Scalable Keyframe Extraction Using One-Class Support Vector Machine

YoungSik Choi[1] and Sangyoun Lee[2]

[1]Department of Computer Engineering, Hankuk Aviation University,
200-1 Hwajun-Dong Dukyang-Gu
Kyonggi-Do, Koyang-City, Korea
choimail@hau.ac.kr
[2]Service Development Laboratory, Korea Telecom,
17 Woomyeon-Dong Seocho-Gu
Seoul, Korea,
leesy@kt.co.kr

Abstract. In this paper, we present a scalable keyframe extraction method using one-class support vector machine. Keyframe extraction seeks to generate "good" images that best represent underlying video content and provide content-based access points. Criteria for "good" images play a major role for keyframe extraction process. Extracting "good images" can be viewed as detecting "novel images" among all the frames within a video. Therefore, keyframe extraction reduces to novelty detection problem. We describe how to efficiently solve the novelty detection problem using one-class support vector machine. We also present an algorithm of extracting keyframes in a scalable way so that one can access a video from coarse to fine resolution. We demonstrate the performance of our algorithm on several different types of videos.

1 Introduction

The advances in video data capturing, compression, storage, and communication technologies have made vast amounts of video data available in consumer and enterprise applications. This phenomenal growth of video data has brought the need for efficient video access mechanism. Efficient video access requires more than connecting with data banks and delivering data via networks. It requires improving the accessibility and usability of video data to the point that one can efficiently and effectively search, browse, organize, and manage video data as textual data. This challenge has attracted researchers from various disciplines and has formed a new research area, so called video content analysis and retrieval. Various applications involving education, entertainment, journalism, medical video libraries, and multimedia information services can benefit from this emerging technology.

Among various research areas in video content analysis and retrieval, video abstraction is one of the most important research topics [8]. Video abstraction is the process of creating an abridged version of video, which should be much smaller and

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2660, pp. 491–499, 2003.

still preserve essential information about the original video content. This abstraction process is similar to extracting keywords or creating summaries in text document processing. Video abstraction is especially useful and important when even a short video of a few minutes' duration has the vast amount of data.

There exist several methods for abstracting video content: skims, highlights, and summaries. A video skim is a condensed representation of the original video, containing image and audio sequences [4]. Highlights only convey interesting parts of the video and thus involve detection of particular events in the video [8]. A video summary extracts structural and semantic information about the video, and normally represented by a collection of keyframes [1][2][3]. The temporal order of extracted keyframes can be visualized in a spatial domain so that one can quickly grasp the main content of video. Therefore, keyframe extraction plays an important role in a video summary. This paper will focus on the keyframe extraction schemes.

Keyframe extraction seeks to select good images that best represent the underlying video content and provide content-based access points to video content. The challenge is how to automatically determine which frames are most representative for a given video. The representational power of a video summary largely depends on the criteria for selecting keyframes. The criteria for keyframes vary with respect to the target video at which the video summarization methods aim. The process of keyframe extraction for long videos focuses on extracting semantic and structural information about the whole video content. Extracted keyframes are a collection of images, probably in a hierarchical order that represent events, scenes, or stories. On the other hand, the keyframe extraction process for relatively short videos seeks to select a small number of images that best represent dynamic visual content. Various applications can benefit from this approach. For example, generating thumbnail images for video clips in web-search services can eliminate the painstaking downloading of the entire video clips to check whether the retrieved video clips are the desired ones. Furthermore, condensing video message will be valuable especially in wireless multimedia messaging services.

In [13], the video is segmented into shots and then the first frame of each shot is determined as a keyframe. Although this approach seems a natural way of extracting keyframes, the number of keyframes for each shot is limited to one, regardless of the visual complexity of the shot. That is, a shot boundary image is not necessarily representative for the rest of images. Several algorithms have been proposed to overcome this limitation [11]. Another class of keyframe extraction is to choose the keyframes based on motion metric. In [10], the optical flow of each flow is first computed, and then a simple motion metric is computed. Finally, by analyzing the metric as a function of time, the frames at the local minima of motion are selected as the keyframes. In [14], a domain specific keyframe extraction method is presented using sophisticated global motion and gesture analysis. Mosaic-based keyframe generation is based on detecting specific camera motions [15]. In [9], temporal variations of feature vectors are used to select keyframes. First, several features are extracted from each frame, forming a feature vector trajectory. Then, the frames of local minimal and maximal curvatures in the feature trajectory are chosen as the keyframes. The curvatures are computed as the magnitude of the second derivative of the feature vector trajectory.

Selecting the representative frames can be viewed as detecting the novel frames that best describe visual content. The representatives are novel in that not all frames within a sequence are descriptive, but only some of them descriptive. Therefore, the

keyframe extraction problem becomes the novelty detection problem. In this paper, we propose a scalable shot-based keyframe extraction method based on one-class support vector machine [7]. We first define a measure for compatibility of a frame with its neighboring frames using the temporal variations of its visual features. We select the frames of locally maximal compatibility measure as the novel frames. We next extract the keyframes out of the selected frames according to the novelty of their visual features. Detection of novel features is made by one class support vector machine, which is well known for novelty detection problems [6]. One-class support vector machine can find the feature vectors on the surface of sphere with minimal radius, which encloses all the feature vectors. The feature vectors on the surface are called support vectors. We choose the frames as the keyframes whose features are mapped into the surface of the smallest sphere. Furthermore, we can peel off the surface and obtain the support vectors from yet another enclosing sphere, smaller than the previous. The new support vectors can be added up to the previously obtained support vectors. In this manner, we can obtain a series of keyframe sets, each of which represents video content in its own level of detail.

In the rest of the paper, we briefly describe the one class support vector machine and present the proposed method. We also present the experimental results of our algorithm on several different video clips.

2 One Class Support Vector Machine

A classical unsupervised learning is density estimation. Assuming that the unlabeled observations $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ were generated i.i.d according to some unknown distribution, the task is to estimate its density. However, there are several difficulties to this task. First, a density need not always exist: there are distributions that do not possess a density. Second, estimating densities is known to be a hard task. In many applications it is enough to estimate the support of a data distribution instead of the full density. One class SVMs avoids solving the harder density estimation problem and concentrate on the simpler task [6], i.e. estimating quantities of the distribution, i.e. its support. So far there are two independent algorithms to solve the problem in a kernel feature space. They differ slightly in spirit and geometric notation [5][6]. For brevity, we will focus the approach of [5] as it is more in the line of this paper.

Suppose we are given a data set containing N points, $S = \{\mathbf{x}_j, j = 1, \ldots N\}$, with $S \subseteq \mathrm{X}$ and $\mathrm{X} \subseteq \mathrm{R}^{\mathrm{d}}$. Using non-linear transform ϕ from X to some high dimensional feature-space, one-class SVM seeks the smallest sphere of radius R, enclosing all the points $\phi(\mathbf{x}_j)$. This is described by the constraints.

$$\left\|\phi(\mathbf{x}_j) - \mathbf{a}\right\|^2 \le R^2 \quad \forall j,$$

where $\|\cdot\|$ is the Euclidean norm and $\mathbf{a}$ is the center of the sphere. Introducing slack variable ξ_j forms soft constraints.

$$\left\|\phi(\mathbf{x}_j) - \mathbf{a}\right\|^2 \le R^2 + \xi_j \quad \forall j \tag{1}$$

with $\xi_j \geq 0$. One-class SVM solves the problem by minimizing the following objective function with inequality constraints (1).

$$R^2 + C\sum \xi_j \tag{2}$$

where C is a constant and $C\Sigma\xi_j$ is the penalty term. One can solve this problem introducing the Lagrangian multipliers $\beta_j \geq 0$ and $\mu_j \geq 0$ as follows.

$$L = R^2 - \sum_j (R^2 + \xi_j - \left\|\phi(\mathbf{x}_j) - \mathbf{a}\right\|^2)\beta_j - \sum_j \xi_j \mu_j + C\sum_j \xi_j \tag{3}$$

Setting to zero the derivative of L with respect to R, $\mathbf{a}$, and ξ_j, respectively, results in

$$\sum_j \beta_j = 1 \tag{4}$$

$$\mathbf{a} = \sum_j \beta_j \phi(\mathbf{x}_j) \tag{5}$$

$$\beta_j = C - \mu_j . \tag{6}$$

The KKT complementary conditions lead to

$$\xi_j \mu_j = 0 , \tag{7}$$

$$(R^2 + \xi_j - \left\|\phi(x_j) - a\right\|^2)\beta_j = 0 . \tag{8}$$

From equation (6) we have $0 \leq \beta_j \leq C$. If β_j is zero, μ_j is C and ξ_j is zero. From (8), $R^2 - \left\|\phi(\mathbf{x}_j) - \mathbf{a}\right\|^2 \geq 0$ and thus point $\mathbf{x}_j$ is inside or on the surface of the sphere. If $0 < \beta_j < C$, $0 < \mu_j <$ C and ξ_j is zero. From (8), $R^2 - \left\|\phi(\mathbf{x}_j) - \mathbf{a}\right\| = 0$, and therefore, point $\mathbf{x}_j$ is on the surface of the sphere. Such a point will be referred to as a support vector. If $\beta_j = C$, μ_j is zero and $\xi_j \geq 0$. Then, $R^2 - \left\|\phi(\mathbf{x}_j) - \mathbf{a}\right\| \leq 0$ from (8) and thus, point $\mathbf{x}_j$ lies outside the sphere. This will be called a bounded support vector.

One can reformulate equation (3) as a function of the variables β_j by substituting equations (4), (5), and (6) into L. This leads to

$$W = \sum_j \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_j)\beta_j - \sum_{i,j} \beta_i \beta_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) , \tag{9}$$

with $0 \leq \beta_j \leq C$, (4), and (5). One can replace the dot product in equation (9) by a Mercer Kernel. Throughout the paper, we use the Gaussian kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-0.5\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / \sigma^2) , \tag{10}$$

with a scale parameter σ^2.

One can solve equation (9) using the Quadratic Programming. However, we use the SMO-like (Sequential Minimal Optimization) method [7] since (9) has relatively

simple constraints. The SMO repeats the procedure of selecting two training points and updating the corresponding β_j's with box constraints [7] until all training points satisfy the KKT conditions. We next briefly describe the elementary updating rules and the overall optimization procedure.

Suppose that we chose two points whose β's are β_1^* and β_2^*, respectively. We want to update the two variables so as to optimize equation (9) while unchanging the sum of the two variables. That is, we want not to violate the linear constraint (4) while updating the variables. Therefore, equation (9) becomes a function of β_2. Note that the equation can be re-written as a function of β_2 using the fact that $\beta_1 + \beta_2 = \beta_1^* + \beta_2^*$ = constant. Setting to zero the derivative of (9) with respect to β_2 yields

$$\beta_2 = \beta_2^* + \frac{O_1 - O_2}{K_{11} + K_{22} - 2K_{12}}, \tag{11}$$

where $O_i = \sum_j \beta_j K(\mathbf{x}_i, \mathbf{x}_j)$ and $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. SMO selects a first β for the elementary optimization step in one of the following two ways: (1) SMO scan over the entire data set until it finds a variable violating a KKT condition. Then, SMO choose the next variable β_j according to $j = \arg\max |O_i - O_j|$. (2) Same as (1) except that SMO scans over non-bounded support vectors. Refer to [7] for the detailed procedure.

3 Scalable Keyframe Extraction

The scalable keyframe extraction method proposed in this paper consists of three phases: (1) Selecting the locally representative frames based on a compatibility measure, (2) Detecting the novel frames among the frames obtained from phase (1) using one-class SVM, and (3) Constructing a scalable keyframe-based video summary.

3.1 Compatibility Measure

We denote a given frame sequence as $\{s_1, \ldots, s_N\}$ and its corresponding visual feature vector sequence as $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where s_i and $\mathbf{x}_i$ are i-th frame and its visual feature vector, respectively, and N is the total number of frames within a given sequence. We define the compatibility measure C_i of frame $\mathbf{x}_i$ with its neighbors as follows:

$$C_i = \left(\sum_{j \in W} \frac{1}{|W|} \phi(\mathbf{x}_j) \right) \cdot \phi(\mathbf{x}_i) = \frac{1}{|W|} \sum_{j \in W} K(\mathbf{x}_i, \mathbf{x}_j) \tag{12}$$

where W represents a set of s_i's neighboring shots, $|W|$ cardinality. Note that $K(\mathbf{x}_i, \mathbf{x}_j)$ is the Gaussian kernel function as defined in equation (10). Equation (12) computes the dot product of $\phi(\mathbf{x}_i)$ and the mean value of its neighbors in the Gaussian kernel space. That is, the compatibility of $\mathbf{x}_i$ with its neighbors is defined as the similarity to its

neighbors in the Gaussian kernel space. It is worthwhile to note that the Gaussian kernel function is very similar to the bell-shaped fuzzy membership function and therefore, our compatibility measure can be considered as a fuzzy compatibility measure [12]. We compute the compatibility for each frame in a given sequence. We select the frames of locally maximal compatibility measures, each of which best represents its neighbors in a compatibility sense in (12). Note that the visual feature in our compatibility definition could be obtained from color, texture, shape of the salient object in the frame, or the combination of the above. In this paper, we select the color histogram of a frame as our visual feature, although other visual features can be readily integrated into the compatibility measure. The color histogram used is a 6×6×6 RGB color histogram. Figures 1 shows the compatibility measures for a hand movement video clip of 380 frames obtained from the Web. Black dots represent local maxima in the Figures. We fixed the scale parameter σ^2 in equation (10) into the standard deviation of the feature vectors within a given video.

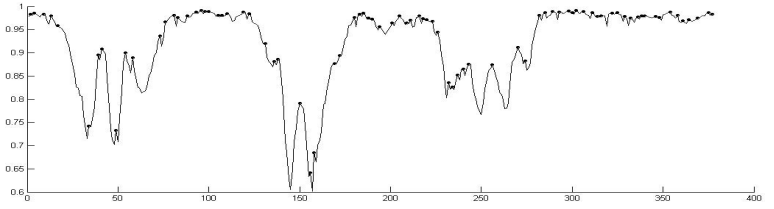

Fig. 1. Compatibility measurements for a hand movement video clip of 380 frames

3.2 Scalable Novelty Detection

One-class SVM described in section 2 has been successfully used in the novelty detection problems [6]. One can control the fraction of outliers and support vectors by varying the value of C. Let $C = 1/(nv)$ where n is the number of training samples and $v \in (0, 1]$. The following statements hold (1) v is an upper bound on the fraction of outliers and (2) v is a lower bound on the fraction of support vectors. These statements directly follow from $0 \le \beta_j \le C$ and $\sum \beta_j = 1$. If we set C to 1, then one-class SVM does not allow the outliers and finds the smallest sphere that encloses all the training data.

The proposed scalable keyframe detection algorithm sets C to 1 and finds the support vectors on the surface of the smallest sphere that encloses all the training points. The obtained support vectors form a set of keyframes. After finding the support vectors on the outer surface and removing them from the training points, the algorithm seeks a new set of support vectors from the rest of training samples. That is, we peel off the surface and obtain yet another enclosing sphere, smaller than the previous. The support vectors from the new enclosing sphere can be added up to the previously

obtained support vectors. In this manner, we can obtain a series of keyframe sets, each of which represents visual content in its own level of detail. Figure 3 shows the result on a synthesized data set. The line in Figure 3 represents the contour of the surface of the sphere in each level.

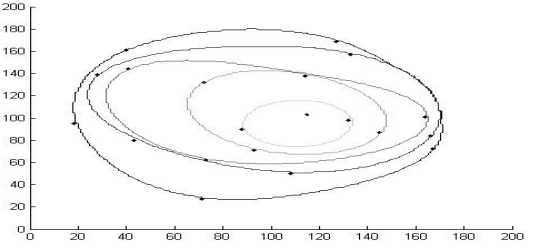

Fig. 2. Scalable Novelty Detection: Contours of the Surfaces of the Smallest Enclosing Spheres

4 Experimental Results

We have experimented with several video clips from the Web. However, we present one of them for the sake of space. In this experiment, we used an MPEG-1 video clip of 380 frames. The $6 \times 6 \times 6$ color histogram in RGB color space was used as a visual feature. The scale factor σ^2 in the Gaussian Kernel function was adaptively adjusted to the standard deviation of a given video clip. Filtering the original sequence with the compatibility measure made 80 frames left. Then, we applied one-class SVM to the remaining 80 frames. Figure 3 shows the result. In order to see the effectiveness of the result, we uniformly sampled every 20 frame from the original sequence. Figure 3(a) shows the uniformly sampled frame sequence. In Figure 3(a), the hand movements are not well captured because the hand movements occur in a burst way within the sequence. Figure 3(b), (c), and (d) show the results from our method. We can see the hand movements well captured even in the first level.

5 Conclusions

In this paper, we presented a scalable keyframe extraction method. Our method is based on the observation that the keyframe extraction problem can be interpreted as the novelty detection problem. For temporal outlier detection, we defined the compatibility measures. For novelty detection in visual features, we used one-class support vector machine. Moreover, we showed how to present a keyframe-based video summary in a scalable fashion. Several experiments on real video clips show the effectiveness of the proposed algorithm.

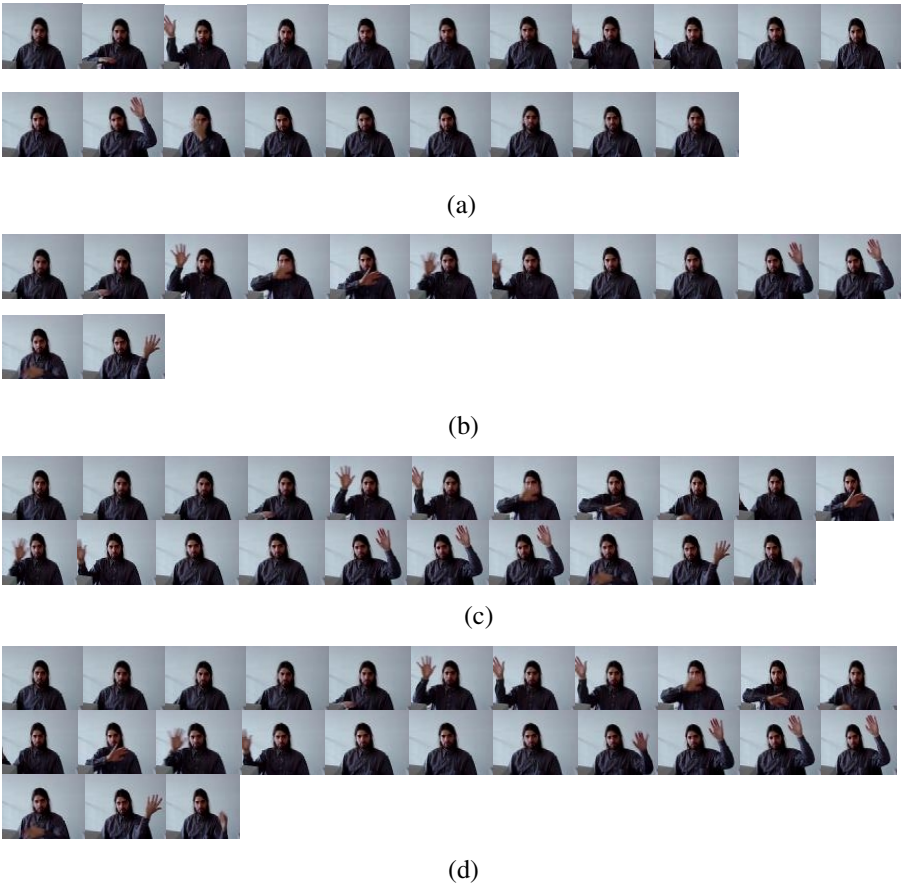

Fig. 3. (a) Result from sampling every 20 frame, (b) Result from our method at level 1, (c) Result from our method at level 2, and (d) Result from our method at level 3

Acknowledgement. This research was supported by IRC (Internet Information Retrieval Research Center) in Hankuk Aviation University. IRC is a Kyounnggi-Province Regional Research Center designatd by Korea Science and Engineering Foundation and Ministry of Science & Technology.

References

1. M Yeung and Boon-Lock Yeo, and Bede Liu “Extracting Story Units from Long Programs for Video Browsing and Navigation”, Proceedings of IEEE International Conference on Multimedia Computing and Systems1996, pp. 296–305.

2. Shingo Uchihashi and, et al, “Video Magna: Generating Semantically Meaningful Video Summaries”, Proceedings of ACM International Conference on Multimedia 1999, pp. 383–391.
3. A. Hanjalic and et al, “Automated High-Level Movie Segmentation for Advanced video-Retrival Systems”, IEEE Transactions On Circuits and Systems for Video Technology, Vol. 9, No. 4, June 1999, pp. 580–588.
4. Michael A. Smith and Takeo Kanade, “Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques”, Proceeding of IEEE, pp.775–781. 1997
5. Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik, “Support Vector Clustering”, Journal of Machine Learning Research 2 (2001) 125–137
6. Klaus Robert Muller, Sebastian Mika, Gunnar Ratsch, Koji Tsuda, and Bernhard Scholkopf, “An Introduction to Kernel-Based Learning Algorithms”, IEEE Transactions on Neural Networks, Vol. 12, No.2, March, 2001, pp.181–202.
7. Nello Cristianini and John Shawe-Taylor, “An Introduction to Support Vector Machines and other kernel-based learning methods”, Cambridge University Press 2000.
8. Nevenka Dimitrova, Hong-Jiang Zhang, Behzad Shahraray, Ibrahim Sezan, Thomas Hunag, and Avideh Zakhor, “Applications of Video-Content Analysis and Retrieval”, IEEE Multimedia, July-September 2002, pp. 42–55.
9. A. D. Doulamis, N. Doulamis, and S. Kollias, “Non-sequential Video Content Representation Using Temporal Variation of Feature Vectors”, IEEE Transactions on Consumer Electronics, Vol. 46, No.3, August 2000, pp.758–768.
10. W. Wolf, “Key frame selection by motion analysis”, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing”, 1996, pp.1228–1231.
11. P.O. Grelsle and T. S. Hunag, “Gisting of video documents: A key frames selection algorithm using relative activity measure,” The 2nd Int. Conf. on Visual Information Systems, 1997.
12. YoungSik Choi, Sun Jeong Kim, and Sangyoun Lee, "Hierarchical Shot Clustering for Video Summarization," Computational Science – ICCS 2002, LNCS 2331.
13. Y. Taniguchi, “An intuitive and effiecient access interface to real-time incoming video based on automatic indexing”, Proc. of ACM Multimedia, Nov. 1995, pp.25–33.
14. S. X. Ju, et. al., “Summarization of video-taped presentations: automatic analysis of motion and gestures”, IEEE Transactions on CSVT, 1998.
15. Y. Taniguchi, et. al., “Panorama Excerpt: extracting and packing panoramas for video browsing”, Proc. of the 5th ACM International Multimedia Conference, pp. 427–436, Nov. 1997.