly no matter how large and how observable the shadows would be. Also in Figure 28, we gave the testing results of another scene where motorcycles and riders were precisely detected (shown in Figures 28(c) and 28(d)). Figure 29 provided the compared results processed by our proposed algorithm with those by the representative methods which could have been accessible

through internet for the highway sequences of videos, which showed a much better processed result by our proposed algorithm (right columns).

5.1.1. Occlusions Caused by Shadows. Here, we would demonstrate some examples of occlusions caused by the influences of shadows. In Figures 30(a), 30(e), 30(g) and 30(i), two vehicles (or motorcycle—vehicle) were detected in the same rectangle due to the influences of shadows, and Figure 30(c) indicated an even worse case in which vehicles were framed together on account of light shadows. From all the figures in the right column of Figure 30, it was apparent to see, our method could correctly detect the foreground objects under the influences of shadows. Moreover, we could still have the correct results (shown in Figure 30(j)) in the foreground object detection for a more difficult case (shown in Figure 30(i)) that three shadowed regions were detected as one foreground object. That was to say, our proposed algorithm could handle the problems of occlusions caused by shadows which have been always considered as tough tasks.

5.1.2. Discussions of Gray Level-Based Method. In Section 3.2, we used darkening factors to enhance the performance and reliability of the proposed algorithm. We hence in here made the comparisons of experimental results by applying and not applying our proposed approach mentioned in Section 3.2. Figure 31 showed a conspicuous example in which green/red rectangles represented the foreground objects/detected objects. Figures 31(a), 31(c), and 31(e) were the results without using the foreground-pixel-extraction approach by gray level-based shadow removal. In other words, the only feature for extracting foreground objects was

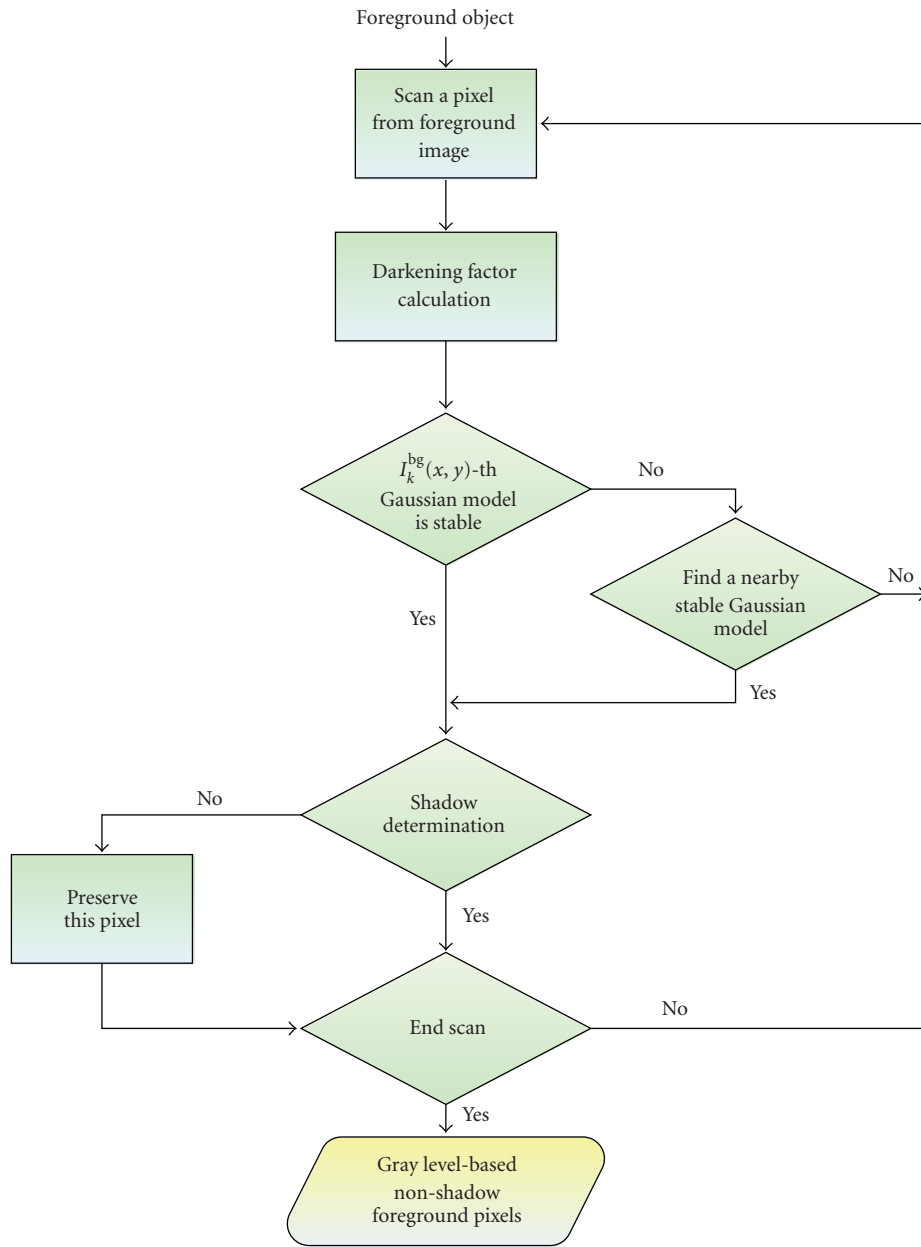
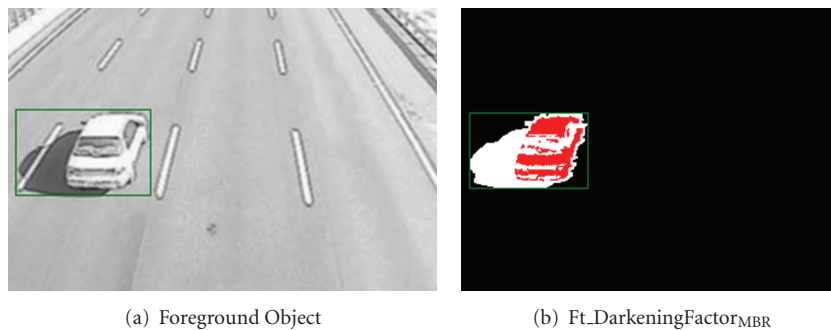


FIGURE 19: Flowchart of the nonshadowed pixel determination.



(a) Foreground Object

(b) Ft_DarkeningFactor_MBR

FIGURE 20: An example of gray level-based nonshadow foreground-pixel extraction.

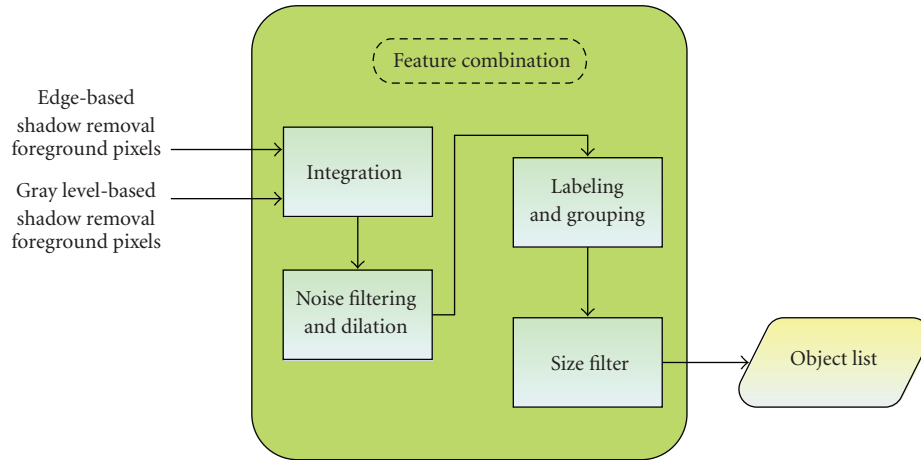


FIGURE 21: Flowchart of feature combination.

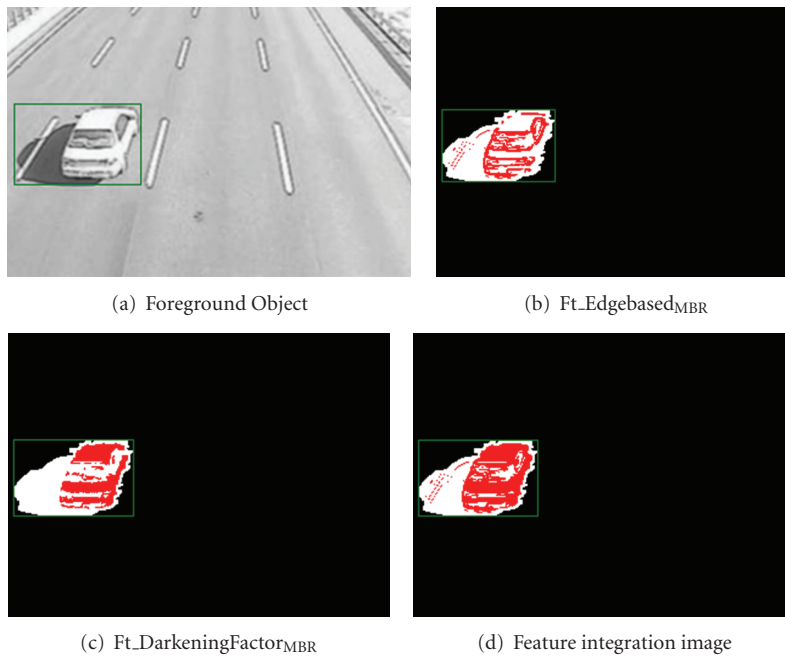


FIGURE 22: An example after integration.

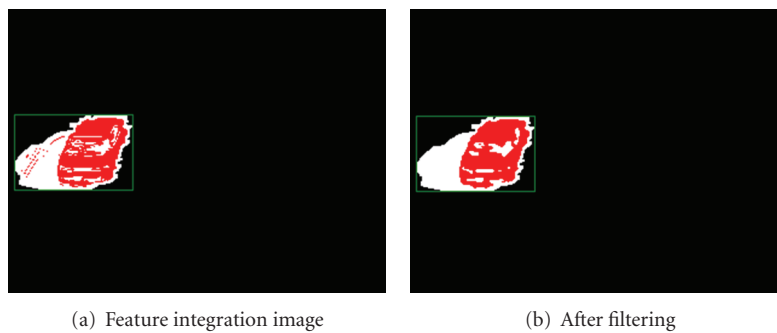


FIGURE 23: Filtering by median filter for feature-integration images.



FIGURE 24: The result of dilation operations on Figure 23(b).

TABLE 1: Descriptions of testing videos for vehicle counting.

| Testing video | Scene | Shadow description | Video FPS |
|---------------|------------|--------------------|-----------|
| Video1 | Highway | Obvious and Large | 30 |
| Video2 | Highway | Light and Large | 30 |
| Video3 | Expressway | Obvious and Large | 25 |

obtained from the foreground-pixel-extraction approach by edge-based shadow removal that we introduced in Section 3.1. But it would bring about the failure of detections when the objects were edgeless (or textureless). As the images shown in the left column of Figure 31, the car roofs could be regarded as edgeless, which might result in that the detected object were in pieces or that only the rear bumper could be detected. Figures 31(b), 31(d), and 31(f) exhibited the better results which were obtained from our introduced structure of feature combination including the information of edge-based and gray level-based shadow removal.

5.2. Vehicle Counting. To prove the validity and versatility of the proposed approach, we tried to apply our developed algorithm to vehicle counting, one of the popular applications in ITS. We had 3 testing videos for vehicle counting, and the scenes and properties of shadows in each video were arranged in Table 1. Figure 32 showed the scene of each video for vehicle counting. In order to illustrate the compared and statistical results in a more convenient way, we had Video1 in 6 sectors, Video2 in 13 sectors, and Video3 in 2 sectors, respectively. Each of the sectors was about 2 minutes. We had 4 lanes for both Video1 and Video2, and 2 lanes for Video3. In Table 2, we gave the number of passing vehicles on each lane for each video manually.

We calculated the accuracy rate for each sector by (7) to give the comparisons in a more reasonable manner.

$$\text{Accuracy rate} = \left[1 - \left(\frac{|N_{\text{program}} - N_{\text{manual}}|}{N_{\text{manual}}} \right) \right] \times 100\%, \quad (7)$$

where N_{manual} and N_{program} represented the number of vehicles which were obtained from manual operations and our programs, respectively. As Table 3 showed, we could obtain

the compared consequences of average accuracy rates by using the foreground object-detection results with/without applying our proposed algorithm as the inputs in the experiments of vehicle counting. Table 4 indicated the average accuracy rate for each video in all lanes. From the testing results, we would also list two kinds of failed examples in Figure 33 to indicate some erroneous detected results in the vehicle counting application, which may be reasonable to illustrate the failed conditions in a more quantized manner. One possible condition which might result in the false consequences came from the much more complicated textures within the detected shadows and was illustrated in Figure 33(a). Figure 33(a) revealed that the algorithm failed to provide a correct result in vehicle counting due to the overcomplicated edge information reflected in detected shadows. The major reason can be easily observed that the shadow of some specific object was not successfully detected and eliminated. In fact, the research of shadow detection and removal based upon all the processes by image processing only has been a tough issue since we did not try to make use of any other information from some useful sensors but cameras. Moreover, this paper aimed at developing a practical algorithm in image processing procedures to efficiently remove the shadowing effect before dealing with the applications of ITS, which would have less impact on the performance of shadow removal and make the influences dependent on some specific application. Owing to the rare appearance of such conditions in a longer recorded file of videos, the detected error rate in counting vehicles could be kept lower to a satisfactory range. As for the other possible condition that might result in the false detection results, Figure 33(b) illustrated this kind of example and also showed the false detection consequence in the vehicle counting application. This occlusion case for two cars caused the false counting result because the shadows would be too unapparent to be correctly detected and the two detected objects were moving simultaneously. This kind of problem should be categorized to another research field of image processing, yet not the issue that we have focused on in this paper. Since this phenomenon may result from many other conditions such as the dynamic behavior of moving vehicles, discussions of different shadows, and influences of light reflection, we would rather concentrate on developing a more practical and automatic algorithm in shadow removal. Those experimental results revealed that our proposed algorithm could not only work well in vehicle counting but also improve the performance of any applications under constraints of shadows.

6. Conclusions

We in this paper present a real-time and efficient moving shadow removal algorithm based on versatile uses of GMM, including the background removal and development of features by Gaussian models. Our algorithm innovates to use the homogeneous property inside the shadowed regions, and hierarchically detects the foreground objects by extracting the edge-based, gray level-based features, and

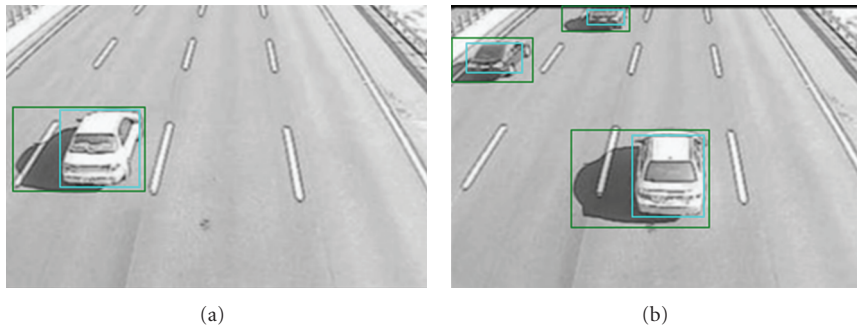


FIGURE 25: Examples of final located real objects.

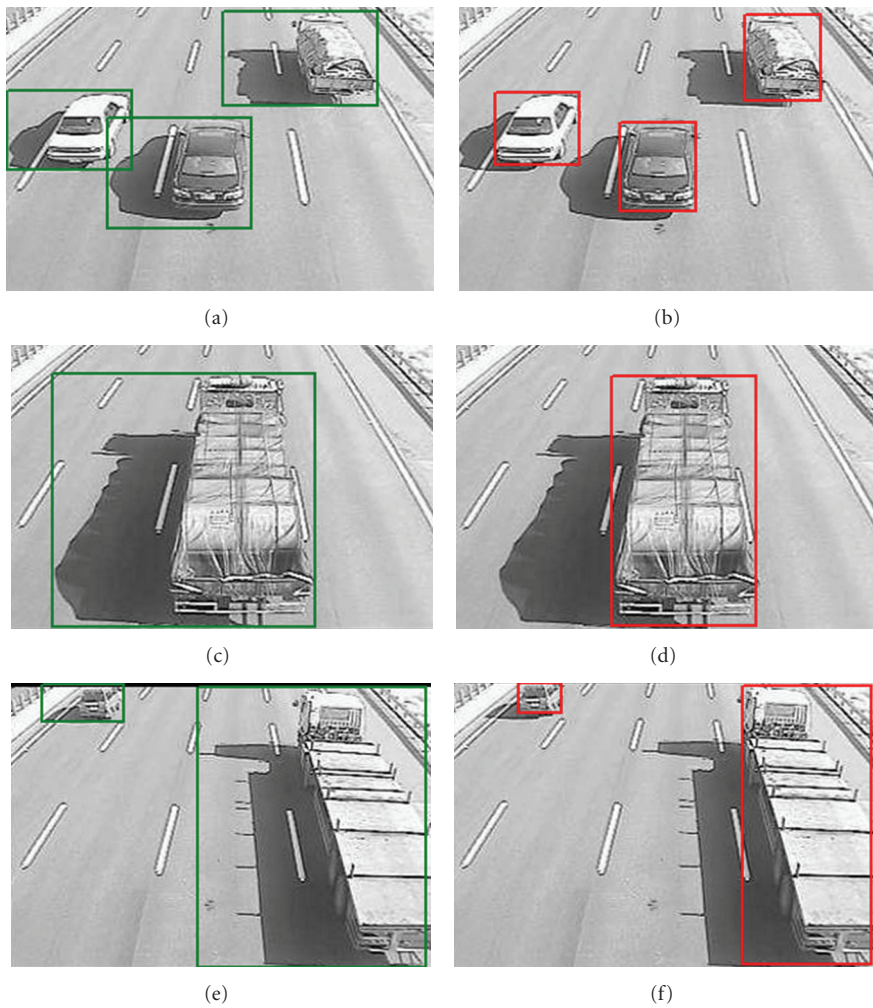


FIGURE 26: Experimental results of foreground object detection.

TABLE 2: Number of passing vehicles in each lane.

| Testing video | Partition number | Lane1 (vehicles) | Lane2 (vehicles) | Lane3 (vehicles) | Lane4 (vehicles) |
|---------------|------------------|---------------------|---------------------|---------------------|---------------------|
| Video1 | 6 | 102 | 189 | 116 | 89 |
| Video2 | 13 | 464 | 505 | 373 | 261 |
| Video3 | 2 | 58 | 75 | — | — |

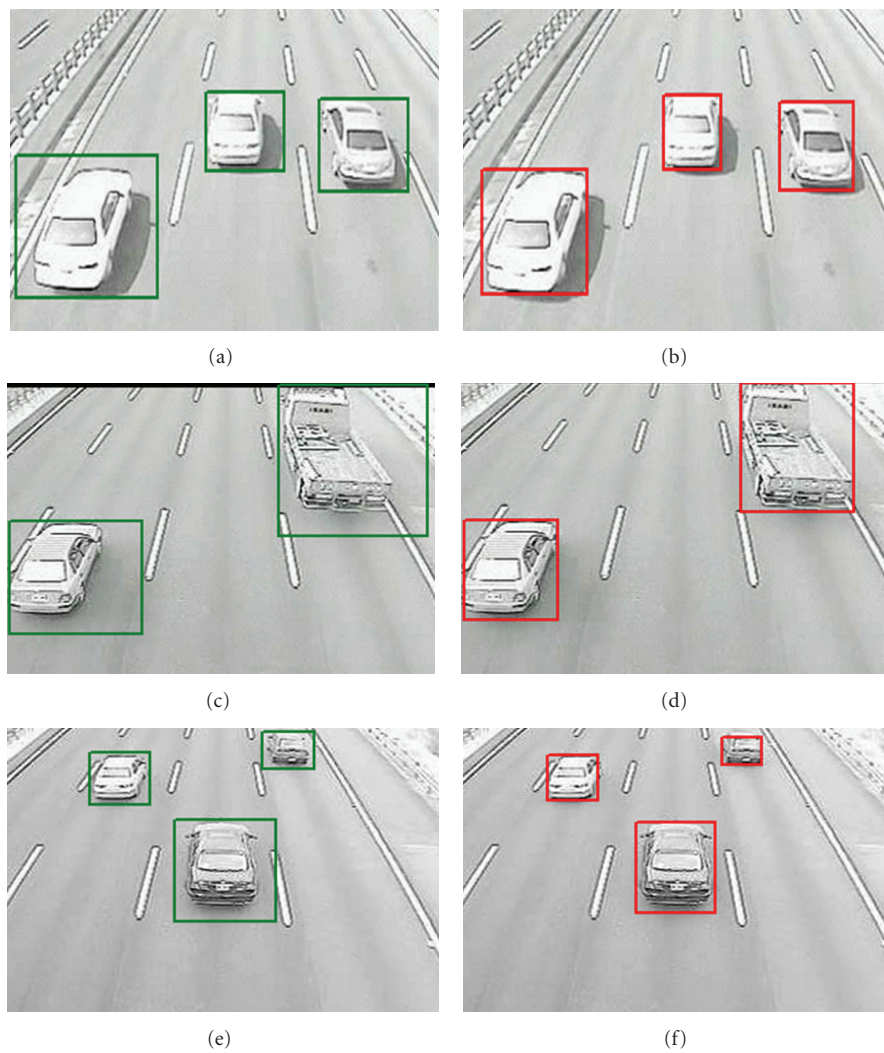


FIGURE 27: Experimental results for different kinds of shadows.

TABLE 3: Vehicle counting results.

| Testing videos | Compared methods | Lane1 | Lane2 | Lane3 | Lane4 |
|----------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| | | (Average Accuracy rate) | (Average Accuracy rate) | (Average Accuracy rate) | (Average Accuracy rate) |
| Video1 | Without Shadow Removal | 81.58% | 97.50% | 96.57% | 82.29% |
| | With Proposed Algorithm | 100% | 99.02% | 97.22% | 100% |
| Video2 | Without Shadow Removal | 92.88% | 96.27% | 95.55% | 89.51% |
| | With Proposed Algorithm | 97.59% | 99.31% | 99.68% | 99.26% |
| Video3 | Without Shadow Removal | 95.16% | 97.14% | — | — |
| | With Proposed Algorithm | 100% | 100% | — | — |

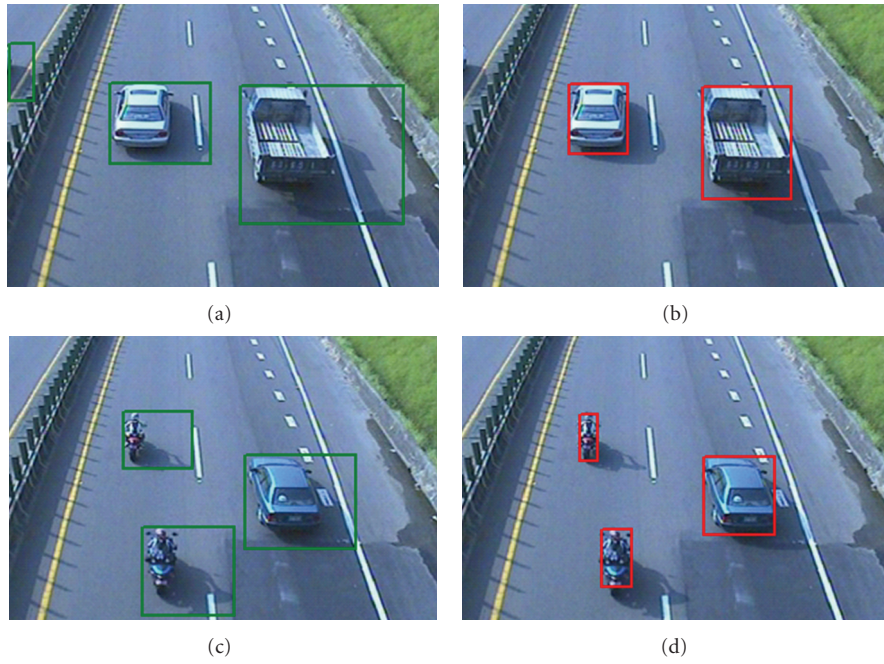


FIGURE 28: Experimental results of foreground object detection.

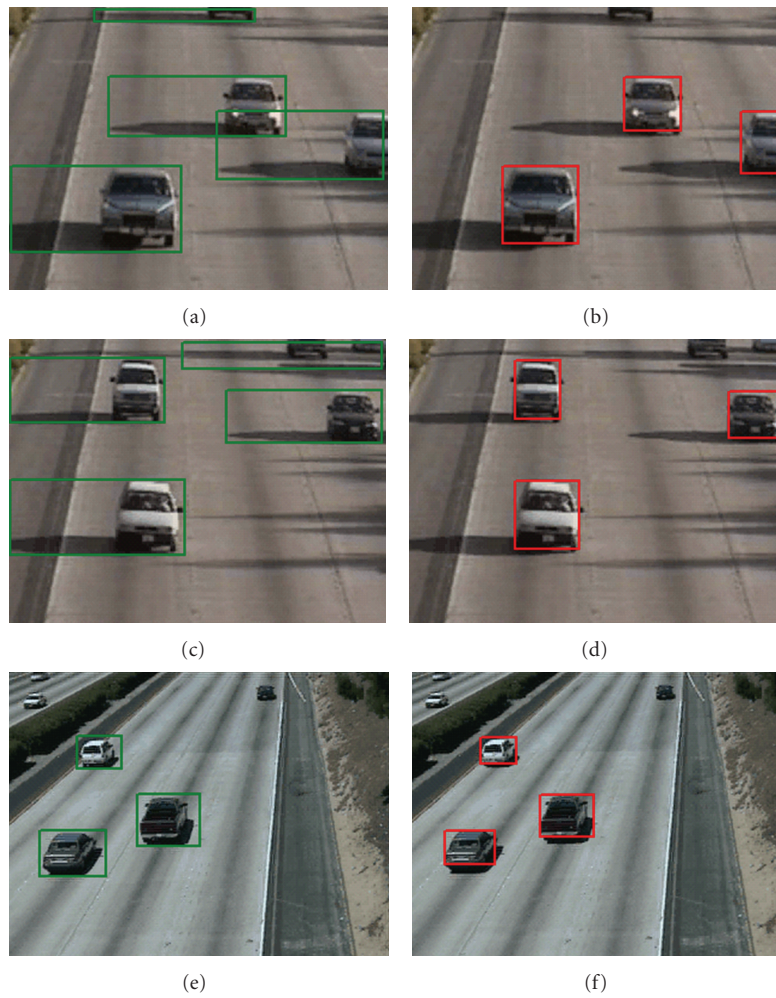


FIGURE 29: Experimental results of foreground object detection.

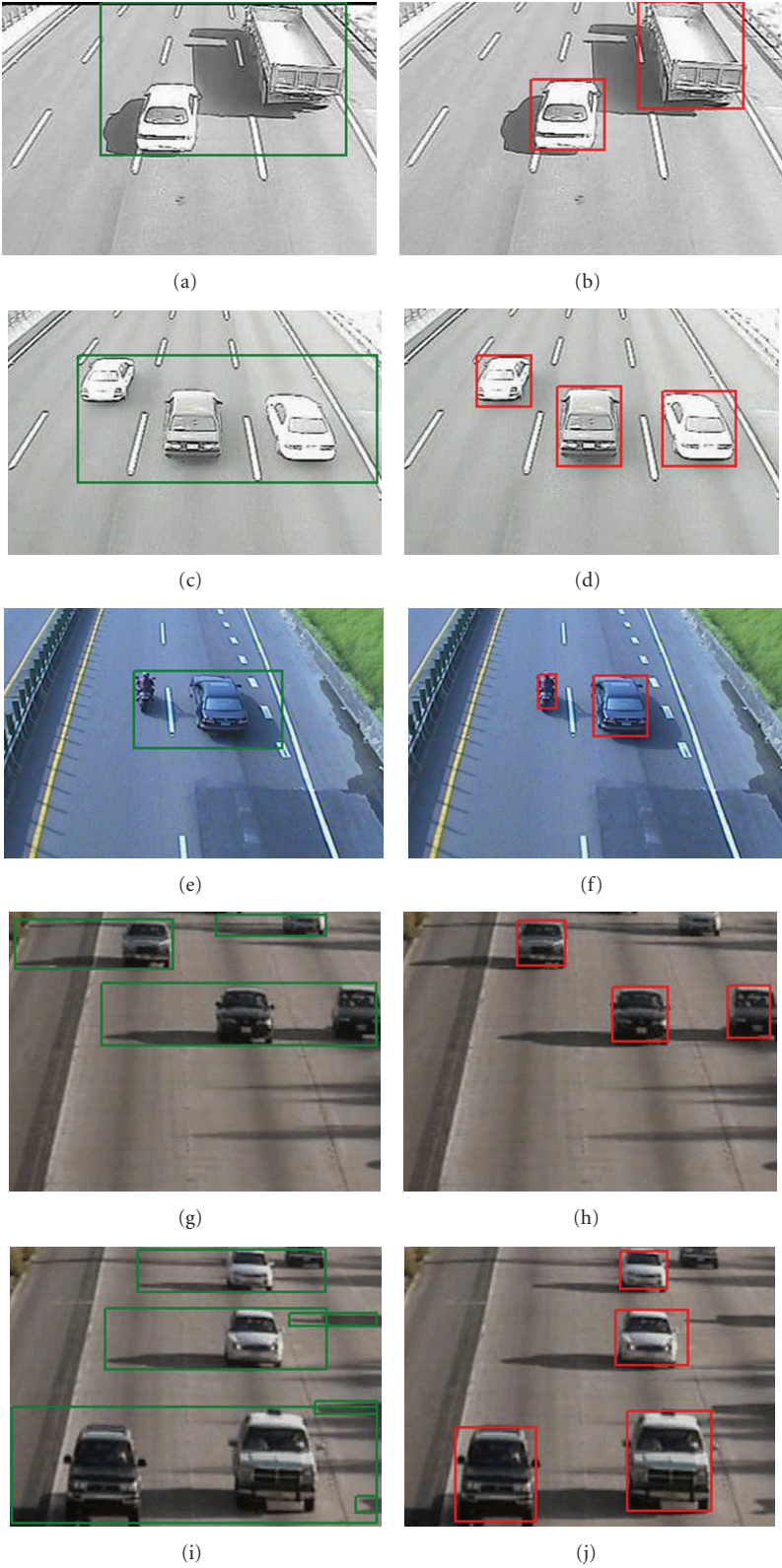


FIGURE 30: Experimental results under the occlusion situation.

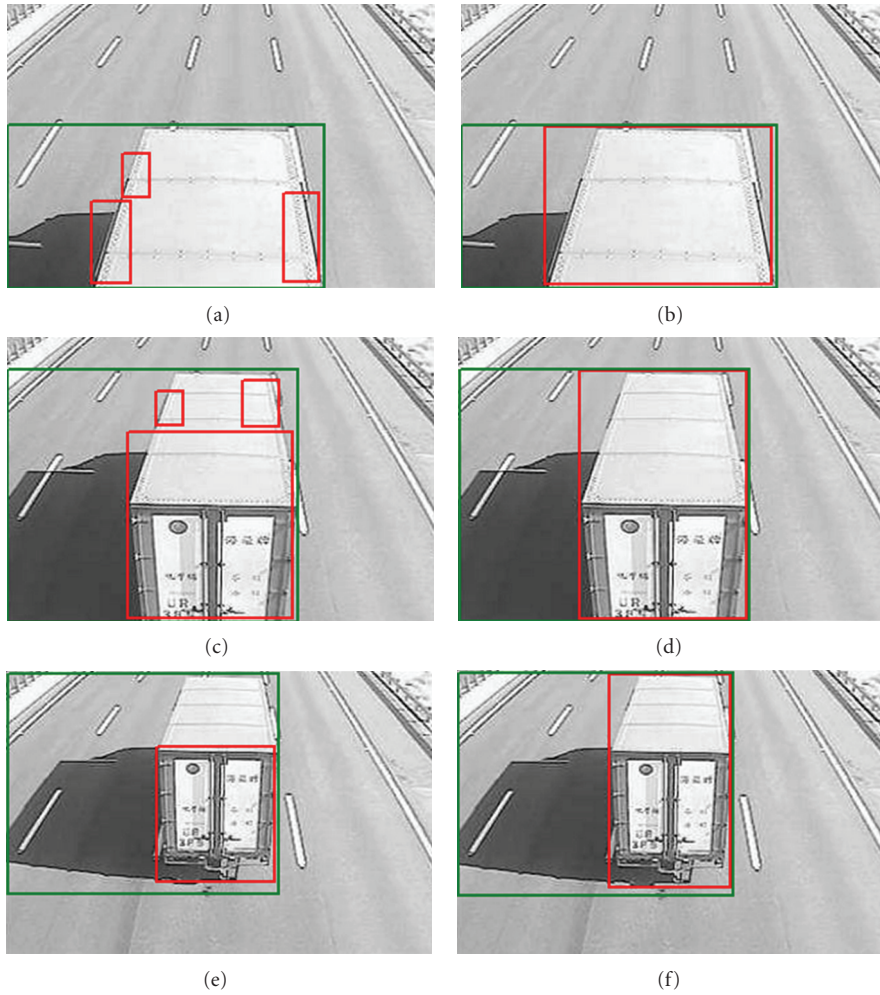


FIGURE 31: Compared results of not applying and applying gray level-based shadow removal foreground-pixel extraction.

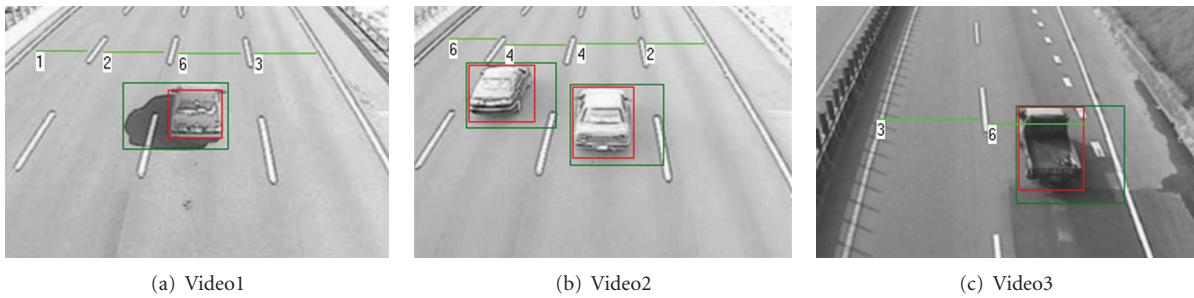


FIGURE 32: Scenes of videos for vehicle counting.



FIGURE 33: Some failed examples of image frames for (a) the much-texture case and (b) the special occlusion case.

TABLE 4: Average accuracy of all lanes in each video.

| Testing Video | | Average Accuracy Rate |
|---------------|-------------------------|-----------------------|
| Video1 | Without Shadow Removal | 89.49% |
| | With Proposed Algorithm | 99.06% |
| Video2 | Without Shadow Removal | 93.55% |
| | With Proposed Algorithm | 98.96% |
| Video3 | Without Shadow Removal | 96.15% |
| | With Proposed Algorithm | 100% |

feature combination. Our approach can be characterized by some original procedures such as “pixel-by-pixel maximization”, subtraction of edges from background images in the corresponding regions, adaptive binarization, boundary elimination, the automatic selection mechanism for shadow-potential regions, and the Gaussian darkening factor model for each gray level.

Among all these proposed procedures, “pixel-by-pixel maximization” and subtraction of edges from background images in the corresponding regions deal with the problems which result from the shadowed regions with edges. Adaptive binarization and boundary elimination are developed to extract the foreground-pixels of nonshadowed regions. Most significantly, we propose the Gaussian darkening factor model for each gray level to extract nonshadow pixels from foreground objects by using the information of gray levels, and integrate all the useful features to locate the real objects without shadows. Finally, in comparison with the previous approaches, the experimental results show that our proposed algorithm can accurately detect and locate the foreground objects in different scenes and various types of shadows. What’s more, we apply the presented algorithm to vehicle counting to prove its capability and effectiveness. Our algorithm indeed improves the results of vehicle counting and it is also verified to be efficient with the prompt processing speed.

Acknowledgment

This work was supported in part by the Aiming for the Top University Plan of National Chiao Tung University, the Ministry of Education, Taiwan, under Contract 99W962, and supported in part by the National Science Council, Taiwan, under Contracts NSC 99-3114-E-009 -167 and NSC 98-2221-E-009 -167.

References

- [1] W. Zhang, X. Z. Fang, X. K. Yang, and Q. M. J. Wu, “Moving cast shadows detection using ratio edge,” *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1202–1214, 2007.
- [2] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detecting moving objects, ghosts, and shadows in video streams,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337–1342, 2003.
- [3] M.-T. Yang, K.-H. Lo, C.-C. Chiang, and W.-K. Tai, “Moving cast shadow detection by exploiting multiple cues,” *IET Image Processing*, vol. 2, no. 2, pp. 95–104, 2008.
- [4] A. Cavallaro, E. Salvador, and T. Ebrahimi, “Shadow-aware object-based video processing,” *IEEE Proceedings: Vision, Image and Signal Processing*, vol. 152, no. 4, pp. 398–406, 2005.
- [5] K.-T. Song and J.-C. Tai, “Image-based traffic monitoring with shadow suppression,” *Proceedings of the IEEE*, vol. 95, no. 2, pp. 413–426, 2007.
- [6] N. Martel-Brisson and A. Zaccarin, “Learning and removing cast shadows through a multidistribution approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1133–1146, 2007.
- [7] A. J. Joshi and N. P. Papanikolopoulos, “Learning to detect moving shadows in dynamic environments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2055–2063, 2008.
- [8] A. J. Joshi and N. Papanikolopoulos, “Learning of moving cast shadows for dynamic environments,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA ’08)*, pp. 987–992, May 2008.
- [9] A. Leone, C. Distanto, and F. Buccolieri, “A texture-based approach for shadow detection,” in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS ’05)*, pp. 371–376, September 2005.
- [10] M. Mohammed Ibrahim and R. Anupama, “Scene adaptive shadow detection algorithm,” *Proceedings of World Academy of Science, Engineering and Technology*, vol. 2, pp. 1307–6884, 2005.
- [11] J.-W. Hsieh, S.-H. Yu, Y.-S. Chen, and W.-F. Hu, “Automatic traffic surveillance system for vehicle tracking and classification,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 179–187, 2006.
- [12] C. Benedek and T. Szirányi, “Bayesian foreground and shadow detection in uncertain frame rate surveillance videos,” *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 608–621, 2008.
- [13] M. Xiao, C.-Z. Han, and L. Zhang, “Moving shadow detection and removal for traffic sequences,” *International Journal of Automation and Computing*, vol. 4, no. 1, pp. 38–46, 2007.
- [14] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’99)*, vol. 2, pp. 246–252, June 1999.
- [15] P. Suo and Y. Wang, “An improved adaptive background modeling algorithm based on Gaussian mixture model,” in *Proceedings of the 9th International Conference on Signal Processing (ICSP ’08)*, pp. 1436–1439, October 2008.
- [16] J. Sauvola and M. Pietikäinen, “Adaptive document image binarization,” *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.
- [17] F. Shafait, D. Keysers, and T. M. Breuel, “Efficient implementation of local adaptive thresholding techniques using integral images,” in *Document Recognition and Retrieval XV*, vol. 6815 of *Proceedings of SPIE*, San Jose, Calif, USA, January 2008.
- [18] P. L. Rosin and T. Ellis, “Image difference threshold strategies and shadow detection,” in *Proceedings of the 6th British Machine Vision Conference*, 1994.
- [19] Y. Wang, K.-F. Loe, and J.-K. Wu, “A dynamic conditional random field model for foreground and shadow segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 279–289, 2006.
- [20] I. Mikić, P. C. Cosman, G. T. Kogut, and M. M. Trivedi, “Moving shadow and object detection in traffic scenes,” in *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 1, pp. 321–324, 2000.