Accurate Silhouette Extraction of a Person in Video Data by Shadow Evaluation

Zhengping Wang, Bok-Suk Shin, and Reinhard Klette

***Abstract*—This paper presents a new shadow detecting method for silhouette extraction of a person in gray-level video sequences. We use a shadow evaluator to verify each raw shadow pixel that was detected by Gaussian distribution analysis. The evaluator considers a raw shadow pixel initially to be a *fake shadow pixel*, and marks it as a *silhouette pixel* if it is enclosed or semi-enclosed by *moving occlusion boundaries* of a person. Those were extracted by subtracting edges in the current frame from edges of the background. We also propose a silhouette compensation technique to recover some missing (i.e. removed) silhouette pixels by using a similarity criterion between silhouette pixels and their neighbors. Experimental results show us that the proposed algorithm keeps a silhouette of a person more accurate compared to other methods. Methods advocated by other researchers in YUV or RGB color space, typically remove silhouette pixels as shadow if the color of these pixels is similar to that of the surrounding background.**

***Index Terms*—Shadow removal, silhouette detection, shadow evaluator, silhouette compensation.**

I. Introduction

Shadows cause serious problems for silhouette extraction of a person in video sequences. The *silhouette* is an image region defined by the circumscribing *occlusion boundary*. Accurate detection and the elimination of moving cast shadows is still a challenge besides numerous efforts in this area with substantial achievements. Some of the proposed methods try to detect cast shadows in an input video sequence by using color information (such as in RGB, YUV, or HSL color space) while others focus on methods depending on gray-level inputs only.

A real-time shadow detection and elimination method, based on the YUV-color space, has been proposed in [1]. In this space, the saturation of a pixel value $\mathbf{u} = (Y, U, V)$ is interpreted as being the Euclidean distance of point $\mathbf{u}$ to the origin $\mathbf{o}$, and the hue is interpreted as being the angle between the segment $\mathbf{ou}$ and the $\mathbf{U}$-axis. Shadows are then detected based on saturation hue.

A method based on the RGB color space is proposed in [2], and is further discussed in [3], [4]. Here, mean μ and standard deviation σ of pixel values are calculated, and then brightness distortion and chromaticity distortion of each RGB pixel value based on μ and σ. Those pixels with large brightness distortion and small chromaticity distortion are regarded as shadow. A very similar method is proposed in [5] also exploiting brightness and color distortion.

These color-based methods follow the basic observation that shadow pixels have high brightness distortion and a low variance in color or chromaticity. In conclusion, real silhouette pixels also falsely detected as being shadow if their color value is close to that of the background. Unfortunately, this situation appears often for moving objects in real-world scenarios.

Paper [6] advocated an algorithm based on value differences between foreground and background for detecting shadow in gray-level video sequences. In this algorithm, candidate shadow pixels are detected by normalized cross-correlation (NCC) [7], then local statistics of ratios of pixel values are used to refine the detected shadow area. The disadvantage of this algorithm is that incorrect detection happens when some foreground objects or background regions are homogeneously textured.

In [8], a shadow-removal method is proposed which is based on object contour information. First, foreground is separated from background by a Gaussian mixture model (GMM), and the gradient-based background subtraction method of [9] is the employed for the detection of contours in the GMM-detected foreground. Finally, shadow is marked as a region of pixels in the exterior of object contours using the assumption that there are no contours in shadow areas. This technique is suitable for many scenes but cannot detect shadows that are cast on floors or walls that have “complicated textures”.

Some proposed shadow detection systems use models for relationships between light sources and objects to retrieve shadows; for example, see [10]-[12]. These methods depend on good estimates for shape and position of light source, moving objects, and background; the methods also require knowledge about the intensity of the light source, the ambient illumination, and the camera geometry. The application of these algorithms is limited because such information cannot be easily acquired in most scenes.

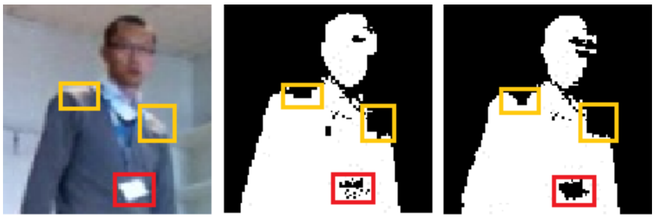

Fig. 1. Shadow removal by methods based on color space. *Left*: Input frame. *Middle*: Detected silhouette using the YUV method [1]. *Right*: Detected silhouette using the RGB method [2].

When testing shadow-removal methods based on color-space models for extracting accurate silhouettes of a person in video sequences, we found out that it is difficult for these methods to distinguish shadow pixels from silhouette

Manuscript received December 9, 2013; Revised March 14, 2014.

The authors were with the University of Auckland, New Zealand (e-mail: wzhengping@gmail.com).

DOI: 10.7763/IJCTE.2014.V6.913

pixels if silhouette pixels have a similar color as the background. This is not surprising, because these methods detect shadow by assuming color differences between foreground and background pixels. Fig. 1 shows cases where methods based on YUV or RGB color space misclassify shoulder pixels, but label pixels in the name tag as shadow. The nametag is white, and the color of the shoulder drifts to white because of lighting. In both cases, the silhouette color is similar to the background, and this "confused" those shadow removal methods; by not identifying true shadow correctly they cause defects in the resulting silhouettes.

In order to resolve such problems, this paper proposes a shadow-detection and removal technique that supports accurate extractions of silhouettes of a person in gray-level video sequences. The proposed method detects true shadow by using image intensity distributions and moving occlusion boundaries of a person. Our experimental results show that the proposed method can distinguish true shadow near silhouettes successfully even if silhouette pixels have a similar color (only shown as gray-level in our videos) as the background.

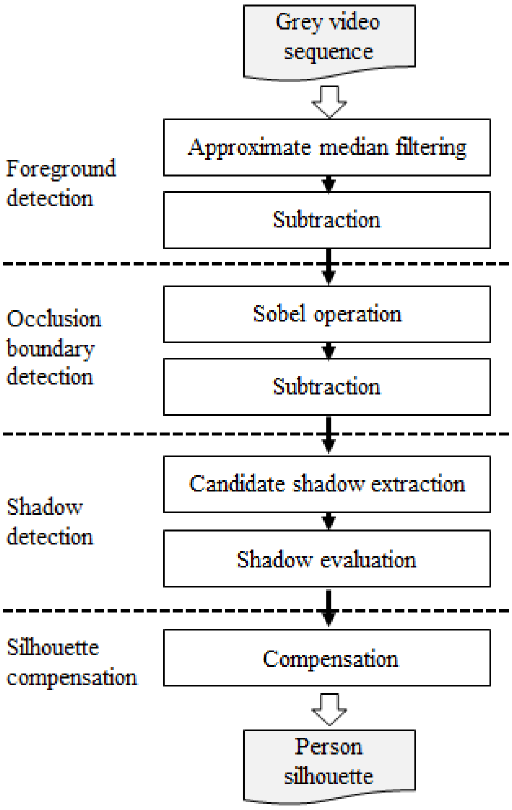

Fig. 2. Sketch of the proposed algorithm.

Fig. 2 outlines the basic architecture of the proposed algorithm. The input is gray-level video sequences. First, foreground is detected by subtracting the current frame from the background which is modeled by the *approximate median filter*, see [13]-[16]. Then, background edges are detected using the Sobel operator on the modeled background. The Sobel operator is also used to extract raw occlusion boundaries of a person in the detected foreground image. Then, the "true occlusion boundaries" of a person are extracted by subtracting the "raw occlusion boundaries" of a person from the background edges. After that the system exploits the Gaussian distribution to extract candidate shadows in the foreground. At this stage, some pixels in the silhouette of a person might also be falsely marked as being shadow. For each candidate shadow pixel, the system uses a *shadow evaluator* for evaluating those pixels again that had been falsely labeled as silhouette pixels before. Here we detect pixels on occlusion boundaries of a person by adjacent features of those pixels. Finally, the system defines and processes a *compensation window* for having a silhouette of a person more completely (more accurately) detected.

The paper is structured as follows. Section II describes the YUV and RGB methods, which were advocated by other researchers, and which are employed in our research for comparison. Section III details the proposed algorithm that was already briefly outlined above. Experimental results and a comparative analysis are given in Section IV. Section V concludes.

II. Color-Based Shadow Detection Methods

We recall here two previously defined methods for later use in our comparative experimental analysis.

A. YUV Method

We employ the YUV shadow detecting method of [1]. In this method, at first only foreground pixels are checked for identical hue (i.e. supposed to be shadow). The luminance difference $\Delta Y = Y_{image} - Y_{background}$ is limited by an upper threshold Y_{max} to avoid positive difference values because a shadow always reduces the image intensity Y_{image}. The difference ΔY must be negative for object assignment. Afterwards, by comparing U and V an appropriate arctan-approximation is chosen. Here, U or V are also limited by a second threshold ε to control an intensity shift caused by a shadow. Finally, $\Delta\alpha$ is chosen as a threshold for hue. For more details, see [1].

B. RGB Method

The RGB shadow detecting method of [2] is our second algorithm for comparative experiments. For each pixel, a brightness distortion α_i is defined as follows:

$$\alpha_i = \frac{\dfrac{I_R(i)\mu_R(i)}{\sigma_R^2(i)} + \dfrac{I_G(i)\mu_G(i)}{\sigma_G^2(i)} + \dfrac{I_B(i)\mu_B(i)}{\sigma_B^2(i)}}{\left(\dfrac{\mu_R(i)}{\sigma_R(i)}\right)^2 + \left(\dfrac{\mu_G(i)}{\sigma_G(i)}\right)^2 + \left(\dfrac{\mu_B(i)}{\sigma_B(i)}\right)^2} \quad (1)$$

A *chromaticity distortion* D_i is defined as follows:

$$D_i = \left(\left(\frac{I_R(i) - \alpha_i \mu_R(i)}{\sigma_R(i)}\right)^2 + \left(\frac{I_G(i) - \alpha_i \mu_G(i)}{\sigma_G(i)}\right)^2 + \left(\frac{I_B(i) - \alpha_i \mu_B(i)}{\sigma_B(i)}\right)^2 \right)^{1/2} \quad (2)$$

where $I_R(i)$, $I_G(i)$, and $I_B(i)$ represent the values of the red, green, and blue channel at the i^{th} pixel in RGB color space of the currently observed color vector. Furthermore, $\mu_R(i)$, $\mu_G(i)$, and $\mu_B(i)$, and $\sigma_R(i)$, $\sigma_G(i)$, and $\sigma_B(i)$ are means and standard deviations of these color components, respectively.

Shadow is detected by the following decision rule:

$$S(x,y,t)=\begin{cases}1 & \text{if } \alpha_i>\gamma \quad and \quad D_i<\beta \\ 0 & \text{others}\end{cases} \tag{3}$$

where γ and β are thresholds for brightness distortion and chromaticity distortion, respectively. Pixel *i* is a shadow pixel if *S*(*x*, *y*, *t*) = 1, and it is a silhouette pixel if *S*(*x*, *y*, *t*) = 0.

III. Proposed Method

We specify our novel technique previously already briefly outlined in Fig. 2.

A. Foreground Detection

We use the *approximate median filter* as proposed in [13]. It has been applied to a wide range of background subtraction scenarios since its initial description.

Let *I* be the mapping of all input pixels, and *B* of the background pixels. For detected background pixels, we use Equ. (4) for a background update:

$$B(x,y,t)=\begin{cases}B(x,y,t-1)++ \\ \qquad \text{if } I(x,y,t)>B(x,y,t-1) \\ B(x,y,t-1)-- \\ \qquad \text{if } I(x,y,t)<B(x,y,t-1)\end{cases} \tag{4}$$

Only those pixels are updated which were detected as being background. The initial value of *B*(*x*, *y*, 0) is *I*(*x*, *y*, 0).

For the modeled background, we employ then a simple subtraction approach for foreground extraction. A foreground mapping *F* is defined as follows:

$$F(x,y,t)=\begin{cases}1 & \text{if } |I(x,y,t)-B(x,y,t-1)|>\sigma_t \\ 0 & \text{otherwise}\end{cases} \tag{5}$$

where the initial value of *F*(*x*, *y*, 0) is 0, that means all the pixels are regarded as background at the beginning. A point at (*x*, *y*) in a frame *t* is detected as a foreground when the difference between its current value *I*(*x*, *y*, *t*) and background value *B*(*x*, *y*, *t* - 1) is bigger than a threshold σ_t, which is the standard deviation of intensity of input frame *t*. For completeness, we also provide the definitions

$$\sigma_t=\sqrt{\left(\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}(I(x,y,t)-\mu_t)^2\right)/n} \tag{6}$$

$$\mu_t=\left(\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}I(x,y,t)\right)/n \tag{7}$$

In Eqs. (6) and (7), *W* and *H* are width and height of the input frame, respectively, and *n* = *W* × *H* is the number of pixels in the input frame.

B. Detection of Occlusion Boundary of a Person

We use the Sobel operator [17], [18] for collecting edge information. This information is used to evaluate candidate shadows in a shadow detection step. The Sobel operator is applied to the first frame of a given input video sequence for defining background edges $E_{background}$ with a low threshold α in order to extract as many background edges as possible, while it is also applied to each foreground image to extract the current initial occlusion boundaries E_{ini} of a person by applying a high threshold α; see Equ. (8).

Let *E* be the edge map, E_H the horizontal Sobel edge convolution result, and E_V the vertical Sobel edge convolution result. The process equations are as follows:

$$E(x,y,t)=\begin{cases}1 & \text{if } |E_H(x,y,t)|+|E_V(x,y,t)| \\ & \qquad >\alpha\sigma_t/2 \\ 0 & \text{otherwise}\end{cases} \tag{8}$$

We use the standard deviation σ_t as defined in Equ. (6), and α is a control parameter. In our experiments for background edge detection, α was set to be 2, and it was set to be 4 for initial detection of an occlusion-boundary of a person.

True occlusion-boundaries E_{true} of a person are detected by subtracting initial occlusion boundaries of a person from background edges as follows:

$$E_{true}(x,y,t)=\begin{cases}1 & \text{if} \\ & E_{ini}(x,y,t)-E_{background}(x,y,t)>0 \\ 0 & \text{otherwise}\end{cases} \tag{9}$$

C. Shadow Detection

In order to detect shadow as accurate as possible, we use two steps for extracting shadow. First, the Gaussian distribution is exploited to detect candidate shadow pixels. Then, we use a shadow evaluator to verify each candidate shadow or extracting the *true shadow*.

1) Candidate shadow detection

In the foreground detection process, the silhouette of a person and some shadow pixels are extracted as foreground successfully. We suppose that the distribution of intensities of all pixels, which belong to the detected foreground, forms a Gaussian distribution. Values of silhouette pixels of a person are close to the peak of this Gauss distribution. Shadow pixel values are expected to be far away from the mean of the Gauss distribution. These assumptions are confirmed by the example shown in Fig. 3.

Fig. 3 shows a current input gray-level frame, the detected foreground by a foreground detection process, the intensity distribution (i.e. the gray-level histogram) of all foreground pixels, the detected silhouette of a person and the intensity distribution of all pixels within this silhouette, the detected shadow and the intensity distribution of all shadow pixels. The intensity distributions of silhouette or shadow pixels are similar to Gaussian distributions. Most of the shadow pixel intensity is larger than the shifted mean $\mu+\sigma$. Thus, we use parameters $\mu-\sigma$ and $\mu+\sigma$ for extracting most of the shadow pixels. At the same time some of the silhouette pixels are also falsely extracted as being shadow. These misclassified pixels will be again relabeled as silhouette pixels in the next shadow evaluation step.

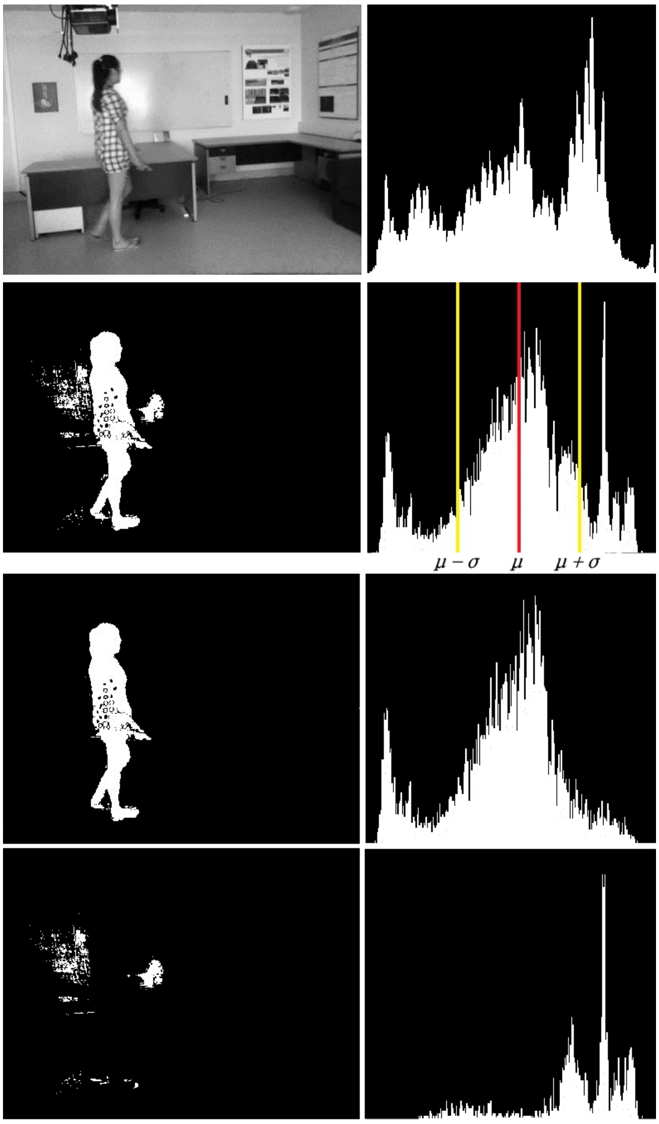

Fig. 3. Foreground, silhouette, and shadow intensity distributions; diagrams are differently scaled in vertical direction. *Top*: Current input frame and gray-level histogram (intensity distribution). *Second row*: Detected foreground and intensity distribution within this foreground. *Third row*: Silhouette and intensity distribution within the silhouette. *Bottom*: Shadows and intensity distribution within shadow regions.

Based on the above formulated assumptions and analysis results, a *candidate shadow* (CS) is detected at pixels which have a larger difference to the intensity mean μ of the detected foreground, formally expressed as follows:

$$S(x,y,t)=\begin{cases}1 & \text{if } F(x,y,t)=1 \\ & \quad \text{and } |I(x,y,t)-\mu|>\sigma \\ 0 & \text{otherwise}\end{cases} \tag{10}$$

where μ is the mean intensity of all foreground pixels which were detected at the first stage, and σ is the standard deviation. Both are calculated following Eqs. (11) and (12), respectively:

$$\mu=\left(\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}F(x,y,t)\times I(x,y,t)\right)/m \tag{11}$$

$$\sigma=\sqrt{\left(\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}F(x,y,t)\times(I(x,y,t)-\mu)^2\right)/m} \tag{12}$$

where *m* is the number of foreground pixels in frame *t*; this number is simply calculated as follows:

$$m=\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}F(x,y,t) \tag{13}$$

Let *I*(*x*, *y*, *t*) be again the pixel value at point (*x*, *y*) in input frame t. Foreground *F*(*x*, *y*, *t*) is calculated by Equ. (5). The pixel is a CS pixel if *S*(*x*, *y*, *t*) equals 1, otherwise the pixel is invalid for CS.

A Gaussian distribution model is used for the extraction of CS. While detecting shadow, some silhouette pixels can be incorrectly labeled as being shadow. For improving such incorrect assignments of shadow pixels and for extracting all valid shadow pixels, we use standard deviation σ as threshold. The following shadow evaluation process corrects misclassified pixels.

2) *Shadow evaluation*

The shadow evaluation step makes use of the detected occlusion boundary E_{true} of a person. We use the rule that a true shadow pixel should be a pixel which is not enclosed or semi-enclosed by pixels on an occlusion boundary of a person. We use this rule to verify each candidate shadow pixel. The motivation is that we assume shadow to be outside of the occlusion boundary of a person. We designed a *shadow evaluator* to figure out the enclosure situation of each candidate shadow pixel.

For each candidate shadow pixel at (*x*, *y*, *t*), the evaluator tries to detect its *enclosure index C*(*x*, *y*, *t*). Value *C*(*x*, *y*, *t*) represents the number of directions in which there are occlusion boundary pixels, and the distance of those from (*x*, *y*, *t*). The distance to the first met occlusion-boundary pixel is evaluated whether it is smaller than a given threshold T_t. Our shadow evaluator tests the usual eight directions of a pixel at (*x*, *y*, *t*) to obtain *C*(*x*, *y*, *t*); see Fig. 4. If *C*(*x*, *y*, *t*) is larger than four, then the candidate shadow pixel is considered to be a falsely classified shadow pixel, and it is relabeled as being a silhouette pixel.

In Fig. 4, *P*1, *P*2 and P3 are evaluated as being true shadow by our shadow evaluator because *C*(*P*1) is 4, *C*(*P*2) is 3, and *C*(*P*3) is 2. *N*1 to *N*9 are evaluated as being fake shadow pixels as *C*(*N*1), *C*(*N*2), and *C*(*N*3) are all 5, *C*(*N*4), *C*(*N*5), and *C*(*N*6) are all 6, *C*(*N*7) and *C*(*N*8) are 7, and *C*(*N*9) is 8. These values are all larger than 4.

The threshold T_t for the enclosure index, for analyzing frame *t*, is determined by the maximum value of foreground height or width. The calculation can use horizontal and vertical projection histograms of the detected foreground; see Fig. 5 for an example. Threshold T_t is then defined as follows:

$$T_t=\max\{h_t(c),\, w_t(r)\} \tag{14}$$

$$h_t(c)=\sum_{y=0}^{H-1}F(c,y,t) \tag{15}$$

$$w_t(r)=\sum_{x=0}^{W-1}F(x,r,t) \tag{16}$$

where *F*(*x*, *y*, *t*) is defined as in Equ. (5).

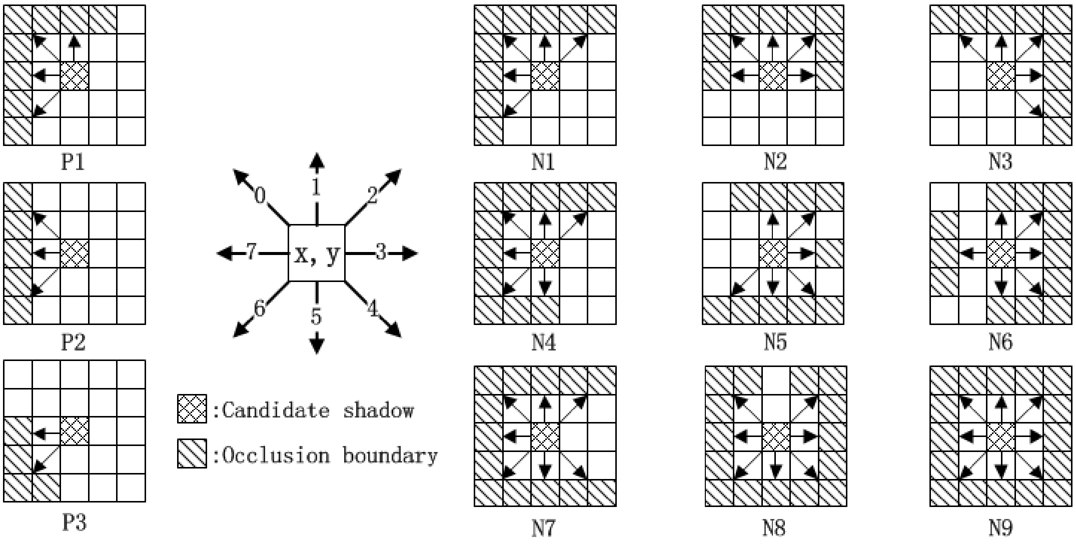

Fig. 4. Shadow evaluator. *P*1, *P*2 and *P*3 are true shadow pixels, and *N*1 to *N*9 are classified as being false shadow pixels.

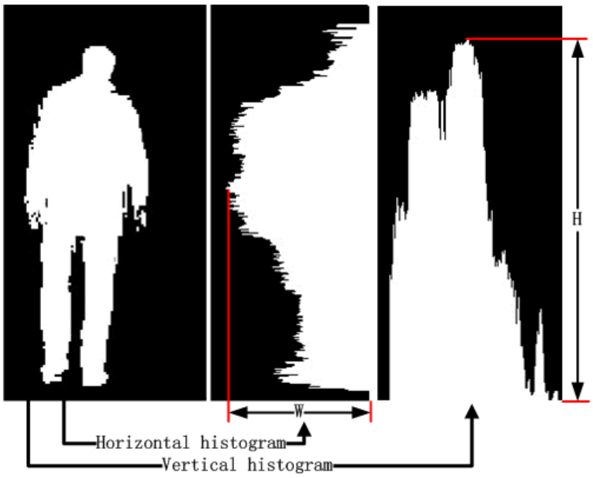

Fig. 5. Extracted foreground and calculation of height and width based on projection histograms.

D. Silhouette Compensation

Some silhouette pixels are not detected as foreground correctly in the background subtraction stage because their intensity is very similar to that of the background. In order to have these misclassified pixels "back", a silhouette compensation algorithm is used in our system. It is based on the similarity between background pixels and their foreground neighbours.

Let p_x be a silhouette pixel, p_y be an adjacent background pixel of p_x, p_z is an adjacent pixel of p_y in an $n \times n$ neighborhood window (NW), with p_y as reference pixel at the centre of this window. An adjacent background pixel p_y should also be a silhouette pixel if p_y has more than $(2n^2) / 5$ neighbors which are silhouette pixels in its $n \times n$ neighborhood window, and the intensity difference between values at p_x and p_y is reasonably small.

Our procedure for silhouette compensation is specified by the following equations:

$$f(p_y) = \begin{cases} 1 & if \quad d(p_x, p_y) < \alpha \\ & \quad and \quad g(p_y) > (2n^2)/5 \\ 0 & else \end{cases} \tag{17}$$

$$d(p_x, p_y) = |I(p_x) - I(p_y)| \tag{18}$$

$$g(p_y) = \sum_{p_z \in NW} h(p_z) \tag{19}$$

$$h(p_z) = \begin{cases} 1 & if \ p_z \text{ is foreground pixel} \\ 0 & if \ p_z \text{ is background pixel} \end{cases} \tag{20}$$

where α is an experimentally optimized threshold, and $I(p_x)$ is the gray value at pixel p_x. Pixel p_y is a silhouette pixel if $f(p_y)$ is equal to 1. The condition $d(p_x, p_y) < \alpha$ is used to ensure that only pixels which are similar to already known adjacent foreground pixels are relabeled as being silhouette pixels. The restriction $g(p_y) > (2n^2) / 5$ is employed to control that no further background pixels are relabeled as being foreground pixels again.

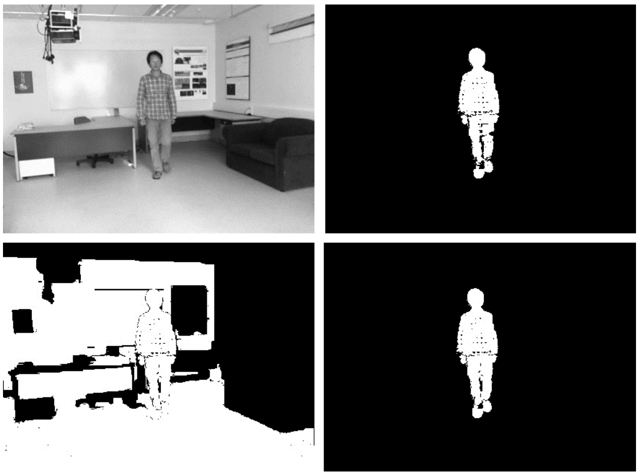

Fig. 6. Silhouette compensation. *Top left*: Current frame. *Top right*: Silhouette before compensation. *Bottom left*: Compensated silhouette without restriction $g(p_y) > (2n^2) / 5$. *Bottom right*: Compensated silhouette with restriction $g(p_y) > (2n^2) / 5$.

Fig. 6 illustrates the process of silhouette compensation also for the case when we are not using the restriction $g(p_y) > (2n^2) / 5$. In this case, background pixels will be labeled as being foreground also when they have similar intensity values with silhouette pixels.

IV. Experimental Results

We apply our method to gray-level video sequences. Experiments have been performed on several videos, each 3 to 4 minutes of length. We detail results below that are representative for the performed experiments.

For comparison with the two described color-model based methods, we record the sequences in color mode. We demonstrate results for recorded indoor sequences that contain one moving person at a time, and a static scene otherwise. By varying lighting in the room we have more or less intense cast shadows. We implemented the YUV and RGB methods for comparison and run those on the same input sequences.

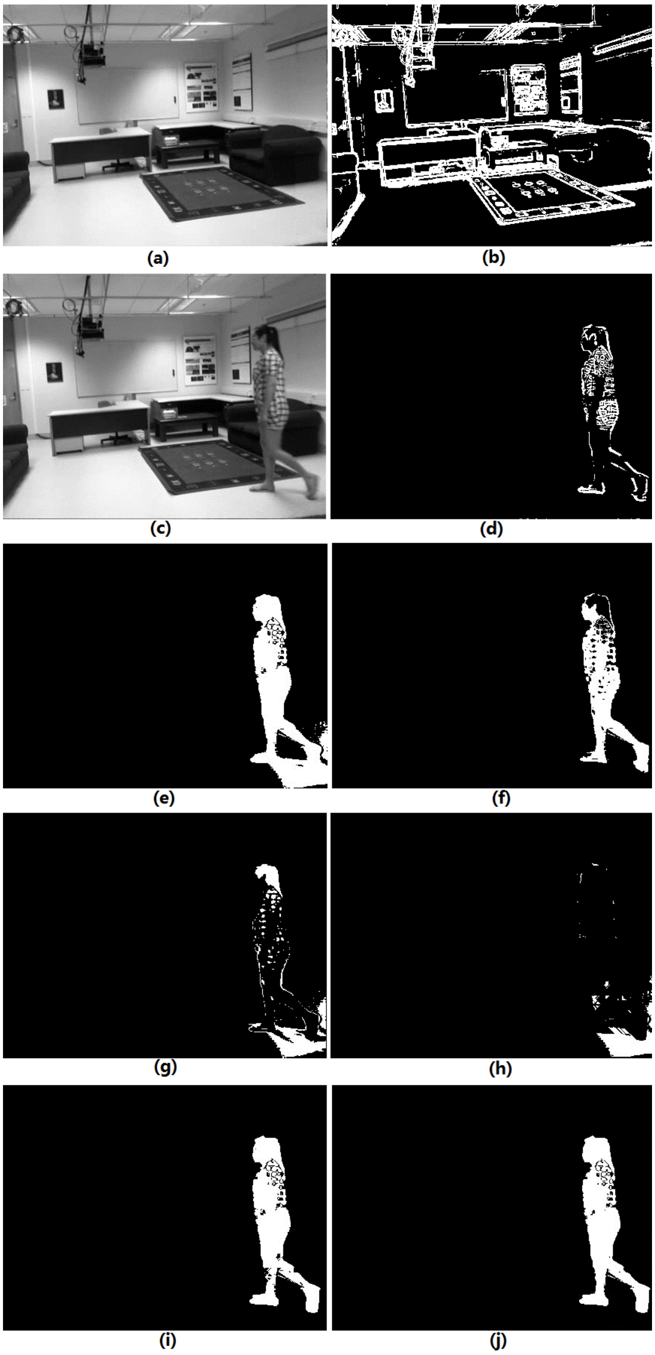

Fig. 7. Experimental results. (a) Background. (b) Background edge. (c) Current frame. (d) Occlusion boundary of a person. (e) Detected foreground by background subtraction. (f) Initial silhouette after shadow removal using Gaussian distribution. (g) Candidate shadow. (h) True shadow after shadow evaluation. (i) Silhouette after shadow evaluation. (j) Silhouette after compensation.

Fig. 7 shows various outputs generated at various stages of the proposed system: (a) is static background in an input sequence, (b) shows detected background edges using the simple Sobel operator, (c) shows a frame of the sequence also showing a moving person, (d) shows the detected occlusion boundary of that person again using the Sobel operator, (e) is the result when detecting foreground by background subtraction, (f) is the initial silhouette after shadow removal based on Gaussian distribution, and (g) shows the extracted candidate shadow pixels.

After shadow removal using the Gaussian distribution, all true shadow pixels are detected successfully. At the same time, several silhouette pixels in the head or foot regions, and in occlusion boundary regions are also misclassified as being shadow pixels if their pixel intensities are far away from the mean μ of the intensities of all foreground pixels. After shadow evaluation using the true occlusion boundary of the person, which was detected by the Sobel operator, these misclassified pixels are relabeled as being silhouette pixels. These stages of the process are illustrated in images (h) and (i). Image (i) shows the result of shadow verification using our shadow evaluator. Finally, image (j) shows the completed silhouette after the compensation process.

TABLE I: Analysis of Experimental Results. Percentages Are Rounded to Nearest Integers

Process	***PN***	***CN***	***CR*** **(%)**	***HR*** **(%)**
Background subtraction	10,544	7,530	71	93
Shadow removal	6,683	6,251	94	77
Shadow evaluation	7,976	7,363	92	91
Compensation	8,146	7,530	92	93
RN **= 8,106**				

We use measures for evaluating our experimental results. In Tables I, II, and III, *PN* is the number of detected silhouette pixels, *CN* is the number of detected correct silhouette pixels, $CR = (CN/PN) \times 100$ is the correct rate, $HR = (CN/RN) \times 100$ is the hit rate, *RN* is the number of true silhouette pixels, which are obtained by manual marking of the input frame.

Table I shows that during the shadow evaluation, the proposed shadow evaluator recognized 1,112 fake shadow pixels. Value *HR* increases from 77% to 91% after the shadow evaluation.

Fig. 8 illustrates a comparison of results obtained with our or with other methods. Subfigure (a) is an input frame in YUV color space for the YUV method. Subfigure (b) is an input frame in RGB color space for the RGB method. Subfigure (c) is the detected foreground by using approximate median filter, (d) and (e) are silhouettes after shadow removal using the YUV method or RGB method, respectively, and (f) is the detected silhouette using the proposed method. Subfigures (g), (h), and (i) are compensated silhouettes for (d), (e), and (f), respectively. Subfigures (f) and (i) show that our proposed algorithm keeps the silhouette more accurate than the YUV and RGB methods, as subfigures (d), (e), (g), and (h) illustrate.

See Fig. 9. In the current input frame (a), light caused that the shoulder area drifts to white. The name card is white originally. In both situations, the YUV and RGB methods fail to distinguish these pixels from true shadow; see (b) and (c). The proposed method recognizes that these pixels are fake shadows; see (d).

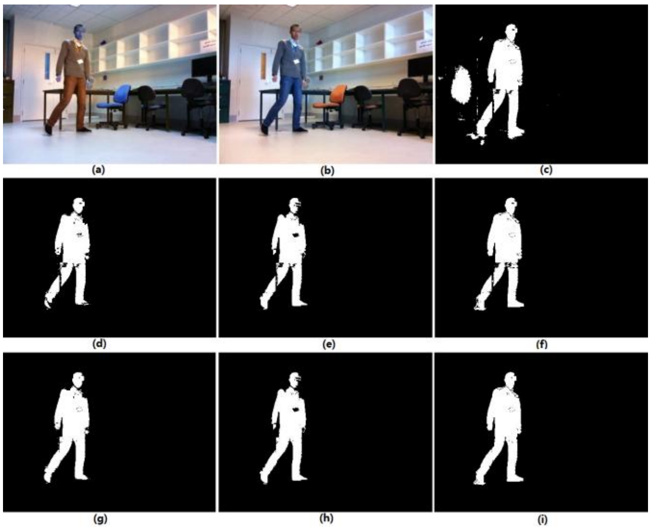

Fig. 8. Experimental results compared with the YUV and RGB methods. (a) Current frame in YUV format. (b) Current frame in RGB format. (c) Detected foreground using a median filter. (d) Silhouette after shadow removal using the YUV method. (e) Silhouette after shadow removal using the RGB method. (f) Silhouette after shadow removal using the proposed algorithm. (g) Silhouette after compensation on (d). (h) Silhouette after compensation on (e). (i) Silhouette after compensation on (f).

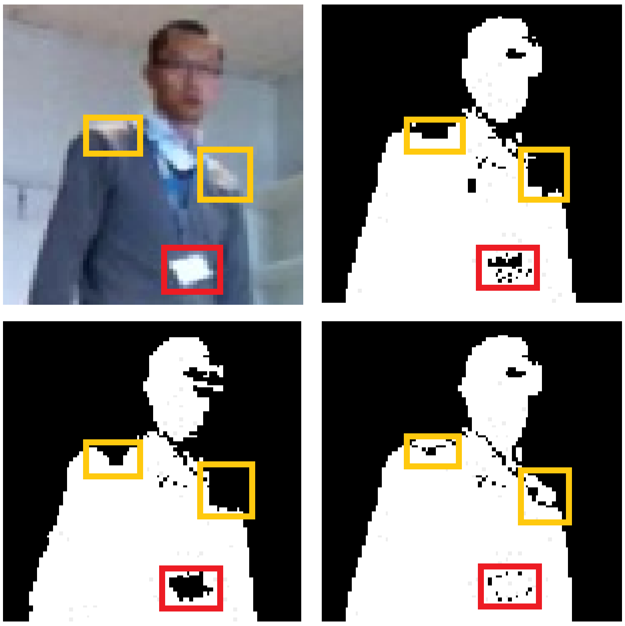

Fig. 9. Shadow removal result. *Top left*: Current input frame. *Top right*: YUV silhouette. *Bottom left*: RGB silhouette. *Bottom right*: Silhouette using the proposed method.

TABLE II: Analysis of Silhouette Extraction Results. Percentages Are Rounded to Nearest Integers

Stage	Method	*PN*	*CN*	*CR* (%)	*HR* (%)
Before compensation	YUV	9,129	8,619	94	90
	RGB	8,773	8,555	98	89
	Proposed	9,626	9,012	94	94
After compensation	YUV	9,545	8,975	94	94
	RGB	9,050	8,787	97	92
	Proposed	9,966	9,313	93	97
RN = 9,596					

Table II lists results for silhouette extraction using the defined measures. Before compensation, the proposed algorithm extracted 94% of silhouette pixels, while the YUV and RGB methods only detect 90% and 89%, respectively. The effectiveness of the advocated compensation algorithm is also confirmed by the fact that the hit rate (*HR*) of proposed, YUV, or RGB method increases to 97%, 94%, or 92%, respectively. There is a minor decrease in correct rate (CR) if the silhouette was compensated.

Table III states that only 142 silhouette pixels are misclassified as shadow by the proposed method. There are 535 and 599 misclassified pixels for the YUV and RGB methods, respectively. The correct shadow removal rate of the proposed method equals 95%, and this supersedes those of the YUV and RGB by more than ten percent. Let *RPN* be the number of pixels removed by one of the three shadow removal methods (i.e., *RPN* = 12,638 – *PN*). Let *RSN* be the number of removed true shadow pixels, and *FSN* stands for the number of removed fake shadow pixels. We have that *RSN* = (12,638 – 9,154) – (*PN* – *CN*) and *FSN* = *RPN* – *RSN*, where *RCR* = (*RSN* / *RPN*) × 100 is the correct shadow removal rate.

TABLE III: Analysis of Shadow Removal Results. Percentages Are Rounded to Nearest Integers

Method /Stage	*PN*	*CN*	*RPN*	*RSN*	*FSN*	*RCR*
Before Shadow Removing	12,638	9,154	-	-	-	-
YUV	9,129	8,619	3,509	2,974	535	85%
RGB	8,773	8,555	3,865	3,266	599	85%
Proposed	9,626	9,012	3,012	2,870	142	95%
RN = 9,596						

V. Conclusions

This paper presented a new shadow removal algorithm in the context of silhouette detection for moving people in gray level video sequences. The algorithm employs an approximate median filter for modeling the background, uses the Gaussian distribution and the simple Sobel operator for detecting edge information to extract true shadow in the foreground. We generate the foreground by applying a straightforward background-subtraction scheme between current frame and modeled background.

The crucial idea of the proposed algorithm is that true shadow should not be enclosed or semi-enclosed by occlusion boundaries of the moving person. Experimental results show that the presented method can distinguish true shadow from falsely detected shadow, even if the intensity values are very similar to that of the background. In such cases, the YUV and RGB methods are not able to detect silhouette pixels properly.

Acknowledgment

The authors thank N. Amarasinghe for helpful discussions about the presented subject, and a special thanks goes to Dongwei Liu and Xiaojie Liu for experiment videos capture.

REFERENCES

[1] O. Schreer, I. Feldmann, U. Golz, and P. Kauff, "Fast and robust shadow detection in videoconference applications," in *Proc. EURASIP-IEEE Symp. VIPromCom*, 2002, pp. 371–375.

[2] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Proc. IEEE ICCV FRAME-RATE Workshop*, 1999, pp. 1–19.

[3] X. Chen, Z. He, D. Anderson, J. M. Keller, and M. Skubic, "Adaptive silhouette extraction and human tracking in complex and dynamic environments," in *Proc. ICIP*, 2006, pp. 561–564.

[4] I. Huerta, M. Holte, T. Moeslund, and J. Gonza`lez, "Detection and removal of chromatic moving shadows in surveillance scenarios," in *Proc. ICCV*, 2009, pp. 1499–1506.

[5] T. Thongkamwitoon, S. Aramvith, and T. H. Chalidabhongse, "An adaptive real-time background subtraction and moving shadows detection," in *Proc. IEEE Int. Conf. ICME*, 2004, vol. 2, pp. 1459–1462.

[6] J. C. S. Jaques, C. R. Jung, and S. R. Musse, "Background subtraction and shadow detection in grayscale video sequences," in *Proc. SIBGRAPI*, 2005, pp. 189–196.

[7] D. Grest, J.-M. Frahm, and R. Koch, "A color similarity measure for robust shadow removal in real time," in *Proc. Vision Modeling Visualization*, 2003, pp. 253–260.

[8] M. Shoaib, R. Dragon, and J. Ostermann, "Shadow detection for moving humans using gradient-based background subtraction," in *Proc. IEEE Conf. ICASSP*, 2009, pp. 773–776.

[9] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Proc. IEEE Workshop Motion Video Computing*, 2002, pp. 22–27.

[10] D. Koller, K. Daniilidis, and H.-H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Computer Vision*, vol. 10, pp. 257–281, 1993.

[11] J. Stauder, R. Mech, and J. Ostermann, "Detection of moving cast shadows for object segmentation," *IEEE Trans. Multimedia*, vol. 1, pp. 65–76, 1999.

[12] Z. Liu, K. Huang, T. Tan, and L. Wang, "Cast shadow removal with

[13] GMM for surface reflectance component," in *Proc. ICPR*, 2006, vol. 1, pp. 727–730.

[14] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision Applications*, vol. 8, no. 187–193, 1995.

[15] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 25, pp. 1337–1342, 2003.

[16] F. Cardile, G. Iannizzotto, and F. L. Rosa, "A vision-based system for elderly patients monitoring," in *Proc. Human System Interactions*, 2010, pp. 195–202.

[17] W. K. Pratt, *Digital Image Processing*, New York: Wiley, 1978.

[18] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE J. Solid-State Circuits*, vol. 23, no. 358–367, 1988.

Zhengping Wang received the B.S. degree in computer science from Shijiazhuang University of Economics, China in 2002, and the M.S. degree in computer application technology from the Beijing Institute of Technology, China in 2005. He is currently working towards the M.S. degree at the Department of Computer Science, The University of Auckland, in the .enpeda.. (Environment Perception and Driver Assistance) research group. His present research interests are object tracking, feature extraction, motion tracking and video processing.

Bok-Suk Shin is a PhD graduate of Pusan National University, Korea. Since February 2011, she is a post-doc researcher at the Computer Science Department, The University of Auckland, in the .enpeda.. (Environment Perception and Driver Assistance) research group. She has 10 years of experience in research, project design, and teaching in data analysis, pattern recognition, computer vision, 3D visualization, and 3D game development. So far, her more than 30 publications have been dominantly on track recognition for small species, with a current shift towards 3D computer vision, especially in the area of vision-based driver assistance.

Reinhard Klette is a fellow of the Royal Society of New Zealand and a professor at The University of Auckland. He is on the editorial board of the International Journal of Computer Vision. He was the founding editor-in-chief of the Journal of Control Engineering and Technology in 2011 - 2013, and an associate editor of IEEE PAMI in 2001-2008. He (co-)authored more than 250 publications in peer-reviewed journals or conferences, and books on computer vision, image processing, geometric algorithms, and panoramic imaging. He presented more than 20 keynotes at international conferences. Springer London (UK) publishes in January 2014 his book on "Concise Computer Vision".