Shot Boundary Detection Based on SVM Optimization Model

Xuemei Sun*, Yiming Zhang, Xueya Hao and Weidong Min

School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin, China

Abstract: Categorizing the consecutive video frames into shots is the first step for content-based video retrieval. Recently, more and more research has made use of support vector machine to improve the performance of shot boundary detection. However, there has not been a uniform standard for selecting parameters of support vector machine kernel so that it relies on numerous experiences to try, which is not only time-consuming, but also can hardly obtain satisfactory results. In this paper, two novel algorithms for shot boundary detection are proposed, which based on support vector machine optimized by particle swarm and Tabu search respectively. The features are organized into a multi-dimension vector by using the method of sliding window. Experimental results show the effectiveness and robustness of the proposed algorithms, and the performance of support vector machine optimized by Tabu search is better than that of Particle swarm optimization algorithm.

Keywords: Particle Swarm Optimization (PSO), Shot boundary detection, Support vector machine (SVM), Tabu search (Tabu).

1. INTRODUCTION

As more and more video data are generated, spread, and accessed all over the world, the efficient methods for retrieving and searching video data according to people's needs are becoming more and more important and urgent. In the content-based video retrieval, the first step is to segment a video sequence into single shot, which is comprised of a sequence of consecutive frames shot by the same camera [1-3]. There are two basic types of shot transitions: cut transition and gradual transition. A cut transition often takes place over a single frame and a gradual transition which has several forms occurs over a sequence of frames gradually.

Over the past decade, shot boundary detection (SBD) has obtained a considerable amount of research, and a large variety of methods have been proposed to address the problem. Lately, more and more research has made use of machine learning methods to improve the performance of SBD. Reference [4] presents a unified model for detecting different types of video shot transitions. Based on the proposed model, it formulates frame estimation scheme using the previous and the next frames. Unlike other shot boundary detection algorithms, instead of properties of frames, frame transition parameters and frame estimation errors based on global and local features are used for boundary detection and classification. Reference [5] proposes an efficient one-pass algorithm for shot boundary detection and a cost-effective anchor shot detection method with search space reduction, which are unified scheme in news video story parsing.

Reference [6] adopts KNN classification, naive bayes probability classification methods and SVM to categorize sequence frames into cuts and non-cuts; for the non-cuts, it makes use of wavelet denoising method to detect gradual frames, thus completing the whole shot segmentation. Reference [7] presents a novel feature which is consistent with human visual attention, and it constructs the feature into a multi-dimension vector to be categorized with SVM. Furthermore, it uses the brightness feature to assist detection. Reference [8] presents an algorithm for shot boundary detection based on SVM in compressed domain. It uses sliding window method to organize the features into a multi-dimension vector, and then segments a video into shots by the classifying model, which completes shot boundary detection finally.

However, there has not been a uniform standard for selecting parameters of SVM kernel so that it relies on experience or numerous trials, which is not only time-consuming, but also can hardly obtain satisfactory results.

In this paper, algorithms for SBD based on support vector machine (SVM) respectively optimized by PSO-SVM and Tabu-SVM optimized classification models are proposed. Firstly, features of color histogram and wavelet texture are extracted from videos, and then organized into a multi-dimension vector by using the method of sliding window. Following that, the PSO and Tabu algorithms are utilized to implement the simulation and iterative optimization towards parameters of SVM kernel function respectively, then the models trained by the approximately optimal parameters are applied to judge and classify the frames of videos, thus SBD is completed. Experimental results show the effectiveness and robustness of the proposed algorithms.

2. FEATURE EXTRACTION

Color histogram represents the ratio that different color occupies in an original frame and can be extracted by computing the number of pixels in each color bin. Color histogram has been proved to have a good effect on detecting shot

1874-4443/14 2014 Bentham Open

transitions with global camera motion and local object motion. To make sure whether two shots are separated with a hard cut or a gradual transition, we need to compute the inter-frame differences. We calculate the histogram bin value for each single channel belonging to the original frame, and utilize the λ^2 distance to define the dissimilarities.

$$d\left(H_j,H_{j+n}\right)=\left(\sum_{i=1}^{k}\frac{\left(H_j(i)-H_{j+n}(i)\right)^2}{\max\left(H_j(i),H_{j+n}(i)\right)}\right) \tag{1}$$

Where i denotes the bin index, H_j represents the color histogram.

Brightness variance has quite a high accuracy for detecting dissolve and fade in/fade out. When the two kinds of gradual transitions happen, the brightness variance value shows two characteristics. On the one hand, the transition frames show an obvious drop in contrast [9]. On the other hand, the brightness variance shows a continuous and identical change between the two adjacent shots; meanwhile the brightness variance differs little within a shot. The definition of brightness variance is:

$$S_i=\sum_{(R,G,B)}\sum_{i=1}^{M}\sum_{j=1}^{N}(\bar{p}-p_{i,j})^2 \tag{2}$$

Where $\bar{p}$ represents the average of all pixels from a single channel.

Corner is a kind of contour feature and rather robust to the object motion in the frame. As a kind of key point of frame, corner has the advantages of being stable and having a large amount of information. Compared with another contour feature edge change ratio, its computation load is many times smaller. Generally speaking, corner has a strong robustness for image rotation, image translation and image noises. We can control the number of corners in a frame by modifying or adjusting the minimum quality factor and smallest distance between every two corners on purpose.

Here we conduct the corner feature measure according to the equation (3):

$$X=\sum_{i=1}^{k}\frac{\left(c_i(x)-c_i(y)\right)^2}{\sqrt{\min\left(c_i(x),c_i(y)\right)}} \tag{3}$$

Where k denotes the corner number of which the horizontal and vertical coordinates are $c_i(x)$, $c_i(y)$. Then the corner difference can be obtained according to the equation (4):

$$D_{i,i+1}=\left|X_i-X_{i+1}\right| \tag{4}$$

In line with MPEG principle, DC coefficient can be obtained by means of conducting DCT transformation and quantization process. First, an image is divided into blocks of 8*8. For the DCT transformation, the first value of a block is direct-current component, its value equals the average value of the corresponding block, and it is called DC coefficient. Then the amount can be obtained, which is the differences between DC coefficients of two co-located blocks exceed some certain threshold, it is taken as frame difference of DC coefficients.

3. ORGANIZATION OF FEATURE VECTOR

3.1. Difference Between the Frames

In order to detect the abrupt cuts and gradual transitions synthetically, computation of dissimilarity for video frames is fixed not only between the adjacent frames, but also between the frames with a certain length l, which is called the inter-frame distance. According to the shot transition styles, the inter-frame distance l selected here has four values, which are 1, 2, 3.

The computation of dissimilarity for frames can be represented as:

$$\begin{aligned}D^{l=1}&=\left[d(H_1,H_2),...,d(H_i,H_{i+1}),...,d(H_{N-1},H_N)\right],\\D^{l=2}&=\left[d(H_1,H_3),...,d(H_i,H_{i+2}),...,d(H_{N-2},H_N)\right],\\D^{l=3}&=\left[d(H_1,H_4),...,d(H_i,H_{i+3}),...,d(H_{N-3},H_N)\right],\end{aligned} \tag{5}$$

3.2. Context Feature Vector

Here we adopt the sliding window strategy to handle the inter-frame differences. According to various shot transition boundaries, the window length is set as 20, 40, 50 respectively. For each l frame, the corresponding feature indicator is centered in the preset sliding window. In addition, to optimally define the content of video frames and shot transitions, the normalization for feature dissimilarities is conducted. After concatenating the feature vectors with distinct l, we can obtain the final feature vector as follows:

$$\begin{aligned}FV(i)=[&C_h^{l=1}(i),B_v^{l=1}(i),C_o^{l=1}(i),D_c^{l=1}(i),\\&C_h^{l=2}(i),B_v^{l=2}(i),C_o^{l=2}(i),D_c^{l=2}(i),\\&C_h^{l=3}(i),B_v^{l=3}(i),C_o^{l=3}(i),D_c^{l=3}(i)]\end{aligned} \tag{6}$$

Where $C_h^{l=f}(i),B_v^{l=f}(i),C_o^{l=f}(i),D_c^{l=f}(i)$ denotes the original feature vector respectively for the color histogram, brightness variance corner and DC coefficients under inter-frame distance $l=f$.

4. SHOT BOUNDARY DETECTION BASED ON SVM

4.1. Support Vector Machine

SVM is a new machine learning method proposed by Vapnik in the 1990s, which is based on the statistical theory and principle of structural risk minimization. The principle of SVM classification is to map the input space into a very high dimensional feature space and construct an optimal separating hyper plane in the corresponding space, and obtain the decision function of classifier [10, 11]. SVM is not only with simple geometric explanation but also elegant formulation as a quadratic optimization problem. Because of the convexity of the quadratic optimization problem with linear and box constraints, the global optimum categorizing and regressing solution can be guaranteed.

In the SVM classification problem, the problem of supervised learning is formulated as follows. Given a set of train-

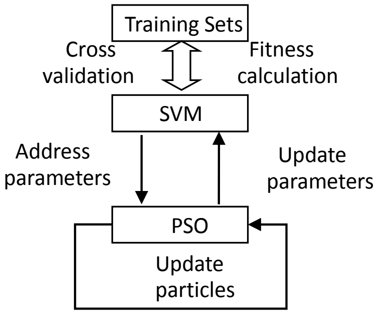

Fig. (1). Process of obtaining SVM mode.

ing data, $\{(x_i, y_i)\}$, where $i = 1, \dots, n$, $x_i \in R^n$, $y_i \in \{-1, +1\}$, and the decision function of SVM can be described as follow:

$$f(x) = sgn(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b) \quad (7)$$

Where α_i ($i = 1, 2, \dots, s$, $s \le l$) denotes Lagrange factor, and $K(x_i, x)$ denotes kernel function. The paper selects radial basis function (RBF) as kernel function, which is defined as:

$$K(x_i, x) = \exp(-\gamma \|x_i - x\|^2), \gamma > 0 \quad (8)$$

The existing studies have shown that the kernel function parameter and penal factor are the key factors affecting SVM's performance. Kernel function parameter mainly influences the complexity of the distribution for sample data in the high-dimension feature space. The change of kernel function parameter demonstrates VC dimension of feature space alters indirectly, and then it affects the fiducially range of SVM, finally leads to the change of structural risk range. Besides, the penalty factor takes charge of the compromise between the maximum interval and classification error. A quite large value for *c* can probably cause the over fitting problem; on the contrary, the under fitting problem will occur when the value of *c* is not large enough.

4.2. Shot Boundary Detection Based on PSO-SVM

Particle swarm optimization is an evolutionary computing technology which is inspired by the artificial life research and simulates the process of migration of group behaviors. It abstracts each individual in the group as a particle without mass and volume. In the iterative process, each particle modifies its direction and velocity by the optimal value produced by it and others; thereby the positive feedback mechanism is formed during the group optimization process. Based on the characteristics above, we select PSO algorithm to optimize two parameters in SVM, completing the iterative optimizations by simulate them into particles.

As we can see in Fig. (**1**), the flow diagram shows the forming process of SVM model. Through the whole operations disposed in the figure, a categorizing model trained by the optimal parameters will be obtained.

Group initialization and parameters settings: The population is consisted of 20 particles whose velocities are produced randomly. Besides, the parameters c and γ is designed to vary in the range of [0,100] and [0, 1000] respectively, with the step is 0.1. c_1, c_2 is set as 1.4 and 1.8 by taking the local search capacity and global search capacity into consideration synthetically.

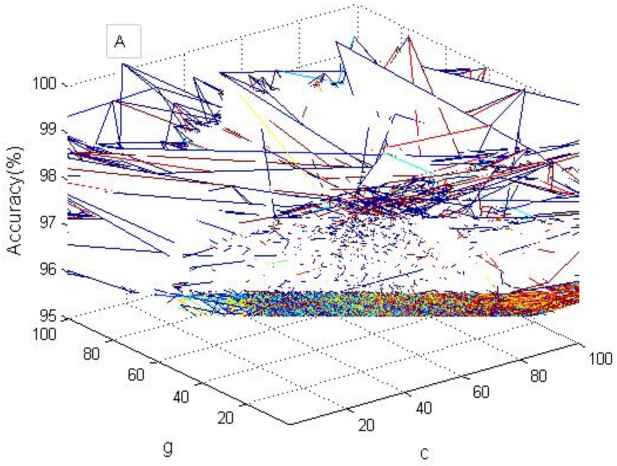

Fig. (2). Parameter optimization process.

Evaluate the fitness: The function $f(c, \gamma) = accuracy$ is adopted to evaluate the fitness. In this paper, the strategy of cross validation is utilized to compute the fitness which equals the value of accuracy of training sets. In other words, the accuracy shows positive correlation with the fitness. The training sets is divided into 3 parts in the cross validating operation.

Update the velocity and position: When the optimal local solution and global solution for the current generation are obtained, the particles are updated according to the velocity and position updating equations, which are defined as:

$$v_{i+1} = \omega v_i + c_1 rand(pbest - x_i) + c_2 rand(gbest - x_i) \quad (9)$$

$$x_{i+1} = x_i + v_{i+1} \quad (10)$$

Where rand denotes a random number between 0 and 1; ω denotes the inertia weight factor which is used to control the effect from the previous particles. x_i and vi represents the particle position and velocity respectively; and *pbest*, *gbest* represents the local optimal and global solutions.

Update the local fitness and global fitness: If the current fitness is prior to the local best fitness, replaces the *pbest* with current value, meanwhile updates position the particle occupies; and if the current fitness is prior to the global best fitness, replaces the *gbest* with current value, meanwhile updates position the particle stays.

Iterative times: If the iteration reaches the iterative times or achieves quite great a solution, then stop the iterations and output the optimal solutions; otherwise, go to step which is evaluate the fitness.

As is shown in Fig. (**2**), the parameter optimization process indicates that different SVM parameters make prominent differences in classification accuracy. The point A in the figure represents the best predictive situation in terms of the optimal parameters. The implementation of parameter optimization is based on LIBSVM [9], in which the parameter γ is represented as *g*.

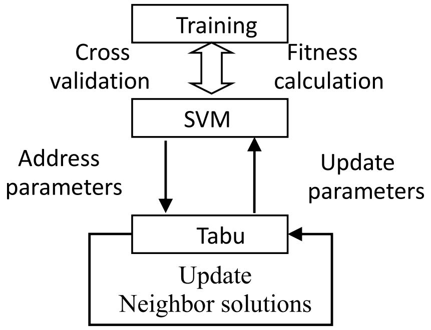

Fig. (3). Building of the classification model.

4.3. Shot Boundary Detection Based on Tabu-SVM

Tabu search was proposed by Glover in 1986. As an extension of the local neighborhood search, it is quite effective to search the optimum global solution. Besides, it is not inclined to fall into a local optimum value because of the existence of tabu list. The main strategies in tabu optimization are set as follows:

Initial solution and fitness function. (c_0, γ_0) denotes an initial solution generated randomly, fitness function $f(c, \gamma) = accuracy$ is adopted to evaluate the fitness, where c denotes the penalty factor and γ denotes the kernel function parameter in SVM.

Neighborhood solutions and tabu object. Around the initial solution (c_0, γ_0), we can get many groups of solutions (c, γ), and each of them will get a fitness value accordingly. Besides, c and γ are designed to vary in the range of [0, 100] and [0, 1000] respectively, with the varying step is 0.1.

Candidate set. The candidate set consists of solutions selected by the fitness function in each generation.

Tabu list. The best solution of the current generation is put into the tabu list whose value refers to the forbidden iterative times, avoiding constant local search. The length of tabu list is set as 10.

Special amnesty. When the fitness value of one taboo solution is higher than the fitness of best solution so far, the solution is allowed to deviate from the tabu list.

Iteration criterion. The strategy adopted here considers two aspects: maximum iterative times and good enough solutions. The pre-set iterative times is 500. As we can see in Fig. (**3**), it shows the construction of classification model via Tabu-SVM. Through the whole evolving procedure displayed, a categorizing model trained by the optimal parameters will be obtained.

5. EXPERIMENTAL RESULTS AND ANALYSES

5.1. Experimental Data

Our algorithm is tested on TREC-2001 video data set. The selected data set is consisted of 30000 frames around; each of the video clips is approximately between 3-minute and 5-minute long. Moreover, it contains conspicuously global camera movements and local object motions with continuous flash scenes, and the frame rate is 29.97fps.

5.2. Performance Measure

To measure the detection performance, we use two conventional performance measures for shot boundary detection, precision and recall [1], which are defined as:

$$Recall = hit / (hit + miss) \tag{11}$$

$$Precision = hit / (hit + false) \tag{12}$$

To have a comprehensive performance measure for comparison, we also use the measure of F_1 in our evaluation. The measure of F_1 is considered to have a favorable compromise between precision and recall, which is defined as:

$$F_1 = \frac{2 * \mathrm{Pr}ecision * \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \tag{13}$$

5.3. Results and Comparisons

Ten clips of videos chosen from the video data set mentioned in the paper are used to test our proposed algorithm, and the partial experimental results derived from six videos are shown as in the (Table **1**).

We detect the shot boundary of the selected video set with the proposed algorithm, then compare the result obtained with the boundary information TREC provided. Furthermore, we compare the experimental results with the results of the algorithms proposed by Reference [6] and Reference [8] under the identical test sets. The result show in table

Table 1. The results of precision and recall.

Video List	*Precision%*		*Recall%*	
	Cut	**Gradual**	**Cut**	**Gradual**
Video 1	98.75	92.75	96.42	85.30
Video 2	96.13	87.32	98.16	90.65
Video 3	96.94	91.27	97.00	81.43
Video 4	97.66	93.46	94.28	68.70
Video 5	98.64	92.77	94.74	87.15
Video 6	99.50	89.92	92.95	80.77

Table 2. Comparative results of F_1.

RUNS	F_1%		SUM_F_1%
	Cut	Gradual	
RUN-1	94.24	69.83	82.04
RUN-2	95.77	83.92	89.85
RUN-3	96.84	84.75	90.78
RUN-4	97.75	85.64	93.52

(2), where RUN-1 represents the algorithm in Reference [6], RUN-2 represents the algorithm proposed by Reference [8], RUN-3 and RUN-4 respectively represent Shot Boundary Detection based on PSO-SVM, Shot Boundary Detection based on Tabu-SVM in this paper.

We can see from the table, the algorithms proposed in this paper are superior to other two methods in the measure of F_1, RUN-1(Reference [6]) method adopts a hierarchical detection method which utilizes two-category classifier on each layer to detect the video shot boundary, and only one layer adopts the classifier based on SVM. RUN-2(Reference [8]) algorithm also uses the support vector machine, but did not support vector machine to optimize the parameters affecting the performance of the algorithm. The algorithms proposed in this paper use the PSO-SVM and Tabu-SVM respectively, archiving good performances in shot boundary detection. Furthermore, the Tabu-SVM preforms better than the PSO-SVM.

CONCLUSION

In this paper, an algorithms for SBD based on support vector machine (SVM) optimized by particle swarm optimization (PSO) and an algorithm for SBD based on support vector machine (SVM) optimized by Tabu search (Tabu) are proposed. The algorithms utilize PSO and Tabu search to optimize the parameters of SVM respectively, then classify the video frames and completes the shot boundary detection by using the models trained by the approximately optimal parameters obtained. Experiments show that the algorithms have excellent performances in terms of precision, recall and F_1 on TREC video set 2001.

CONFLICT OF INTEREST

Financial contributions to the work being reported should be clearly acknowledged, as should any potential conflict of interest.

ACKNOWLEDGEMENTS

This research was partially supported by the National Nature Science Foundation of China (No.61173032), the Tianjin Higher Education Science and Technology Development Fund (No.20100806) and the Natural Science Foundation of Tianjin, China (No. 13JCYBJC15500).

REFERENCES

[1] L. Liang, Y. Liu, H. Lu, X. Xue and Y. Tan, “Enhanced Shot Boundary Detection Using Video Text Information”, *IEEE Transactions on Consumer Electronics,* vol. 51, pp. 580-588, 2005.

[2] M. Birinci and S. Kiranyaz, “A perceptual scheme for fully automatic video shot boundary detection”, *Signal Processing Image Communication*, vol. 29, pp. 410-423, 2014.

[3] Z. Lu and S. Yong, “Fast video shot boundary detection based on SVD and pattern matching”, *IEEE Transactions on Image Processing*, vol. 22, pp. 5136-5145, 2013.

[4] P. Mohanta, S. Saha and B. Chanda, “A model-based shot boundary detection technique using frame transition parameters”, *IEEE Transactions on Multimedia*, vol. 14, pp. 223-233, 2012.

[5] H. Lee, J. Yu and Y. Im, “A unified scheme of shot boundary detection and anchor shot detection in news video story parsing”, *Multimedia Tools and Applications*, vol. 51, pp. 1127-1145, 2011.

[6] Y. Qi, A. Hauptmann and T. Liu, “Supervised Classification for Video Shot Segmentation”, in *Proceedings of Multimedia and Expo International Conference*, pp. 689-692, 2003.

[7] X. Li, G. Xiao, J. Jiang, K. Du and K. Qiu, “shot boundary detection based on SVMs *via* visual attention features”, *Information Technology and Applications*, vol. 2, pp. 484-487, 2009.

[8] J. Cao and A. Cai, “Algorithm for shot boundary detection based on support vector machine in compressed domain”, *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 36, pp. 203-208, 2008.

[9] K. Du, G. Xiao and J. Jiang, “Shot boundary detection algorithm based on multiple video features”, *Computer Engineering*, vol. 35, 2009.

[10] C. Chang and C. Lin, “LIBSVM: a library for support vector machines”, *ACM Transactions on Intelligent Systems and Technology*, vol. 2, 2011.

[11] V. Vapnik, The nature of statistical learning theory, In: Information Science and Statistics *Springer*, 2000, p. 314.

Received: September 22, 2014 Revised: November 03, 2014 Accepted: November 06, 2014