計測自動制御学会東北支部 第233回研究集会 (2006.12.18)
資料番号 233-4

An Improved Technique for LVQ-Based Video Object Extraction

スリヤ　スンペノ†, ハリアディ　モハマド‡, 伊藤康一†, 青木孝文†

Surya Sumpeno†, Hariadi Mochamad‡, Koichi Ito†, Takafumi Aoki†

†東北大学大学院情報科学研究科
‡スラバヤ工科大学, インドネシア

† Graduate School of Information Sciences, Tohoku University, Japan
‡ Department of Electrical Engineering, Sepuluh November Institute of Technology (ITS), Surabaya 60111, Indonesia

キーワード： object extraction, learning vector quantization, video processing

連絡先： 東北大学大学院情報科学研究科，〒980-8579 仙台市青葉区荒巻字青葉6-6-05
e-mail: surya@aoki.ecei.tohoku.ac.jp

概 要

This paper presents a semi-automatic algorithm for video object extraction, in which a semantic object of interest is defined in advance in a key frame provided by human. For ordinary video frames, the specified video object is tracked and segmented automatically using Learning Vector Quantization (LVQ). This paper also presents a technique for improving extraction performance of the proposed algorithm and reducing computation time. Experimental evaluation using MPEG standard test video sequences demonstrates that the proposed algorithm is able to extract the video object with low error.

1. Introduction

Today, multimedia users are no longer satisfied to be just passive observers of visual presentation. They want more experiences by interacting with the content of multimedia data they are viewing. The video standards defined by MPEG-4 and MPEG-7 provide standardized technology for representing and manipulating video data, to address this need[1]. MPEG-4 offers object-based representation and compression of video, thus enabling various content-based functionalities for new types of content-based applications, while MPEG-7 provides us with a structured meta-data description for semantically rich media content, along with support for multimedia database indexing. Object-based video processing is required to fully make use of these advanced functionalities. Furthermore, object-based technology is also important for computer vi-

sion applications including gesture understanding, image recognition, augmented reality, etc. However, extracting the shape information of semantic objects from video sequences is a very difficult task, since this information is not explicitly provided within the video data.

Many video object extraction algorithms have been proposed over the years to overcome this difficulty. These algorithms are generally classified into two types i.e., *automatic extraction* (e.g., [2]) and *semi-automatic extraction* (e.g., [3, 4]). Automatic object extraction is usually based on special characteristics of the scene or on specific knowledge (i.e., *a priori* information) such as colors, textures and motions. However, it is very difficult to automatically extract a semantically meaningful object, since the object may have multiple colors, textures and motions [5]. As as a trade-off between automatic segmentation and manual segmentation, many semi-automatic object extraction methods that incorporate interaction of user have been proposed.

Our algorithm for video object extraction belongs to semi-automatic approach and applicable for generic video sequences [6, 7, 8, 9, 10]. The track mechanism is based on LVQ (Learning Vector Quantization) [11], which provides optimal class decision for distinguishing between the object of interest and the background. We use LVQ codebook vectors to maintain the class of each region for tracking the semantic object. Each pixel of a video frame is represented by a 5-dimensional (5-D) feature vector integrating spatial and color features. Spatial feature refers to pixel position in 2-D coordinates, while color feature is represented by YUV color space components. We choose this combination of color and spatial features because they are generic low-level features that are easily applicable to a wide range of video sequences.

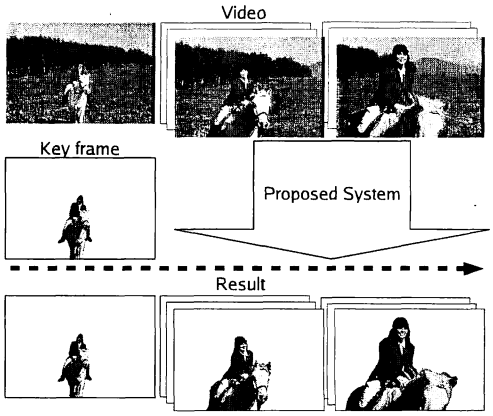

Fig. 1 Illustration of the practical application of proposed algorithm using key frame provided by human.

In this paper, we also present that the extraction performance of the proposed algorithm is improved by tracking the semantic object using only codebook vectors defined around the object. By selectively using codebook vectors, we are able to reduce computation time. Experimental evaluation using MPEG standard test video sequences demonstrates significant extraction performance of the improved algorithm compared with our previous one.

2. LVQ-Based Video Object Extraction System

This section describes a video object extraction algorithm using Learning Vector Quantization (LVQ). Our approach requires a "key" frame in which the semantic object of interest is manually defined by the user. In the key frame, each pixel is represented by a 5-D feature vector with a specific class label (object or background). LVQ is used to approximate the object boundary, where the LVQ codebook vectors are trained by the 5-D feature vectors

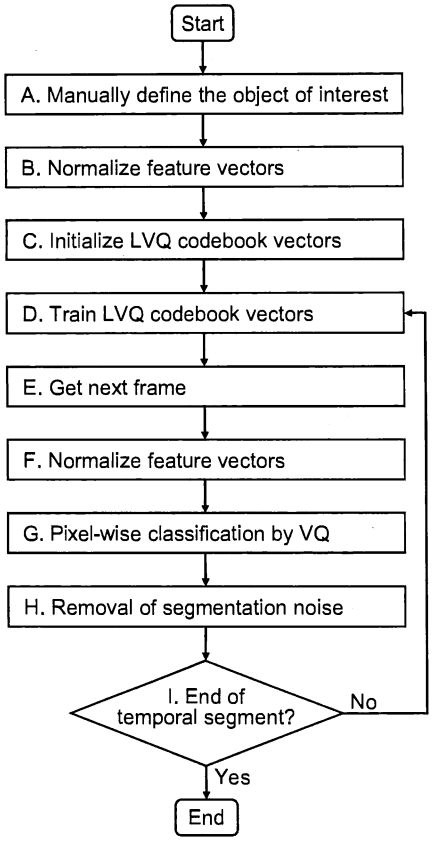

Fig. 2 Flowchart of the proposed system.

of the key frame. In the next frame, the LVQ codebook vectors are used to classify the pixels into object class or background class in order to identify the object of interest thus propagating the semantic information. The classified pixels are then used to update the LVQ codebook vectors for segmenting subsequent frames. This process is repeated until the end of the video sequence.

The system flowchart is shown in Fig. 2. The LVQ codebook vectors are trained in the key frame to approximate the object shape. For the subsequent frames, the object is segmented automatically. For each frame, the LVQ codebook vectors obtained in its previous frame are used to classify the pixels of the current frame into object or background. The classified pixels are then used as training data for supervised learning to update the LVQ codebook vectors for segmenting the next frame, hence propagating the semantic information. This process is repeated until the last frame.

Fig. 3 Codebook vector initialization: (a) codebook vectors used in the previous algorithm, (b) codebook vectors used in the proposed algorithm.

In order to reduce computation time, instead of using all the pixels of the video frame, we previously employed a rectangular window that encloses the object of interest by a small margin[6]. However, the object pixel is sometimes misclassified as background, since the object of interest can not be represented as a rectangular window. In this paper, we use a region fitting the object of interest as shown in Fig. 3. where the region has a small margin. This region is generated from the extraction result of the previous frame which is applied the morphological filter (dilation) to have a margin. We can reduce the number of codebook vectors, reduce the computation time and improve the extraction performance of the algorithm.

Described below are the steps of the flowchart in Fig. 2.

A. Manually define the object of interest

The semantic object is manually defined with user assistance in a particular key frame. Each pixel of the key frame is manually classified as either "object" or "background". This manual segmentation provides the initial training data for the

LVQ codebook vectors in Step D.

B. Normalize feature vectors

Each pixel of the key frame is represented by a 5-D feature vector $x = (\tilde{p}, \tilde{q}, K\tilde{y}, K\tilde{u}, K\tilde{v})$, where pixel coordinates (p, q) and YUV color components (y, u, v) are integrated together to form a single feature vector after normalization process to prevent domination by any one feature [6]. The parameter K is used to adjust the balance between pixel coordinates and YUV color components.

C. Initialize LVQ codebook vectors

Initialization of codebook vectors is done by randomly choosing N pixels from the key frame as codebook vectors. Each codebook vector is assigned the majority class label of pixels within its Voronoi region.

D. Train LVQ codebook vectors

Training of codebook vectors is performed to learn the shape of the segmented object. The codebook vectors are trained using OLVQ1 algorithm [11, 6]. We use 20,000 total training steps and the initial learning rate $\alpha_i(0)$ is 0.4 for all codebook vectors m_i.

E. Get the next frame

The next frame in the temporal segment is loaded.

F. Normalize feature vectors

The feature vectors of the loaded video frame are normalized.

G. Pixel-wise classification by VQ

Pixel-wise classification of the video frame into object class or background class is done to create the segmentation result. The VQ-based classification is carried out using the codebook vectors trained in the previous frame. Each pixel is labeled object or background depending on the class of its nearest codebook vector.

H. Removal of segmentation noise

Color similarities between the background and the object may sometimes result in segmentation noise in the form of small, scattered disjoint regions. This is because the projection of a 5-D Voronoi region onto 2-D image plane is not necessarily continuous. To reduce this noise, we introduce a post-processing step using median filtering for removing the small regions. Thus, we obtain the segmented object.

I. End of temporal segment?

The algorithm terminates if the end of the temporal segment is reached. Otherwise, the segmentation result of Step G is used to update the LVQ codebook vectors for classifying the next frame (repeat from steps D to H).

3. Experiments and Discussion

In our experiments, we evaluate our proposed algorithm using the following MPEG standard test video sequences: *Claire*, *Horse Riding* and *Table Tennis*. The object of interest is manually defined in the first frame, and then the proposed algorithm is used to automatically extract the object of interest. The number of codebook vectors depends on the size of the object to be extracted.

The extraction accuracy is evaluated by comparing the extraction result of our proposed algorithm with that of manual segmentation (ground truth). We manually extracted all frames of each video sequence that was used in our experiments. The extraction error for each frame is given by the following formula:

$$\text{error} = \frac{\text{MO} + \text{AB}}{\text{OB}} \times 100\% \qquad (1)$$

where MO (Missing Object) is number of pixels of misclassified object as background, AB (Added

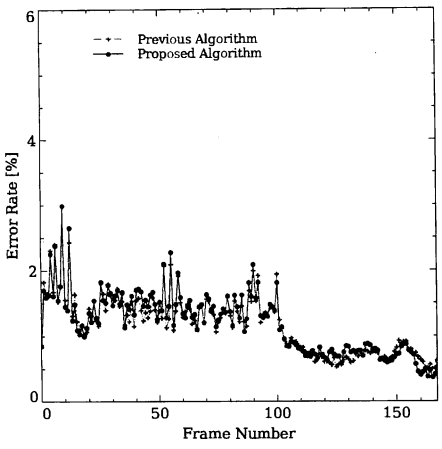

Fig. 4 Frame-by-frame error rate for *Claire* sequence.

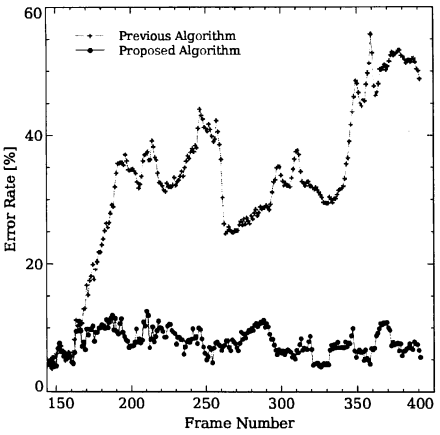

Fig. 5 Frame-by-frame error rate for *Table Tennis* sequence.

Background) is number of pixels of misclassified background as object, and OB is ground truth object's [12]. The extraction quality is directly dependent on the value of the parameter K [6]. In this paper, we employ the best value of K determined empirically. As for the previous algorithm, $K = 3.7$ for *Claire*, $K = 3.6$ for *Horse Riding*, and $K = 0.7$ for *Table Tennis*, respectively. As for the proposed algorithm, $K = 3.2$ for *Claire*, $K = 2.7$ for *Horse Riding*, and $K = 1.2$ for *Table Tennis*, respectively.

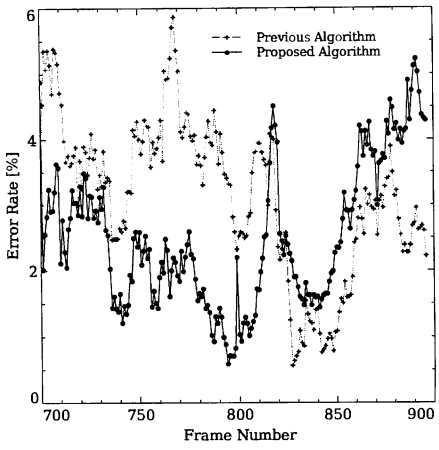

Fig. 6 Frame-by-frame error rate for *Horse Riding* sequence.

Figures 4, 5 and 6 show the frame-by-frame error rate for each sequence. The proposed algorithm exhibits lower error rate than that of the previous algorithm. Table 1 shows the mean error rate of all the frames for each video sequence. Figures 7, 8 and 9 show some images of extracted video object along with the original images.

Table 2 shows the average number of codebook vectors for previous algorithm and the proposed algorithm. The number of codebook vectors used in the proposed algorithm is much less than that of the previous algorithm. Table 3 shows the computation time of the previous algorithm and the proposed algorithm. In the cases of *Claire* and *Horse Riding*, the computation time of the proposed algorithm is much faster than that of the previous algorithm.

As is observed in the above experiments, the proposed algorithm is useful for extracting the video object compared with the previous algorithm.

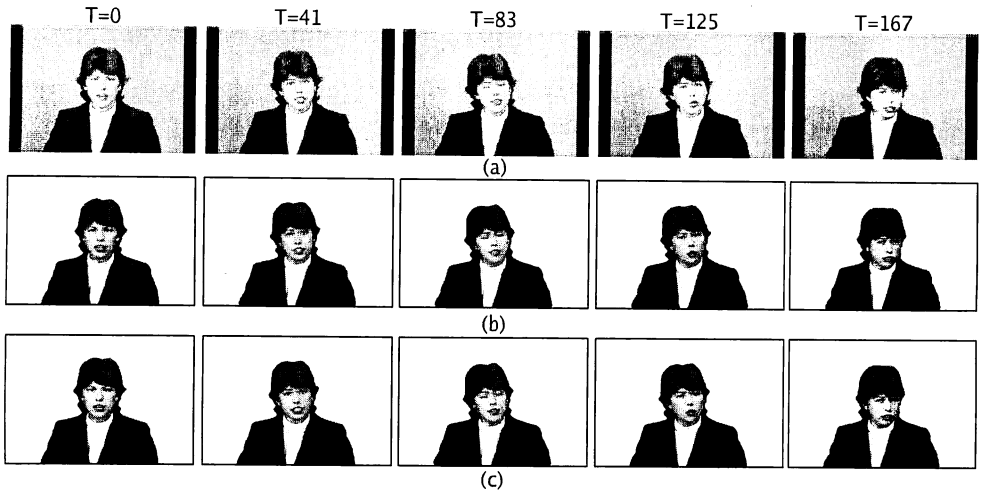

Fig. 7 Object extraction results for *Claire* sequence: (a) original images, (b) extraction results of the previous algorithm and (c) extraction results of the proposed algorithm.

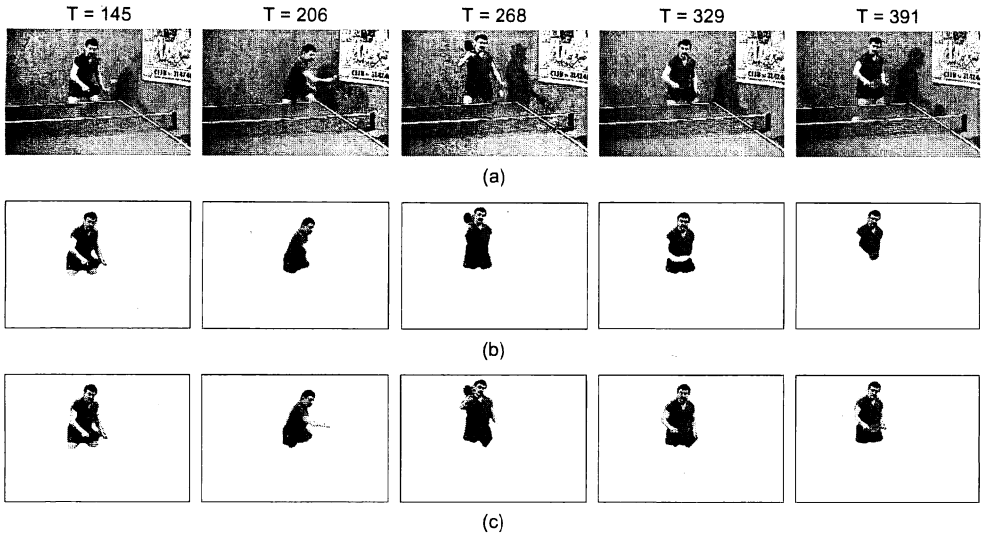

Fig. 8 Object extraction results for *Table Tennis* sequence: (a) original images, (b) extraction results of the previous algorithm and (c) extraction results of the proposed algorithm.

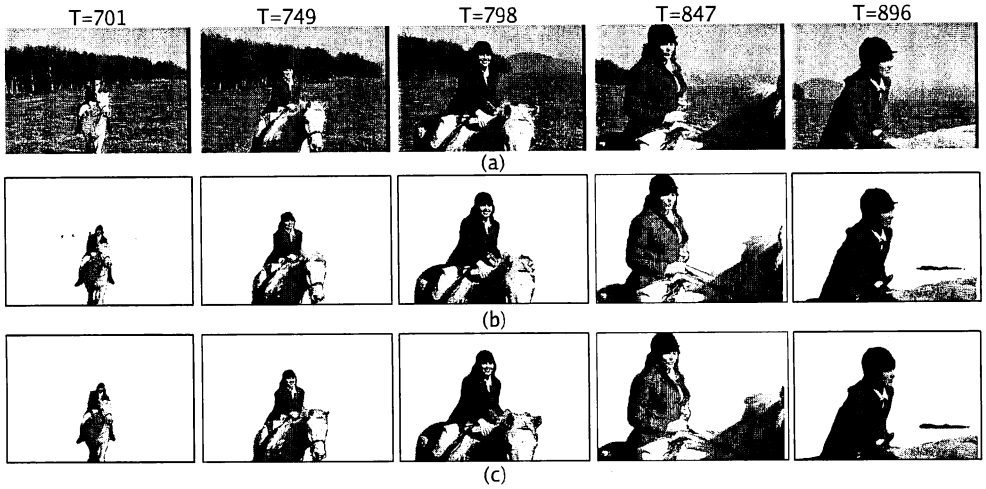

Fig. 9 Object extraction results for *Horse Riding* sequence: (a) original images, (b) extraction results of the previous algorithm and (c) extraction results of the proposed algorithm.

Table 1 Performance evaluation for video sequences.

Sequence	Previous Algorithm Mean Error [%]	Proposed Algorithm Mean Error [%]
Claire	0.60	0.61
Table Tennis	32.60	7.81
Horse Riding	3.12	2.53

Table 2 Average number of codebook vectors.

Sequence	Previous Algorithm	Proposed Algorithm	Reduction Rate [%]
Claire	1450	578	40
Table Tennis	332	120	36
Horse Riding	952	290	30

Table 3 Computation time.

Sequence	Previous Algorithm [sec./frame]	Proposed Algorithm [sec./frame]
Claire	5.49	2.76
Table Tennis	1.94	1.88
Horse Riding	4.10	1.54

4. Conclusion

This paper present a video object extraction algorithm using Learning Vector Quantization (LVQ). Each pixel of a video frame is represented by a 5-D feature vector combining both spatial and color information. The object of interest is manually defined by a user in the key frames. For frames between the key frames, LVQ codebook vectors are used to classify the pixels into object or background. The object extraction performance is improved by selecting the codebook vectors around the object of interest. We have demonstrated that the proposed algorithm can archive consistent extraction of an object of interest.

参考文献

1) H. Kosch, Distributed Multimedia Database Technologies supported by MPEG-7 and MPEG-21, CRC Press, 2003.

2) H. Xu, A.A. Younis, and M.R. Kabuka, "Automatic moving object extraction for content-based applications," IEEE Trans. on Circuits Syst. Video Technol., vol.14, no.6, pp.796–812, June 2004.

3) S. Sun, D.R. Haynor, and Y. Kim, "Semiautomatic video object segmentation using vsnakes," IEEE Trans. on Circuits Syst. Video Technol., vol.13, no.1, pp.75–82, Jan. 2003.

4) C. Toklu, A.M. Tekalp, and A. Tanju Erde, "Semi-automatic video object segmentation in the presence of occlusion," IEEE Trans. on Circuits Syst. Video Technol., vol.10, no.4, pp.624–629, June 2000.

5) A.C. Bovik, The Hand Book of Image and Video Processing, Academic Press Limited, 1st edition, 1998.

6) M. Hariadi, H.C. Loy, and T. Aoki, "Semi-automatic video object segmentation using LVQ with color and spatial features," IEICE Trans. on Inf. Syst., vol.E88-D, no.7, pp.1553–1560, July 2005.

7) M. Hariadi, H.C. Loy, and T. Aoki, "LVQ-based video object segmentation through combination of spatial and color features," Proc. of TENCON, vol.A, pp.211–214, Nov. 2004.

8) M. Hariadi, H.C. Loy, and T. Aoki, "Integrating spatial and color features for LVQ-based video object segmentation," Proc. of ITC-CSCC, pp.7F3P–38–1–7F3P–38–4, July 2004.

9) M. Hariadi, A. Harada, T. Aoki, and T. Higuchi, "An LVQ-based human motion segmentation," Proc. of APCCAS 2002, vol.II, pp.171–176, Oct. 2002.

10) M. Hariadi, A. Harada, T. Aoki, and T. Higuchi, "Pixel-wise human motion segmentation using learning vector quantization," Proc. of 7th International Conference on Control, Automation, and Vision, pp.1439–1444, Dec. 2002.

11) T. Kohonen, Self-Organizing Maps 3rd edition, Springer-Verlag, 2001.

12) P. Villegas, X. Marichal, and A. Salcedo, "Objective evaluation of segmentation masks in video sequences," WIAMIS 99 workshop, Berlin, May 1999, pp 85-88, May 2006.