On the Construction of a Color Model for Real-time Visual Moving Objects Segmentation

Chin-Hwa Kuo*, Ping-Huang Wu and Tay-Shen Wang

Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C.

Abstract

A color model, called RGB-Ellipse, is developed in this paper. The proposed color model takes advantages of CIE-L*a*b*94 and CMC(*l*:*c*) in determining color differences. Furthermore, the transformation between RGB and RGB-Ellipse is linear. As a result, we are able to manipulate the noise tolerance processing as well as the computation efficiency in dealing with color differences. By using the above features, we develop a real-time moving objects segmentation scheme. The developed segmentation scheme consists of two main steps: (1) region seed determination and region growing and (2) region-based change detection and background update. The results from a visual surveillance system are given to highlight the value of the proposed color model and moving objects segmentation scheme.

***Key Words*:** Image Segmentation, Color Model, Surveillance System

1. Introduction

Visual objects segmentation has been a field of research for several decades. Many image and video applications can benefit from using the segmentation technique. At present, due to the improvements in computing power and the Internet, research in this field has become even more active. In this paper, we propose a color model, called RGB-Ellipse. We make use of the proposed color model as the fundamental basis in computing the color differences in the design of moving objects segmentation schemes. Results in a visual surveillance system are given to highlight the value of the proposed color model and moving objects segmentation scheme.

In the still image case, the existing object segmentation schemes can be classified into two categories: (1) boundary-based segmentation and (2) region-based segmentation. The discontinuities of color and/or luminance signals are used in the boundary-based edge detection algorithm to allocate the position of edge [1–3]. In the above case the design of filters affects the quality of the resulting edge detection. In the region-based segmentation techniques, the neighborhood pixels with similar color or luminance are collected to form a region. This idea was further extended in [4,5], using not only color but also texture information. In the two approaches above, the boundary-based segmentation scheme may fail to constitute a contour due to the smooth transition characteristic of natural images. In other words, there are pixels that cannot be classified into any particular region. This problem does not occur in region-based segmentation schemes. Both boundary-based and region-based segmentation schemes may divide the same object into different regions. In addition, different objects are allocated in the same region. Thus, a post-processing scheme is usually required to deal with these issues.

In the continuous images case, object motion provides additional dimension information for object segmentation. In MPEG-1 and H.26x, motion estimation schemes are applied to identify the motion vectors. The coherent motion blocks are viewed as the same object

*Corresponding author. E-mail: chkuo@mail.tku.edu.tw

[6–8]. However, in the case of non-rigid objects, the block motion vector is usually not consistent with object motion, and the object boundary cannot be obtained accurately. The above approach may not yield a satisfactory result. In addition to the block motion vector, the contour obtained in the edge detection process can also be used as a feature for motion estimation [9,10]. The computational result of object contour depends on the boundary obtained in the computation process and the contour mapping in the temporal domain. Furthermore, the fragmented boundary curves need to be resolved to reduce the complexity in dealing with contour. These processes require high computational intensity. Therefore, the above approach is usually used in non-real time applications. In [11], Deng and Manjunath utilize hierarchical concepts in dealing with the texture in image segmentation. After several times region merges/splits, a satisfactory result may be obtained with a considerable amount of computational work. The temporal information is not taken into account in their work.

Background subtraction is another common continuous images segmentation technique. It detects and extracts the different portions of a new image and a background image based on the differences of the comparison unit. According to the comparison unit, the corresponding change detections are classified into pixel-based [3, 12–15], block-based [3], and region-based approaches [3]. The pixel-based change detection scheme is sensitive to noise. As a result, ambiguity of the foreground and background occurs frequently. Block-based schemes are not suitable to deal with objects with arbitrary shapes.

A region-based approach may overcome this difficulty. The regions obtained from edge detection are used as a comparison unit. Therefore, it has the advantage of extracting arbitrary shapes in an image. Both block-based and region-based change detection schemes outperform pixel-based change detection in dealing with noise. In [16], the concept of change detection is used in the object segmentation in the case of stationary background video. By comparing the differences between a new image and a background image, the areas that are subject to changes are detected. A post processing for shadow elimination is also included in their work. Furthermore, code optimization is taken into account to achieve a real time 30 fps in CIF video. Integration of boundary-based and region-based change detection schemes is proposed in [17]. It makes use of the characteristics of image boundary to allocate the possible origins of objects. Followed by region growing schemes, objects are identified. In [18], the traditional Lab color space model is used as color metric. A preprocessing scheme for reducing the complexity of still images is developed. As a result, the advantage of this preprocessing is shown in the region-based image segmentation. A pixel-based approach to compute the changes in temporal domain based on the Gaussian model is shown in [19]. This method gives good results if the contaminated noise is within a certain level.

Briefly speaking, a good quality video segmentation scheme requires significant computational effort at the present time. This often applies in non-real time applications. In the case of real time applications, we may resort to use low resolution video formats, reduce processing quality requirements, or only consider gray level images. For instance, in the digital surveillance system, the above treatments are common [11].

The proposed moving objects segmentation scheme for a surveillance system consists of two main steps: (1) *region seed determination and region growing*, (2) *region-based change detection*. Note that it is computational intense to compute the regions of a given frame. In our approach, we make use of the temporal information to determine the seed of a region to ease the computation loading. Then, a region growing scheme is applied to allocate the corresponding region. Each region has a representative color. The change detection is executed by comparing the color change in a region with the background frame in the same geometric location. Note also that in order to resolve the weakness of the Euclidean distance in determining the color difference, we construct a new color representation model, called RGB-Ellipse. The proposed RGB-Ellipse color model is the foundation to compute the color difference of two regions. The proposed color model is similar to the HSI representation. However, the transformation between RGB and RGB-Ellipse color spaces is linear. We are able to take advantage of noise tolerance processing as well as the efficiency in dealing with color difference computation. By using the proposed segmentation scheme, we implemented a visual networked surveillance system.

The remaining parts of this paper are organized as follows. The proposed RGB-Ellipse color model is presented in Section 2. We describe the color difference computation scheme and indicate the strength of this model. Region based moving object segmentation is shown in Section 3. We show the corresponding design steps in detail. Application and result that make use of the proposed approaches are illustrated in Section 4. Conclusions are given in Section 5.

2. The RGB-Ellipse Color Model Overview

Determining the color difference between two colors is not an easy issue. Both the Colour Measurement Committee of the Society of Dyers Colorists (CMC) and the International Committee on Illumination (CIE) organizations have proposed schemes, CMC(*l*:*c*) and CIE-L*a*b*94 respectively, to measure color difference. These models make use of weighted Lightness, Chroma and Hue as fundamental components to compute the color difference of two colors. The difference between CIE-L*a*b*94 and CMC(*l*:*c*) depends on the way that ellipsoid dimensions are calculated. A large amount of experimental tests are performed to obtain the results of human visual response in both CMC(*l*:*c*) and CIE-L*a*b*94. However, the determination of color difference based on the above models requires intensive computational effort. Note that the output signals of common CCD cameras are machine dependent. The output signals are usually contaminated by noise. The computation of color difference directly based on these signals may lead to ambiguous results. In this section, we introduce the *RGB-Ellipse color model* as the basis for the development of moving objects segmentation scheme. The proposed model make use of the human perceptual cognition idea from the CIE-L*a*b*94 and CMC(*l*:*c*) in dealing with color difference. It is a linear transformation between RGB and RGB-Ellipse models. The computational effort is significantly reduced. As a result, the RGB-Ellipse color model provides us with a tool to deal with the issues of shadow elimination and noise resistance in the image segmentation and the change detection processes.

In this section, we briefly describe the CMC(*l*:*c*) and CIE-L*a*b*94 models. Then, we introduce the proposed RGB-Ellipse model. In the last two subsections, we illustrate the concepts of shadow elimination and noise resistance in the RGB-Ellipse model.

2.1 CMC(*l*:*c*)

The CMC(*l*:*c*) color model was standardized by CMC in 1988. The color difference in this model is computed based on CIE-LCH color space, which makes use of lightness (L), chroma (C), and hue (H) as basis. As we mentioned, the output signals of common CCD are machine dependent. However, CMC(*l*:*c*) transfers the representation from a machine dependent domain into a machine independent domain. Under such a coordinate, we obtain a more precise color difference comparison. The transformation from RGB to CIE-LCH consists of three steps, as follows. First, the color representation transfers from RGB to XYZitu by Eq. (1), from XYZitu to CIE-L*a*b*94 by Eq. (2), and then from CIE-L*a*b*94 to CIE-LCH by Eq. (3).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.430574 & 0.341550 & 0.178325 \\ 0.222015 & 0.706655 & 0.071330 \\ 0.020183 & 0.129553 & 0.939180 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

$$\begin{cases} L^* = \begin{cases} 116 \times \sqrt[3]{Y/Y_n} & if \quad Y/Y_n > 0.008856 \\ 903.3 \times Y/Y_n & otherwise \end{cases} \\ a^* = 500 \times \left[f\left(\sqrt[3]{X/X_n}\right) - f\left(\sqrt[3]{Y/Y_n}\right) \right] \\ b^* = 200 \times \left[f\left(\sqrt[3]{Y/Y_n}\right) - f\left(\sqrt[3]{Z/Z_n}\right) \right] \\ f(t) = \begin{cases} \sqrt[3]{t} & for \quad t > 0.008856 \\ 7.787 \times t + 16/116 & otherwise \end{cases} \end{cases} \quad (2)$$

$$\begin{cases} L^* = L^* \quad in \quad CIE - L^*a^*b^* \\ C^* = \sqrt{a^{*2} + b^{*2}} \\ H^* = \tan^{-1}(b^*/a^*) \end{cases} \quad (3)$$

By computing the color difference of two points (R_1, G_1, B_1) and (R_2, G_2, B_2), we calculate the above transformation process to obtain (L_1, C_1, H_1) and (L_2, C_2, H_2), respectively. Then, we get the color difference ΔE^* of any

two points by Eq. (4).

$$\begin{cases} \Delta E^* = \sqrt{\left(\dfrac{\Delta L^*}{l \times S_l}\right)^2 + \left(\dfrac{\Delta C^*}{c \times S_c}\right)^2 + \left(\dfrac{\Delta H^*}{S_h}\right)^2} \\ S_l = \begin{cases} 0.511 \ \text{when} \ L^* > 16 \\ \dfrac{0.040975 \times L^*}{1 + 0.01765 \times L^*} \ \text{otherwise} \end{cases} \\ S_c = \dfrac{0.0638 \times C^*}{1 + 0.0131 \times C^*} + 0.638 \\ S_h = \left(f \times T + 1.0 - f\right) \times S_c \\ f = \sqrt{\dfrac{C^{*4}}{C^{*4} + 1900}} \\ T = \begin{cases} 0.56 + \left|0.2 \times \cos(h + 168°)\right| & \text{if} \quad 164° \le h \le 345° \\ 0.36 + \left|0.4 \times \cos(h + 35°)\right| & \text{otherwise} \end{cases} \end{cases} \tag{4}$$

where ΔL^*, ΔC^*, and ΔH^* represents the differences between L_1 and L_2, C_1 and C_2, and $H_1$and H_2, respectively. C^* denotes the value of hue at the reference point. l and c are input parameters used to determine the size of the ellipsoid. CMC standard suggests that if ΔE^* is less than one, which represents two colors are not distinguishable by human visual perception. CMC also suggests that uses l and c to change the shape of the corresponding ellipsoid. Different l and c affect luminance and chroma, respectively. As shown in Figure 1, the common l:c chosen consists of 1:1, 1.4:1, and 2:1.

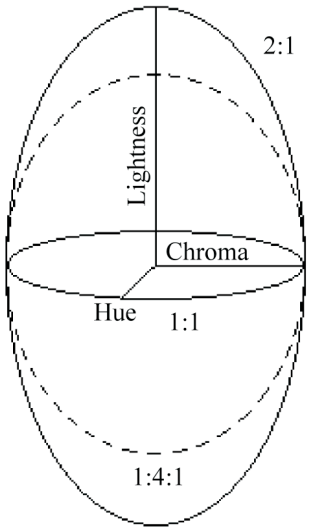


Figure 1. The ellipsoids in CIE-LCH for different l:c.

2.2 CIE-L*a*b*94

CIE-L*a*b*94 was standardized by CIE in 1994. The basic idea of this proposal is similar to CMC(l:c), which makes use of CIE-LCH color space for comparison. The computing of color difference in CIE-L*a*b*94 is shown in Eq. (5).

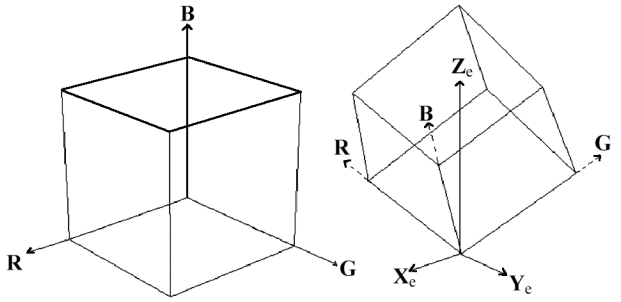
$$\begin{cases} \Delta E^* = \sqrt{\left(\dfrac{\Delta L^*}{k_L \times S_l}\right)^2 + \left(\dfrac{\Delta C^*}{k_C \times S_c}\right)^2 + \left(\dfrac{\Delta H^*}{k_H \times S_h}\right)^2} \\ S_l = 1.0 \\ S_c = 1.0 + 0.045 \times C^* \\ S_h = 1.0 + 0.015 \times C^* \\ C^* = \sqrt{C_1^* \times C_2^*} \end{cases} \tag{5}$$

Similar to CMC(l:c), CIE-L*a*b*94 also results in an ellipsoid to represent just noticeable difference (JND) region. The computation of color differences in the case CIE-L*a*b*94 is simpler than CMC(l:c). It depends on the three input parameters k_L, k_C, and k_H, which are fixed for different points. Meanwhile, the change of ellipsoid only depends on chroma. The reasons of these differences are simply because different models have different advantages and serve for different applications.

2.3 RGB-Ellipse Color Model

In dealing with color representation for processing, the HSI color representation has advantages over the RGB color representation. However, the transformation between the RGB color space to the HSI color space is a nonlinear process. As a result, the noise obtained in the image capture process becomes complicated to render. The proposed RGB-Ellipse color representation is similar to the HSI representation, and the transformation between the RGB to the RGB-Ellipse model is linear. Therefore, the RGB-Ellipse model has an advantage over the HSI color representation system.

The transformation between RGB to RGB-Ellipse ($X_eY_eZ_e$) is illustrated in Figure 2. The vector (1,–1,0) is used as a rotation axis. Using the right-hand rotation rule, we rotate the vector (1,1,1) in the RGB space to the place parallel to (0,0,1) in the $X_eY_eZ_e$ space. In this manner, the Z-axis in the RGB-Ellipse model represents the luminance and X- and Y-axis represent the chrominance plane,



(a) RGB color space (b) RGB-Ellipse color space

Figure 2. The transformation between the RGB space and the RGB-Ellipse ($X_eY_eZ_e$) color space.

respectively. The above transformation can be stated as Eq. (6).

$$\begin{bmatrix} X_e \\ Y_e \\ Z_e \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(1-\cos\theta)+\cos\theta & \frac{-1}{2}(1-\cos\theta) & \frac{-1}{\sqrt{2}}\sin\theta \\ \frac{-1}{2}(1-\cos\theta) & \frac{1}{2}(1-\cos\theta)+\cos\theta & \frac{-1}{\sqrt{2}}\sin\theta \\ \frac{1}{\sqrt{2}}\sin\theta & \frac{1}{\sqrt{2}}\sin\theta & \frac{1}{\sqrt{2}}\sin\theta \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (6)$$

$$\cos\theta = \frac{\sqrt{3}}{3}, \quad \sin\theta = \frac{\sqrt{6}}{3}$$

After plugging in the corresponding coefficients in the above equation, Eq. (6) can also be represented as Eq. (7).

$$\begin{cases} X_e = 0.789 \times R - 0.211 \times G - 0.577 \times B \\ Y_e = -0.211 \times R + 0.789 \times G - 0.577 \times B \\ Z_e = 0.577 \times (R + G + B) \end{cases} \quad (7)$$

From Eq. (7), it is clear that the RGB-Ellipse model is similar to the HSI color representation. The computation of the color difference in this paper is based on the above equation as shown in Eq. (8). Note that Eq. (8) also forms an ellipsoid Just Noticeable Difference region. Similar to the parameters *l* and *c* in the CMC(*l*:*c*) color model, the coefficients *a*, *b*, and *c* determine the shape of the corresponding ellipsoid. These coefficients also relate to the allowable tolerance in the $X_eY_eZ_e$ axis.

$$\Delta E^* = \sqrt{\frac{\Delta X_e^{\,2}}{a} + \frac{\Delta Y_e^{\,2}}{b} + \frac{\Delta Z_e^{\,2}}{c}} \quad (8)$$

As shown in the above equation, the computation of color difference in our model is able to count on the stretch of the illuminant axis, i.e., the control parameter *c*. In this manner, we take the advantage of the CMC(*l*:*c*) and CIE-L*a*b*94, which establish the human perceptual color difference model. Meanwhile, we avoid the significant computational load in the CMC(*l*:*c*) and CIE-L*a*b*94 models.

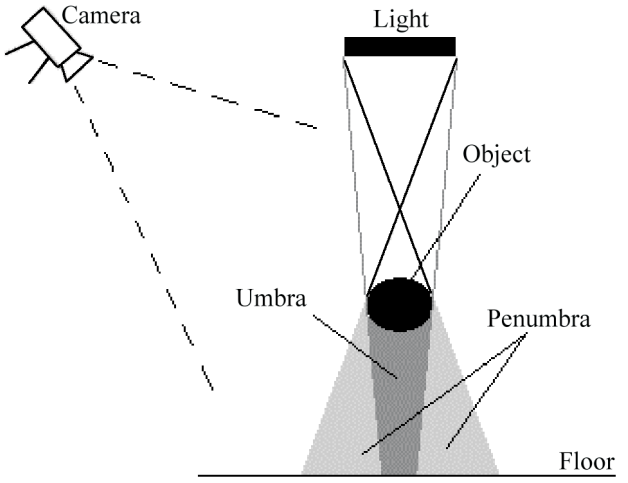


Figure 3. The relationship between light source, object, and shadow.

2.4. Shadow Elimination

Instead of shadow detection, our work aims to eliminate the influence of shadows in the process of object segmentation. Note that the color of a light source affects the color of shadow. As shown in Figure 3, the incident angle of a light source and the distance between the object and the light source affect the resulting shape of shadow. Furthermore, the receiving brightness is not all the same on the background. The umbra is darker than the penumbra.

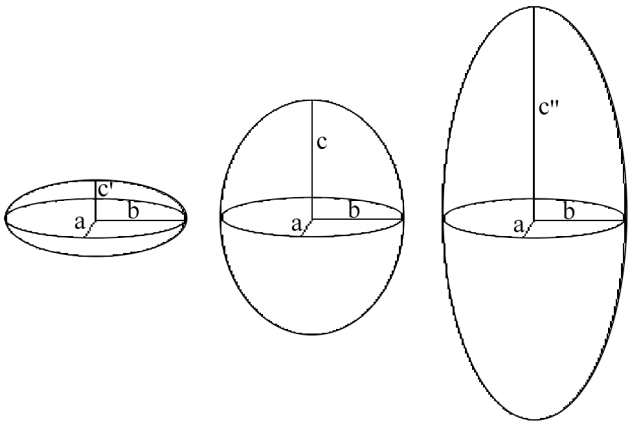
In this paper, there are two cases when we apply the RGB-Ellipse model to compute the color difference:

1. Comparison: To detect the differences between different images, the shadow cannot be treated as changes. In order to have more tolerance, the ellipsoid JND needs to extend along the Z_e (illuminant) axis.
2. Region growing: The shadow is not part of an object. To separate object and shadow, the ellipsoid JND needs to squeeze along the Z_e -axis.

The above two cases can be illustrated further in Figure 4. The volume of JND is increasing from Figure 4a to 4c due to the selection of the parameter *c*, which is the control parameter of the illuminant. Note that the chroma remains the same.

2.5 Noise Tolerance

The captured signals of CCD camera are contami-



(a) Squeeze *c* case (b) Normal case (c) Extend *c* case

Figure 4. The effect of control parameter *c* to the volume of JND in RGB-Ellipse model.

nated by noise. Assuming the noise distribution is Gaussian, we further modify Eq. (8) as follows.

$$\begin{cases} \Delta E^* = \sqrt{\dfrac{\Delta X_e^{\,2}}{a'} + \dfrac{\Delta Y_e^{\,2}}{b'} + \dfrac{\Delta Z_e^{\,2}}{c'}} \\ a' = \left(\dfrac{\tau + k_x}{\tau}\right)^2 \times a \\ b' = \left(\dfrac{\tau + k_y}{\tau}\right)^2 \times b \\ c' = \left(\dfrac{\tau + k_z}{\tau}\right)^2 \times c \end{cases} \tag{9}$$

where τ is a predetermined threshold of ΔE^*. k_x, k_y and k_z denote the standard derivation in $X_eY_eZ_e$ axis, respectively.

In summary, the proposed RGB-Ellipse model uses parameters *a* and *b* to deal with chrominance component and parameter c to take care of luminance and shadow. A fine adjustment of c is able to filter shadow effect in the image processing. As we increase the standard derivations (k_x, k_y, k_z) in Eq. (9), we are able to increase the size of the corresponding ellipsoids. In other words, we use these parameters to wrestle the noise in common camera.

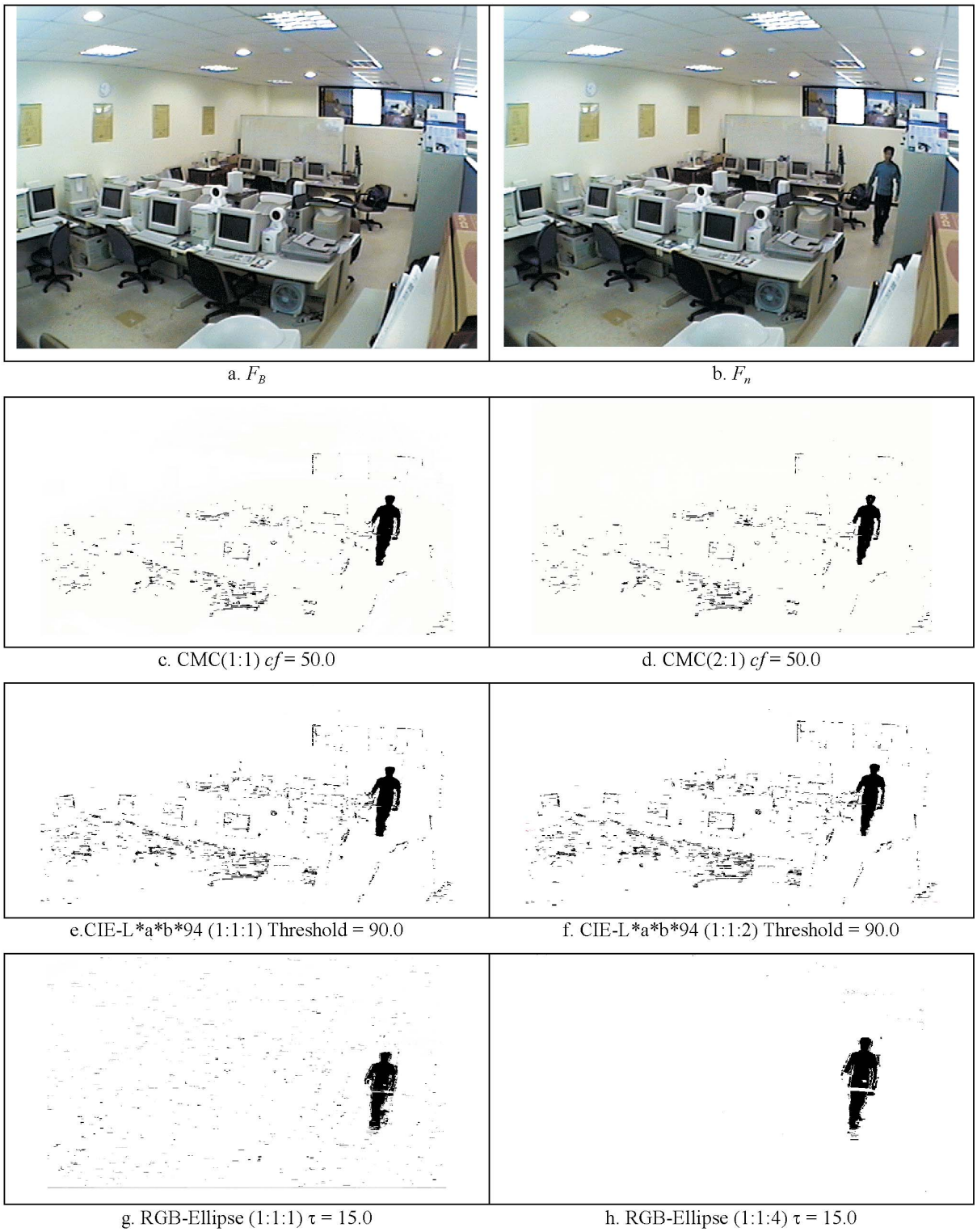
An experimental video is used to verify the above scheme. The results of shadow elimination and noise tolerance of CMC(*l*:*c*), CIE-L*a*b*94, and RGB-Ellipse models are shown in Figure 5. Figure 5a and Figure 5b show the background image of the monitoring site and the present image, respectively. The results of CMC(1:1), CMC(2:1), CIE-L*a*b*94(1:1:1), CIE-L*a*b*94(1:1:4), RGB-Ellipse(1:1:1), and RGB-Ellipse(1:1:4) are shown in Figure 5c, 5d, 5e, 5f, 5g, and 5h, respectively. Both of CMC(*l*:*c*) and CIE-L*a*b*94 produce fine results regarding to shadow elimination. In particular, these models take human perceptual cognition characteristics into account and are more sensitive to the dark area. However, the RGB-Ellipse has the advantage in dealing with noise. The Euclidean distance is used in Figure 5g, i.e., sphere model (1:1:1), to describe the color difference. The ellipse model (1:1:4) is more tolerable to luminance noise than the sphere model (1:1:1). This result is also indicated by comparing Figure 5h with Figure 5g.

Note that what represented in Figure 5 is actually not to show that RGB-Ellipse is better than CMC(*l*:*c*) or CIE-L*a*b*94. Observe the color of those people and the background, CMC(*l*:*c*) and CIE-L*a*b*94 strengthen low luminance part differentiate ratio, and produce better result. However, because of this strengthen, it would also easily produce false alarm in darker areas. In addition, we just use a common CCD sensor camera, it is not fine tuned for the experiment. Human perception-oriented color space may not be helpful for intruder detection. But in this paper, the purpose is not only limited to detect if one pixel is belonged to an intruder, but also include the detection and removal of the shadow area. If we stretch properly of linear sampling RGB space, the shadow can be tolerated. In other words, to get result similar to CMC(*l*:*c*) and CIE-L*a*b*94, the proposed color space will also be focused on shadow elimination as human perception behavior.

3. A Real-time Moving Objects Segmentation Scheme

A real-time moving objects segmentation scheme for surveillance systems is described in this section. The proposed segmentation scheme is able to separate moving objects with clear contours in real time which facilitates object-based tracking and analysis. The proposed scheme consists of three stages (see Figure 6), namely, (1) seed determination and region growing, (2) region-based change detection, and (3) background update. Seeds, the possible points that have the potential to form regions are allocated first. Then, the neighborhood points of seed that have the "same" color are collected. Region constitutes a basic comparison unit for the change detection in the second stage. The regions that do not be-

long to background are considered as an object; we only consider the portion subjects to changes thus eliminating many unnecessary objects searching processes. The third stage is applied to deal with the case when the applications background may be changed. This phenomenon occurs frequently in common visual surveillance systems.



a. F_B b. F_n

c. CMC(1:1) $cf = 50.0$ d. CMC(2:1) $cf = 50.0$

e.CIE-L*a*b*94 (1:1:1) Threshold = 90.0 f. CIE-L*a*b*94 (1:1:2) Threshold = 90.0

g. RGB-Ellipse (1:1:1) $\tau = 15.0$ h. RGB-Ellipse (1:1:4) $\tau = 15.0$

Figure 5. Results of shadow elimination and noise tolerance.

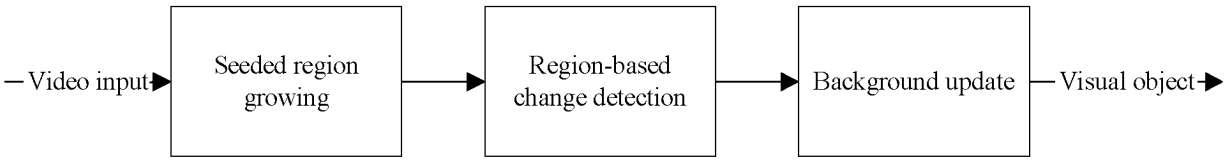


Figure 6. The designed moving object segmentation scheme.

The details will be presented in the following two subsections.

3.1 Seed Determination and Region Growing

The proposed region-based segmentation is an intuitive approach. The main idea is that the neighborhood pixels are collected together, if their colors are the "same". To implement the above idea, the seeds need to be determined first. Next, the RGB-Ellipse color model developed in Section 2 is applied to compute the color difference between the seed and its neighborhood pixels. Note that in an image of resolution 640 × 480, thousands of regions may be formed in arbitrary shapes. As a result, the object segmentation process becomes very time consuming. However, in the case of surveillance systems, only moving objects are of interest. If we are able to allocate the seed of the region inside the corresponding moving object, the result obtained after the region growing process may just convey the object of concern. In other words, we only focus on the regions that object are subject to change. Computation time is saved significantly in this situation.

There are two different cases in the determination of seeds. One case is when the background scene is available and does not change. In the other case, the background image is difficult to obtain without any disruption in some applications. The captured images always contain moving objects that may not be of interest.

Case 1. the background scene is available

We apply the background-based seeds determination scheme when the background scene is available. That is, for pixels that color differences are greater than a predetermined threshold, the corresponding pixels are selected in the Seed-List. Most pixels that belong to objects could be found this way. However, there are regions where colors are close to background. These regions require small thresholds to precede the region growing method.

Case 2. the background scene is not available

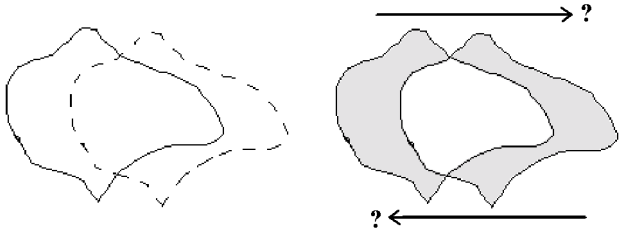


Figure 7. The color difference area in two consecutive frames.

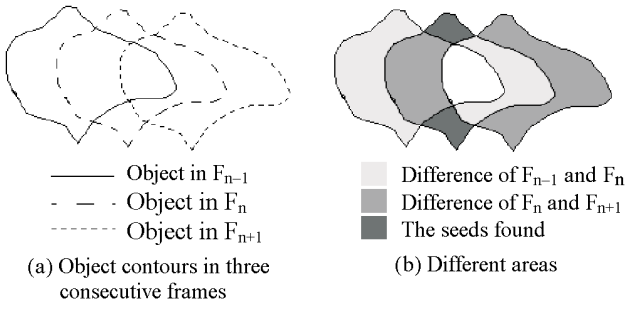


Figure 8. The different areas in three consecutive frames.

We apply the motion-based seeds determination scheme when the background image is not available. Let's consider the case of object moving in the consecutive images. As shown in Figure 7, we indicate the different regions of two images caused by the object's motion. The object contours are shown in Figure 7a. After the computation of color difference, the changed areas are shown in gray in Figure 7b. One cannot tell whether the changed areas belong to the background or to the moving object.

We make use of three consecutive images to resolve the above problem. The notations F_{n-1}, F_n, and F_{n+1} denote the previous image, the present image, and the next image. F_B represents the background image and does not contain any undesired motion objects. As shown in Figure 8, the light gray area indicates the color differences between F_{n-1} and F_n (hereafter we denote this area as ΔF). The dark gray area indicates the color differences between F_n and F_{n+1} (hereafter we denote this area as $\Delta F'$). The darkest area in Figure 8b represents the inter-

section of ΔF and $\Delta F'$. The point in the darkest areas is chosen as growing seed to identify the object.

To implement the above idea, we compute the RGB-Ellipse color difference on the same geometrical location of three consecutive frames. The computation sequence is from upper scan-line to lower scan-line and from left to right. If the color difference is greater than a predetermined threshold between the frame F_{n-1} and F_n, the color difference between F_n and F_{n+1} is also computed. This point is put into the SeedList, if both color differences are greater than a predetermined threshold. Otherwise, this point is ignored. After all the points in an image are taken into account, we have all seeds stored in the SeedList.

The region-growing step is carried out after the seed determination step. Each point inside the SeedList is viewed as an original point. First, the color differences between the original point and its four closest neighborhood points are computed. If the color difference is less than a predetermined threshold, the corresponding neighborhood point is put into the same region. Second, we update (R_{Mean}, G_{Mean}, B_{Mean}), which denotes the average value of RGB in the region, when new point is added in the region. The average value (R_{Mean}, G_{Mean}, B_{Mean}) is utilized as the comparison basis in the determination process. The above two steps are executed recursively until all the closest neighborhood points are evaluated. Then, the next seed, which does not cover the previous region growing step, in the SeedList is selected in the following region growing step. The above step is terminated when all the points in the SeedList are covered.

Our experimental experiences indicated that, in the practical applications, the boundary between object and the background is not so easy to separate. The SeedList obtained may contain points allocated to the background. To avoid this false alarm problem, a post processing assessment which inspects the ratio of the number of seeds and the growing area is included in the design scheme. If the ratio is over the threshold, this growing region is abandoned.

3.2 Region-based Change Detection and Background Update

In some cases, only applying the region growing algorithm may lead to false results, such as (1) regions belonging to the background F_B may be treated in the foreground F_n and (2) regions include the shadowed areas. Therefore, a region-based change detection scheme is introduced to eliminate the above problems. The scheme is similar to background subtraction, but the operation unit is region, instead of pixel. First, the resulting regions in F_n are projected to F_B, and then the mean color values in the corresponding regions are computed. Second, we compare the color difference by using Eq. (9). If the difference is greater than a predetermined threshold, the resulting region is classified as a part of a foreground object. Otherwise, it belongs to the background.

The background update is used to take care of the situation when the background image obtained may still be subject to change. Note that the background image does not process in the region-growing phase; the background update is a pixel based approach. First, an image buffer F'_B is allocated to track the possible pixels where colors may be changed. Then, we compute the RGB-Ellipse color differences for the remaining pixels, i.e., pixels that are not included in any object, between F_B and F_n. If the color difference is less than a threshold, this pixel is updated to F_B, otherwise, to F'_B. Meanwhile, a counter array is also included to record the update time of F'_B. If the update time is over a certain number of seconds, the pixels in the image buffer will replace F_B and then reset F'_B and the counter array. The recorded data in the image buffer F'_B provide us with information to track object appearing in F_n and object disappearing in F_B.

4. Implementation and Results

A digital surveillance system based on the designed moving object segmentation scheme is implemented. Common digital surveillance systems demand the following features: (1) real time processing, (2) moving objects detection (includes object entries and exists), (3) monitoring camera may be fixed or pan periodically. In this section, we describe our implementation results. We integrate (1) visual image object segmentation, (2) object histogram comparison, (3) motion tracking, (4) real-time transmission, and (5) digital storage/query techniques in designing a novel networked visual monitoring system. The detail surveillance system architecture can be found in our previous work [20]. In this paper, we extend our previous result. A novel color model is developed as the basis to enhance the image segmentation algorithm in dealing with shadow and noise elimination. The devel-

oped system has the ability to perform the following functions:

1. Auto-tracking: A tracking camera tracks the intruder motion at the monitoring site then, it takes a series of shots to capture the main characteristics of the intruders. Instead of periodically scanning the monitoring site, the designed system captures the intruder's features actively. If no intruders appear, the system will not record any useless images.
2. Web-based remote control and monitoring: Users are able to view the monitoring site from common web browsers on the Internet. Authorized users are allowed to set the tracking mode and to manipulate the tracking camera to desired location remotely.
3. Digital storage/query: The system provides two types of query. One is based on time. The other is based on event. Thus, the events of interest can be retrieved randomly and efficiently.

A testing video sequence, in which a man wearing a dark jacket enters a lighted reading room is chosen to illustrate the results of the proposed scheme. Our demand is to separate the intruder from the background image to facilitate the object tracking process. The frame size is 640 × 480 with sampling rate 10 fps. We illustrate the detailed experimental results of the proposed algorithm in Figure 9. The three consecutive images are shown in Figure 9a, 9b, and 9c. Seeds are obtained in Figure 9d after the seed determination process. The results of region growing are shown in Figure 9e. In the regions that seed color closes to the background image, we have false alarm regions. These false alarm regions can significantly be reduced by using the algorithm as described in section 3.1 (see Figure 9f). Then, we apply the change detection algorithm to compute the color differences between F_n and F_B in these regions; the false alarm regions can be further reduced (see Figure 9g). We obtain the final result by connecting the neighborhood regions (see Figure 9h). The intruder is clearly identified.

Another testing video is chosen to illustrate the global results of the proposed algorithm (see Figure 10). In the first part, we illustrate the quality of segmentation results. There are five frames, Frames 55, 61, 79, 217, and 301. The second part starts from Frame 403. Another man brings a bottle and comes into the room. He leaves the bottle on the desk. As shown in Frame 433, the bottle will be detected. The third part starts from Frame 499. We show the occlusion case. The proposed algorithm can identify a multiple objects situation.

In our experimental implementation, by using Pentium III 1.2 GHz, we are able to process 15 to 20 fps under im-

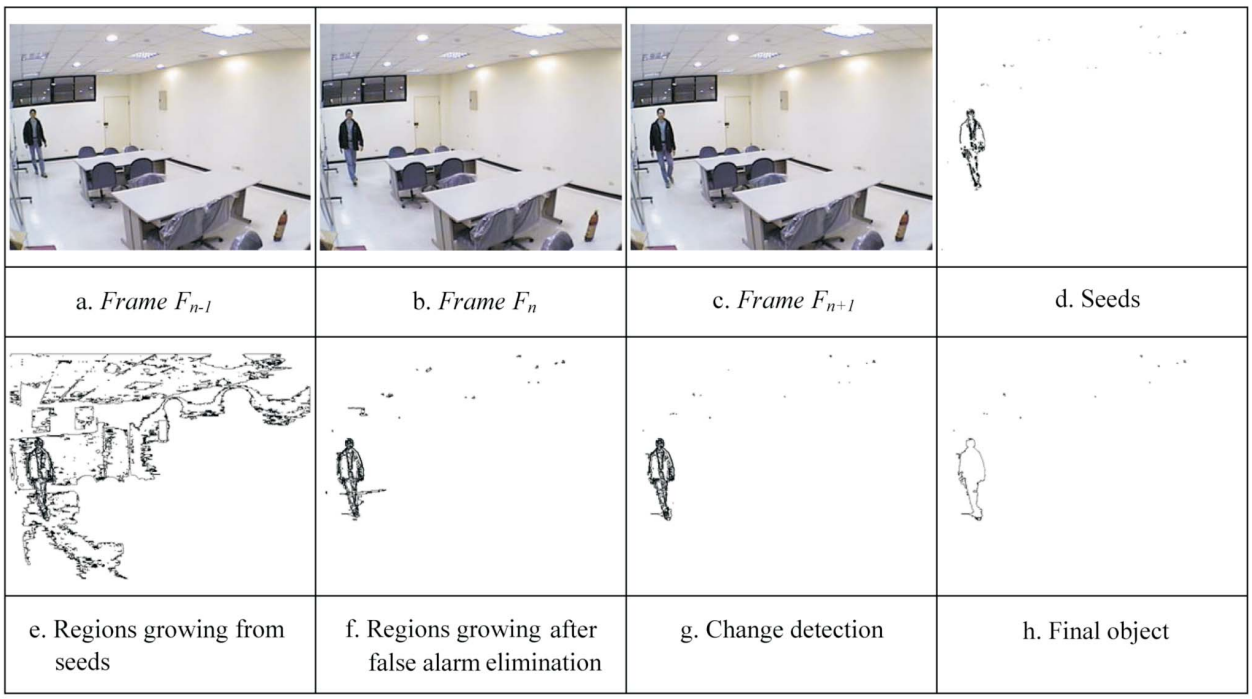


Figure 9. The experimental results of the proposed moving object segmentation algorithm.

Figure 10. The experimental results of a given testing video.

age size 320 × 240 and color depth 24bits. This result is suitable for use in most real time digital surveillance system.

5. Conclusions

Object segmentation plays a crucial role in modern image and video processing. In this paper, we have designed a real-time moving object segmentation scheme for surveillance system. The segmentation scheme consists of two main stages, namely, (1) Seed determination and region growing and (2) region-based change detection and background update. In order to compute the color difference in our scheme efficiently and effectively, we propose a novel color model, RGB-Ellipse. Taking advantage of CIE-L*a*b*94 and CMC(l:c) in determination of color differences and the linear transformation from RGB to RGB-Ellipse, the proposed color model allows us to manipulate noise tolerance and shadow elimination issues in a straightforward manner. These features make the proposed scheme very attractive.

To illustrate the value of our approach, we have implemented the proposed moving object segmentation scheme in a networked surveillance system. Even without a code optimization process, our implementation and simulation results indicate that the proposed scheme is able to achieve image segmentation at a frame rate of 16 fps with 320 × 240, 24-bits true color image. That is suitable for many real-time applications, such as gesture capturing in the distance learning classroom or conferencing room.

Acknowledgments

The present work is partly supported by National Science Council Taiwan R.O.C. under the contract No. NSC 92-2520-S-032-002.

References

[1] Canny, J., "A Computational Approach to Edge Detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 679–698 (1986).

[2] Ma, W. Y. and Manjunath, B. S., "Edge Flow: A Framework of Boundary Detection and Image Segmentation," *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 744–749 (1997).

[3] Sarkar, S. and Boyer, K. L., "Optimal, Efficient, Recursive Edge Detection Filters," *Proc. of 10th Int'l Conf. on Pattern Recognition*, Vol. 1, pp. 931–936 (1990).

[4] Christopoulos, C. A., Philips, W., Skodras, A. N. and Cornelis, J., "Segmented Image Coding: Techniques and Experimental Results," *Signal Processing: Image Communication*, Vol. 11, pp. 63–80 (1997).

[5] Elgammal, A., Duraiswami, R., Harwood, D. and Davis, L. S., "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," *Proc. of IEEE*, Vol. 90, pp. 1151–1163 (2002).

[6] Dufaux, F. and Moscheni, F., "Segmentation Based Motion Estimation for Second Generation Video Coding Techniques," L. Torres and M. Kunt, editors, *Video Coding: The Second Generation Approach*, Chapter 6, Kluwer Academic Publishers, Boston, MA, U.S.A. pp. 219–263 (1996).

[7] Dufaux, F., Moscheni, F. and Lippman, A., "Spatio-Temporal Segmentation Based on Motion and Static Segmentation," *Proc. of ICIP'95, IEEE*, Vol. 1, pp. 306–309 (1995).

[8] Fan, J., Yau, K. Y., Elmagarmid, A. K. and Aref, W. G., "Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing," *IEEE Trans. on Image Processing*, Vol. 10, pp. 1454–1466 (2001).

[9] Deklerck, R., Corneils, J. and Bister, M., "Segmentation of Medical Images," *Image and Vision Computing*, Vol. 11, pp. 486–503 (1993).

[10] Kim, C. and Hwang, J. N., "Fast and Automatic Video Object Segmentation and Tracking for Content-Based Applications," *IEEE Trans. on Circuit and System for Video Tech.*, Vol. 12, pp. 122–129 (2002).

[11] Deng, Y. and Manjunath, B. S., "Unsupervised Segmentation of Color-Texture Regions in Images and Video," *IEEE Trans. on Pattern Analysis and Machine Learning*, Vol. 23, pp. 800–810 (2001).

[12] Etoh, M., "Promotion of Block Matching: Parametric Representation for Motion Estimation," *Proc. 4th ICPR*, Vol. 1, pp. 282–285 (1998).

[13] Gao, H., Siu, W. C. and Hou, C. H., "Improved Techniques for Automatic Image Segmentation," *IEEE Trans. on Circuit and System for Video Tech.*, Vol. 11, pp. 1273–1280 (2001).

[14] Harville, M., Gordon, G. and Woodfill, J., "Foreground Segmentation Using Adaptive Mixture Models in Color and Depth," *Proc. of the IEEE Workshop on Detection and Recognition of Events in Video*, pp. 3–11 (2001).

[15] Rosin, P. L., "Thresholding for Change Detection," *Proc. of 6th Int'l Conf. on Computer Vision, IEEE*, pp. 274–279 (1998).

[16] Chien, S. Y., Ma, S. Y. and Chen, L. G., "Efficient Moving Object Segmentation Algorithm using Background Registration Technique," *IEEE Trans. on Circuit and System for Video Tech.*, Vol. 12, pp. 577–586 (2002).

[17] Foresti, G. L. and Regazzoni, C. S., "A Change-Detection Method for Multiple Object Localization in Real Scenes," *Proc. of Int'l Conf. on Industrial Electronic, Control and Instrumentation*, pp. 984–987 (1994).

[18] Gibbins, D., Newsam, G. N. and Brooks, M. J., "Detecting Suspicious Background Changes in Video Surveillance of Busy Scenes," *Proc. of 3rd IEEE Workshop on Application of Computer Vision*, pp. 22–26 (1996).

[19] "http://www.komoto.com.tw/html/idx-info.html" (2005).

[20] Kuo, C. H., Wang, T. S. and Wu, P. H., "Design of Networked Visual Monitoring Systems," *The Tamkang Journal of Science and Engineering*, Vol. 2, pp. 149–161 (1999).

Manuscript Received: Mar. 2, 2005
Accepted: May 9, 2005