Title of the Paper: An efficient method of cast shadow removal using multiple features

Authors: Chu Tang, M.Omair Ahmad, and Chunyan Wang

Email: chu_tan, omair, chunyan@ece.concordia.ca

Communicating Author

Name: Dr. M. Omair Ahmad

Phone: 514-848-2424 Ext.3075

E-mail: omair@ece.concordia.ca

Fax: 514-848-2802

Address: Department of Electrical & Computer Engineering
Concordia University
1455 de Maisonneuve Boulevard West
Montreal, Quebec H3G 1M8 CANADA

An Efficient Method of Cast Shadow Removal Using Multiple Features

Chu Tang, M. Omair Ahmad and Chunyan Wang

Abstract

Features of images are often used for cast shadow removal. A technique based on using only a single feature cannot universally distinguish an object pixel from a shadow pixel of a video frame. On the other hand, the use of multiple features increases the computational cost of a shadow removal technique considerably. In this paper, an efficient yet simple method for cast shadow removal from video sequences with static background using multiple features is developed. The basic idea of the proposed technique is that a simultaneous use of a small number of multiple features, if chosen judiciously, can reduce the similarity between object and shadow pixels without an excessive increase in the computational cost. Using the features of gray levels, color composition and gradients of foreground and background pixels, a method is devised to create a complete object mask. First, based on each of the three features, three individual shadow masks are constructed, from which three corresponding object masks are obtained through a simple subtraction operation. The object masks are then merged together to generate a single object mask. Each of the three shadow masks is created so as to cover as many shadow pixels as possible, even if it results in falsely including in them some of the object pixels. As a result, the subsequent object masks may lose some of these pixels. However, the object pixels missed by one of the object masks should be able to be recovered by at least one of the other two, since they are generated based on features complementary to the one used to construct the first one. The final object mask obtained through a logical OR operation of the three individual masks can, therefore, be expected to include most of the object pixels. The proposed method is applied to a number of video sequences. The simulation results demonstrate that the proposed method provides a mechanism for shadow removal that is superior to some of the recently proposed techniques without imparting an excessive computational cost.

Keywords: *Shadow removal, Object mask, Object segmentation*

I. Introduction

Segmentation of moving objects from a video sequence is essential in many vision-based applications, such as object recognition, video surveillance and security. For the purpose of segmenting a moving object, the frame of a video sequence containing an object can be regarded to consist of the moving object of interest, its cast shadow and the background. Segmenting a moving object is essentially a construction of an object mask that requires removal not only of the background but of the cast shadow. Segmentation of a moving object from a video sequence is a difficult task in view of the fact that cast shadow moves along with the object even if the background is static.

For a video captured with a fixed camera, Gaussian mixture model (GMM) has been conventionally used to build a background model and then used to generate a static background and a foreground that consists of the moving objects and its cast shadow [1]. In order to remove the cast shadow from the foreground, features such as gray levels, colors and textures have been used in the literature. The method in [2] uses only one feature, namely, the gray level, for the removal of cast shadow. However, if the gray levels of some of the pixels of the object and those of the corresponding background pixels are similar, then, in this method, these object pixels may also be regarded as part of the shadow. The method in [3] has used the fact that the color compositions of the cast shadow pixels are similar to that of the corresponding background pixels. However, some object pixels may be mistaken as shadow pixels when their color composition is similar to that of the background. Thus, it is difficult to distinguish between an object pixel and a shadow pixel by using a single feature when these two pixels are similar with respect to this feature. In order to overcome this problem, multiple features have been used for shadow removal [4]-[7]. In systems using multiple features, multiple modules, one of each feature, are employed. In [4] and [5], the shadow removal schemes have been developed, in which modules corresponding to the individual features operate in a sequential manner. A drawback of these schemes is that a useful information neglected by a single module could be completely lost or an unnecessary information could propagate through the entire sequential system. In [6] and [7], architectures for shadow removal have been proposed in which the individual modules operate in parallel. These parallel shadow removal systems provide a superior performance, since each module operating on the same original foreground image can more effectively produce results complementary to one another.

The methods of cast shadow removal using parallel architectures mentioned above are based on choosing a set of complementary features, constructing an object mask corresponding to each of the features chosen, and finally merging them into a single mask. The performance accuracy of these methods depends predominantly on the effectiveness of the techniques in utilizing the complementarity of the features used, and their complexity depends not only on the number of features employed, but also on the simplicity of schemes employed for creating the individual object masks and merging them into a single object mask. In this paper, a simple and efficient method for the removal of cast shadow from a segmented foreground of a frame of video sequences is developed by using the gray level, color composition and gradient features of the image pixels. Similar to other existing methods, the proposed scheme constructs individual masks corresponding to each of the features used. However, unlike existing methods, a greater emphasis is placed in the proposed method in maximizing the detection of shadow and object pixels in creating the respective masks. This is so done by maximizing the likelihood of including each shadow pixel in all of the shadow masks through the use of appropriate threshold parameters even if it does result in falsely including an object pixel in some of shadow masks. At the same time, it is ensured that the likelihood of getting the same object pixel to appear simultaneously in all the shadow masks is minimized by an efficient utilization of the complementarity of the features. The proposed method developed based on this strategy is shown to increase the performance of shadow removal as well as to reduce the complexity.

II. Proposed Method

The guiding principle in choosing an appropriate set of features is that an object pixel and a shadow pixel must not be similar to each other simultaneously with respect to all the features chosen. Thus, as the number of features chosen becomes increasingly large, it becomes increasingly less likely that an object pixel is found similar to a shadow pixel simultaneously with respect to all the features chosen. However, the use of a very large set of features would inevitably be computationally very expensive. Hence, the challenge is to employ a small number of features chosen judiciously, and still be able to distinguish an object pixel from a shadow pixel effectively. In the proposed method, we use a set of features consisting of gray levels, color composition and pixel gradients. With this choice for the set of features, an object pixel and a shadow could be very similar, for example, with respect to the gray level and color composition if their three RGB color components are similar. However, it is very unlikely that the two pixels are also similar with respect to the third feature, namely, the pixel gradients. Our objective is to develop a simple scheme for shadow removal using this set of features.

The proposed shadow removal scheme is depicted through the block diagram of Figure 1. In this scheme, three modules, each operating on the same foreground image and using one of the three features, is used to produce three binary shadow masks. The shadow masks are then used to create the corresponding object masks by a subtraction operation (an XOR operation) of the individual shadow masks from the foreground mask obtained through a binarization operation of the foreground image. Finally, the three object masks are merged by carrying out a logical OR operation to generate an overall object mask.

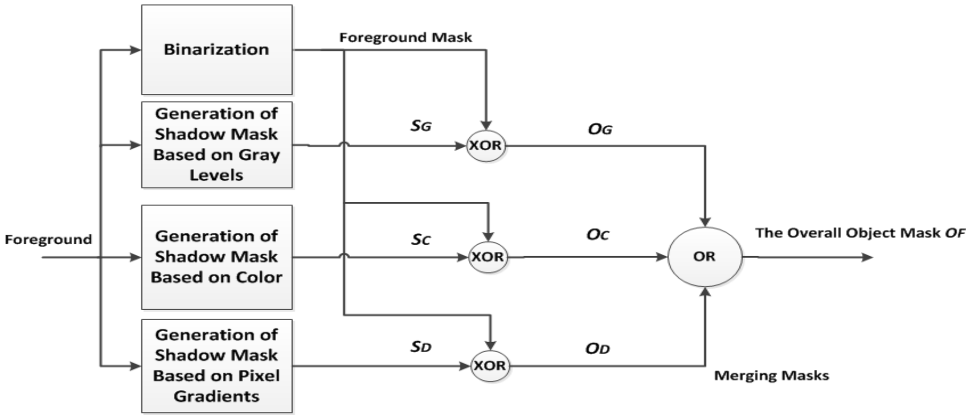


Figure 1 Scheme of the proposed method for shadow removal, where S_G, S_C and S_D denote the three shadow masks, and O_G, O_C and O_D denote the three subsequent object masks.

The objective of the proposed method is to generate an overall (final) object mask that captures all the object pixels and does not include any of the pixels belonging to the shadow. Since the final object mask O_F is obtained via a logical OR operation of the three intermediate object masks, none of the object masks, O_G, O_C and O_D, must, therefore, include a shadow pixel and an object pixel must be included in at least one of them. In order to achieve this goal, the three shadow masks, S_G, S_C and S_D, must meet the following two requirements:

(1) Each shadow mask must capture all the shadow pixels;

(2) An object pixel must not be mistakenly detected as a shadow pixel in all the three shadow masks simultaneously.

In view of the fact that three different features are used to create the three shadow masks, the condition (2) is less stringent in the sense that the second condition would likely be met more easily than the first one. Hence, we should make sure that each shadow mask captures as many shadow pixels as possible even if this is achieved at the expense of some of the object pixels being mistakenly detected as shadow pixels.

2.1. Detection of Shadow Pixels Based on Gray Levels

In the proposed method, the detection of shadow pixels using gray level information is based on the luminance enhanced method [2], which is adapted to capture as many shadow pixels as possible even at the expense of mistaking an object pixel to be a shadow pixel.

It is observed that the gray level difference between the background and shadows is generally smaller than that between the objects and the background. Adding a small positive constant δ to all the non-zero pixels in the foreground image of the tth frame $Y_F^{(t)}$, we have

$$Y_E^{(t)}(i,j) = Y_F^{(t)}(i,j) + \delta \tag{1}$$

If δ is appropriately chosen, the gray level difference between the background and shadow will become zero or get reduced, whereas, that between the background and the objects still remain significant. If $B^{(t)}$denotes the background image of the tth frame, the difference between the modified foreground and the background is given by

$$Y_D^{(t)}(i,j) = \left| Y_E^{(t)}(i,j) - B^{(t)}(i,j) \right| \tag{2}$$

A threshold $\varepsilon_G(i,j)$ is calculated based on $Y_D^{(t)}$ and $B^{(t)}$ as

$$\varepsilon_G(i,j) = \varepsilon_0 \bullet \left(1 + \frac{Y_D^{(t)}(i,j)}{(B^{(t)}(i,j)+1)} \right) \tag{3}$$

where ε_0 is a small positive constant. Then, a shadow mask is computed as

$$S_G^{(t)}(i,j) = \begin{cases} 1, \text{ if } Y_D^{(t)}(i,j) < \varepsilon_G(i,j) \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

The gray level difference between the background and shadow is generally not a constant, and varies from area to area. Thus, using a very small δ would generally not reduce the difference $Y_D^{(t)}$ given by (3) to zero. However, a large value δ may produce a $Y_E^{(t)}$ in which the gray level difference between some pixels of the objects and those of the background may be comparable to the gray level difference between the shadow and the background. In other words, $Y_D^{(t)}(i,j)$ at a pixel position in the object may be similar to that in parts of the shadow. Ideally, the value of δ should be made signal dependent, instead of keeping it constant. This, however, would increase the complexity of the proposed method. As mentioned previously, the emphasis in constructing the shadow mask is on a complete coverage of the shadow pixels, even if this is achieved at the risk of mistaking some of the object pixels as shadow pixels. The various steps of the gray-level-based detection process are given in Algorithm 1.

2.2. Detection of Shadow Pixels Based on Color Composition

The construction of the shadow mask $S_G^{(t)}$ in Section 2.1 is aimed at capturing all the shadow pixels; however, it may also include some of the object pixels having gray levels similar to that of the corresponding background pixel. Thus, the subsequent object mask $O_G^{(t)}$ resulting from this shadow mask will miss these pixels. The color feature of these pixels, however, may be different from that of the background pixels. Thus, in such a situation, we can make use of the color feature in the formation of the shadow mask without mistaking these object pixels as shadow pixels.

It is known that under normal illumination condition, the color composition of an individual pixel inside the main body of a shadow remains approximately the same as that of the corresponding pixel in the background. This characteristic of shadow formation has been referred to as color invariance property [3]. The ratio of the three color components, R, G and

B, of a pixel satisfying this property is the same as that of the corresponding background pixel. However, the pixels on or near the boundary of the shadow, in general, do not satisfy this property. Therefore, a shadow formed under normal illumination condition have two regions: the umbra region consisting of the majority of the pixels in the interior of the shadow that satisfy the color invariance property, and the penumbra region consisting of a minority of the border shadow pixels for which this property is not satisfied. Using the color information of the un-shadowed background, one can identify the umbra shadow pixels in the foreground. In the proposed method, we use the RGB color space instead of the HSV color space in order to simplify the computation.

Algorithm 1 Computation of a shadow mask based on pixel gray levels

```
Input: The background image $B^{(t)}$ and the foreground image $Y_F^{(t)}$. Set parameters $\delta$ and $\varepsilon_0$.

Output: A binary shadow mask $S_G^{(t)}$ for time $t$.

Begin:
At time $t$, get $B^{(t)}$ and $Y_F^{(t)}$ from the GMM algorithm.
if $Y_F^{(t)}(i,j) \neq 0$ then
    Compute the enhanced pixel value $Y_E^{(t)}(i,j)$ using (1).
    Compute the value difference $Y_D^{(t)}(i,j)$ using (2).
    Compute the modulated threshold $\varepsilon_G(i,j)$ using (3).
else
    ;
end if
Compute the value of $S_G^{(t)}(i,j)$ using (4).
```

Representing the three color components of the background (i.e., the scene without the object or shadow) by R_B, G_B and B_B, and those of the foreground of the tth frame by $R_F^{(t)}$, $G_F^{(t)}$ and $B_F^{(t)}$, we can calculate the following three ratios corresponding to each of the three color components.

$$R_R^{(t)}(i,j) = \frac{R_F^{(t)}(i,j)}{R_B(i,j)} \tag{5}$$

$$R_G^{(t)}(i,j) = \frac{G_F^{(t)}(i,j)}{G_B(i,j)} \tag{6}$$

$$R_B^{(t)}(i,j) = \frac{B_F^{(t)}(i,j)}{B_B(i,j)} \tag{7}$$

Since the three ratios for a pixel in an object vary considerably from one another than that in the case when the pixel is in the shadow, we can use the variance of the three ratios to distinguish an object pixel from the shadow pixel in the foreground. Denoting the variance of the three ratios corresponding to an (i, j)th pixel in the tth frame by $V^{(t)}(i, j)$, a shadow mask based on color feature can be constructed as

$$S_C^{(t)}(i,j) = \begin{cases} 1, \ if \ V^{(t)}\left(i,j\right) \leq \varepsilon_C \\ 0, \ otherwise \end{cases} \tag{8}$$

where ε_C is a pre-specified small positive constant. The object pixels having gray levels similar to the corresponding background pixels, and therefore, mistakenly included in $S_G^{(t)}$ will not be included in the shadow mask $S_C^{(t)}$ as long as they have color compositions different from the corresponding background pixels. The steps of the process of the color-composition-based detection of shadow pixels are formally given as Algorithm 2.

Algorithm 2 Computation of a shadow mask based on color composition

```
Input: The color background and the color foreground image. Set parameter $\varepsilon_C$.

Output: A binary shadow mask $S_C^{(t)}$ for time $t$.

Begin:
At time t, get the color foreground according to the gray level foreground produced by the GMM algorithm.
Represent the three color components of the background by $R_B$, $G_B$ and $B_B$ in RGB system.
Represent the three color components of the foreground by $R_F^{(t)}$, $G_F^{(t)}$ and $B_F^{(t)}$ in RGB system.

if $R_F^{(t)}(i,j) \neq 0$ or $G_F^{(t)}(i,j) \neq 0$ or $B_F^{(t)}(i,j) \neq 0$
    Compute the three ratios $R_R^{(t)}\left(i,j\right)$, $R_G^{(t)}\left(i,j\right)$ and $R_B^{(t)}\left(i,j\right)$ using (5), (6) and (7).
else
    ;
end if
Compute the variance $V^{(t)}(i,j)$ of $R_R^{(t)}\left(i,j\right)$, $R_G^{(t)}\left(i,j\right)$ and $R_B^{(t)}\left(i,j\right)$.

Compute the value of $S_C^{(t)}(i,j)$ using (8).
```

2.3. Detection of Shadow Pixels Based on Pixel Gradients

If an object pixel has both the color composition and gray level similar to those of the corresponding background pixel, this pixel will be included in both gray level and color-based shadow masks. Therefore, another technique for a shadow mask construction based on a feature that can handle this kind of situation more effectively needs to be developed. We now develop such a technique based on pixel gradient feature of the image.

It is known that the pixel gradient information of the object is usually very different from the background. It is assumed that there are more gray level variations in the object than that in the background. One can, therefore, use the gradient of the foreground image to detect shadow pixels. We use the Sobel operator to calculate a gradient. Then, the magnitude of the gradient $G^{(t)}$ is used to construct a shadow mask as

$$S_D^{(t)}(i,j)=\begin{cases}1, & if\ G^{(t)}(i,j)\le \varepsilon_D\\ 0, & otherwise\end{cases} \tag{9}$$

where ε_D is a pre-specified small positive constant. Algorithm 3 gives the steps of the process of the pixel-gradients-based detection of shadow pixels.

As discussed earlier, our objective in the construction of the shadow masks is to capture simultaneously in all of them as may shadow pixels as possible. The achievement of this goal can be facilitated by choosing the threshold parameters ε_0, ε_C and ε_D to have reasonably large values. However, a choice of large values for these parameters may lead to many of the object pixels to be mistakenly included in one or more of the shadow masks. This would be acceptable as long as the same object pixel is not included in the shadow masks S_G, S_C and S_D simultaneously. Note that an object pixel being included in all the shadow masks would happen in the case when gray level and color composition of such a pixel are the same as that of the corresponding background pixel as well as this object pixel has a low gradient value. Our proposed technique is, therefore, based on the assumption that such a situation would occur infrequently for most of the real images.

2.4. Creation of the Final Object Mask

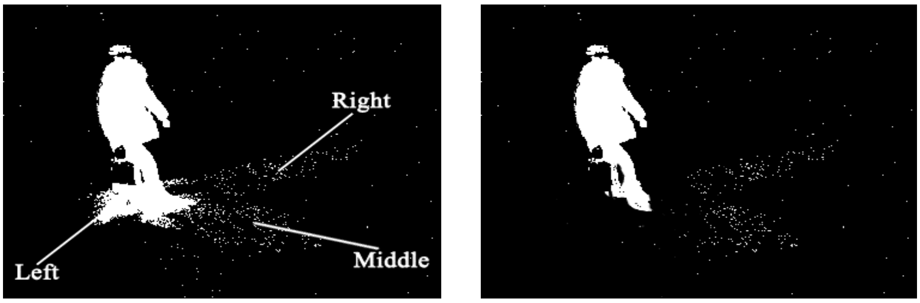
As explained earlier, each of the three shadow masks, obtained by using a distinct feature, is subtracted from the foreground mask in order to create a corresponding object mask. Recall that our objective is to include each object pixel in at least one of the three object masks, $O_G^{(t)}$, $O_C^{(t)}$ or $O_D^{(t)}$. Note that an object pixel has a greater likelihood to be captured by $O_D^{(t)}$ than by $O_G^{(t)}$ or $O_C^{(t)}$. The reason for this is as follows: Two pixels, one belonging to the object and the other to the shadow, are more contrasted from each other when their gradient features are used than in the case of using gray level or color features, since in the computation of the gradient of a pixel, its local neighborhood information, instead of the pixel information, is used.

As shown in Figure 1, the three object masks, O_G, O_C and O_D are combined using a logical OR operation in order to obtain a single object mask O_F. As stated earlier, our objective is not to miss out a shadow pixel in any of the three shadow masks, so that none of the object masks would have included in it a shadow pixel. Therefore, in order to include all the shadow pixels in each of the shadow masks, we propose to use reasonably large values for the threshold parameters ε_0, ε_C and ε_D. We now discuss the following two situations in which despite the choices of large values for ε_C and ε_D, the shadow masks S_C and S_D may still miss some of the shadow pixels, and therefore, subsequently, included in O_C and O_D:

(1) In the construction of the shadow mask based on color, we have used the color invariance property, that is, the color composition of a pixel is not affected by the shadow. It means that the three components of the color are modulated by a same positive constant. However, in some cast shadow areas, such as in penumbra regions, the three color components of a pixel are modulated differently because of different light sources forming the projection of the object. A choice of a small value for ε_C may be sufficient to detect most of the umbra pixels, whereas a relatively

large value of ε_C may not be adequate to detect some of the penumbra pixels. Such pixels may, therefore, be detected as object pixels and excluded from S_C, and consequently, get included in the subsequent object mask O_C. Figure 2 depicts an example of such a situation.

```
**Algorithm 3** Computation of a shadow mask based on pixel gradients

Input: The background image $B^{(t)}$ and the foreground image $Y_F^{(t)}$. Set parameter $\varepsilon_D$.

Output: A binary shadow mask $S_D^{(t)}$ for time t.

Begin:

At time t, get $B^{(t)}$ and $Y_F^{(t)}$ from the GMM algorithm.

if $Y_F^{(t)}(i,j) \neq 0$ then
    Compute the magnitude of the gradient $G^{(t)}(i,j)$ using SOBEL operator.
else
    ;
end if
Compute the value of $S_G^{(t)}(i,j)$ using (9).
```



(a) (b)

Figure 2 (a) Foreground mask of a frame of a video sequence. (b) Object mask resulting from the method described in Section 2.2.

Figure 2(a) shows the foreground mask of a frame in which there are shadows marked as “Left”, “Middle” and “Right”, resulting from the projections of the object from three different light sources. The left shadow is primarily formed by the dominant light source, whereas the middle and right shadows are formed by the other two. The left shadow can be considered as the one formed under normal illumination condition consisting of an interior umbra region and the boundary penumbra region. On the other hand, the other two shadows are formed, respectively, from the two distant light source situated on the left of the object and their pixel values are affected from the illumination of all the three light sources. The R, G and B components of a pixel in these shadows are modulated differently. Thus, such a pixel does not satisfy the color invariance property. As a result, these two shadows are overwhelmingly of the penumbra type. Figure 2(b) is the corresponding object mask resulting from the use of the method of forming shadow mask described in Section 2.2. It is seen that the interior pixels in the left shadow (umbra region) are almost completely removed, whereas a large number of small clusters of pixels in the penumbra regions of these shadows still remain in the object mask. These pixels could be removed if the value of the threshold parameter ε_C is further increased, but this would be done at the expense of losing a very large number of the object pixels from the object mask.

(2) The shadow mask S_D has been constructed using the pixel gradients. The construction of this mask is based on the assumption that the variations in gray levels of the object pixels are larger than that of the shadow pixels. However, in reality, some of the shadow pixels, especially the edge pixels, having relatively large gradients, would be regarded as object pixels and consequently excluded from the shadow mask S_D, included in the subsequent object mask O_D. Figure 3 illustrates an example of such a situation. Figure 3(a) is an original gray level frame of a sequence and Figure 3(b) is the corresponding foreground mask; Figure 3(c) shows the object mask resulting from the use of the method of forming a shadow mask described in Section 2.3. It is seen from Figure 3(c) that the most shadow pixels have been successfully removed, but there are some pixels with a structure of continuous or discontinuous thin lines that are, as a matter of fact, shadow pixels, which still remain in this object mask. A comparison of Figure 3(c) with Figure 3(a) shows that these line pixels correspond to the edges within the shadowed background or the boundary of the shadow itself, where gradient values of the pixels are usually quite large.


(a) (b) (c)

Figure 3 (a) Original frame. (b) Corresponding foreground mask. (c) Object mask resulting from the method described in Section 2.3.

It is clear that for above two cases, O_C and O_D would be affected in the sense that they will also have some shadow pixels considered as object pixels. Therefore, a direct OR operation on the pixels of O_G, O_C and O_D would result in a final object mask O_F including these shadow pixels. As discussed above, the shadow pixels appearing in O_D and O_C have, respectively, a structure of thin lines and isolated clusters of some small numbers of pixels. We, therefore, propose to make use of the morphological opening operation [8] in order to remove these artifacts from O_F. To apply the morphological opening operation, the merging operation on the three individual object masks in Figure 1 is modified as shown in Figure 4. Since O_C and O_D could be affected by the inclusion of the two types of shadow pixels, these two masks are first combined into a single object mask O_{CD} using a logical OR operation and then subjected to a morphological opening operation with a pre-specified coefficient to generate a subsequent object mask O_{CDM}. This mask is finally combined with O_G using a logical OR operation to obtain the final object mask O_F.

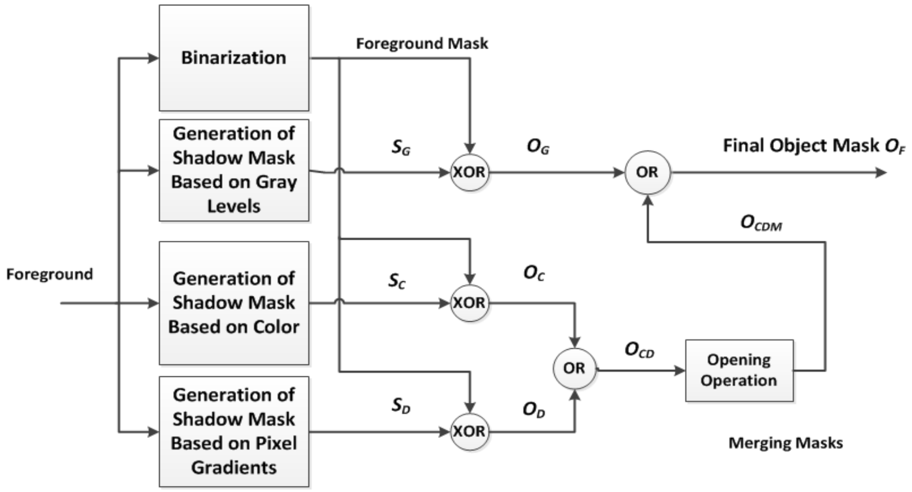


Figure 4 Modified scheme of the proposed method.

III. Performance Evaluation

The proposed method is applied to remove shadow in a number of video sequences to assess its performance and to compare the results with those obtained by applying the methods in [2] and [7]. Figure 5 illustrates the results of applying the proposed method and those given in [2] and [7] on the video sequence *S3_T7_A* from the PETS database [9]. This sequence contains 2370 frames. Figures 5(a) and (b) show the 147th and 291st original frames of this sequence and the corresponding foreground masks. Figure 5(c) shows the ground truth mask for the object. Figures 5(d), (e) and (f) are the corresponding object masks resulting from the methods of [2] and [7] and that using the proposed method, respectively. A comparison of the masks in Figures 5(c) and (d) shows that the method of [2] misses many of the object pixels that have gray levels similar to that of the background. Similarly, as seen from Figures 5(c) and (e), a number of object pixels are missed by using the method of [7] due to their color or texture being similar to the corresponding background pixels. A comparison of the masks in Figures 5(c)-(f) shows that the proposed method is the best among all the three methods considered in capturing object pixels and in removing the shadow pixels.

Figure 6 illustrates the results of applying the proposed method and those given in [2] and [7] on two video sequences *Intelligentroom* [10] and *Hall_Monitor* [11]. Figure 6(a) shows the 300th and 41st original frames of the two video sequences, respectively. Figures 6(b) and (c) show the corresponding foreground masks and ground truth masks for the objects, whereas Figures 6(d), (e) and (f) show the corresponding object masks resulting from the methods of [2] and [7] and that using the proposed method. It is clearly seen that, among the three methods, the proposed one gives the most complete object masks.

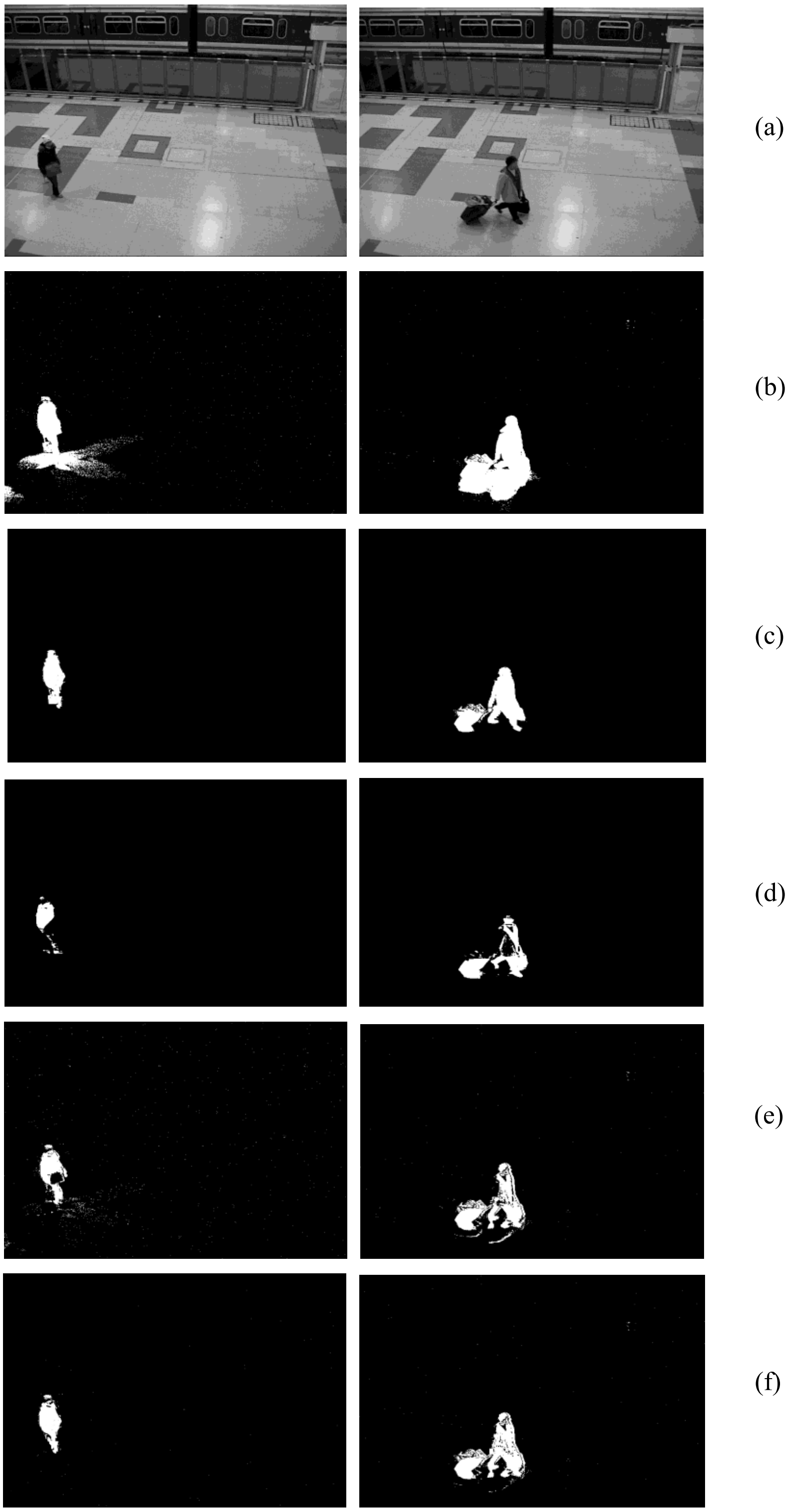


Figure 5 (a) Original 147th frame (left) containing Object 1 and 291st frame (right) containing Object 2 of sequence *S3_T7_A* from the PETS database. (b) Foreground masks generated using GMM. (c) Ground truth masks for the objects. (d) Object masks obtained by applying method [2]. (e) Object masks obtained by applying method [7]. (f) Object masks obtained by applying the proposed method.

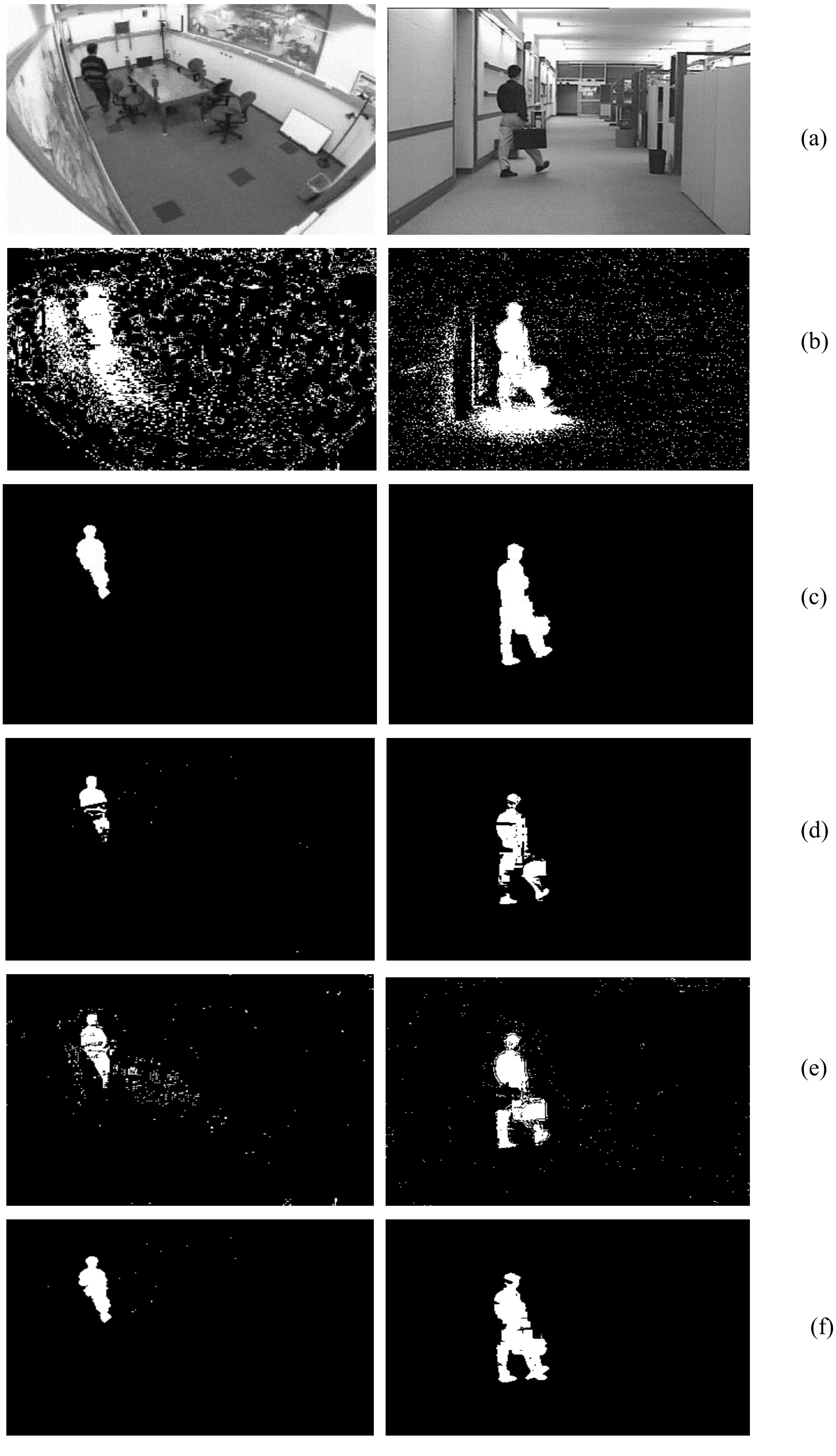


Figure 6 (a) Original 300th frame (left) of sequence *Intelligentroom* and 41st frame (right) of sequence *Hall_Monitor*. (b) Foreground masks generated using GMM. (c) Ground truth masks for the objects. (d) Object masks obtained by applying method [2]. (e) Object masks obtained by applying method [7]. (f) Object masks obtained by applying the proposed method.

In order to provide a quantitative performance evaluation of the proposed and the other two methods considered in this paper, we use the false alarm rate (FAR) and tracker detection rate (TRDR) [12] defined below as performance metrics.

$$FAR = \frac{FP}{TP + FP} \tag{10}$$

$$TRDR = \frac{TP}{TP + FN} \tag{11}$$

where TP is true positive, FP false positive, FN false negative. Table 1 gives the quantitative results of these metrics for the three methods applied to the video sequence frames chosen for the illustration of visual performance. From this table, it is seen that the proposed method, when applied to frame 291 of the *S3_T7_A* sequence, gives a slightly higher FAR value than those obtained by using [2] and [7], since, as seen from the right image in Figure 5(f), there are some shadow pixels in the object mask that have been falsely detected as object pixels. However, for the other three frames used for performance evaluation, the proposed method gives the lowest FAR values, indicating that it misclassifies only a small number of shadow pixels as object pixels. Also, the proposed method gives the highest TRDR values among the three methods for all the four frames used for evaluation, which shows that the method proposed provides a more accurate detection of the object pixels among the three methods.

Table 1 Results of performance evaluation

Sequence Frame	Method [2]		Method [7]		Proposed Method	
	FAR	TRDR	FAR	TRDR	FAR	TRDR
147th frame of *S3_T7_A* sequence of database PETS	0.289	0.385	0.388	0.719	0.196	0.868
291st frame of *S3_T7_A* sequence of database PETS	0.169	0.625	0.137	0.769	0.217	0.876
300th frame of *Intelligentroom* sequence	0.183	0.707	0.354	0.846	0.112	0.911
41st frame of *Hall_Monitor* sequence	0.102	0.795	0.197	0.803	0.098	0.889

The computation times of the proposed method and those of [2] and [7] are obtained by applying these methods to the *S3_T7_A* sequence from the PETS database as well as to the *Intelligentroom* and *Hall_Monitor* sequences using a Windows platform based PC with a 2.83 GHz Intel Core Quad CPU and 8 GB RAM using MATLAB codes. The results are shown in Table 2. It is seen from this table that the proposed method on average takes about 29% more times than the times taken by method of [7]. Thus, the proposed method provides a shadow removal performance superior to that of [7] without an excessive computational cost. It is to be noted that even though the method [2] takes much lower computation time, its performance is much lower in comparison to that of the other two.

Table 2 Average computation time per frame (in seconds)

Sequence	Frame Size	Method [2]	Method [7]	Proposed Method
S3_T7_A sequence containing Object 1 [9]	720×576	0.82	1.91	2.49
S3_T7_A sequence containing Object 2 [9]	720×576	0.88	1.99	2.58
Intelligentroom [10]	320×240	0.17	0.48	0.61
Hall_Monitor [11]	352×288	0.25	0.68	0.86
Average computation time		0.53	1.27	1.64

IV. Conclusion

Use of multiple features is essential for removing cast shadows in frames of video sequences, since a single feature cannot effectively distinguish an object pixel from a shadow pixel. However, the complexity of a cast shadow removal technique relies heavily on the number and the nature of the features used. In this paper, a low-complexity scheme for cast shadow removal has been developed by effectively utilizing the complementary nature of gray level, color composition and gradient features of the pixels of a video frame. Each of the features is used to construct a respective shadow mask corresponding to the same video frame. By devising a very simple scheme, the individual shadow masks are used to obtain the corresponding object masks that are then finally merged into a single object mask. The main idea in developing the proposed scheme for providing an accurate object mask has been in minimizing the likelihood of missing a shadow pixel in each of the shadow masks even if it does result in falsely detecting some of the object pixels as shadow pixels but at the same time, ensuring that the same object pixels do not get falsely included in all the shadow masks. Challenges arising from the formation of cast shadows from the use of multiple illumination sources or from the similarity of the gradient feature of an object pixel and that of a pixel at the edge of the shadow have also been discussed and a simple scheme using morphological operation has been proposed to overcome these problems. The proposed scheme and two other recently reported schemes for cast shadow removal have been applied to the frames of three different video sequences. Subjective and objective results obtained through these experiments have demonstrated the effectiveness and superiority of the proposed scheme at a reasonable computation cost.

References

[1] C. Stauffer and W.E.L. Grimson,“Adaptive background mixture modules for real-time tracking”, in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, vol.2, Jun. 1999, pp. 246-252.

[2] C.-T. Chen, C.-Y. Su and W.-C. Kao,“An enhanced segmentation on vision-based shadow removal for vehicle detection”, in *Proc. Int. Conf. Green Circuits and Systems*, Jun. 2010, pp. 679 - 682.

[3] E. Salvador, A. Cavallaro and T. Ebrahimi,“Cast shadow segmentation using invariant color features”, *Comput.Vision and Image Understand.*, vol. 95, no.2, pp. 238-259, Jun.2004.

[4] K. Nakagami and T. Nishitani,“The study on shadow removal on transform domain GMM foreground segmentation”, in *Proc. Int. Symp. Communications Information Technol.*, Oct.2010, pp. 867-872.

[5] Y.-F. Su and H.H. Chen, “A three-Stage approach to shadow field estimation from partial boundary information”, *IEEE Trans. Image Process.*, vol.19, no.10, pp.2749-2760, Oct. 2010.

[6] M.-T. Yang, K.-H. Lo, C.-C Chiang and W.-K Tai, “Moving cast shadow detection by exploiting multiple cues”, *Image Process.IET,* vol. 2, pp. 95 -104, Apr. 2008.

[7] C. Wang and W. Zhang, “A robust algorithm for shadow removal of foreground detection in video surveillance”, in in *Proc. Asia-Pacific Conf. Information Process.*, vol.2, Jul. 2009, pp.422-425.

[8]D.Vernon.(1999).*MachineVision*.[Online].Available:http://homepages.inf.ed.ac.uk/rbf/BOOKS/VERNON/vernon.htm [Oct. 11, 2012].

[9] http://www.cvg.rdg.ac.uk/PETS2006/data.html

[10] http://www.openvisor.org/video_details.asp?idvideo=114

[11] http://trace.eas.asu.edu/yuv/

[12] J.Black, T. Ellis and P. Rosin, “A novel method for video tracking performance evaluation”, in *Proc. Int. Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Oct. 2003, pp. 125-132.

[13] E. Arbel and H.Hel-Or, “Shadow removal using intensity surfaces and texture anchor points”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no.6 , pp. 1202-1216, Jun. 2011.

[14] W.T. Wintringham, Bell Telephone Laboratories: *Colorimetry and color television* [M]. Inc., Murray Hill, N.J. Current version: 08, Jan. 2007.