# A Semi-Automatic Video Segmentation Algorithm using Virtual Blue Screens

Jong-Han Shin*, Daehee Kim and Yo-Sung Ho

Dept. of Information and Communications
Kwangju Institute of Science and Technology (K-JIST)

## ABSTRACT

In order to support the object-based coding in the MPEG-4 video coding standard, we need to represent each video frame in terms of video object planes (VOPs). Segmentation is an essential operation to generate such VOPs. In this paper, we propose a semi-automatic segmentation algorithm using a concept of the virtual blue screen (VBS) that is defined as an image where its background is colored by a particular value, such as blue. After we extract moving objects in the first frame using VBS, we predict locations of the moving objects in the following frames via VBS. Experiment results show that the proposed method reduces the computational complexity significantly, while providing reasonably good segmentation results under an assumption that the video sequence has a still background.

**Keywords:** Semi-automatic Video Segmentation, Object Tracking, Virtual Blue Screen, Object Extraction

## 1. INTRODUCTION

The MPEG-4 video coding standard provides many new features for multimedia applications and enables interactivity with visual objects in video sequences. A scene in MPEG-4 is viewed as a composition of video objects (VOs) with intrinsic properties, such as shape, motion, and texture. VOs are semantic objects which can be useful in a variety of new digital video applications including content-based video communication, internet multimedia, digital movie studio, digital video database, remote sensing, and robot vision. Such a content-based representation of a video sequence is an essential key to enable interactivity with objects for a variety of multimedia applications [1].

For the content-based video representation, object segmentation is a prerequisite. However, identification and extraction of video objects is not an easy task because physical objects are generally not homogeneous with respect to image features, such as color, luminance, and optical flow [1]. Recently, many segmentation algorithms for moving object extraction have been proposed. Automatic segmentation methods based on spatial-temporal operations are proposed in [2]; however, automatic segmentation algorithms are premature to obtain satisfactory segmentation results because automatic segmentation itself is an ill-posed and open problem. Semi-automatic segmentation is straightforward and improves segmentation performance for non-real time applications because simple interaction could be very useful to adapt the segmentation procedure spatially or temporally and to change some significant segmentation parameters based on segmentation results [3].

Previous semi-automatic segmentation algorithms usually consist of threes steps [4]. In the first step, the user can identify semantic objects by user's pointing devices. For the given initial object boundary, we define interior and exterior boundaries to limit the search area of the real object contour. In the third step, uncertain pixels between interior and exterior boundaries are classified to one of the neighboring objects [5]. Although current semi-automatic segmentation techniques can extract moving objects from the initial frame, they are somewhat inefficient in the inter-frame segmentation because they need to find all possible candidate blocks in the given search range to obtain the best matched block.

* jhsin@kjist.ac.kr; phone +82-62-970-2263; fax +82-62-970-2204; http://www.kjist.ac.kr; Visual Comm. Lab., K-JIST, 1 Oryong-dong, Buk-gu, Kwangju, 500-712, Korea



Visual Communications and Image Processing 2002,
C.-C. Jay Kuo, Editor, Proceedings of SPIE Vol. 4671 (2002) © 2002 SPIE · 0277-786X/02/$15.00

In this paper, we propose a new semi-automatic video segmentation algorithm using a virtual blue screen (VBS) to utilize information of object motion and region boundaries assuming a still background. In order to extract moving objects from the current frame efficiently, we define VBS as a reference image where the background is colored by a particular value. Since it is very simple and easy to extract moving objects in the blue screen, we can exploit the concept of the blue screen by constructing an image with the given background color and estimate VBS over the following frames. The proposed semi-automatic segmentation algorithm consists of two parts: intra-frame segmentation and inter-frame segmentation. The intra-frame segmentation includes construction of VBS, region labeling and region decision. The inter-frame segmentation completes the overall segmentation by predicting moving objects defined in the intra-frame segmentation and reconstruct VBS for the next image frame.

## 2. THE SYSTEM OVERVIEW

In our segmentation algorithm, we first identify semantic objects and construct VBS, and then predict moving objects using VBS for the next frame segmentation. Fig. 1 illustrates main operations of the proposed semi-automatic segmentation scheme. Before we start video segmentation, we can remove random noises in the video sequence by applying an appropriate lowpass filter. After the pre-processing operation, the input image $f_k$ is segmented based on temporal motion information and VBS of the previous frame, as shown in Fig. 1.

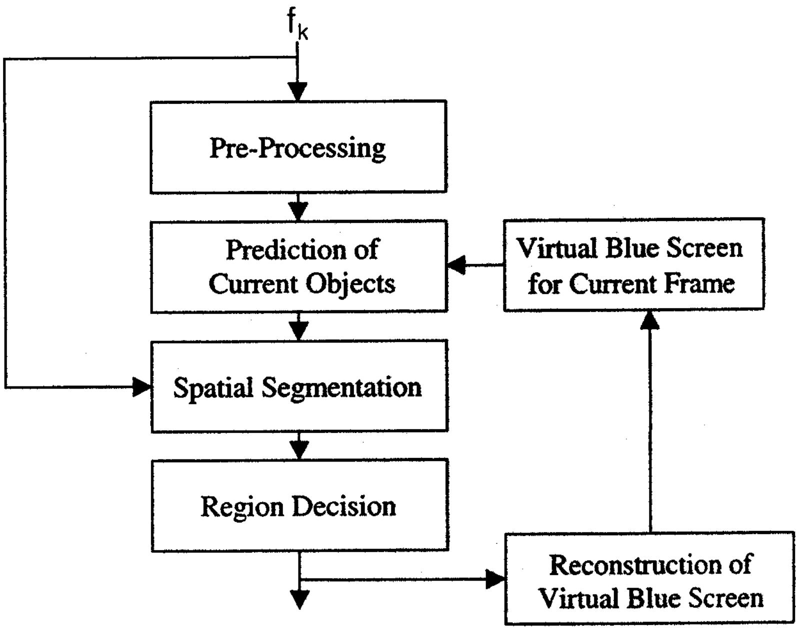



**Figure 1.** The Proposed Segmentation Algorithm

Since image sequences usually entail intensity changes, we employ differences of intensity values between two successive image frames to predict motion regions of moving objects [6]. A local window slides over the difference image. The squared sum of differences is compared to a predefined threshold value. This approach usually leads to the test static, such as the chi-squared distribution or the $F$-distribution. In our algorithm, we utilized the $F$-distribution as a probability distribution model for the test statistic.

In order to obtain regions belonging to moving objects in the current frame, we split the input image $f_k$ and its prediction image into homogeneous regions by a watershed algorithm with morphological filtering [7]. Then, we compare two spatially segmented images to extract moving objects from the current frame. Finally, we reconstruct VBS for the next frame using the extracted moving objects.

## 3. MOVING OBJECT EXTRACTION

### 3.1 INITIAL OBJECT EXTRACTION

A virtual blue screen (VBS) that is used to extract moving objects in the current frame is a reference image obtained from the previous frame. We can construct VBS by assigning a particular value to regions where we are not interested in. As shown in Fig. 2, we do not need to outline the interested region precisely. It is sufficient for us to distinguish object regions from the background region roughly. We can construct VBS easily with an aid of the graphic user interface (GUI). Fig. 2 shows the value inserted in the background region and the constructed VBS.

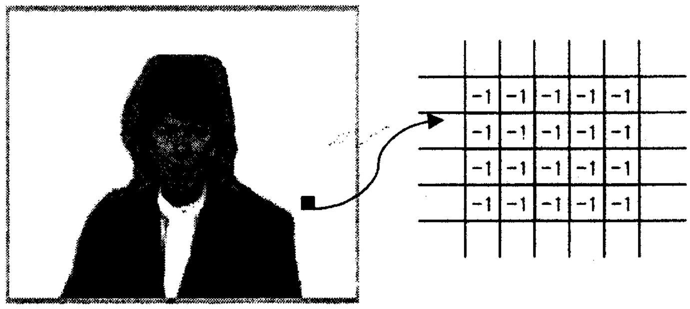



**Figure 2.** Virtual Blue Screen

Once we have VBS, we can extract moving objects in the initial frame by region decision. In order to segment the initial frame, we partition the current input image and VBS into homogeneous regions by a watershed algorithm [7]. Because the watershed algorithm splits each image into homogeneous regions, we can have common regions between the current input image and VBS. We then determine regions of moving objects according to rules described in Sect. 3.3.

### 3.2 MOVING OBJECT PREDICTION

We can apply inter-frame segmentation on the result of the intra-frame segmentation in the consecutive frames following the first frame, newly appeared objects, or scene changes. In inter-frame segmentation, the user-defined object is usually segmented by object tracking. However, it takes relatively long time to calculate motion vectors because we employ the block-matching algorithm (BMA). If the background is fixed, we can reduce computational complexity considerably by utilizing a change detector in the neighborhood region of object boundaries.

#### 3.2.1 STATISTICAL HYPOTHESIS

In order to predict locations of moving objects, we can use a statistical hypothesis test. In general, motion of a moving object entails intensity changes in magnitude; therefore, intensity change is one of the important cues for locating moving objects both in the time and in the space [2]. The intensity change can be represented by the difference image between two successive images, both of which include moving objects. The intensity difference in the stationary background between two consecutive frames is often modeled as the normal distribution [2].

For the prediction process, a statistical hypothesis test is used based on a variance test to detect intensity change. The idea of this approach is that the intensity variation of the moving object is different from that of the stationary background. If we define $\sigma_1^2$ and $\sigma_2^2$ as the variances of the background and the moving object regions, respectively, the hypothesis on variances can be written as

$$\begin{aligned} H_0 &: \sigma_1^2 = \sigma_2^2 \\ H_1 &: \sigma_1^2 < \sigma_2^2 \end{aligned} \tag{1}$$

The null hypothesis $H_0$ implies that there is no change between two consecutive points. The alternative hypothesis $H_1$ is that if we test a statistical hypothesis on a moving object region, the variance of the moving object is larger than that of the background so that there is a change between two consecutive points. This operation exploits the fact that the region of the moving object usually entails large changes in intensity, resulting in a large value of the variance estimate for the pixel difference in the region between two consecutive frames [2]. In this paper, we adopt the $F$-distribution as a statistical distribution model. Let $X_1, X_2, \ldots\ldots, X_n$ be independent identically distributed (*iid*) random variables with a normal distribution $N[\mu, \sigma^2]$. The random variable

$$(n-1)S^2/\sigma^2 = \sum_{i=1}^{n}(X_i - \bar{X})/\sigma^2 \tag{2}$$

has the chi-squared distribution with ($n$-1) degree of freedom. If we define $V_1$ and $V_2$ as chi-squared random variables with $n_1$ and $n_2$ degrees of freedom, respectively, the random variable $V$

$$V = \frac{V_2/n_2}{V_1/n_1} \tag{3}$$

has the $F$-distribution with $n_1$ and $n_2$ degrees of freedom. Because the random variable $S_1^2/\sigma_1^2$ and $S_2^2/\sigma_2^2$ have the chi-squared distributions with ($n_1$-1) and ($n_2$-1) degrees of freedom, respectively, the random variable $V$ can be written as

$$V = \frac{S_2^2/\sigma_2^2}{S_1^2/\sigma_1^2} \tag{4}$$

Under the null hypothesis, that is, $\sigma_1^2 = \sigma_2^2$, the random variable $V$ is reduced to the ratio $S_1^2/S_2^2$. The examination of the ratio performs the hypothesis test established on variance comparison. When the null hypothesis is true, we can then assert $S_1^2$ and $S_2^2$ to be close in a certain value. Therefore, if the random variable $V$ is much larger than 1, we declare that a change has occurred. Otherwise, we declare that no change occurs.

### 3.2.2 PREDICTION

We first define some variables. Let $C(k)$, $P(k)$, $C_p(k)$ and $B(k)$ denote the current input image, the previous image, the current prediction image and VBS, respectively, where $k$ represents the pixel position. We can define $B(k)$ by

$$B(k) = \begin{cases} P(k) & \text{if a pixel of } P(k) \text{ is in the object region} \\ BLUE & \text{if a pixel of } P(k) \text{ is not in the object region} \end{cases} \tag{5}$$

where the '*BLUE*' is the symbolic value of the background color.

The value of $B(k)$ is actually the same as that of VBS, as shown in Fig. 2. $C(k)$ and $P(k)$ are compared at the position where $B(k)$ has the '*BLUE*' value. If the variance ratio of the differences between $C(k)$ and $P(k)$ at that position exceeds some threshold value, we declare that significant change has occurred and we assign the pixel value of $C(k)$ to $C_p(k)$ at the same position. Otherwise, '*BLUE*' is assigned.

In order to make a more reliable decision, we evaluate a set of differences inside a small decision region, instead of only one pixel. We do not need to consider all pixels assigned as '*BLUE*' when making a decision. We only consider pixels within the predefined width, as shown in Fig. 3, because the motion of a moving object is likely to occur in the neighborhood of object boundaries. In this way, we can reduce computational time significantly relative to a method based on motion vectors to predict the location of objects. The pixel values of $C(k)$ in the position where $B(k)$ does not have the value of '*BLUE*' are copied to $C_p(k)$ in the same position. Fig. 4 shows the prediction image of a moving object.

$$C_p(k) = \begin{cases} BLUE & \text{if } V < threshold \\ C(k) & \text{if } V \geq threshold \end{cases} \tag{6}$$

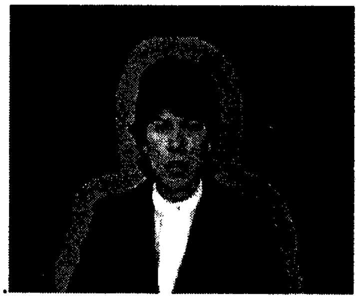
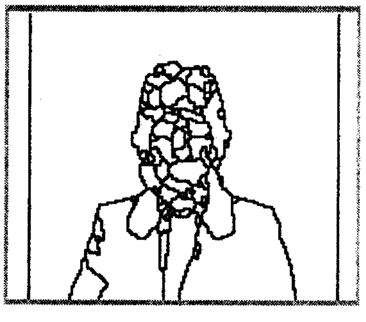


**Figure 3.** Search Range

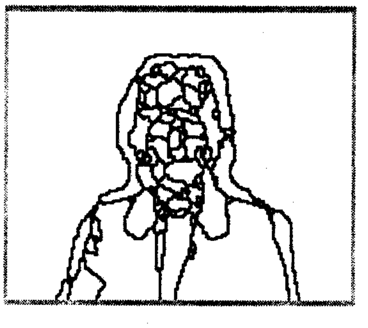
**Figure 4.** Prediction Image

### 3.3 OBJECT EXTRACTION

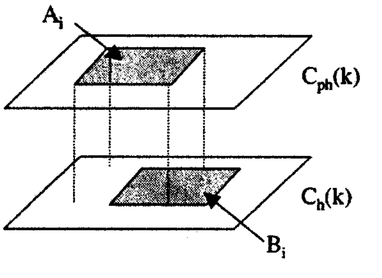
Fig. 5 shows label masks of $C(k)$ and $C_p(k)$, respectively, and the mapping process of two masks. We define $C_h(k)$ as the label mask of $C(k)$, and $C_{ph}(k)$ as that of $C_p(k)$. $C_{ph}(k)$ includes object regions of $C_h(k)$ and additional regions due to $C_p(k)$. In order to generate a complete object mask $M(k)$, we need to decide whether the additional region of $C_{ph}(k)$ belongs to the foreground or not. For example, we can assume that region $A_i$ in $C_{ph}(k)$ is overlaid on region $B_i$ in $C_h(k)$, as shown in Fig. 5(c). Since we employ a simple mapping process, some parts of $A_i$ may be overlaid to the outside of $B_i$. Thus, we need to calculate the size of the overlapped region between $A_i$ and $B_i$. Once the size of the overlapped region is obtained, we can easily determine whether $A_i$ belongs to the actual object region or not for the given threshold value. Fig. 6 shows the detected additional regions and the complete change mask.



(a) Label Mask of $C(k)$

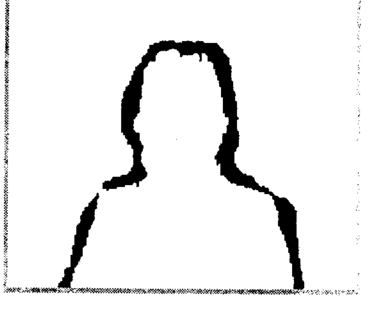


(b) Label Mask of $C_p(k)$




(c) Mapping Process

**Figure 5.** Label Masks and Mapping Process



(a) Additional Regions

(b) Complete Change Mask

**Figure 6.** Object Mask

## 4. EXPERIMENTAL RESULTS

We have tested the proposed algorithm with MPEG-4 test sequences that contain moving foreground objects in the still background. The video sequences are of the QCIF format. The proposed algorithm can extract moving objects accurately once the outline of the object is defined in the first frame. Assuming that the background is stationary and the mean square error (MSE) is used to estimate motion vectors, we can observe that complexity of the proposed algorithm is much less than that of the typical semi-automatic segmentation algorithm [3], as shown in Table 1.

| Method using Motion Vector | Method using Change Detection |
|---|---|
| Number of Blocks : 156<br>Block Size : 9×9<br>Search Range : H/V [-5,5] | Number of Pixels : 2899<br>Window Size : 9×9 |
| Number of Operations<br>156×81×121 = 1528956 | Number of Operations<br>2899×81 = 234819 |

**Table 1.** Complexity Comparison

The variance used in the change detection is calculated by Eq. (2), while MSE used in the motion vector estimation is defined by

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(f_{j+i} - f_{j-1+i})^2 \tag{7}$$

where $f_j$ represents a current input frame.

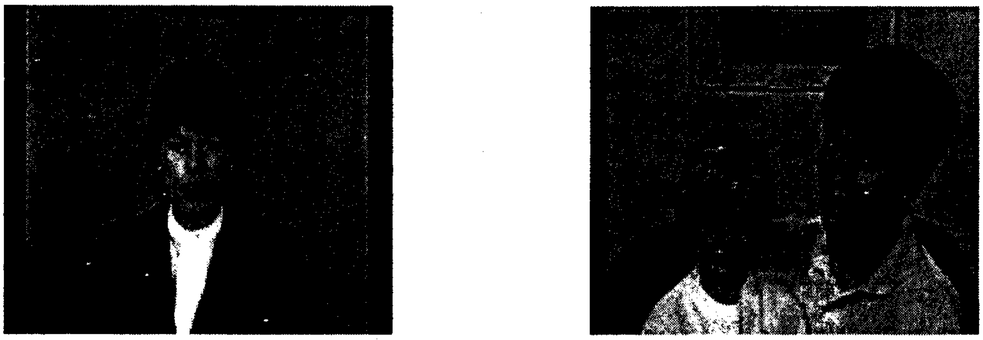
As shown in Table 1, the proposed algorithm has reduced the number of operations to about 20% of the number of MSE computations in the inter-frame segmentation. Fig. 7(a) and Fig. 7(b) show a partitioned image by 9×9 block for motion vector estimation and a search range for change detection, respectively.

(a) Blocks for Motion Estimation (b) Search Range

**Figure 7.** 3th Frame of MOTHER AND DAUGHTER

An automatic segmentation method for VOP generation usually depends on the intensity difference between two subsequent frames and decides a changed region as the foreground and an unchanged region as the background. Therefore, it cannot extract a complete object in the first frame of the video sequence because of insufficient information about the object, and may need a long transition time until the complete moving object is found. On the other hand, a semi-automatic segmentation method with user assistance in the first frame can extract a moving object without missing the information of the object from the initial state. Therefore, if the whole object contour is appeared in the initial frame segmentation by the user's aid, the change detection used in the automatic segmentation method can be an efficient operation because it can reduce computational complexity significantly, as described earlier.

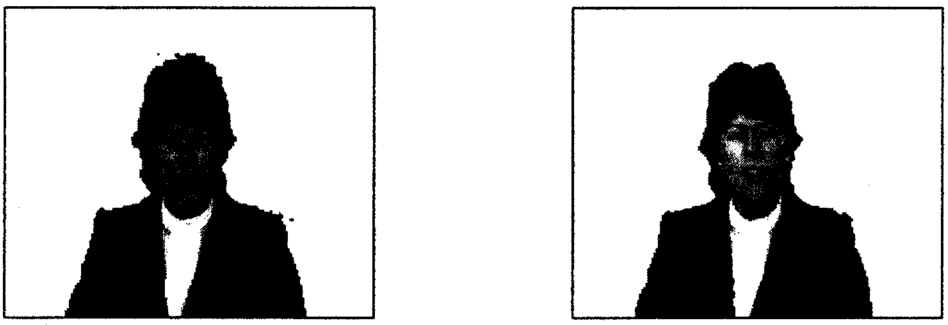
Fig. 8 shows the initial frames of CLAIRE and MOTHER AND DAUGHTER sequences.



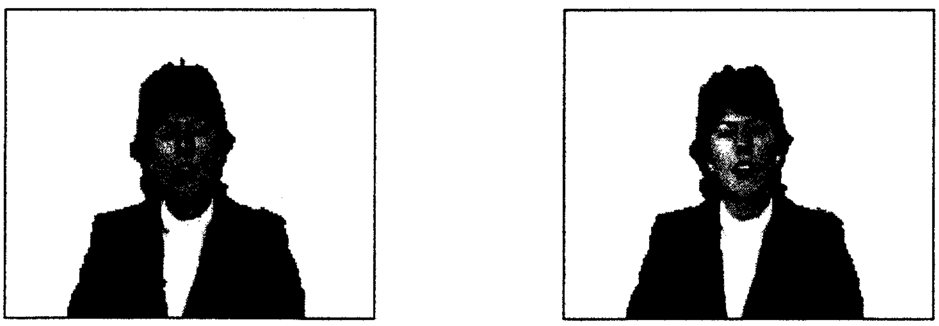
(a) CLAIRE sequence (b) MOTHER AND DAUGHTER sequence

**Figure 8.** Initial Frame

Fig. 9 and Fig. 10 show segmentation results for their sequences both by intra-frame and by inter-frame segmentation. Fig. 11 shows segmentation results of AKIYO sequence by inter-frame segmentation.



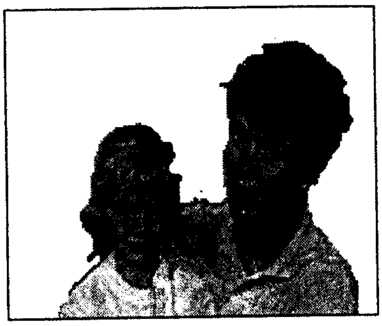
(a) $0^{th}$ Frame (b) $6^{th}$ Frame



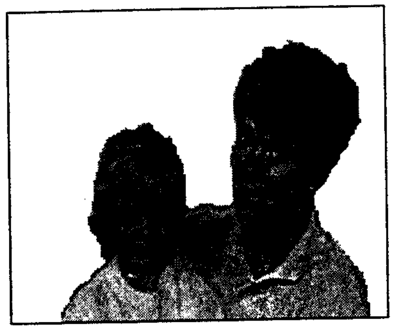
(c) $12^{th}$ Frame (d) $15^{th}$ Frame

**Figure 9.** Segmentation Results (CLAIRE)

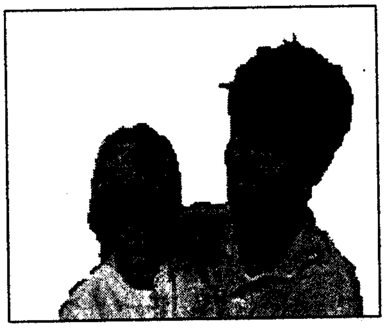
Fig. 9(a) and Fig. 10(a) show intra-frame segmentation results of CLAIRE and MOTHER AND DAUGHTER sequences, respectively. The other figures in Fig. 9 and Fig. 10 show segmentation results in inter-frame by the proposed method. We can observe that they have nice object contours without losing object information. However, in Fig. 10 and Fig. 11, segmentation results look a bit poor because the human and the background in boundary regions have a similar color. Therefore, boundary regions between the object and the background are merged into a homogenous region by a watershed algorithm. Such a problem can be solved by improving a region-growing algorithm, such as the watershed algorithm.



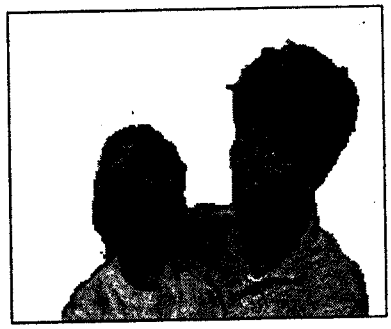
(a) $0^{th}$ Frame



(b) $9^{th}$ Frame



(c) $12^{th}$ Frame



(d) $15^{th}$ Frame

**Figure 10 .** Segmentation Results (MOTHER AND DAUGHTER)

(a) $51^{th}$ Frame

(b) $87^{th}$ Frame

(c) $138^{th}$ Frame

(d) $201^{th}$ Frame

**Figure 11 .** Segmentation Results (AKIYO)

## CONCLUSIONS

Segmentation is recognized as an ill-posed problem and still remains unsolved. However, the MPEG-4 standard assumes that VOPs in video sequence are available. Good segmentation algorithms for natural images and video sequences enable to realize potential applications of MPEG-4. In this paper, we have proposed a semi-automatic VOP generation method to support object-based coding. The proposed algorithm utilizes the concept of the blue screen. For the intra-frame segmentation, we construct a virtual blue screen manually and predict it in the successive frames to track moving objects for the inter-frame segmentation. We have reduced computational complexity significantly using the change detection instead of motion vectors for the inter-frame segmentation. In addition, our algorithm shows good performance under the assumption of a still background.

## ACKNOWLEDGEMENT

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through the Ultra-Fast Fiber Optical Network (UFON) Research Center at kwangju Institute of Science and Technology (K-JIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK 21) project.

## REFERENCES

1. MPEG Video Group, "MPEG-4 Video Verification Model Version 10.1," *ISO/IECJTC1/SC29/WG11/W3464*, March 1998.

2. M. Kim, J. Choi, D. Kim, H. Lee, M. Lee, C. Ahn and Y. Ho, "A VOP Generation Tool: Automatic Segmentation of Moving Objects in Image Sequences based on Spatio-Temporal Information," *IEEE Trans. Circuits and System For Video Techonology*, vol. **9**, no. 8, Dec. 1999.

3. J.G. Choi and et al., "A User-Assisted Segmentation Method for Video Object Plane Generation," *ITC-CSCC '98*, vol. 2, no. 1, pp. 7-10, July 1998.

4. C. Gu and M. Lee, "Semiautomatic Segmentation and Tracking of Semantic Objects," *IEEE Trans. On Circuit and System for Video Technology*, vol. **5**, pp. 572-584. August 1988.

5. D. Park, H. Yoon and C. Won, "Fast Object Tracking in Digital Video," *IEEE Transaction on Consumer Electronics*, vol. **46**, no. 3, August 2000.

6. ISO/IEC JTC1/SC29/WG11, MPEG-4 Visual Final Committee Draft, Dublin, Ireland, July 1998.

7. P. Salembier and M. Pardas, "Hierarchical Morphological Segmentation for Image Sequence Coding," *IEEE Trans. Image Processing*, vol. **3**, no. 5, pp. 639-651. Sept. 1994.

8. T. Aach, A. Kaup and R. Mester, "Statistical Mode-based Change Detection in Moving Video," *Signal Processing*. vol. **31**, no. 2, pp. 165-180, March 1993.

9. T. Aach and A. Kaup, "Bayesian Algorithm for Adaptive Change Detection in Image Sequences using Markov Random Fields," *Signal Processing*, vol. 7, no. 2, pp.147-160, May 1995.