Far East Journal of Experimental and Theoretical Artificial Intelligence
Volume 4, Number 2, 2009, Pages 73-87
Published Online: March 5, 2010
This paper is available online at http://www.pphmj.com
© 2009 Pushpa Publishing House

A FIGURE EXTRACTION AND SYNTHESIS SYSTEM USING LEARNING VECTOR QUANTIZATION NEURAL NETWORKS

CHUAN-YU CHANG and ZONG-YU TSAI

Department of Computer Science and Information Engineering
National Yunlin University of Science and Technology
Taiwan
e-mail: chuanyu@yuntech.edu.tw

Abstract

Extracting complete figures from videos in complicated environments is difficult. A novel figure extraction and synthesis system with capability of extracting figures from consecutive frames in a messy environment is proposed in this paper. A figure template is constructed based on the face detection results and some image processing techniques. Features of figure and background are extracted from the figure images. By means of these features, a learning vector quantization neural network (LVQNN) is applied to classify the uncertain regions into figural and non-figural objects. The extracted figure can be further synthesized into an optional cinestrip. Experimental results show the proposed method successfully extracts the figure object from a complex background environment.

1. Introduction

Moving object extraction is widely used in many applications, such as video

---

Keywords and phrases: moving object extraction, human shape extraction, learning vector quantization neural networks.

This work was supported by the National Science Council Taiwan, R. O. C., under the grants NSC 96-2221-E-224-070.

Communicated by Shun-Feng Su

Received November 15, 2009

compression, sports reporting, surveillance and traffic management system. Researchers have proposed many methods to solve the problems of the moving object extraction [2, 5]. However, two potential problems, (1) how to specify the initial moving object and (2) how to extract objects from a complicated background, are still challenge works. This study attempts to overcome these problems by some existing methods. Chang et al. [1] proposed a face tracking method capable of recognizing faces under various illumination conditions and outlining regions with recognized faces as the figure image. We apply the tracking method to acquire possible figure image and prevent further computation on the whole image. Accordingly, we can decrease the computation cost and increase performance of the figure extraction. The extracted figure image contains both the figural region and the non-figural (background) region. To extract figural region from the figure images efficiently, we construct a figure template by averaging the obtained figural regions in a preprocessing procedure. By means of using two thresholds, the figure template is categorized as figure, non-figure and unknown regions. The unknown region is further classified as figural and non-figural regions using a learning vector quantization neural network (LVQNN) in this study. The LVQNN developed by Kohonen [6, 7] is a useful tool for pattern classification. The network is trained with the coefficients of the discrete cosine transform (DCT) of the regions acquired from figural and non-figural regions. Once the training process is completed, the trained networks can be used to classify the unknown region. The classified figural regions cooperating with the extracted figural region are taken as the figure object that we extracted from the video of moving object.

Experimental results demonstrate that the proposed method can effectively extract the figure object in a complex environment (background).

This paper is organized as follows. The preprocess and the construction of figure template are briefly described in Section 2. The proposed method is presented in Section 3. Section 4 demonstrates the proposed approach using some experimental results. Conclusions are briefly drawn in Section 5.

2. The Proposed Scheme

To deal the problem mentioned in Section 1, we propose a figure (human shape) extraction strategy in the following two subsections. Figure 1(a) shows the schematic diagram of the proposed scheme. Figure 1(b) is the procedure for constructing the figure template. We first apply a robust face detection method [1] to locate the possible figure image from frames (images) sequence of basketball-shooting videos. For a real-time consideration, the color figure images of pixel-size 200×240 are transformed into 256-gray-level images with the same pixel-size. The extracting process consists of two major procedures: (1) a four-step operation for acquiring figural regions from the figure images, and (2) an averaging operation performed on the figural regions to construct the figure template.

2.1. Figure image construction

Before the major procedure begins, we apply the face detection [1] to locate the figure image. The method can locate face images in $w \times h$ pixels as shown in Figures 2(a) and 2(b). We assume that the width and height of a body area are proportion to that of the face area in a reasonable ratio. Hence, we define w_1 as $1.5 \times w$ and w_2 as $1.5 \times w_1$, and, similarly, h_2 as $2.5 \times h$ and h_1 as $1.5 \times h$. Under these assumptions, we can acquire the figure image once the face detection [1] is completed. Figure 2(c) is a possible target figure image. The extracted figure image based on Figure 2(c) is shown as Figure 2(d). It is linearly interpolated to resolution of 200×240 pixel-size.

2.2. The four-step operation

Step 1: Minimum within-group variance: The minimal within-group variance [9] is used to roughly separate the figure images obtained by [1] into two groups, figural and non-figural groups. The minimal within-group variance is determined as follows:

$$T = \arg\min_{g}[\sigma_1^2(g) + \sigma_2^2(g)], \tag{1}$$

where g ranges from 0 to 255, and σ_1 and σ_2 are variances of two groups. Through equation (1), we obtain a threshold T which is used to acquire the suspicious figural regions.

Step 2: Subtraction: The second step subtracts the corresponding region of the suspicious figural regions in the initial frame from the figure region obtained in Step 1. The subtraction is defined as

$$F(x, y, t) = \begin{cases} 0, & \text{if } |f(x, y, t) - f(x, y, t_0)| \leq th_d, \\ 1, & \text{otherwise,} \end{cases} \tag{2}$$

where $f(x, y, t)$ and $f(x, y, t_0)$ are the intensity of the figure region at position

(x, y) in frames t and t_0, respectively, and th_d is the threshold. Note that the t_0 frame is a background image without any moving object.

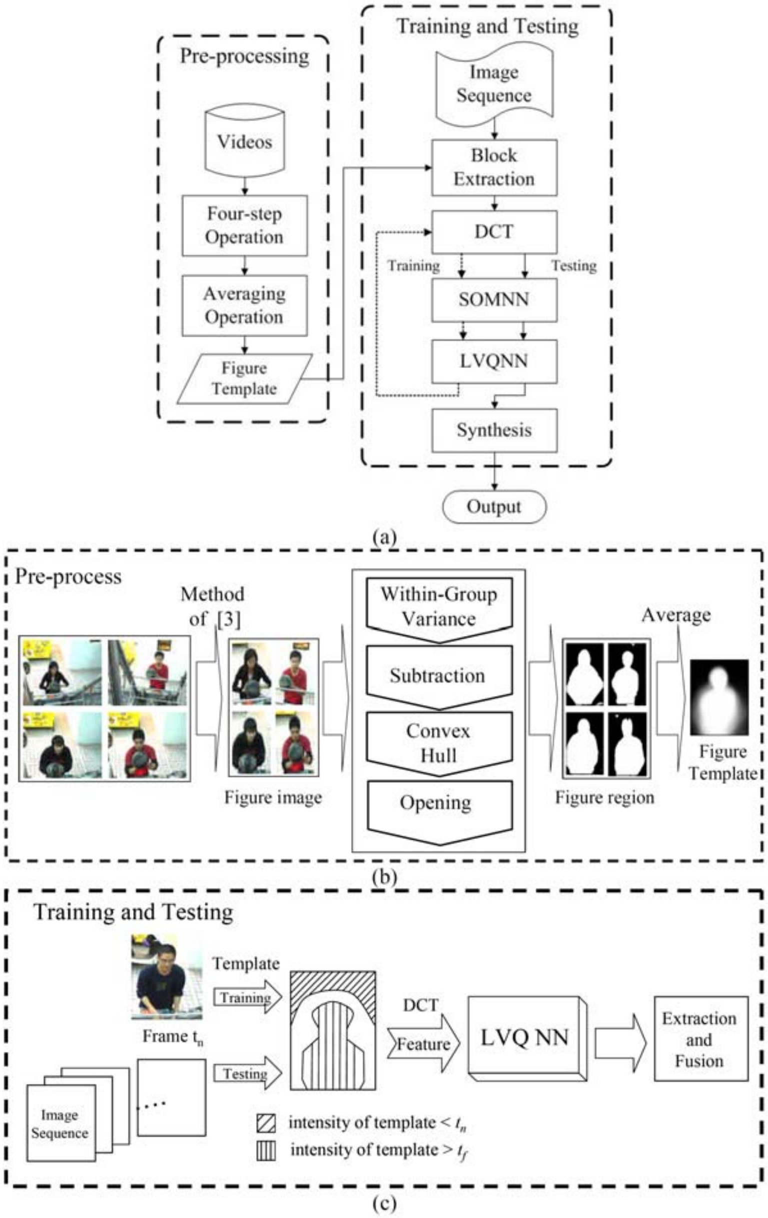
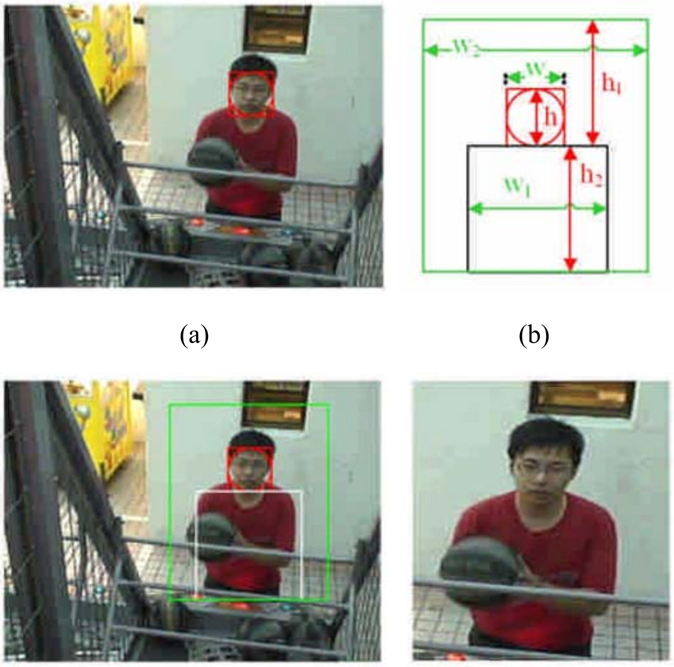


Figure 1. (a) The schematic diagram of the proposed scheme. (b) Constructing the figure template. (c) Fusion of the extracted figure region with selected image.



(a) (b)

(c) (d)

Figure 2. The figure image extraction. (a) The detected face in a video frame. (b) The reasonable ratio on general frames. (c) The reasonable target based on the assumption. (d) The extracted figure image.

The morphologic operations are tools for extracting image components. They are useful in the representation and description of region shape [4]. Two morphologic operations including convex hull and opening are adopted to refine the figure region.

Step 3: Convex hull: Convex hull is a morphological process which finds a minimum set includes all convex in the original region such that any two points in the set lie entirely within the set. The processes for obtaining the convex hull, $C(A)$, of a set A is described as follows:

Let B^i, $i = 1, 2, 3, 4,$ represent the four structuring elements as shown in Figure 3. The procedure consists of iteratively applying the hit-or-miss transform to A with B^i; when no further changes occur, we perform the union with A and call the result

D^i. The procedure is defined as implementing the equation:

$$X_k^i = (X_{k-1} \circledast B^i) \cup A, \quad i = 1, 2, 3, 4 \text{ and } k = 1, 2, 3, \cdots, \tag{3}$$

where $X_k^i = A$, and the operator $\circledast$ denotes the morphological hit-or-miss transform. Let $D^i = X_{\text{conv}}^i$. Then the convex hull of A is obtained by

$$C(A) = \bigcup_{i=1}^{4} D^i. \tag{4}$$

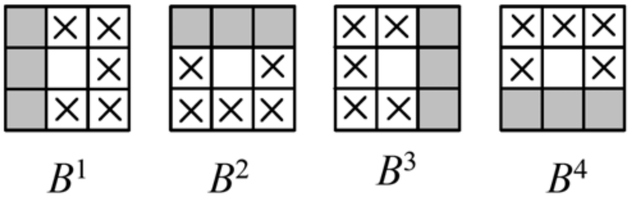


Figure 3. Structuring elements. Graygrids stand for 1 while the × entries indicate do not care.

After the convex hull operation, not only the figure region but also the messy fractions are tend to compact. Thus, an opening operation is applied to smooth the contour of an object, to break narrow isthmuses, and to eliminate thin protrusions.

Step 4: Opening: The opening of set A by structuring element B, denoted as $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B. \tag{5}$$

That is, the opening A by B is the erosion of A by B, followed by a dilation of the result by B. After the opening operation, the smooth figure region *fig* is obtained.

2.3. Average operation

A figure template *Fig* is created by averaging the figure regions as follows:

$$Fig(x, y) = 255 \times \frac{1}{N} \sum_{i=0}^{N} fig_i(x, y), \tag{6}$$

where $fig_i(x, y)$ denotes the figure region in the *i*th figure image. It should be noted that the template *Fig* is a gray scale image.

3. Figure Extraction

Based on the figure images and the figure template obtained in previous section, we can roughly locate the figures from frames. The schematic diagram of the proposed method is shown in Figure 1(c).

3.1. Image feature extraction

The gray scale figure template can be classified into three parts, figure, non-figure, and unknown regions, by two thresholds th_f and th_n. If the intensity of a pixel is larger than the threshold th_f, then the pixel is regarded as the figure region. If the intensity of a pixel is smaller than the threshold th_n, then the pixel is regarded as the non-figure region. If the intensity of a pixel is between th_f and th_n, then the pixel will be regarded as the unknown region. To obtain features for further classification, a labeling process is performed on the figure template using a sliding window method of 8×8 pixel mask and of two pixel-wise sliding. The blocks are labeled using the following criteria:

$$\begin{cases} B_{i,f}, & \text{if all } b_i(x, y) > th_f, \\ B_{i,n}, & \text{if all } b_i(x, y) < th_n, \\ B_{i,u}, & \text{otherwise,} \end{cases} \quad \begin{array}{l} x = 0, \ldots, 7;\ y = 0, \ldots, 7; \\ \text{and } i = 0, \ldots, N-1, \end{array} \tag{7}$$

where $b_i(x, y)$ denotes the intensity of pixel (x, y) in the *i*th block, *N* is the number of blocks; $B_{i,f}$, $B_{i,n}$ and $B_{i,u}$ are the regions in the *i*th block that are labeled as figure, non-figure and unknown, respectively.

We perform the discrete cosine transform (DCT) to obtain the DCT coefficients as features of each block. The coefficients $D(u, v)$ are obtained using the following transformation:

$$D(u, v) = \frac{1}{4} c(u) c(v) \sum_{i=0}^{7} \sum_{j=0}^{7} b_i(x, y) \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}, \tag{8}$$

where $c(u) = 1/\sqrt{2}$ for $u = 0$; $c(v) = 1/\sqrt{2}$ for $v = 0$; $c(u) = 1$ for $v \neq 0$; and $c(v) = 1$ for $v \neq 0$. $b_i(x, y)$ is the intensity of pixel (x, y) in the *i*th block.

The coefficients are organized as the features of the blocks using zigzag scan.

3.2. Categorization with learning vector quantization neural network (LVQNN)

The LVQNN [6, 7] is a supervised learning neural network that can classify input vectors based on vector quantization. The architecture of the LVQNN is shown in Figure 4. The LVQNN is adopted to classify the unknown region $B_{i,u}$ into figure and non-figure regions. In addition, the lower bands in DCT generally contain most of information and the corresponding coefficients are larger than other bands. Thus, the lower-band coefficients of DCT in blocks of $B_{i,f}$ and $B_{i,n}$ are applied to train the LVQNN. After the training process, the unknown region $B_{i,u}$ can explicitly be classified by the LVQNN.

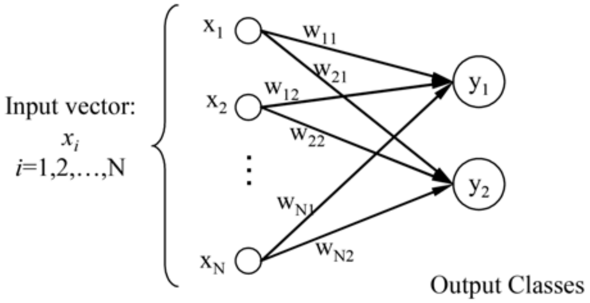


Figure 4. The architecture of LVQNN.

We use the first ten elements of zigzag scan on the DCT coefficients of $B_{i,f}$ and $B_{i,n}$ as the input vectors. In the LVQNN, the input vectors are consisted of thirty DCT coefficients of colors Red, Green and Blue in each block; and the output consists of two classes of figure and non-figure. Then let the synaptic weights between the output neuron j and the inputs $\boldsymbol{w}_j$ as

$$\boldsymbol{w}_j = [w_{j1}, w_{j2}, ..., w_{jN}], \quad j = 1, 2, \tag{9}$$

where N is the dimension of the input vector. For each input vector $\boldsymbol{x}_i$, $\boldsymbol{w}_j$ is determined by the following equation:

$$\min_{\forall j} d(\boldsymbol{x}_i, \boldsymbol{w}_j) = \min_{\forall j} \| \boldsymbol{x}_i, \boldsymbol{w}_j \|_2^2. \tag{10}$$

Let $C_{\boldsymbol{w}_j}$ be the class that is associated with the weight vector $\boldsymbol{w}_j$, and $C_{\boldsymbol{x}_i}$ be the class label of input vector $\boldsymbol{x}_i$ to the network. The weight vector $\boldsymbol{w}_j$ is adjusted in the following manner:

$$\text{If } C_{\boldsymbol{w}_j} = C_{\boldsymbol{x}_i}, \text{ then } \boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) + \mu(k)[\boldsymbol{x}_i - \boldsymbol{w}_j(k)]. \tag{11}$$

$$\text{If } C_{\boldsymbol{w}_j} \neq C_{\boldsymbol{x}_i}, \text{ then } \boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) - \mu(k)[\boldsymbol{x}_i - \boldsymbol{w}_j(k)], \tag{12}$$

where $0 < \mu(k) < 1$ (the learning rate parameter).

We use the labeled figure and non-figure blocks to update the weight of the LVQNN. Once the training process is completed, the unknown block can be categorized using the LVQNN.

4. Experimental Results

Three basketball-shooting videos of different players and different situations are used to evaluate the performance of the proposed method. Besides, two other videos obtained in different environment are also demonstrated. The proposed method was implemented using Borland C++ Builder 6.0 on a system with an Intel Pentium IV 2.8 GHz processor and 1 GB RAM.

In the experiments, the parameters th_d, th_f and th_n, are set as 35, 254 and 20, respectively. The dimension N is 30. The learning rate is 0.05. The maximum iteration is 300 in the LVQNN in this study.

Demonstration on basketball-shooting videos: We demonstrate the proposed scheme which can correctly extract the figure and into different background images in the following cases.

Case I: A player plays street basketball in simple background. It is demonstrated using Figure 5. Figure 5(a) is a video frame where a person is playing street basketball. Figure 5(b) shows the figure image obtained using Chang's method [1]. Figure 5(c) is the classification result of the unknown region. The green regions are labeled as background. Figure 5(d) shows the captured player's figure, which is composed of classification result of the unknown region and the certain figure region. Figure 5(e) shows the extracted figure is synthesized to an optional brick wall.

Case II: A player plays street basketball in background connected with the player in the image. Figure 6(a) is a player in a messy environment. Figure 6(b) is the obtained figure. The player's figure was captured completely. This case shows the proposed method can extract the complete figure even in a complex background. Figure 6(c) shows the synthesized image.

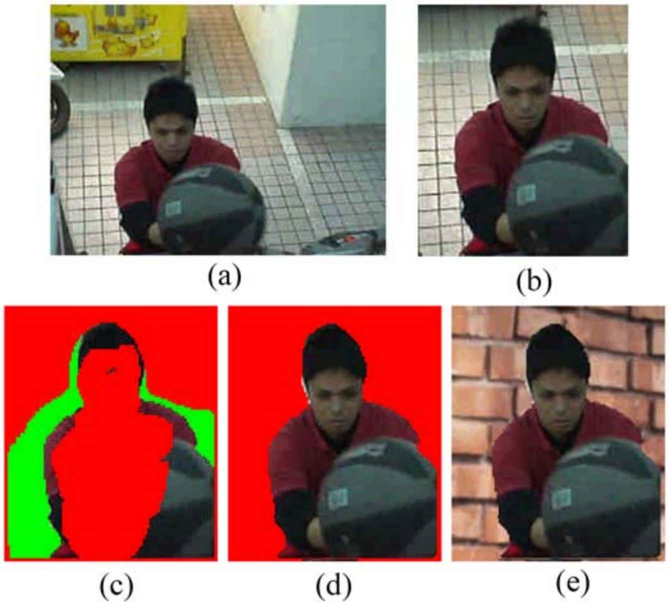


Figure 5. Figure extraction and synthesis in a simple background. (a) Original frame. (b) Figure image. (c) The classification result of unknown region. (d) The complete figure. (e) The synthesized result.

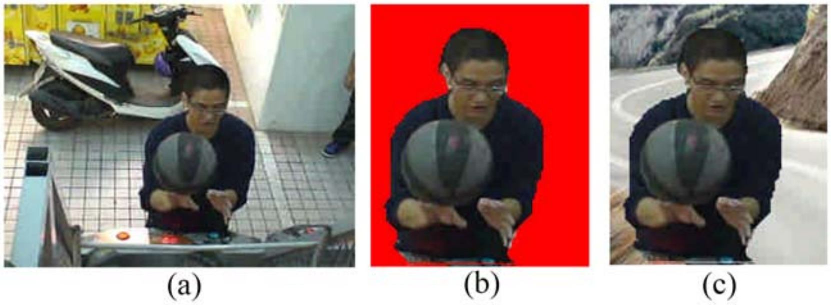
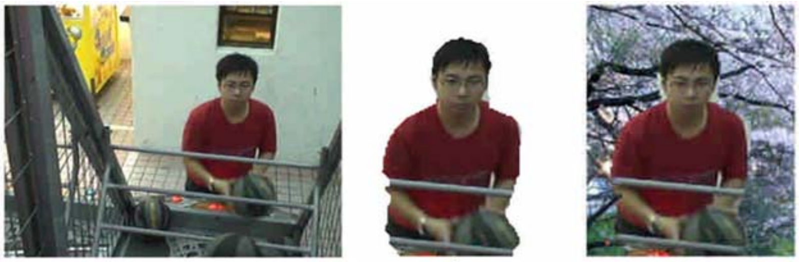


Figure 6. Figure extraction and synthesis in a messy environment. (a) Original frame. (b) The extracted complete figure. (c) The synthesized result.



(a) (b) (c)

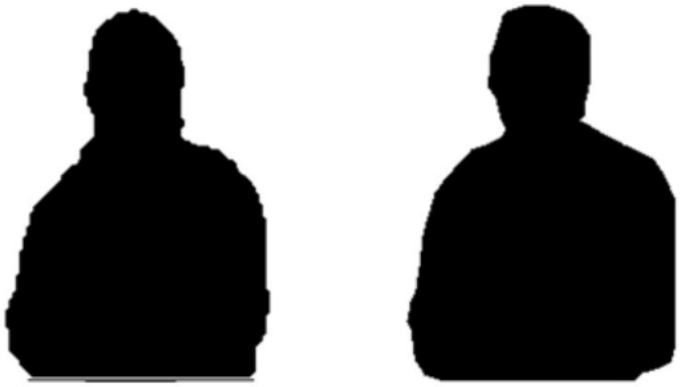
Figure 7. Figure extraction and synthesis in another messy environment. (a) Original frame. (b) The extracted complete figure. (c) The synthesized result.

Case III: A player plays street basketball in some background close to the player's head in the image. Figure 7(a) is a player in a messy environment. Figure 7(b) is the obtained figure. The player's figure was captured completely. This case shows the proposed method can extract the complete figure in a complex background where an object is near the player's head. Figure 7(c) shows the synthesized image.

Performance evaluation basketball-shooting video: In order to evaluate the acquired figure, we compared the extracted figure with that extracted using the manually outlined contour. Accuracy, sensitivity and specificity are adopted to evaluate the performance. Table 1 shows the performance of the proposed method with five video sequences. The average accuracy, sensitivity and specificity of these five cases are 94.32, 92.71 and 96.50, respectively. Therefore, the system can successfully extract figures from images under complex background environment. Figure 8(a) shows the figures extracted by the system. Figure 8(b) is the figure outlined by the engineer. The result shows that the proposed scheme can accurately extract the figure in the basketball-shooting videos.

Table 1. The performance of the proposed method in different video sequences

	Accuracy (%)	Sensitivity (%)	Specificity (%)
Case 1	93.85	91.48	97.66
Case 2	95.41	93.23	97.99
Case 3	93.15	92.95	93.41
Case 4	94.75	93.07	97.08
Case 5	94.46	92.8	96.35
Average	94.32	92.71	96.50



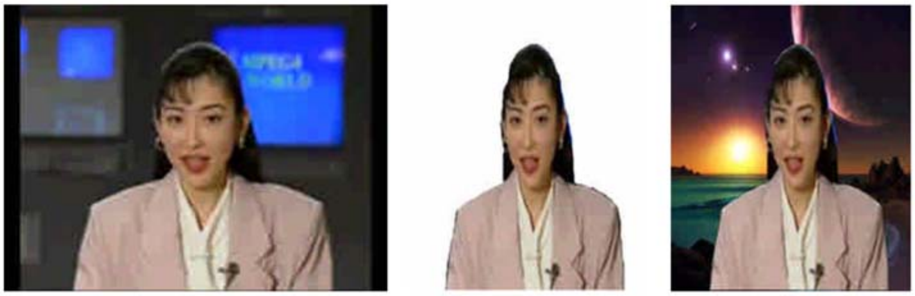
(a) (b)

Figure 8. The extracted figures. (a) The figure extracted by the system. (b) The extracted figure that is manually outlined.

Demonstration on other videos: To compare the proposed method with others, we compare the color video Akiyo and the gray-level video Claire to demonstrate the generality and accuracy of the proposed method. Frames of the videos are demonstrated as follows:

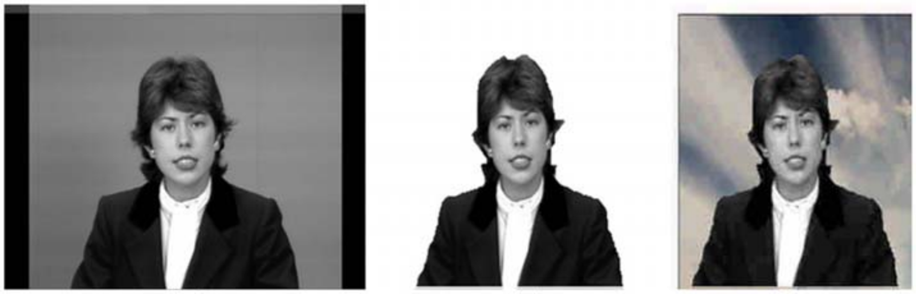
Video frame Akiyo: Akiyo is a video of a reporter. The color of background and foreground differ greatly. However, motion of the foreground in the video stream is not obvious. In addition, the color of background is similar to the reporter's hair in the image. Figure 9(a) is an original image. Figure 9(b) demonstrates the figure that is completely extracted using the proposed method. The demonstration reveals that the background environment does not affect the result in the proposed method. Figure 9(c) shows the synthesized image.

Video frame Claire: Claire is gray-level video of a reporter in front of a gray screen. The background is plain and its color is very similar to part of the person's clothes. The motion of the foreground in the video stream is not obvious. Figure 10(a) is the video frame of Claire. Figure 10(b) demonstrates the figure that is completely extracted in gray-level frame. The demonstration reveals that the proposed approach can deal with most general image whether the image is color or not. Figure 10(c) shows the synthesized image.



(a) (b) (c)

Figure 9. Figure extraction and synthesis in video frame of Akiyo. (a) An original frame of video Akiyo. (b) The extracted complete figure. (c) The synthesized result.



(a) (b) (c)

Figure 10. Figure extraction and synthesis in video frame of Claire. (a) An original frame of video Claire. (b) The extracted complete figure. (c) The synthesized result.

Performance evaluation using other videos: To evaluate the extraction accuracy quantitatively of Doulamis et al.'s method [3], Kompatsiaris and Strintz's method [8], and the proposed scheme, we use the *error* index defined as the ratio of the number of mismatched pixels and the number of total pixels in an image, i.e.,

$$error = \frac{|M_s - M_r|}{N_{pix}}, \tag{13}$$

where M_s is the number of the pixels in the manually outlined region, M_r is the number of the pixels outlined by respective methods, and N_{pix} is the number of pixels in the image. A comparison on error index for the streams of Akiyo and Claire among the three approaches is shown in Table 2. From the results in Table 2, we discover that the proposed method obtains the smallest *error* on the error index of these methods. The proposed scheme extracts the figures more accurately than the other two methods. Once the figure is correctly extracted, they can easily be synthesized to other images.

Table 2. The *error* rate of respective methods

	Doulamis et al. [3]	Kompatsiaris and Strintz [8]	Proposed
Akiyo	0.065111	0.178600	0.006694
Claire	0.066880	0.228703	0.005014

5. Conclusion

An effective method of figure extraction and synthesis is proposed in this paper. By a specifically designed figure template, figure images can be obtained from the frames. With the DCT and the LVQNN the figure region, as a moving image object, is detected. Experimental results show the proposed method can completely extract figures in a messy environment and synthesize to an optional cinestrip successfully.

References

[1] C. Y. Chang, Y. C. Tu and H. H. Chang, Adaptive color space switching based approach for face tracking, Lecture Notes in Computer Science 4223 (2006), 244-252.

[2] S.-Y. Chien, S.-Y. Ma and L.-G. Chen, Efficient moving object segmentation algorithm using background registration technique, IEEE Trans. Circuits and Systems for Video Technology 12(7) (2002), 577-586.

[3] N. Doulamis, A. Doulamis and S. Kollias, Improving the performance of MPEG compatible encoding at low bit rates using adaptive neural networks, J. Real-Time Imaging, Academic Press, 6(5) (2000), 327-345.

[4] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Prentice Hall, New Jersey, 2002.

[5] C. Kim and J.-N. Hwang, Fast and automatic video object segmentation and tracking for content-based applications, IEEE Trans. Circuits and Systems for Video Technology 12(2) (2002), 122-129.

[6] T. Kohonen, Learning vector quantization for pattern recognition, Technical Report TKK-F-A601, Helsinki University of Technology, Finland, 1986.

[7] T. Kohonen, Improved versions of learning vector quantization, Proceedings of the International Joint Conference on Neural Networks, Vol. 1, pp. 545-550, San Diego, CA, 1990.

[8] I. Kompatsiaris and M. G. Strintz, Spatiotemporal segmentation and tracking of objects for visualization of videoconference image sequences, IEEE Trans. Circuits and Systems for Video Technology 10(8) (2000), 1388-1402.

[9] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Systems, Man and Cybernetics 9(1) (1979), 62-66.