

Fast Video Segmentation Algorithm With Shadow Cancellation, Global Motion Compensation, and Adaptive Threshold Techniques

Shao-Yi Chien, Yu-Wen Huang, Bing-Yu Hsieh, Shyh-Yih Ma, and Liang-Gee Chen, *Fellow, IEEE*

Abstract—Automatic video segmentation plays an important role in real-time MPEG-4 encoding systems. Several video segmentation algorithms have been proposed; however, most of them are not suitable for real-time applications because of high computation load and many parameters needed to be set in advance. This paper presents a fast video segmentation algorithm for MPEG-4 camera systems. With change detection and background registration techniques, this algorithm can give satisfying segmentation results with low computation load. The processing speed of 40 QCIF frames per second can be achieved on a personal computer with an 800 MHz Pentium-III processor. Besides, it has shadow cancellation mode, which can deal with light changing effect and shadow effect. A fast global motion compensation algorithm is also included in this algorithm to make it applicable in slight moving camera situations. Furthermore, the required parameters can be decided automatically, which can enhance the proposed algorithm to have adaptive threshold ability. It can be integrated into MPEG-4 videophone systems and digital cameras.

Index Terms—Adaptive threshold, background registration, global motion compensation, MPEG-4 camera systems, object extraction, shadow cancellation, video segmentation.

I. INTRODUCTION

THE MPEG-4 standard [1] has been taken as the most important standard for multimedia and visual communication and will be applied to many real-time applications, such as video phones, video conference systems, and smart camera systems. The most important function of MPEG-4 video part is content-based coding, which can support content-based manipulation and representation of video signal and random access of video objects (VO). To support this, video sequences are encoded object by object rather than frame by frame, and the shape information of each video object is required. Automatic video segmentation is the technique to generate shape information of video objects from video sequences. It is very important in a real-time MPEG-4 camera system with content-based coding scheme, since the shape information is required for shape

coding, motion estimation, motion compensation, and texture coding, as shown in Fig. 1 [2]. Without automatic video segmentation, MPEG-4 content-based coding scheme cannot be realized in real-time applications. Consequently, a real-time automatic video segmentation system that can produce good segmentation results is urgently required.

Several video segmentation algorithms have been proposed [3]. They can be classified into three types: edge information based video segmentation, image segmentation based video segmentation, and change detection based video segmentation. Edge information based algorithms [4], [5] first apply Canny edge detector to find edge information of each frame and then keep tracking these edges. A morphology motion filter is also applied to find edges belonging to foreground objects. Next, a filling technique can connect edge information to generate final object masks. This method can deal with both still camera and moving camera situations; however, the computation load is very large. Image segmentation based algorithms first apply image segmentation algorithms, such as watershed transform [6], [7] and color segmentation [8], [9] on each frame to separate a frame into many homogeneous regions. By combining motion information derived with motion estimation, optical flow, or frame difference, regions with motion vectors different from the global motion are merged as foreground regions. These algorithms often can give segmentation results with accurate boundaries, but the computation load for image segmentation and motion information calculation is also high, and the region merging process often has many parameters to set. Both these two kinds of algorithms are too complex to be integrated into a real-time system. Change detection based segmentation algorithms [10]–[12] threshold the frame difference to form change detection mask. Then the change detection masks are further processed to generate final object masks. The processing speed is high, but it is often not robust. The segmentation results are suffered from the uncovered background situations, still object situations, light changing, shadow, and noise. The robustness can be promoted by a lot of postprocessing algorithms [10]; however, complex postprocessing will make the efficiency of less computation lost. Besides, these algorithms cannot deal with moving camera situations, and the threshold of change detection is very critical and cannot be automatically decided. These reasons make this kind of algorithms not practical for real applications.

In this paper, a fast video segmentation algorithm for MPEG-4 camera systems is proposed. The algorithm has four

Manuscript received February 17, 2002; revised January 27, 2003. This work was supported by National Science Council of Taiwan, R.O.C. under Grant NSC91-2219-E-002-045. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Radu Serban Jasinschi.

S.-Y. Chien, Y.-W. Huang, B.-Y. Hsieh, and L.-G. Chen are with DSP/IC Design Lab, Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: shoayi@video.ee.ntu.edu.tw; yuwen@video.ee.ntu.edu.tw; bingyu@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw).

S.-Y. Ma is with the Vivotek, Inc, Taipei County 235, Taiwan, R.O.C. (e-mail: syma@ieee.org).

Digital Object Identifier 10.1109/TMM.2004.834868

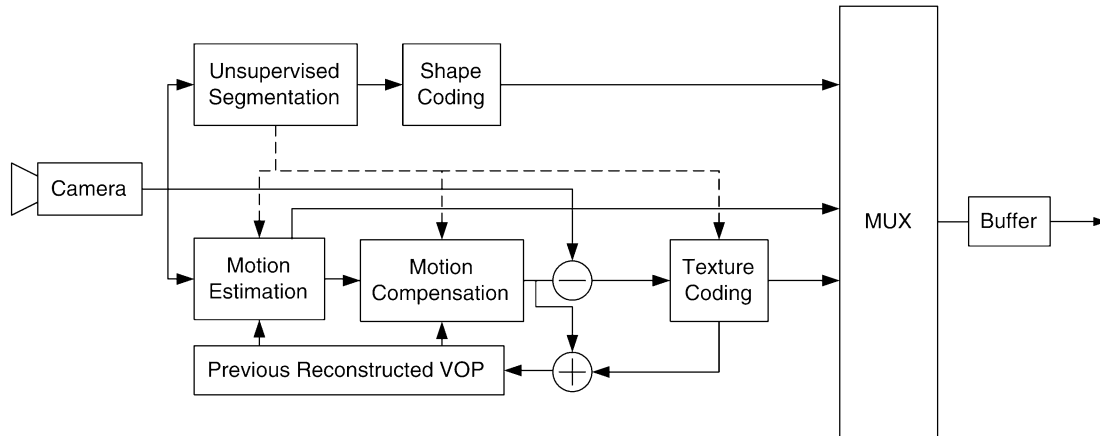


Fig. 1. Block diagram of MPEG-4 encoding systems.

modes: baseline mode, shadow cancellation mode, global motion compensation mode, and adaptive threshold mode. It is based on our previous work using change detection [12]. With background registration technique, this algorithm can deal with uncovered background and still object situations. An efficient postprocessing algorithm can improve segmentation results without large computation overhead. Moreover, it has a shadow cancellation mode, in which light changing effect and shadow effect can be suppressed. A global motion compensation mode can deal with slightly moving camera situations with low computation load. Furthermore, an adaptive threshold mode is also proposed to decide the threshold automatically.

Efficient implementation of this algorithm is also proposed in this paper. Since recent microprocessors, such as general purpose microprocessors, digital signal processor, and embedded processors, improve their multimedia capability by single instruction multiple data (SIMD) technique, this algorithm is optimized on SIMD architecture. Besides, fast morphological operations are also integrated in this algorithm with bit-parallel technique [13].

This paper is organized as follows. The overview of the proposed segmentation algorithm is shown in Section II. Sections III–VI describe baseline mode, shadow cancellation mode (SC mode), global motion compensation mode (GMC mode), and adaptive threshold mode (AT mode), respectively. Next, the efficient implementation of the proposed algorithm is shown in Section VII. The experimental results are shown in Section VIII. Finally, Section IX gives a conclusion of this paper.

II. SYSTEM OVERVIEW

The block diagram of the proposed video segmentation algorithm is shown in Fig. 2. It contains four main parts: *GMC*, *Gradient Filter*, *Threshold Decision*, and *Video Segmentation Baseline*. It also has four modes. Each mode is a combination of the four main parts. The selection of modes is done manually by the users according to the shoot situations.

The baseline mode is designed for ideal situations, in which no light changing effect and shadow effect occur, the camera

is still, and the environment is stable. In this mode, *GMC* and *Gradient Filter* are turned off, and *Video Segmentation Baseline* is turned on. A change detection and background registration based video segmentation algorithm is applied.

The shadow cancellation mode (SC mode) is designed for the situations when light changes and shadow exists. *Gradient Filter* and *Video Segmentation Baseline* are turned on in this mode, and *Postprocessing* is modified.

When cameras are held by hands, slight motion of cameras is inevitable, and conventional change detection based algorithms cannot be applied. The global motion compensation mode (GMC mode) is designed for this situation. In GMC mode, *Gradient Filter*, *GMC*, and *Video Segmentation Baseline* are turned on, and the background information is compensated.

The adaptive threshold mode (AT mode) can be applied with all the three modes described above. When environment changes dramatically, for example, light sources are changed, the camera is turned on at the first time, or automatic gain controller (AGC) of the camera is working, *Threshold Decision* should be turned on. It can decide the optimal threshold automatically. The detail of each mode is described in Sections III–VI.

III. BASELINE MODE

Baseline mode is designed for stable situations. That is, the camera is still, and there is no light changing and no shadows. It is based on change detection and background registration technique. Unlike other change detection algorithms, the change detection mask here is not only generated from the frame difference of current frame and previous frame but also from the frame difference between current frame and background frame, which can be produced by background registration technique. Since the background is stationary, it is well-behaved and more reliable than previous frame. Besides, still objects and uncovered background problems can be easily solved under this scheme.

The block diagram of baseline mode is shown in Fig. 3. There are five parts in baseline mode: *Frame Difference*, *Background Registration*, *Background Difference*, *Object Detection*, and *Postprocessing*.

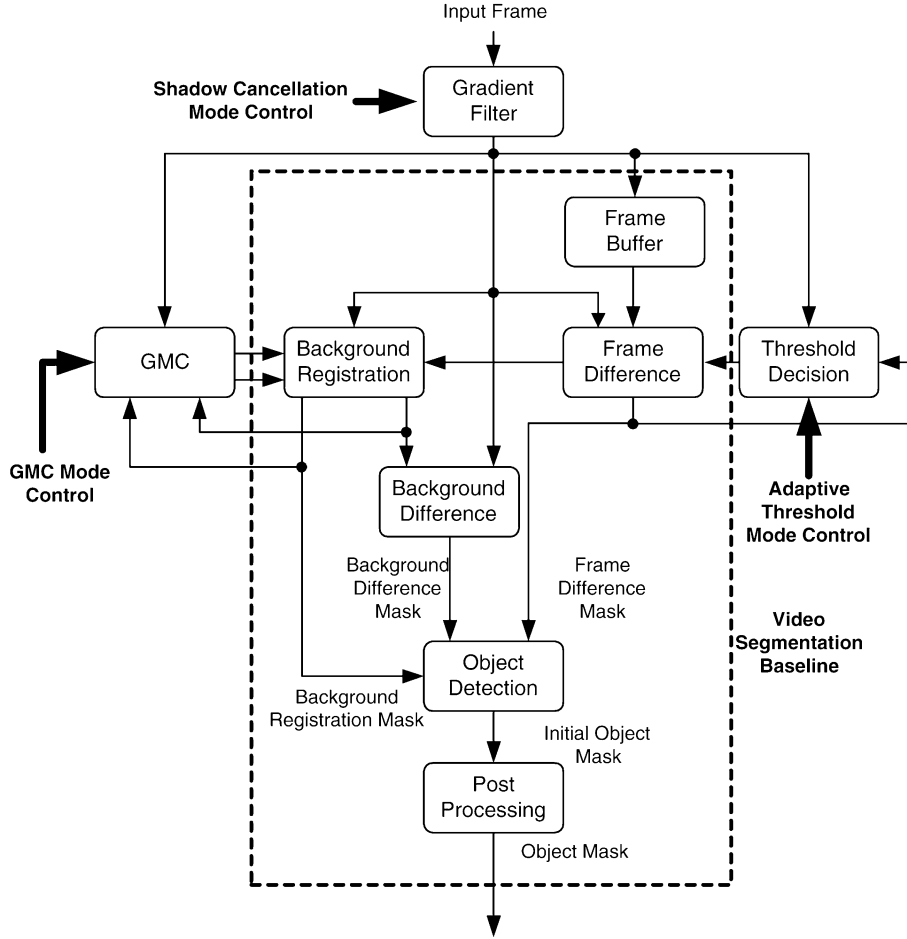


Fig. 2. Block diagram of the proposed algorithm.

A. Frame Difference

In *Frame Difference*, the frame difference between current frame and previous frame, which is stored in *Frame Buffer*, is calculated and thresholded. It can be presented as

$$FD(x, y, t) = |I(x, y, t) - I(x, y, t - 1)| \quad (1)$$

$$FDM(x, y, t) = \begin{cases} 1 & \text{if } FD \geq Th \\ 0 & \text{if } FD < Th \end{cases} \quad (2)$$

where I is frame data, FD is frame difference, and FDM is *Frame Difference Mask*. Note that there is a parameter Th needed to be set in advance. The method to decide the optimal Th is shown in Section VI. Pixels belonging to FDM are viewed as “moving pixels.”

B. Background Registration

Background Registration can extract background information from video sequences. According to FDM , pixels not moving for a long time are considered as reliable background pixels. The procedure of *Background Registration* can be shown as

$$SI(x, y, t) = \begin{cases} SI(x, y, t - 1) + 1, & \text{if } FDM = 0 \\ 0, & \text{if } FDM = 1 \end{cases} \quad (3)$$

$$BG(x, y, t) = \begin{cases} I(x, y, t), & \text{if } SI(x, y, t) = Fth \\ BG(x, y, t - 1), & \text{else} \end{cases} \quad (4)$$

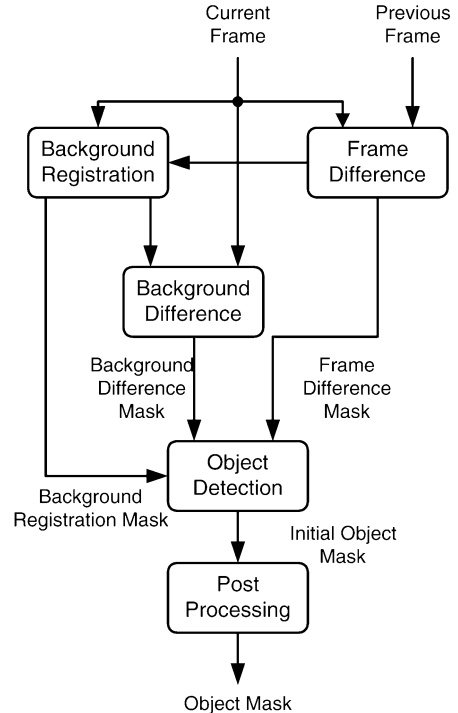


Fig. 3. Block diagram of baseline mode.

$$BI(x, y, t) = \begin{cases} 1, & \text{if } SI(x, y, t) = Fth \\ BI(x, y, t - 1), & \text{else} \end{cases} \quad (5)$$

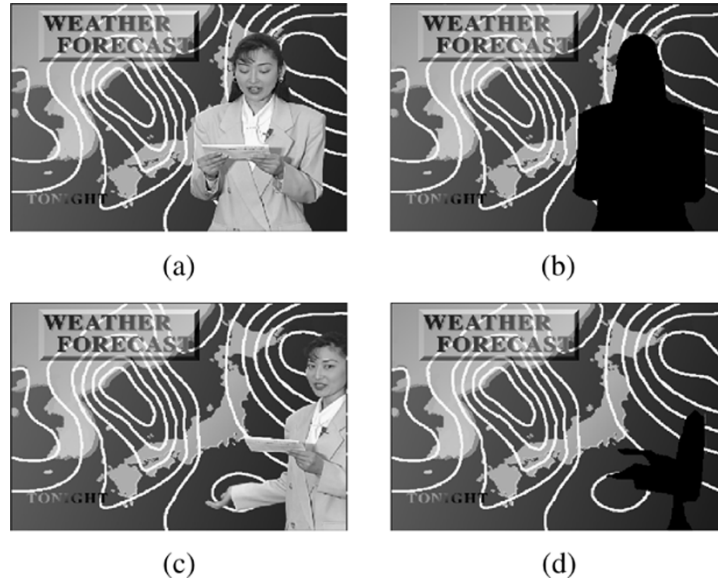


Fig. 4. Illustration of background registration technique. (a) *Weather* at #50; (b) background information at #50; (c) *Weather* at #100; (d) background information at #100.

where *SI* is *Stationary Index*, *BI* is *Background Indicator*, and *BG* is the background information. The initial values of *SI*, *BG*, and *BI* are all set to “0.” *Stationary Index* records the possibility if a pixel is in background region. If *SI* is high, the possibility is high; otherwise, it is low. If a pixel is “not moving” for many consecutive frames, the possibility should be high, which is the main concept of (3). When the possibility is high enough, the current pixel information of the position is registered into the background buffer *BG*, which is shown as (4). Besides, *Background Indicator* is used to indicate whether the background information of current position exists or not, which is shown as (5). Note that (3)–(5) also imply that a background updating ability is also included in *Background Registration*, that is, if background changes, new background information will be updated into the background buffer.

Fig. 4 shows the results of *Background Registration*. The original frames and the registered background information are shown in Fig. 4(a)–(d), respectively. In Fig. 4(b) and (d), the black parts indicate the regions where background information is not available until that time, which is caused by being covered by the foreground object. Obviously, more and more background information is available as more and more frames are input into the system. Since the number of frames of sequence *Weather* is 100, Fig. 4(d) shows the total background information we can get from the sequence. The black parts in Fig. 4(d) are the parts which are covered by the foreground object in entire sequence.

C. Background Difference

The procedure of *Background Difference* is similar to that of *Frame difference*. What is different is that the previous frame is substituted by background frame *BG*. After *Background Difference*, another change detection mask named *Background Differ-*

TABLE I
SITUATIONS OF OBJECT DETECTION

Situation	FDM	BDM	BI	IOM
Stationary	0	–	0	0
Moving	1	–	0	1
Background	0	0	1	0
Moving object	1	1	1	1
Still object	0	1	1	1
Uncovered background	1	0	1	0

ence Mask (BDM) is generated. The operations of *Background Difference* can be shown by

$$BD(x, y, t) = |I(x, y, t) - BG(x, y, t - 1)| \quad (6)$$

$$BDM(x, y, t) = \begin{cases} 1, & \text{if } BD \geq Th \\ 0, & \text{if } BD < Th, \end{cases} \quad (7)$$

where *BD* is background difference, *BG* is background frame, and *BDM* is *Background Difference Mask*, respectively.

D. Object Detection

Both of *FDM* and *BDM* are input into *Object Detection* to produce *Initial Object Mask (IOM)*. The procedure of *Object Detection* can be presented as the following equation.

$$IOM(x, y, t) = \begin{cases} BDM(x, y, t), & \text{if } BI(x, y, t) = 1 \\ FDM(x, y, t), & \text{else.} \end{cases} \quad (8)$$

This process can deal with the six situations shown in Table I, where “–” means “not available.” Note that the last two situations are easily misclassified by other change detection based segmentation algorithms, where *BDM* information is not available. In other algorithms, the still objects are often taken as background objects because they are not included in *FDM*, and the uncovered background is often taken as foreground object because it is included in *FDM*. Both of these two situations need complex postprocessing algorithms to compensate the mis-classification, which are not needed in the proposed algorithm.

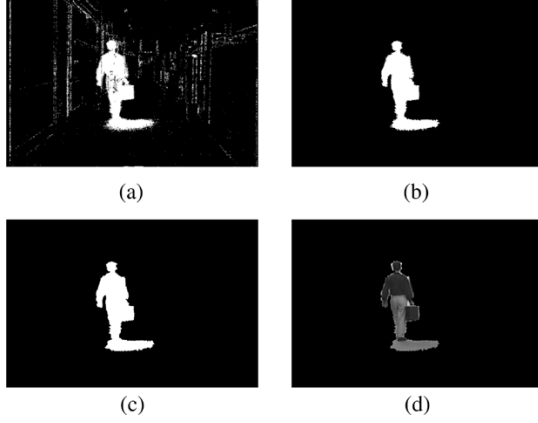


Fig. 5. Illustration of postprocessing. (a) Initial object mask; (b) after noise elimination; (c) after morphological closing operation; (d) generated VOP.

E. Postprocessing

The *Initial Object Mask (IOM)* generated by *Object Detection* has some noise regions because of irregular object motion and camera noise. Also, the boundary may not be very smooth. Therefore, there are two parts in *Postprocessing*: noise region elimination and boundary smoothing.

The connected component algorithm [14] can mark each connected region with a special label. Then we can filter these regions by their area. If the area of a region is small, it may be a noise region and can be eliminated. Background regions, which are indicated by “0” in *IOM*, are first filtered, that is, background regions with small area are eliminated. This process eliminates holes in the change detection mask, which often occur especially when the texture of foreground objects is insignificant. Then foreground regions, which are indicated by “1” in *IOM*, are then filtered. This process removes noise regions. Next, the morphological close–open operations [15] are applied to smooth the boundary of object mask. In addition, *Stationary Index* is further revised with *IOM* by

$$SI(x, y, t) = 0, \text{ if } IOM(x, y, t) = 1. \quad (9)$$

This process can avoid still objects to be registered into the background buffer.

Fig. 5 shows the effect of *Postprocessing*. Fig. 5(a) is *IOM*, where the white parts are those indicated by “1” in *IOM*, and the black parts are those indicated by “0” in *IOM*. After noise region elimination, the mask can be improved as shown in Fig. 5(b). After boundary smoothing, the improved mask is shown in Fig. 5(c). Finally, the generated VOP is shown in Fig. 5(d).

IV. SHADOW CANCELLATION MODE

Conventional change detection algorithms usually cannot give acceptable segmentation results when light changes, or shadows exist. Shadow regions are always falsely detected as parts of foreground regions, and the whole frame will be regarded as foreground object if light changes. In these situations, shadow cancellation mode (SC mode) is preferred. Unlike the other complex algorithms to get rid of shadow regions [16], in SC mode, the influence of light changing and shadows can

be suppressed, and good segmentation results are maintained without large computation overhead. In Sections IV-A and B, the effects of light changing and shadow are discussed and analyzed. Based on the analyzes, a shadow cancellation algorithm is proposed in Section IV-C.

A. Light Changing and Shadow Effects

The image luminance at time t can be modeled as the following equation.

$$I_t(x, y) = E_t(x, y)\rho_t(x, y) \quad (10)$$

where $I_t(x, y)$ is the luminance of the pixel (x, y) at time t , $E_t(x, y)$ is the irradiance, and $\rho_t(x, y)$ is the reflectance of the object surfaces [16]. For the foreground objects, this equation is modified as

$$I_t^F(x, y) = E_t^F(x, y)\rho_t^F(x, y) \quad (11)$$

where $E_t^F(x, y)$ is the irradiance to the foreground objects, and $\rho_t^F(x, y)$ is the reflectance of the foreground object surfaces. On the other hand, for the background objects, this equation is rewritten as

$$I_t^B(x, y) = E_t^B(x, y)\rho_t^B(x, y) \quad (12)$$

where $E_t^B(x, y)$ is the irradiance to the background objects, and $\rho_t^B(x, y)$ is the reflectance of the background object surfaces. Similarly, for the registered background frame generated with *Background Registration*, the luminance is defined as

$$I_B(x, y) = E_B(x, y)\rho_B(x, y). \quad (13)$$

Note that, the background is assumed to be static, and the background updating operation is ignored, that is, and at any time

$$\rho_t^B(x, y) = \rho_B(x, y). \quad (14)$$

1) *No Light Changing and No Shadow Situations*: When there is no light changing and no shadow, the irradiance of foreground objects, background object, and registered background has the relationship

$$E_t^F = E_t^B = E_B = E. \quad (15)$$

The discriminant function is the function of *Background Difference*. For foreground objects

$$\begin{aligned} BD_t(x, y) &= I_t^F(x, y) - I_B(x, y) \\ &= E_t^F(x, y)\rho_t^F(x, y) - E_B(x, y)\rho_B(x, y) \\ &= E(x, y)\rho_t^F(x, y) - E(x, y)\rho_B(x, y) \\ &= E(x, y) [\rho_t^F(x, y) - \rho_B(x, y)] \neq 0. \end{aligned} \quad (16)$$

The absolute values of BD_t for foreground objects should be large in most situations. For the background object, the discriminant function is

$$\begin{aligned} BD_t(x, y) &= I_t^B(x, y) - I_B(x, y) \\ &= E(x, y) [\rho_t^B(x, y) - \rho_B(x, y)] \\ &= 0. \end{aligned} \quad (17)$$

Therefore, a threshold Th can easily separate foreground objects from background.

2) *Light Changing Effect*: When lighting is changed, we can model the irradiance as

$$\begin{aligned} E_B &= E, \\ E_t^F &= E_t^B = E + c \end{aligned} \quad (18)$$

where c is light changing factor and is a constant. For background object, the dicriminant function is

$$\begin{aligned} BD_t(x, y) &= I_t^B(x, y) - I_B(x, y) \\ &= E_t^B(x, y)\rho_t^B(x, y) - E_B(x, y)\rho_B(x, y) \\ &= \rho_B(x, y) [E_t^B(x, y) - E_B(x, y)] \\ &= c\rho_B(x, y) \neq 0. \end{aligned} \quad (19)$$

Therefore, the background object may also be taken as foreground objects.

3) *Shadow Effect*: When shadow exists, we can model the irradiance as

$$\begin{aligned} E_t^F &= E_B = E, \\ E_t^B &= E + c(x, y) \end{aligned} \quad (20)$$

where c is a function of (x, y) , and is negative in the shadow regions. For background object, the dicriminant function is

$$\begin{aligned} BD_t(x, y) &= I_t^B(x, y) - I_B(x, y) \\ &= c(x, y)\rho_B(x, y). \end{aligned} \quad (21)$$

Therefore, the background object may also be taken as foreground object in the shadow regions.

B. Light Changing and Shadow Effects on the Gradient Images

From Section IV-A, we know that the object mask is not correct under the influence of the light changing factor c when lights change and shadows exist. c is a constant in light changing situation and a function of (x, y) when shadows exist. We think that the effects of the light changing factor can be reduced in the gradient domain, where the DC term of the function can be suppressed. In this section, the light changing and shadow effects are analyzed in the gradient domain. Note that, for simplification, the image signal function $I_t(x, y)$ is simplified to one-dimensional function $I_t(x)$. For two dimension situations, the analysis procedure is the same.

1) *No Light Changing and No Shadow Situations*: When there is no light changing and no shadow, the irradiance of foreground objects, background object, and registered background are the same as (15), and $dE/dx = 0$. For foreground objects, the discriminant function is modified to

$$\begin{aligned} \frac{dI_t^F}{dx} - \frac{dI_B}{dx} &= \frac{d[E_t^F(x)\rho_t^F(x)]}{dx} - \frac{d[E_B(x)\rho_B(x)]}{dx} \\ &= \left(E_t^F \frac{d\rho_t^F}{dx} + \rho_t^F \frac{dE_t^F}{dx} \right) \\ &\quad - \left(E_B \frac{d\rho_B}{dx} + \rho_B \frac{dE_B}{dx} \right) \\ &= \left(E \frac{d\rho_t^F}{dx} + \rho_t^F \frac{dE}{dx} \right) - \left(E \frac{d\rho_B}{dx} + \rho_B \frac{dE}{dx} \right) \\ &= E \frac{d(\rho_t^F - \rho_B)}{dx} + (\rho_t^F - \rho_B) \frac{dE}{dx} \\ &= \frac{d(BD_t)}{dx}. \end{aligned} \quad (22)$$

The absolute gradient values of BD_t should be very large at the boundaries of foreground objects and sometimes large in the inner part of the highly textured foreground objects. For background object, the dicriminant function is

$$\begin{aligned} \frac{dI_t^B}{dx} - \frac{dI_B}{dx} &= \left(E_t^B \frac{d\rho_t^B}{dx} + \rho_t^B \frac{dE_t^B}{dx} \right) \\ &\quad - \left(E_B \frac{d\rho_B}{dx} + \rho_B \frac{dE_B}{dx} \right) \\ &= E \frac{d(\rho_t^B - \rho_B)}{dx} + (\rho_t^B - \rho_B) \frac{dE}{dx} \\ &= 0. \end{aligned} \quad (23)$$

Therefore, a threshold Th can also separate foreground object from background. The discriminant function may be not as good as *Background Difference*; however, the postprocessing can fill the holes in the object mask and eliminate the noise in the background region to maintain the correctness.

2) *Light Changing Effect*: When lighting is changed, we can also model the irradiance as (18), and $dE/dx = 0$. For foreground objects, the discriminant function is modified to

$$\begin{aligned} \frac{dI_t^F}{dx} - \frac{dI_B}{dx} &= \left(E_t^F \frac{d\rho_t^F}{dx} + \rho_t^F \frac{dE_t^F}{dx} \right) \\ &\quad - \left(E_B \frac{d\rho_B}{dx} + \rho_B \frac{dE_B}{dx} \right) \\ &= \left[(E + c) \frac{d\rho_t^F}{dx} + \rho_t^F \frac{d(E + c)}{dx} \right] \\ &\quad - \left(E \frac{d\rho_B}{dx} + \rho_B \frac{dE}{dx} \right) \\ &= E \frac{d(\rho_t^F - \rho_B)}{dx} + c \frac{d\rho_t^F}{dx} + (\rho_t^F - \rho_B) \frac{dE}{dx} \\ &= \frac{dBD_t}{dx} + c \frac{d\rho_t^F}{dx}. \end{aligned} \quad (24)$$

For background object, the dicriminant function is

$$\begin{aligned} \frac{dI_t^B}{dx} - \frac{dI_B}{dx} &= \left(E_t^B \frac{d\rho_t^B}{dx} + \rho_t^B \frac{dE_t^B}{dx} \right) \\ &\quad - \left(E_B \frac{d\rho_B}{dx} + \rho_B \frac{dE_B}{dx} \right) \\ &= \left[(E + c) \frac{d\rho_t^B}{dx} + \rho_t^B \frac{d(E + c)}{dx} \right] \\ &\quad - \left(E \frac{d\rho_B}{dx} + \rho_B \frac{dE}{dx} \right) \\ &= E \frac{d(\rho_t^B - \rho_B)}{dx} + c \frac{d\rho_t^B}{dx} + (\rho_t^B - \rho_B) \frac{dE}{dx} \\ &= c \frac{d\rho_t^B}{dx}. \end{aligned} \quad (25)$$

In general situations, the foreground objects should have more complex texture, that is

$$\left| c \frac{d\rho_t^F}{dx} \right| > \left| c \frac{d\rho_t^B}{dx} \right| \quad (26)$$

and at the boundaries of the foreground objects:

$$\left| \frac{d(BD_t)}{dx} \right| \gg \left| c \frac{d\rho_t^B}{dx} \right| \text{ and } \left| c \frac{d\rho_t^F}{dx} \right| \gg \left| c \frac{d\rho_t^B}{dx} \right|. \quad (27)$$

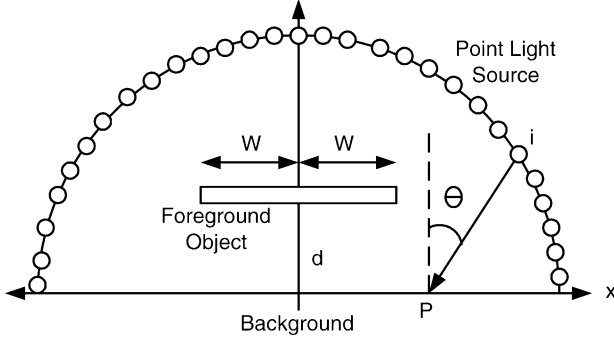
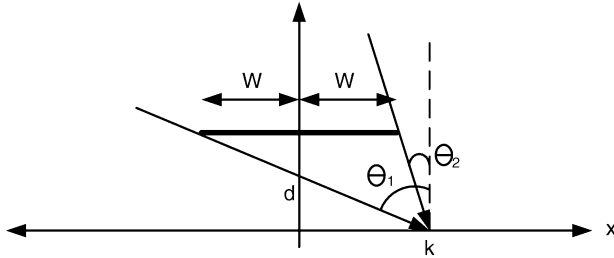


Fig. 6. Lighting model of an indoor environment.

Fig. 7. Angles where light sources are covered by the foreground object at position k .

Therefore, the foreground objects can also be separated from the background in the gradient domain even when light changes.

3) *Shadow Effect*: When shadow exists, we can model the irradiance as

$$\begin{aligned} E_t^F &= E_B = E, \\ E_t^B &= E + c(x). \end{aligned} \quad (28)$$

Before deriving the discriminant function, we first discuss the behavior of the light changing factor $c(x)$. In an indoor environment, several light sources may exist, and reflection of walls and ceiling can be taken as other new light sources. In this complex environment, we can model the light sources as shown in Fig. 6. The background is modeled as a line, and the light sources are modeled as infinite point light sources distributed on a semicircle as shown in Fig. 6. The foreground object is modeled as a line whose length is $2w$, and the distance between the foreground object to the background is d . For a point P on the background and a light source i on the semicircle, the amount of light power received at point P is [17]

$$E = c_A + c_i \cos \theta \quad (29)$$

where c_A is ambient light, which is a constant, c_i is the light power of the light source i , and $c_i \cos \theta$ is the diffusion light. For simplification, c_i is set as a constant c_τ for all the point light sources, and the light fading effect is ignored. Note that this model can also be employed in outdoor environment when the weather is cloudy.

In Fig. 7, for a point k on the background, the input rays of angles between θ_1 and θ_2 are covered by the foreground object, where

$$\theta_1 = -\arctan \frac{w+k}{d} \quad (30)$$

$$\theta_2 = -\arctan \frac{k-w}{d}. \quad (31)$$

The total light power received at point k is

$$\begin{aligned} E(k, w, d) &= \int_{-(\pi/2)}^{\theta_1} c_\tau \cos \theta d\theta + \int_{\theta_2}^{(\pi/2)} c_\tau \cos \theta d\theta + c_A \\ &= c_\tau \left[\sin \left(-\arctan \frac{w+k}{d} \right) - (-1) \right. \\ &\quad \left. + 1 - \sin \left(-\arctan \frac{k-w}{d} \right) \right] + c_A \\ &= c_A + c_\tau \left[\sin \left(-\arctan \frac{w+k}{d} \right) \right. \\ &\quad \left. - \sin \left(-\arctan \frac{k-w}{d} \right) + 2 \right]. \end{aligned} \quad (32)$$

Finally, the light changing factor $c(x)$ is

$$\begin{aligned} c(x) &= E(x, w, d) - E(x, 0, d) \\ &= c_\tau \left[\sin \left(-\arctan \frac{w+x}{d} \right) \right. \\ &\quad \left. - \sin \left(-\arctan \frac{x-w}{d} \right) \right]. \end{aligned} \quad (33)$$

If we set $c_\tau = 1$ and $w = 20$, the curve of the absolute value of light changing factor is shown in Fig. 8. From this figure, it is shown that the curve of $|c(x)|$ becomes smoother as the distance between the foreground object and the background becomes larger. The large values of $|c(x)|$ leads to the mis-classification of background objects in shadow regions as foreground objects.

Based on this model, for background object, the discriminant function in the gradient domain is

$$\begin{aligned} \frac{dI_t^B}{dx} - \frac{dI_B}{dx} &= \left(E_t^B \frac{d\rho_t^B}{dx} + \rho_t^B \frac{dE_t^B}{dx} \right) \\ &\quad - \left(E_B \frac{d\rho_B}{dx} + \rho_B \frac{dE_B}{dx} \right) \\ &= \frac{d\rho_B}{dx} c(x) + \rho_B \frac{dc(x)}{dx}. \end{aligned} \quad (34)$$

The absolute value of $dc(x)/dx$ is shown in Fig. 9. From Fig. 8, the irradiance of background can be separated into three types: illuminated, penumbra, and shadow, as shown in Fig. 10. We can discuss the values of (34) in these three type of regions with Figs. 8 and 9 First, in the illuminated regions,

$$\frac{d\rho_B}{dx} c(x) \approx 0, \quad (35)$$

$$\frac{dc(x)}{dx} \approx 0 \quad (36)$$

therefore

$$\frac{dI_t^B}{dx} - \frac{dI_B}{dx} \approx 0. \quad (37)$$

Next, in the shadow regions:

$$\frac{d\rho_B}{dx} c(x) \neq 0, \quad (38)$$

$$\frac{dc(x)}{dx} \approx 0 \quad (39)$$

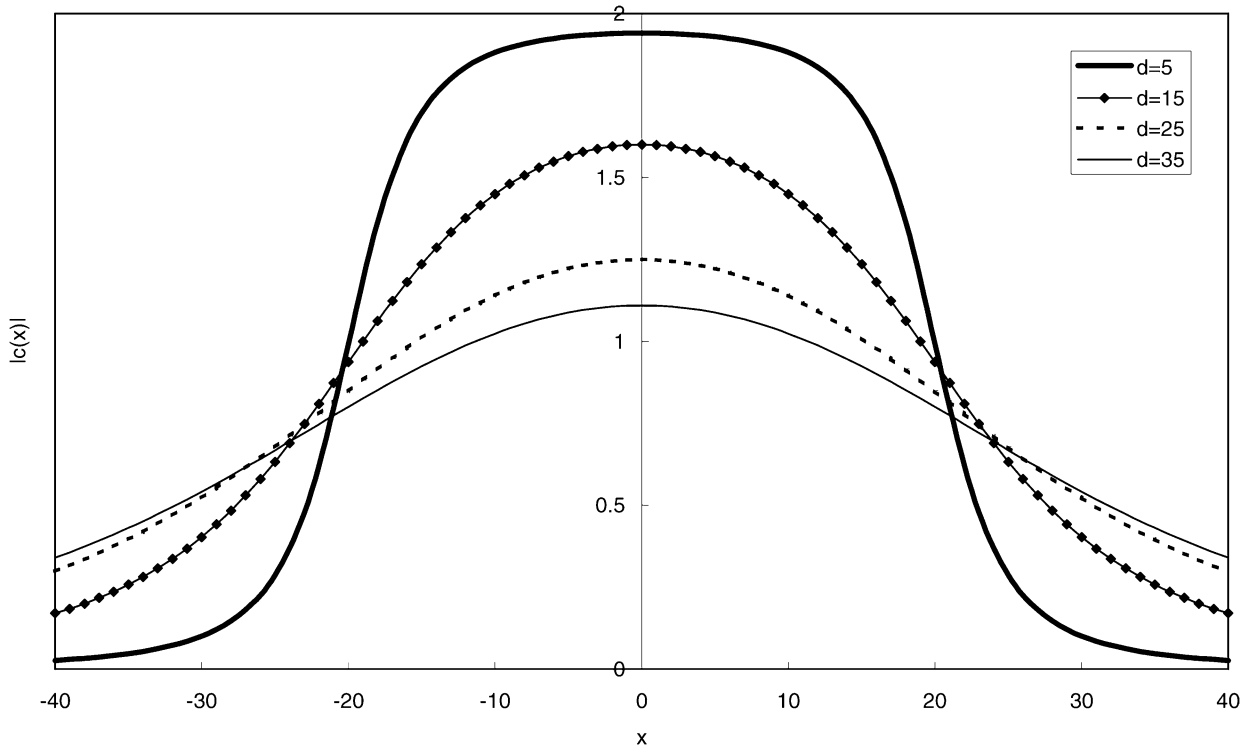


Fig. 8. Absolute value of light changing factor $c(x)$, where $w = 20$.

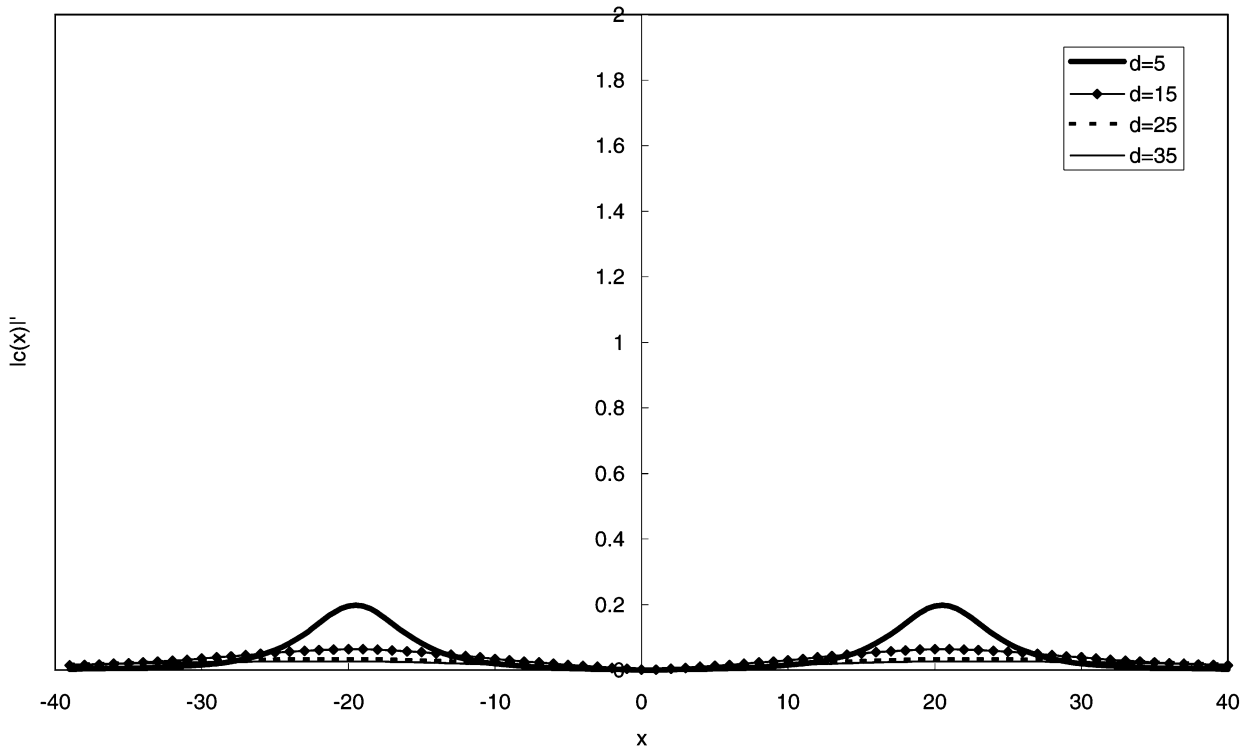


Fig. 9. Gradient value of $|c(x)|$.

and the first term of (34) dominates. If the texture of background region is insignificant, this term can also be ignored. Finally, in the penumbra regions

$$\frac{d\rho_B}{dx}c(x) \neq 0 \tag{40}$$

$$\frac{dc(x)}{dx} \neq 0. \tag{41}$$

In this situation, the second term of (34) may dominate when the distance between the foreground objects and the background is small. This may still cause mis-classification of the boundaries of shadows as foreground objects.

In conclusion, the light changing effect and shadow effect can be simply and effectively reduced in the gradient domain. However, this method has two limitations. First, when the texture

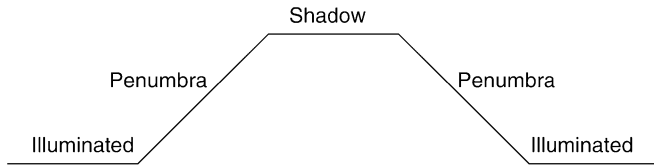


Fig. 10. Irradiance can be separated into three types: illuminated, penumbra, and shadow.

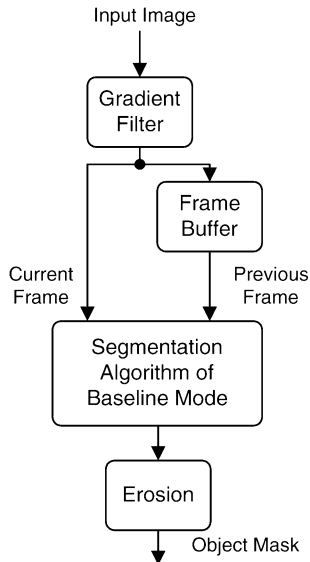


Fig. 11. Block diagram of shadow cancellation mode.

of background regions is complex, this algorithm cannot work well. Second, when the position of foreground objects is very close to that of the background or when parallel light sources are involved, for example, outdoor situations in sunny days, the shadow regions cannot be suppressed.

C. Proposed Shadow Cancellation Algorithm

Based on the analysis, a shadow cancellation algorithm is proposed and is employed in shadow cancellation mode (SC mode). The block diagram of SC mode is shown in Fig. 11. Compared to baseline mode, there are two different parts in SC mode: *Gradient Filter* and *Postprocessing*.

1) *Gradient Filter*: The morphological gradient is chosen because of its simple operations. It can be described in

$$GRA = (I \oplus B) - (I \ominus B) \quad (42)$$

where I is the original image, B is the structuring element of morphological operations, \oplus is morphological dilation operation, \ominus is morphological erosion operation, and GRA is the gradient image. After the gradient operation, the shadow effect will be reduced, and the motion information is still kept.

2) *Postprocessing*: After morphological gradient operation, the edges are thickened. The edge thickening effect is illustrated in Fig. 12. In Fig. 12, compared with ideal edge image, the edge of gradient image is thickened, which may make the segmentation results not accurate at boundaries. To eliminate this effect, a morphological erosion operation should be added at the end of *Post-Processing*.

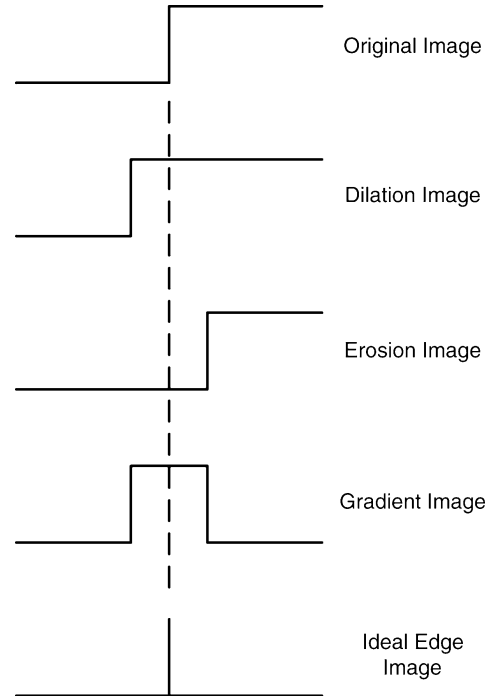


Fig. 12. Edge thickening effect of morphological gradient filter.

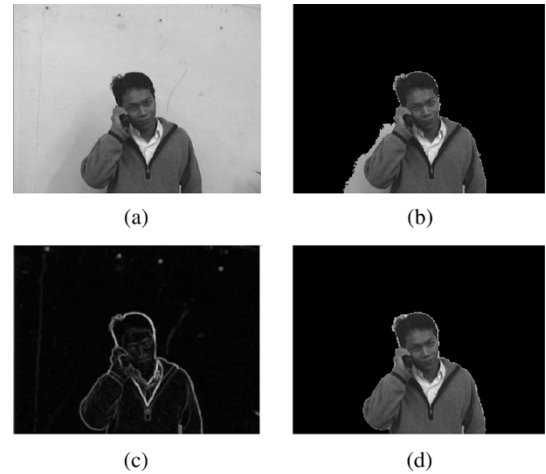


Fig. 13. Illustration of the effect of gradient filter. (a) Original frame; (b) segmentation result of baseline mode; (c) gradient image; (d) segmentation result of SC mode.

An example of SC mode is shown in Fig. 13. It is obvious that after gradient operation, the shadow in Fig. 13(a) is depressed as shown in Fig. 13(c). The segmentation result can be improved from Fig. 13(b)–(d).

V. GLOBAL MOTION COMPENSATION MODE

The same as the algorithms based on conventional change detection, the baseline mode and SC mode are designed for ideal still camera situations. That is, no motion of the camera is allowed. Even a little motion will ruin the segmentation results. That means the camera may need to be set with a tripod and cannot be held by hands; however, this is not the case for most of the real situations. A global motion compensation mode (GMC mode) is designed for slight moving camera situations.

Many global motion estimation algorithms have been proposed. These algorithms are based on the motion model of two (translation model), four (isotropic model), six (affine model), eight (perspective model), or 12 parameters (parabolic model). They can be classified into three types: frame matching, differential technique, and feature points based algorithms [18]. Frame matching algorithm matches the whole frame with the candidate motion parameters to find the global motion vector [19], [20]. The motion parameters can be derived accurately; however, the computation complexity is enormous. The differential method employ Taylor series to expand a criterion function into polynomial equations, such as frame difference of adjacent frames [21], [22]. Then the motion parameters can be derived with linear regression and iteration procedures. This kind of algorithms is more feasible for higher order motion models; however, the computation complexity is also large. The feature points based algorithms [23] first find the motion vectors of the feature points. The global motion vector can then be derived with regression. The computation complexity of this kind of algorithms is much smaller than those of the former two kinds, but the motion vectors are not as accurate as them. All these algorithms have too large computational intensity for real-time systems when higher order motion models are considered. On the other hand, many video segmentation algorithms have integrated GMC [8], [10]. However, the performance of GMC is not perfect, which will introduce errors, especially for change detection based algorithms [24].

To avoid high computation complexity, a fast GMC algorithm should consider only two or four motion parameters, and feature points based algorithms are more suitable. For slight moving camera situations, such as hand-held camera conditions, translation model is sufficient to describe the global motion, and rotation, zoom-in, and zoom-out operations are not considered here. Furthermore, the computation complexity can be further reduced with block based operations rather than pixel based operations. Note that, it is not the same to integrate GMC operation to the proposed background registration based algorithm as to the conventional change detection algorithms. Therefore, a fast GMC algorithm is proposed in this paper with slight motion assumption, and this algorithm is integrated into the proposed video segmentation system.

The block diagram of this algorithm is shown in Fig. 14. There are three parts in GMC: *Feature Blocks Selection*, *Global Motion Estimation*, and *Global Motion Compensation*. The *Background Frame*, *Stationary Index*, and *Background Indicator* are compensated by GMC.

A. Feature Blocks Selection

For the sake of low computation load, block-based operations are adopted. There are two reasons to select some feature blocks to calculate motion vectors rather than to calculate motion vectors of all blocks. First, the motion vector of a block may be different from the real motion. For example, if the texture of a block is insignificant, the motion vector of such block usually cannot tell the real motion. Including these motion vectors in GMC may degrade the correctness. Thus, blocks with higher gradient values are selected as feature blocks, and the gradient information can be obtained in the gradient image, which has

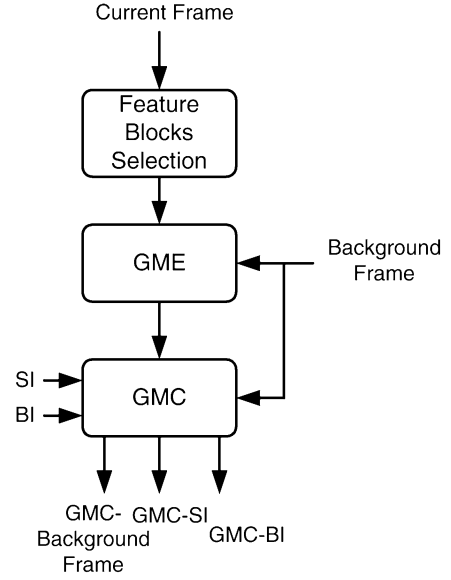


Fig. 14. Block diagram of global motion compensation (GMC).

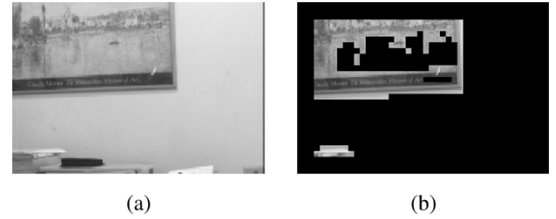


Fig. 15. Feature blocks selection. (a) Original frame; (b) selected feature blocks.

been calculated in SC mode, and no extra computation is required. Second, reducing the number of blocks for motion estimation can also reduce the computation of GMC.

The procedure of feature blocks selection includes two steps. First, blocks near or within foreground objects are excluded. The position of foreground objects can be obtained from previous object mask. Next, accumulate gradient in each block and pick up blocks with higher sum of gradient values as the feature blocks. An example is shown in Fig. 15, in which highly textured blocks are selected.

B. Global Motion Estimation

After *Feature Block Selection*, the motion vector of each feature block between current frame and background frame is calculated. Only global motion model with two parameters are considered here because of the slight motion assumption. Next, the average motion vector of these feature blocks is the global motion vector in this fast algorithm, which can be shown in the following equations:

$$GMV_x = \text{round} \left(\frac{\sum_i MV_x(i)}{\#(\text{feature blocks})} \right) \quad (43)$$

$$GMV_y = \text{round} \left(\frac{\sum_i MV_y(i)}{\#(\text{feature blocks})} \right) \quad (44)$$

where GMV_x and GMV_y are the global motion vectors in horizontal direction and vertical direction, respectively, $MV_x(i)$ and $MV_y(i)$ are the motion vector of the i -th feature block, and $round(\cdot)$ is the rounding function.

C. Global Motion Compensation

After *Global Motion Estimation*, the two global motion parameters are used to compensate the global motion. Unlike the conventional change detection based algorithm, where GMC is applied on the current frame, in the proposed algorithm, background frame, *Stationary Index*, and *Background Indicator* are then compensated. Note that, if GMC is applied on the current frame, the camera position of the current frame and background frame will become further and further, which will reduce the accuracy of GMC and degrade the object masks. The operation of GMC can be expressed by (45)–(47), shown at the bottom of the page, where BG_C , SI_C , and BI_C are compensated background frame, compensated *Stationary Index*, and compensated *Background Indicator*, respectively. Since the global motion parameters are rounded to integer, the computation complexity of this part is very low.

Note that, the compensated background information may be not good enough. The background information can be improved by background updating described in Section III. Besides, the proposed postprocessing algorithm can afford to eliminate the effect of inaccurate GMC.

VI. ADAPTIVE THRESHOLD MODE

The threshold is a very critical parameter for change detection based algorithms. If the optimal threshold cannot be decided automatically, these kinds of video segmentation algorithms are hardly used in real applications. Therefore, the automatic threshold decision is very important in our video segmentation system. Several threshold decision algorithms have been proposed [25], [26]; however, they are not designed for change detection and background registration based video segmentation algorithms, so the performance are not good enough for our algorithm.

The required threshold decision algorithm should have several features. First, it can decide optimal threshold without any other information given by users. Second, since the behavior of background registration based change detection is not the same as conventional change detection algorithm, it should be designed with the consideration of background registration. Moreover, a digitizing effect of digital systems should be also taken into consideration.

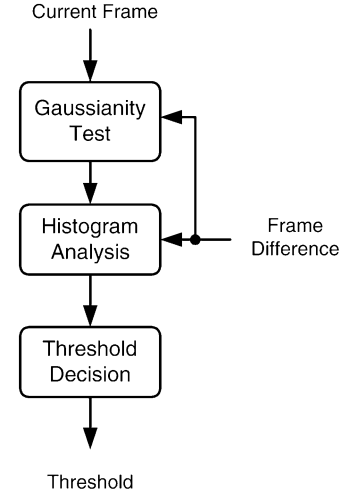


Fig. 16. Block diagram of threshold decision.

To meet these requirements, a threshold decision algorithm is proposed in this section. It is based on the assumption that the camera noise is in zero-mean Gaussian distribution, and the camera noise is the only factor to affect the optimal threshold for background registration. The block diagram of threshold decision is shown in Fig. 16. It includes three parts: *Gaussianity Test*, *Histogram Analysis*, and *Threshold Decision*. In adaptive threshold mode (AT mode), the threshold decision algorithm is executed when environment changes dramatically.

A. Gaussianity Test

Before measuring the parameters for camera noise, background parts of a frame should be indicated automatically because it is very hard to correctly measure the value of camera noise from foreground parts. Since we assume the camera noise is Gaussian distributed, the frame difference of background part should also be Gaussian distributed. Thus, Gaussianity test [27] can be used, which can indicate if a group of values is Gaussian distributed. First, a frame is divided into many blocks. Gaussianity test is then applied in each block to examine if the frame difference in the block is distributed in Gaussian or not. Gaussianity test can be shown as

$$I_k = \frac{I}{M \times N} \sum_{m=1}^M \sum_{n=1}^N FD^k(m, n) \quad (48)$$

$$H(I_1, I_2, I_3, I_4) = I_3 + I_4 - 3I_1(I_2 - I_1^2) - 3I_2^2 - I_1^3 - 2I_1^4. \quad (49)$$

- Gaussian: $|H(I_1, I_2, I_3, I_4)| < Gth$
- Non-Gaussian: $|H(I_1, I_2, I_3, I_4)| \geq Gth$

$$BG_C(x, y) = \begin{cases} BG(x + GMV_x, y + GMV_y), & \text{if } (x + GMV_x, y + GMV_y) \text{ is inside the frame} \\ 0, & \text{else} \end{cases} \quad (45)$$

$$SI_C(x, y) = \begin{cases} SI(x + GMV_x, y + GMV_y), & \text{if } (x + GMV_x, y + GMV_y) \text{ is inside the frame} \\ 0, & \text{else} \end{cases} \quad (46)$$

$$BI_C(x, y) = \begin{cases} BI(x + GMV_x, y + GMV_y), & \text{if } (x + GMV_x, y + GMV_y) \text{ is inside the frame} \\ 0, & \text{else} \end{cases} \quad (47)$$

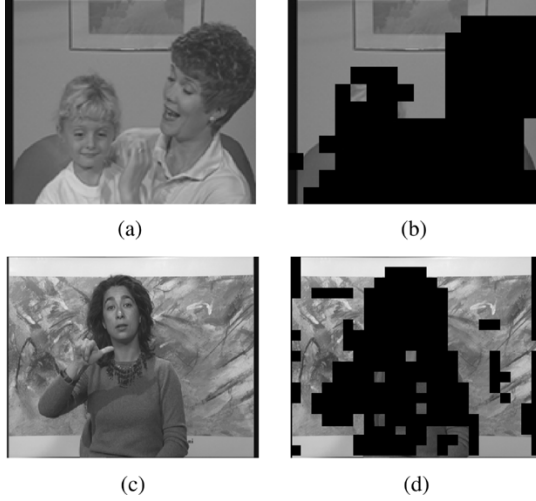


Fig. 17. Illustration of Gaussianity test. (a) *Mother and Daughter #50*; (b) result of Gaussianity test; (c) *Silent Voice #50*; (d) result of Gaussianity test.

If the frame difference in a block is distributed in Gaussian, the block should belong to background region.

Examples of Gaussianity test are shown in Fig. 17. Only blocks whose frame difference is Gaussian distributed are shown in Fig. 17(b) and (d). It shows that Gaussianity test can roughly distinguish background parts and foreground parts. The parameter Gth can be set as a constant because the H values of foreground blocks and background blocks are dramatically different, and Gth can be fixed to "1." Note that although Gaussianity test can already distinguish between background and foreground blocks, it cannot be used for background registration because it can only give rough object mask. Some background blocks may be included in the foreground part, and some foreground blocks may be included in the background part in the rough object mask. The function of Gaussianity test here is to decide the optimal threshold without any information given by users and makes the threshold decision procedure independent to the segmentation procedure to avoid error propagation.

B. Histogram Analysis and Threshold Decision

The optimal threshold Th for background registration with parameter Fth in (4) can be expressed as:

$$Th = \max(FD_1, FD_2, \dots, FD_{Fth}) \quad (50)$$

where FD is a random variable of frame difference in background regions, which is distributed in Gaussian, and Th is a random variable of optimal threshold.

Since the frame difference is distributed in Gaussian, and the digitizing effect of digital systems is considered, the distri-

bution of the absolute value of frame difference in digital domain should be (51), shown at the bottom of the page, where $P_k = P(FD = k)$. The expected value of the optimal threshold should be

$$\begin{aligned} E(Th) &= \sum_{k=0}^{255} k P(Th = k) \\ &= \sum_{k=0}^{255} k \left[\binom{Fth}{1} P(FD = k) P(FD \leq k)^{Fth-1} \right] \\ &= \sum_{k=0}^{255} k \left[\binom{Fth}{1} P_k \left(\sum_{m=0}^k P_m \right)^{Fth-1} \right] \\ &= \sum_{k=0}^{255} k \left\{ (Fth) P_k \left[\sum_{m=0}^k P_m \right]^{Fth-1} \right\}. \end{aligned} \quad (52)$$

Nevertheless, the parameter σ is hard to derived from frame difference information, since the frame difference is an integer in digital systems, the standard deviation of frame difference is different from σ . Another parameter σ^{new} is calculated instead:

$$\sigma^{new} = \sqrt{\sum_{\text{pixels} \in \text{background blocks}} FD^2(m, n)} \quad (53)$$

where FD is the absolute value of frame difference. The parameter σ^{new} can also calculated from (51):

$$\sigma^{new} = \sqrt{\sum_{k=0}^{255} k^2 P_k}. \quad (54)$$

Note that the mean of random variable FD is zero.

By substituting σ with different values, the expected optimal threshold $E(Th)$ can be derived by (51) and (52), and the associated σ^{new} can be derived by (53). Then the threshold decision curve can be drawn by this data as shown in Fig. 18. Finally, in AT mode, *Histogram Analysis* uses the frame difference values of all background blocks indicated by Gaussianity test to calculate the value of σ^{new} by (53), and *Threshold Decision* decides the optimal threshold by the $\sigma^{new} - E(Th)$ threshold decision curve in Fig. 18.

VII. IMPLEMENTATION

The proposed algorithm can be optimized with parallel computing concept. In this section, the optimization of baseline mode is described.

SIMD instructions are included in almost every microprocessor and DSP of personal computers, cellular phones, PDA's, and other IA's. Therefore, *Frame Difference*, *Background*

$$P_k = \begin{cases} \int_{-0.5}^{0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} dx = \int_0^{0.5} \frac{2}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} dx, & \text{for } k = 0 \\ \int_{k-0.5}^{k+0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} dx + \int_{-k-0.5}^{-k+0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} dx = \int_{k-0.5}^{k+0.5} \frac{2}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} dx, & \text{for } k = 1, 2, \dots, 255 \end{cases} \quad (51)$$

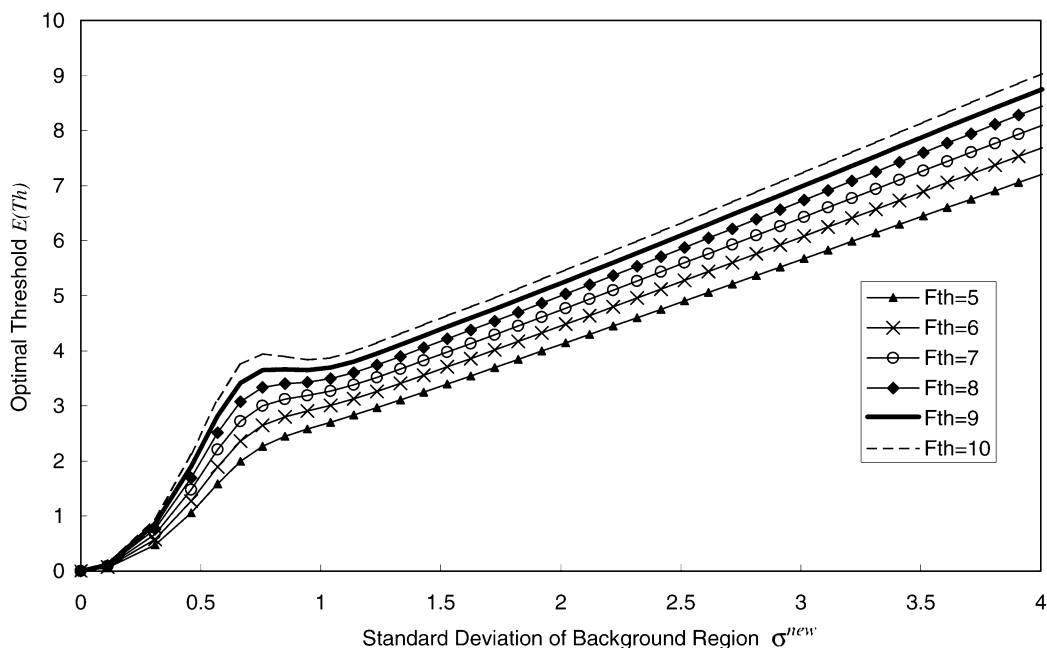


Fig. 18. Threshold decision curve.

Registration, Background Difference, and Object Detection are optimized based on SIMD platform. Intel Pentium processors with MMX instructions [28] are chosen as the target platform since they are the most widespread processors with SIMD architecture. Besides, since the datapath word length of modern processors is 32 bits or longer, a fast binary morphological operations based on pixel parallel technique [13] can also speed up the calculation.

A. Implementation With MMX Technology

Frame Difference, Background Registration, Background Difference, and Object Detection are optimized by Intel MMX instructions [28] because most of these operations can be executed in parallel. One example is shown in Fig. 19, where the frame difference and thresholding operation are implemented by MMX instructions. The MMX applies SIMD technique so four 16-bit data can be manipulated at the same time. Note that the frame difference operation is implemented by combining four MMX instructions: PSUB, PCOMGTW, PXOR, and PAND. PUNPCKLBW unpacks four 8-bit data into four 16-bit data. PSUB, PCOMGTW, PXOR, and PAND calculate the frame difference of these four pixels at the same time. PCMPGTW compares the difference with a preset threshold and generate CDM, which is “255” if the difference is larger or equal than the threshold and is “0” otherwise. Finally, PACKSSWB packs the four 16-bits data into four 8-bits data and writes them to the memory. All the processes need only seven instructions to deal with four pixels simultaneously.

B. Fast Binary Morphological Operations

The most computationally intensive operation in our algorithm is the binary morphological operations in *Postprocessing*. Since most microprocessors have 32-bit or 64-bit datapaths, it is not efficient to store and process a pixel value in byte or word.

Therefore, a 32-bit word is used to present the binary values of 32 pixels so the morphological operations for these 32 pixels are computed in parallel [13]. The pseudo code of such an implementation is shown in Algorithm 1.

Algorithm 1:

```

Efficient Implementation for Binary
Morphological Operation
Dilate(I, width, height)
{
  PackBinaryImage();
  //I: input image; T: frame buffer
  //row operation
  for (y = 0; y < height; y++)
    for (x = 0; x < width/32; x++)
      T[x][y] = I[x][y]
        | (I[x][y] << 1 + I[x + 1][y] >> 31)
        | (I[x][y] >> 1 + I[x - 1][y] << 31);
  //column operation
  for (y = 0; y < height; y++)
    for (x = 0; x < width/32; x++)
      I[x][y] = T[x][y] | T[x][y - 1] | T[x][y + 1];
  UnPackBinaryImage();
}

```

VIII. EXPERIMENTAL RESULTS

A. Proposed Algorithm

The performance of the proposed video segmentation algorithm is tested with many video sequences. The same as other previous works, the quality of segmentation results is evaluated subjectively. A fair evaluation method named “fix frame number test” is used here. That is, the segmentation results of fixed frames of each sequence are picked, rather than chosen by

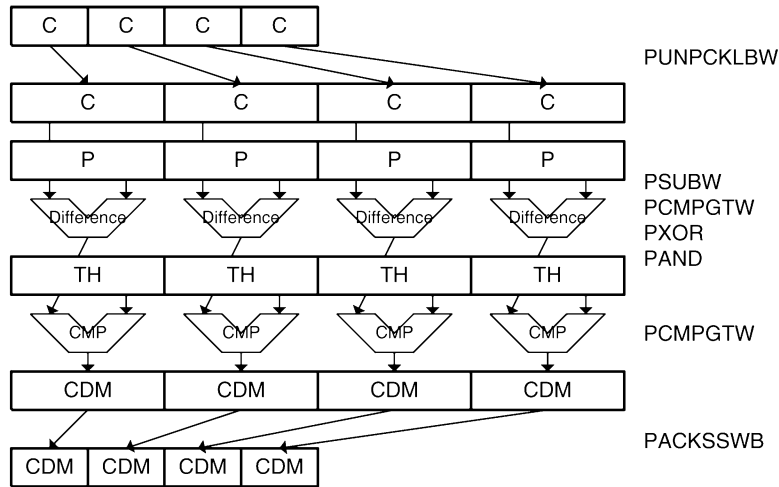


Fig. 19. Implementation with MMX technique.

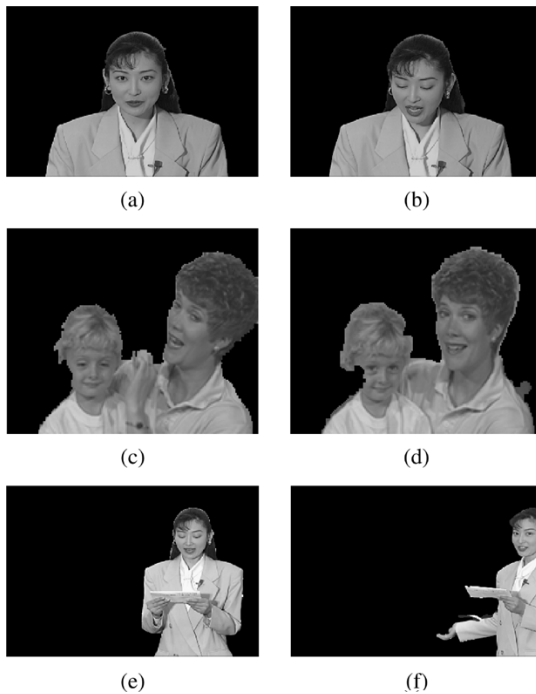


Fig. 20. Segmentation results of baseline mode. (a) *Akiyo* #50; (b) *Akiyo* #100; (c) *Mother and Daughter* #50; (d) *Mother and Daughter* #100; (e) *Weather* #50; (f) *Weather* #100.

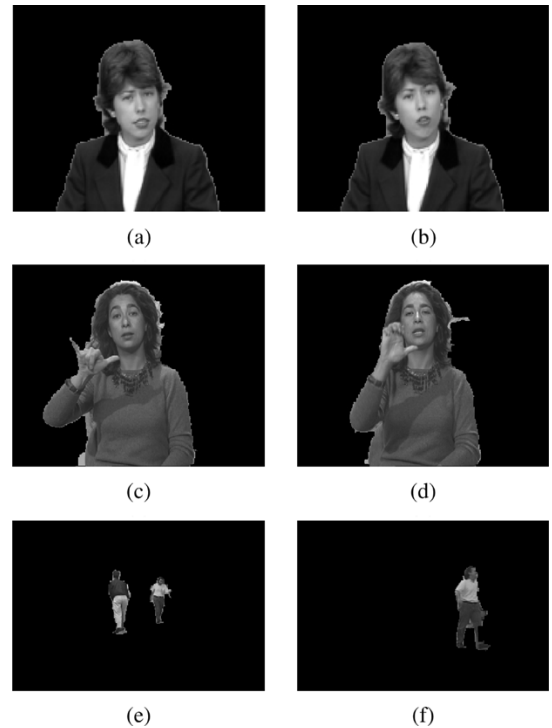


Fig. 21. Segmentation results of SC mode. (a) *Claire* #50; (b) *Claire* #100; (c) *Silent Voice* #50; (d) *Silent Voice* #100; (e) *Hall Monitor* #50; (f) *Hall Monitor* #100.

the algorithm developers themselves. The quality of segmentation results is then evaluated subjectively. In these experiments, frame 50 and frame 100 are chosen in every sequence.

1) *Baseline Mode*: The segmentation results of sequence *Akiyo*, *Mother and Daughter*, and *Weather* are shown in Fig. 20. These sequences are not influenced by shadow and light changing effects. The segmentation results of sequence *Akiyo* and *Mother and Daughter* are shown in Fig. 20(a)–(d) respectively, where the still object problem can be correctly solved. In Fig. 20(e) and (f), the sequence *Weather*, in which the foreground object has large motion, can also be correctly segmented. Note that in Fig. 20(f), some segmentation errors occur near the hand of the reporter. That is because the background information of that region is not yet available.

2) *Shadow Cancellation Mode*: The experimental results of shadow cancellation mode are presented in Figs. 21 and 22. The sequence *Claire*, which is influenced by light changing effect, can be correctly manipulated in SC mode, as shown in Fig. 21(a) and (b). In Fig. 21(c) and (d), the segmentation results of sequence *Silent Voice* are shown. It shows that the shadow effect in *Silent Voice* can be reduced, and good object mask can be generated. The segmentation results of sequence *Hall Monitor* are presented in Fig. 21(e) and (f). Both of the light changing and shadow effects occur in this sequence. The experimental results show that this sequence can also be correctly segmented in SC mode. The shadows of the foreground objects can be cancelled.

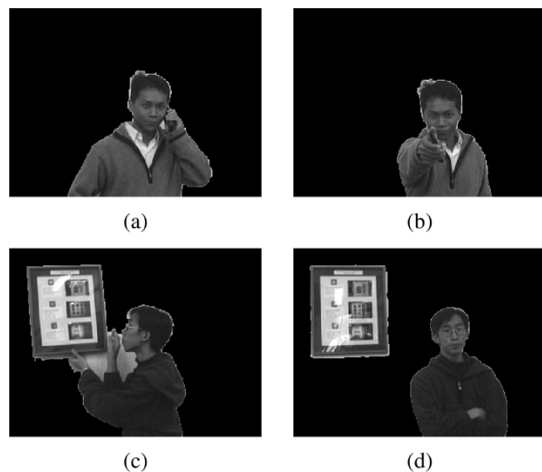


Fig. 22. Segmentation results of SC mode with sequences taken by a general camera. (a) Frank #50; (b) Frank #100; (c) ShaoYi #50; (d) Shao-Yi #100.

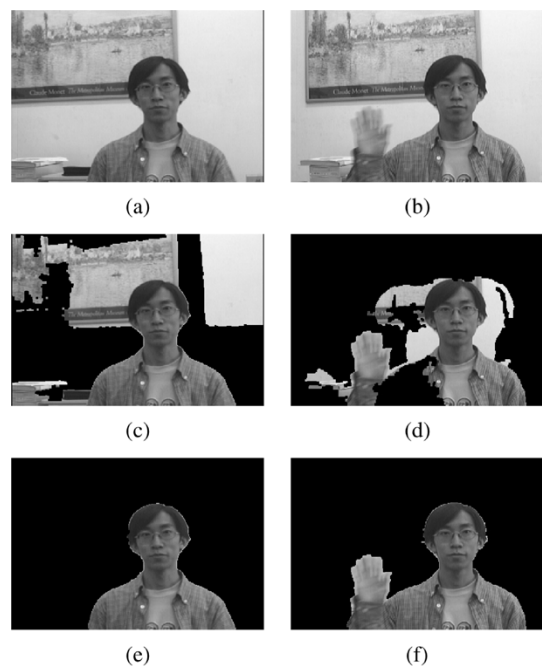


Fig. 23. Segmentation results of GMC mode. (a) ShaoYi Under Moving Camera #120; (b) ShaoYi Under Moving Camera #200; (c) segmentation result of #120 without GMC; (d) segmentation result of #200 without GMC; (e) segmentation result of #120 with GMC; (f) segmentation result of #200 with GMC.

Note that it is also shown that the proposed algorithm can deal with multiple video objects situations.

Some video sequences captured with a general camera are also used as test sequences. The segmentation results of them are shown in Fig. 22. The light sources are fluorescent lamps, and shadows exist in these sequences. The experimental results show that this kind of sequences can also be well segmented. It also means this video segmentation algorithm can be applied in real applications.

3) *Global Motion Compensation Mode*: The experimental results of GMC mode is shown in Fig. 23. In Fig. 23(a) and (b), it can be observed from the background that the camera slightly moves. When baseline mode is applied, the segmentation results are shown in Fig. 23(c) and (d). The segmentation results

are poor, which proves that change detection based algorithms, including the proposed change detection and background registration based algorithm, are easily suffered from moving camera situations. In GMC mode, the results are shown in Fig. 23(e) and (f). The background information are correctly compensated, and good segmentation results are generated.

4) *Adaptive Threshold Mode*: The experimental results of AT mode is shown in all the figures mentioned above because the threshold Th in all experiments is decided by the automatic threshold decision algorithm. The segmentation results show the proposed threshold decision algorithm is suitable for change detection and background registration based video segmentation algorithms.

B. Efficient Implementation

The run-time analysis of the proposed algorithm is shown in Fig. 24. The execution time of direct implementation (baseline mode), optimized implementation (baseline mode), SC mode, and GMC mode are shown. The test platform is a personal computer with a Pentium-III 800 MHz processor, and the test sequences are in QCIF format. After optimized, *Object Detection*, *Frame Difference*, and *Background Registration* can be accelerated with MMX technology, and the binary morphological open-close operations are dramatically accelerated with bit-parallel technique. The processing speed of 40 QCIF frames per second can be achieved in baseline mode, which can achieve real-time requirement. In SC mode, the morphological gradient operation is added, the processing speed is slowed down to 21 QCIF frames/s. In GMC mode, the computation load of motion estimation is large, and the processing speed is 10 QCIF frames/s. Note that *Threshold Decision* is not include in the analysis, since AT mode is turned on only when environment changes or when the camera is just turned on.

Note that the efficient implementation of SC mode and GMC mode are not included here, which is one of our future works.

IX. CONCLUSION

A complete fast video segmentation algorithm is proposed in this paper. It contains four modes: baseline mode, shadow cancellation mode, global motion compensation mode, and adaptive threshold mode. There are six contributions of this work. First, we propose a background registration and change detection based video segmentation algorithm, which can generate satisfying segmentation results with low computation complexity. Second, an effective and simple shadow cancellation algorithm is developed according to the analyzes of light changing and shadow effects in indoor environments. Third, a fast GMC algorithm is proposed and integrated into the background registration based video segmentation algorithm with the shadow cancellation algorithm. Moreover, the probability model of this algorithm is derived, the optimal threshold can be decided according to this model, and Gaussinity test is applied here to interrupt the error propagation. In addition, the efficient implementation method is also developed to make this algorithm achieve real-time requirement on a PC. Finally, the most important contribution is to integrate background

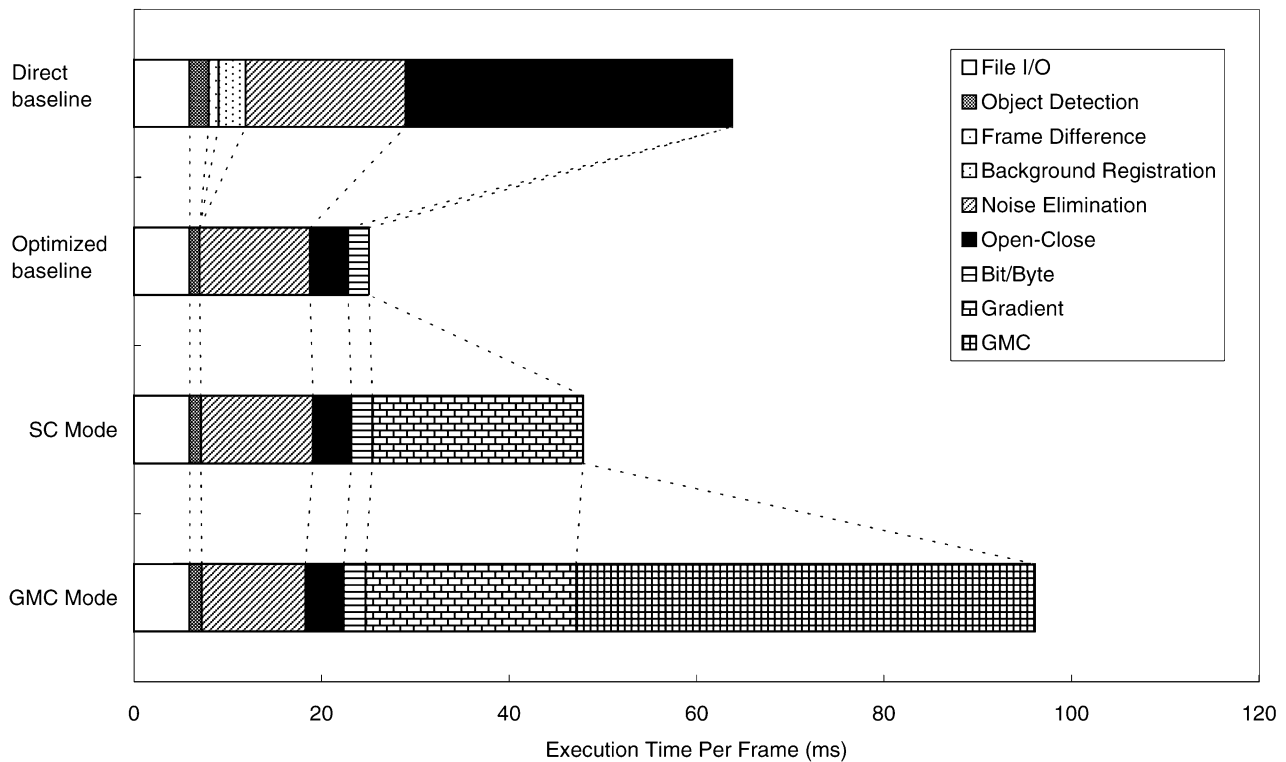


Fig. 24. Run-time analysis of the proposed algorithm.

registration based video segmentation algorithm, shadow cancellation technique, global motion compensation technique, and adaptive threshold technique into a complete video segmentation system. Experiments show that this algorithm can give good segmentation results in general situations and can achieve real-time requirement in baseline mode on a PC with a Pentium-III 800 MHz processor.

There are still some limitations in the proposed segmentation system. The shadow cancellation cannot deal with parallel and strong light sources and may cause errors when the texture of background is significant. The GMC algorithm cannot deal with zoom-in, zoom-out, and rotation, and only slight motion can be well handled with the proposed fast algorithm. Furthermore, the decision to turn on and off each mode of the proposed algorithm is not automatic. It is adjusted manually by users according to the shoot situations. Finally, this algorithm is designed for moving objects segmentation; therefore, for stable and accurate results, the foreground object should not be still for a long time, and the background should never move.

Video segmentation is a key technology for content based video coding, representation, indexing, and retrieval. With integrating this algorithm into digital camera systems, the real-time content based video processing becomes feasible, and intelligent video processing applications can be developed on this platform, which are included in our future works.

REFERENCES

- [1] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 19–31, Feb. 1997.
- [2] *The MPEG-4 Video Standard Verification Model ver. 18.0*, ISO/IEC JTC 1/SC 29/WG11 N3908, 2001.
- [3] *Annex F: Preprocessing and Postprocessing*, ISO/IEC JTC 1/SC 29/WG11 N4350, 2001.
- [4] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 525–538, Dec. 1998.
- [5] —, "Video segmentation for content-based coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1190–1203, Dec. 1999.
- [6] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 539–546, Sept. 1998.
- [7] M. Kim, J. G. Choi, H. Lee, D. Kim, M. H. Lee, C. Ahn, and Y.-S. Ho, "A VOP generation tool: Automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1216–1226, Dec. 1999.
- [8] J. Guo, J. Kim, and C.-C. J. Kuo, "Fast video object segmentation using affine motion and gradient-based color clustering," in *Proc. IEEE Workshop on Multimedia Signal Processing*, 1998.
- [9] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal segmentation and tracking of objects for visualization of videoconference image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1388–1402, Dec. 2000.
- [10] R. Mech and M. Wollborn, "A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera," *Signal Process.*, vol. 66, 1998.
- [11] S.-Y. Ma, S.-Y. Chien, and L.-G. Chen, "An efficient moving object segmentation algorithm for MPEG-4 encoding systems," in *Proc. Int. Symp. Intelligent Signal Processing and Communication Systems 2000*, 2000.
- [12] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "An efficient video segmentation algorithm for real-time MPEG-4 camera system," in *Proc. Visual Communication and Image Processing 2000*, 2000, pp. 1087–1098.
- [13] R. V. D. Boomgaard and R. V. Balen, "Methods for fast morphological image transforms using bitmapped binary images," *CVGIP: Graph. Models Image Process.*, vol. 54, 1992.
- [14] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*. Reading, MA: Addison-Wesley, 1992.
- [15] J. Serra, *Image Analysis and Mathematical Morphology*. London, U.K.: Academic, 1982.
- [16] J. Stauder, R. Mech, and J. Ostermann, "Detection of moving cast shadows for object segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, pp. 65–76, Mar. 1999.

- [17] J. D. Foley, A. v. Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips, *Introduction to Computer Graphics*. Reading, MA: Addison-Wesley, 1994.
- [18] F. Moscheni, F. Dufaux, and M. Kunt, "A new two-stage global/local motion estimation based on a background/foreground segmentation," in *Proc. of ICASSP*, 1995, pp. 2261–2264.
- [19] D. Adolph and R. Buschmann, "1.15 Mbit/s coding of video signals including global motion compensation," *Signal Process.: Image Commun.*, vol. 3, no. 2, 1991.
- [20] H. Nicolas, "New methods for dynamic mosaicking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1239–1251, Aug. 2001.
- [21] M. Hoetter, "Differential estimation of the global motion parameters zoom and pan," *Signal Process.*, vol. 16, 1989.
- [22] S. F. Wu and J. Kittler, "A differential method for simultaneous estimation of rotation, change of scale and translation," *Signal Process.: Image Commun.*, vol. 2, no. 1, pp. 69–80, May 1990.
- [23] A. Smolic, T. Sikora, and J.-R. Ohm, "Long-term global motion estimation and its application for sprite coding, content description, and segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1227–1242, Dec. 1999.
- [24] Y.-W. Huang, S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Analysis of global motion effects on video segmentation," in *Proc. 2000 Asia Pacific Conf. Multimedia Technology and Applications*, 2000.
- [25] N. Habibi, A. Moini, and N. Burgess, "Automatic thresholding for change detection in digital video," in *Proc. Visual Communication and Image Processing 2000*, May 2000, pp. 133–142.
- [26] D.-J. Kim, J.-S. Cho, and D.-J. Part, "Fast computation of the Gaussian mixture parameters and optimal segmentation," in *Proc. Visual Communication and Image Processing 2000*, 2000, pp. 1087–1098.
- [27] M. N. Gurcan, Y. Yardimci, and A. E. Cetin, "Influence function based gaussianity tests for detection of microcalcifications in mammogram images," in *Proc. International Conference on Image Processing 1999*, vol. 3, Oct. 1999, pp. 407–411.
- [28] A. Peleg and U. Weiser, "MMX technology extension to the Intel architecture," *IEEE Micro*, vol. 16, pp. 42–50, Aug. 1996.



Shao-Yi Chien was born in Taipei, Taiwan, R.O.C., in 1977. He received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, in 1999 and 2003, respectively.

During 2003 to 2004, he was a Research Staff member with Quanta Research Institute, Tao Yuan Shien, Taiwan. In 2004, he joined the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, NTU, as an Assistant Professor. His research interests include video segmentation algorithm, intelligent video coding technology, and associated VLSI architectures.



Yu-Wen Huang was born in Kaohsiung, Taiwan, R.O.C., in 1978. He received the B.S. degree in electrical engineering from National Taiwan University (NTU), Taipei, in 2000, where he is currently pursuing the Ph.D. degree in the Graduate Institute of Electronics Engineering. His research interests include video segmentation, moving object detection and tracking, intelligent video coding technology, motion estimation, face detection and recognition, H.264/AVC video coding, and associated VLSI architectures.



Bing-Yu Hsieh was born in Taichung, Taiwan, R.O.C., in 1979. He received the B.S.E.E. and M.S.E.E. degrees from National Taiwan University (NTU), Taipei, in 2001, and 2003, respectively. He joined MediaTek, Inc., Hsinchu, Taiwan, in 2003, where he develops integrated circuits related to multimedia systems and optical storage devices. His research interests include object tracking, video coding, baseband signal processing, and VLSI design.



Shyh-Yih Ma received the B.S.E.E., M.S.E.E., and Ph.D. degrees from National Taiwan University, Taipei, Taiwan, R.O.C., in 1992, 1994, and 2001, respectively.

He joined Vivotek, Inc., Taipei County, in 2000, where he developed multimedia communication systems on DSPs. His research interests include video processing algorithm design, algorithm optimization for DSP architecture, and embedded system design.



Liang-Gee Chen (S'84–M'86–SM'94–F'01) was born in Yun-Lin, Taiwan, R.O.C., in 1956. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 1979, 1981, and 1986, respectively.

He was an Instructor (1981–1986), and an Associate Professor (1986–1988) in the the Department of Electrical Engineering, NCKU. In the military service during 1987 and 1988, he was an Associate Professor in the Institute of Resource Management, Defense Management College. In 1988, he joined the Department of Electrical Engineering, National Taiwan University (NTU), Taipei. During 1993 to 1994, he was Visiting Consultant with DSP Research Department, AT&T Bell Labs, Murray Hill, NJ. In 1997, he was the Visiting Scholar, Department of Electrical Engineering, University of Washington, Seattle. Currently, he is Professor with NTU. His current research interests are DSP architecture design, video processor design, and video coding system. He is the Associate Editor of the *Journal of Circuits, Systems, and Signal Processing* since 1999. He served as the Guest Editor of the *Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, November 2001.

Dr. Chen is a member of Phi Tau Phi. He was the general chairman of the 7th VLSI Design/CAD Symposium. He is also the general chairman of the 1999 IEEE Workshop on Signal Processing Systems: Design and Implementation. He serves as Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY since June 1996, and Associate Editor of the IEEE TRANSACTIONS ON VLSI SYSTEMS since January 1999. He is also the Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING. He received the Best Paper Award from the R.O.C. Computer Society in 1990 and 1994. From 1991 to 1999, he received Long-Term (Acer) Paper Awards annually. In 1992, he received the Best Paper Award of the 1992 Asia-Pacific Conference on Circuits and Systems in VLSI design track. In 1993, he received the Annual Paper Award of Chinese Engineer Society. In 1996, he received the Outstanding Research Award from NSC, and the Dragon Excellence Award for Acer. He was elected as the IEEE Circuits and Systems Society Distinguished Lecturer in 2001–2002.