

Research Article

Video Shot Boundary Recognition Based on Adaptive Locality Preserving Projections

Yongliang Xiao,^{1,2} Limin Xia,¹ Shaoping Zhu,² Dazu Huang,² and Jianquan Xie²

¹ School of Information Science and Engineering, Central South University, Changsha 410083, China

² Department of Information Management, Hunan University of Finance and Economics, Changsha 410205, China

Correspondence should be addressed to Limin Xia; xlm@mail.csu.edu.cn

Received 7 August 2013; Revised 25 September 2013; Accepted 10 October 2013

Academic Editor: SunQun Cao

Copyright © 2013 Yongliang Xiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel video shot boundary recognition method is proposed, which includes two stages of video feature extraction and shot boundary recognition. Firstly, we use adaptive locality preserving projections (ALPP) to extract video feature. Unlike locality preserving projections, we define the discriminating similarity with mode prior probabilities and adaptive neighborhood selection strategy which make ALPP more suitable to preserve the local structure and label information of the original data. Secondly, we use an optimized multiple kernel support vector machine to classify video frames into boundary and nonboundary frames, in which the weights of different types of kernels are optimized with an ant colony optimization method. Experimental results show the effectiveness of our method.

1. Introduction

Video shot boundary recognition is a fundamental process towards video summarization and analysis. There are many boundary recognition methods already presented [1, 2]. Common method to recognize shot boundary is comparing the difference of two adjacent frames with a threshold. In paper [3], the abrupt shot boundary is detected based on an adaptive threshold and gradual transition boundary is detected with a set of standard templates. Warhade et al. [4] detected shot boundary with cross-correlation coefficient, stationary wavelet transform, and combination of local. Thakar and Hadia [5] proposed a new gradual shot detection method in which the threshold can be adaptively determined based on the total information change of video frames. Huo et al. [6] used a statistical model according to the video frame differences to determine the adaptive threshold. In paper [7], the threshold is automatically determined according to the magnitude of color differences quantification. Warhade et al. [8] first extracted structure features from each video frame by using dual-tree complex wavelet transform and then decided the shot boundary based on the spatial domain similarity. In order to reduce the computation, Gao and Ma [9] used color

histogram and mutual information to measure the difference between frames, and then the corner distribution of frames is utilized to exclude most of the false boundaries.

The main disadvantage of these methods is susceptible to the effect of thresholds, which can make a mistake for some complicated long gradual shots. To resolve this problem, people see the video shot recognition as a categorization task. In paper [10], a fuzzy logic method is used to detect shot boundary. This method contains two processing modes, where one is dedicated to detection of abrupt shot and the other for detection of gradual shot. In paper [11], the video feature including HSV (hue, saturation, value), edge orientation, and texture feature is obtained, and then the Kohonen self-organized network is used to recognize shot boundary. Huang et al. [12] classified video frames with a radial basis function neural network. Mohanta et al. [13] used a multilayer perception network to classify video frames based on local features matrix. To improve the recognition performance, Li et al. [14] first removed some frames from the original video which were clearly not shot boundaries, then used a novel SIFT key point matching algorithm to detect shot boundary. Zhao et al. [15] used context feature vector and Tabu-SVM to recognize shot boundary. In paper [16],

the proposed approach first detected general shot boundary with Fisher criterion and then classified the cut and gradual shot with SVM. In order to improve the effect of SVM, Zhao et al. [17] optimized the parameters of SVM with particle swarm method. In addition, Lankinen and Kamarainen [18] detected shot boundary using a visual bag-of-words approach. Donate and Liu [19] extracted salient features from a video sequence and tracked them over time to estimate shot boundaries. Li and Chen [20] recognized shot boundary with macroblock type information which can save a lot of computation cost.

In this paper, we present a novel method to improve the shot boundary recognition accuracy. Firstly, based on the analysis of LPP, we present an adaptive LPP to extract more useful and discriminating features. Secondly, we recognize shot boundary with an optimized multiple kernel support vector machine.

The rest of this paper is organized as follows. Section 2 abstracts the theoretical fundamentals about LPP. In Section 3, we extract the video feature with improved LPP. Section 4 uses an optimized multiple kernel SVM to recognize shot boundary. Some experiments are used to evaluate the presented method in Section 5 and all the paper is concluded in Section 6.

2. Theoretical Fundamentals

2.1. Locality Preserving Projection. Locality preserving projection (LPP) is a dimensionality reduction method which can be explained by the graph theory [21]. Assume there is m -dimensional data point set $X = [x_1, x_2, \dots, x_n]$; we try to find a project matrix A to project these data point into a low-dimensional subspace, and the projection can be expressed as $y_i = A^T x_i$.

The objective function of LPP is as follows:

$$\min \sum_{ij} \|y_i - y_j\|^2 S_{ij}, \quad (1)$$

where the weight matrix S can be defined as follows:

$$S_{ij} = \begin{cases} 1 & \text{if } x_i \text{ is among } K \text{ nearest neighbors of } x_j \\ & \text{or } x_j \text{ is among } K \text{ nearest neighbors of } x_i \\ 0 & \text{else.} \end{cases} \quad (2)$$

Then, the objective function of LPP can be converted into the following minimization problem:

$$\begin{aligned} \min_A \quad & \text{tr} \{A^T X L X^T A\} \\ \text{s.t.} \quad & A^T X D X^T A = 1, \end{aligned} \quad (3)$$

where $D_{ij} = \sum_j W_{ij}$ is a diagonal matrix and $L = D - S$ is a Laplacian matrix.

Lastly, the project matrix A can be obtained by solving a generalized eigenvalue problem as follows:

$$X L X^T A = \lambda X D X^T A. \quad (4)$$

Let the column vectors a_1, a_2, \dots, a_d be the solutions of (4), ordered according to their eigenvalues, $\lambda_1, \lambda_2, \dots, \lambda_d$; we can define the following transformation form:

$$x_i \longrightarrow y_i = A^T x_i, \quad A = [a_1, a_2, \dots, a_d]. \quad (5)$$

2.2. Weight Definition. In the LPP, the weight between two points is defined to be a simple either 1 or 0 or heat kernel, which cannot reflect the class information. Given $X = [x_1, x_2, \dots, x_n]$, let C_i and $N(x_i)$ be the label and k the nearest neighbors of the point x_i , Li et al. [22] presented an orthogonal discriminating projection (ODP), in which the weight S_{ij} between two points is defined as follows:

$$S_{ij} = \begin{cases} \exp\left(-\frac{d^2(x_i, x_j)}{\beta}\right) & \text{if } x_i \in N(x_j), x_j \in N(x_i), C_i = C_j \\ \exp\left(-\frac{d^2(x_i, x_j)}{\beta}\right) \left(1 - \exp\left(-\frac{d^2(x_i, x_j)}{\beta}\right)\right) & \text{if } x_i \in N(x_j), x_j \in N(x_i), C_i \neq C_j \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where $d(x_i, x_j)$ denote the geodesic distance between points x_i and x_j , and β is a parameter which is used as a regular.

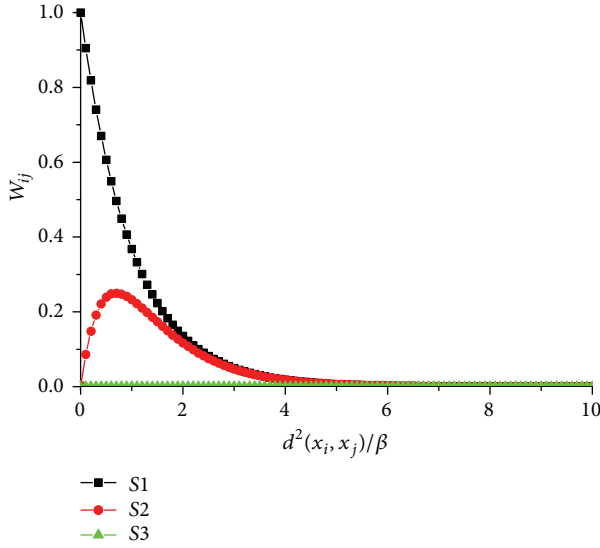
Figure 1 shows that the typical plot of S_{ij} is a function of $d^2(x_i, x_j)/\beta$, where $S_1 = \exp(-d^2(x_i, x_j)/\beta)$, $S_2 = \exp(-d^2(x_i, x_j)/\beta)(1 - \exp(-d^2(x_i, x_j)/\beta))$, $S_3 = 0$.

From Figure 1 we find that S_{ij} is not a monotonically decreasing function of $d^2(x_i, x_j)/\beta$, which is due to the fact that S_2 is a nonmonotonic function. When $0 < d^2(x_i, x_j)/\beta < 0.71$, S_{ij} is monotonically increasing. In the actual applications, S_{ij} should decrease with the increase of $d^2(x_i, x_j)/\beta$. Zhang et al. [23] proposed a modified ODP (MODP) with correlation coefficient $\|(x_i - \bar{x}_i)(x_j - \bar{x}_j)\| / \|(x_i - \bar{x}_i)\| \cdot \|(x_j - \bar{x}_j)\|$ between x_i and x_j .

Obviously, the above definitions about S_{ij} consider only the space structure and not the manifold structure. Meanwhile, the presented k nearest neighbors do not reflect the real information of manifold structure.

3. Video Feature Reduction

In this section, we extract more discriminating video features from the original color, shape, and texture feature. Firstly, we propose an improved LPP. We adaptively select the k nearest neighbors of each point and introduce the model information into the new weight similarity which can make the weight be a monotonically decreasing function of distance. The major merit of ALPP is preserving the local structure and label information of original data. Then we use ALPP to extract more discriminating video features for shot boundary recognition.


 FIGURE 1: Type plot of S_{ij} as a function of $d^2(x_i, x_j)/\beta$.

3.1. Mode Detection. In order to preserve the mode information of data points, we use median-shift method relied on computing the median of local neighborhoods instead of the mean to detect mode. Considering that the median of a set is a point in the set, the method is more robust than mean-shift method. Most importantly, the median-shift method is not a nonparametric method that does not require a prior knowledge of the number of clusters nor does it place any limitations on the shape of the clusters. The process of mode detection is as follows [24].

Suppose we are given a set $S = \{s_1, \dots, s_n\}$, $s_i \in R^d$, $d > 1$; define the Tukey depth of a point to be

$$\text{DEPTH}(s_i) = \min [v^T s_i : v \in R^d, \|v\| = 1], \quad (7)$$

where the Tukey depth of a point is the minimum of its depth along any projection vector v .

Firstly, the median of S is an element with maximal depth

$$\text{MEDIAN}(S) = \max_{s_i \in S} \left(\sum_{v_k \in V} \text{DEPTH}(s_i^T v_k) \right). \quad (8)$$

Then we seek the mode with the median-shift algorithms. For each point we wish to ascend in the direction of the positive gradient of the underlying probability density function. We define the median-shift for point c in set P as

$$c' = \text{MEDIAN}(p \in P \mid \|p - c\| \leq R), \quad (9)$$

where r is a bandwidth parameter. Since c' uses necessarily a point in the dataset, there is no need for multiple iterations in this step. After one iteration all points are linked and we can only go through the list of discovered medians to find a mode. The results of this step are a set of modes representing clusters.

Next, we proceed by iteratively working on the reduced set of modes, replacing the median calculation by weighted median calculation until convergence, where weights are

the number of points mapped to the given mode. The weights are taken into account during the calculation of the depth of each point in the next iteration by modifying the definitions as follows:

$$\text{RANK}(s_i) = \sum_{s_j < s_i} \text{Weight}(s_j) \quad (10)$$

and defining $W = \sum_{s_j} \text{Weight}(s_j)$ as the total weights in the neighborhood of s_i , then

$$\text{DEPTH}(s_i) = \min(\text{RANK}(s_i), W - \text{RANK}(s_i)). \quad (11)$$

Finally, in case of data clustering, and not only mode detection, we map each data point to its closest mode. Let M_i be the model of the point x_i , and let M be the set of M_i mode; we can obtain

$$M = \{M_1, M_2, \dots, M_M\}, \quad (12)$$

where $M_m = \{x_j \mid x_j \in M \wedge \text{model}(x_j) = m\}$.

3.2. Adaptive Neighborhood Selection. Considering the unchangeable k neighborhood not reflecting the mode information of manifold structure, we apply an adaptive strategy to select the neighborhood of each point.

Firstly, we define the manifold adjusted length of line segment

$$L(x_i, x_j) = \begin{cases} \rho^{d(x_i, x_j)} - 1 & x_i \text{ and } x_j \text{ are neighboring points} \\ \text{infinite else,} & \end{cases} \quad (13)$$

where $d(x_i, x_j)$ is the Euclidean distance between x_i and x_j and ρ is a flexing factor. Obviously, this formulation can be utilized to describe the global consistency. In addition, the length of line segment between two points can be elongated or shortened by adjusting the flexing factor ρ [25].

Then, let data points be the nodes of graph $G = (V; E)$ and let $p \in V^l$ be a path of length $l = |p| - 1$ connecting the nodes p_1 and $p_{|p|}$ in which $(p_k, p_{k+1}) \in E$, $1 \leq k \leq |p|$. Let P_{ij} denote the set of all paths connecting nodes x_i and x_j ; the manifold distance metric between two points is defined as follows:

$$D(x_i, x_j) = \min_{p \in P_{ij}} \sum_{k=1}^{|p|-1} L(p_k, p_{k+1}), \quad (14)$$

where $L(a, b)$ denotes the manifold adjusted length of line segment.

Next, the average manifold distance of point x_i is defined as follows:

$$\text{MS}(x_i) = \frac{1}{N_m} \sum_{M_i = M_j} D(x_i, x_j), \quad (15)$$

where N_m is the total number which meets the conditions $M_i = M_j$.

Lastly, the adaptive neighborhood of point x_i is constructed as

$$N'(x_i) = \{x \mid D(x_i, x_j) < MS(x_i) \wedge M_i = M_j\} \quad (16)$$

which shows that the neighborhood of point x_i is adaptively built with the points where the distance is shorter than the average distance.

The major merit of the adaptive neighborhood selection method can be summarized as follows.

- (1) The manifold distance metric can measure the geodesic distance along the manifold, which can elongate the distance among data points in different regions of high density and simultaneously shorten that in the same region of high density.
- (2) The neighborhood of each point is different from others, which is decided by the local density of the origin space. When the local density of x_i is lower, the neighborhood is higher and vice versa.

3.3. Improve Weight Definition. In order to resolve the problem which is described in Section 2.2, we improve the weight definition between two points x_i and x_j as follows:

$$S_{ij} = \begin{cases} \exp\left(-\frac{d^2(x_i, x_j)}{\beta_1}\right) & \text{if } x_i \in N'(x_j), x_j \in N'(x_i), C_i = C_j \\ \exp\left(-\frac{d^2(x_i, x_j)}{\beta_1}\right) \exp\left(-\frac{d^2(x_i, x_j)}{\beta_2}\right) & \text{if } x_i \in N'(x_j), x_j \in N'(x_i), C_i \neq C_j \\ 0 & \text{otherwise,} \end{cases} \quad (17)$$

where $d(x_i, x_j)$ denotes the distance between points x_i and x_j , β_1 and β_2 are the regular parameters, and C_i is the label of the point x_i for definition of $N'(x_i)$ please refer to (16).

Figure 2 shows that the typical plot of S_{ij} is a function of $d^2(x_i, x_j)/\beta$, $\beta = \beta_1 = \beta_2$, where $S_1 = \exp(-d^2(x_i, x_j)/\beta_1)$, $S_2 = \exp(-d^2(x_i, x_j)/\beta_1) \exp(-d^2(x_i, x_j)/\beta_2)$, and $S_3 = 0$.

Similar to paper [26], let $\exp(-d^2(x_i, x_j)/\beta_1)$ be the local weight, and let $\exp(-d^2(x_i, x_j)/\beta_2)$ be the intermode discriminating weight. The new weight definition can be viewed as the local weight and discriminating weight. It means that the discriminating similarity reflects both the local neighborhood structure of model and label information of the data set.

The properties and the corresponding advantages of the improved weight definition can be summarized as follows.

- (1) The improved weight definition make use of the label information and model information to preserve the manifold information, which is very important for classification.
- (2) Since the value of $\exp(-d^2(x_i, x_j)/\beta_2)$ ranges from 0 to 1, no matter how far the two points are, the intermode similarity can be limited in certain ranges.

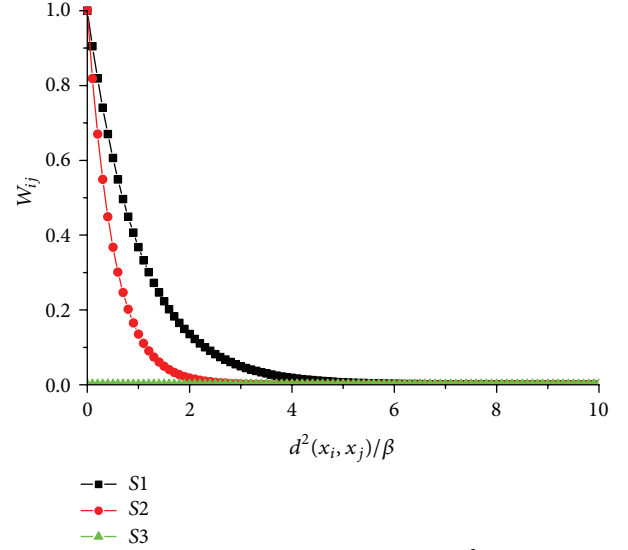


FIGURE 2: Type plot of S_{ij} as a function of $d^2(x_i, x_j)/\beta$.

- (3) With the decrease of the geodesic distance, $\exp(-d^2(x_i, x_j)/\beta_2)$ is a decrease, which means that two near points from different modes have a smaller similarity.
- (4) Note that S_1 and S_2 always decrease when x_i and x_j are far apart and they increase when x_i and x_j are close. Thus, S_{ij} is a monotonically decreasing function of $d^2(x_i, x_j)/\beta$.
- (5) The use of mode prior probabilities makes the newly designed discriminating similarity S_{ij} more suitable for perverting the local structure information of original data.

3.4. Video Feature Reduction. In order to enhance the discriminating information for shot boundary recognition, we hope to combine the label information and model information to improve the discriminating ability and preserve the local neighborhood structure of the original data.

Due to introducing the similarity matrix S , we define the local scatter matrix as follows [23]:

$$\begin{aligned} J_L(A) &= \sum_{i=1}^n \sum_{j=1}^n S_{ij} (y_i - y_j) (y_i - y_j)^T \\ &= \sum_{i=1}^n \sum_{j=1}^n S_{ij} (A^T x_i - A^T x_j) (A^T x_i - A^T x_j)^T \quad (18) \\ &= A^T S_L A, \end{aligned}$$

where $S_L = \sum_{i=1}^n \sum_{j=1}^n S_{ij} (x_i - x_j) (x_i - x_j)^T = XLX^T$, $L = M - S$, $M_{ii} = \sum_j S_{ij}$.

Then we define the nonlocal scatter matrix as follows:

$$\begin{aligned} J_N(A) &= \sum_{i=1}^n \sum_{j=1}^n (1 - S_{ij}) (y_i - y_j) (y_i - y_j)^T \\ &= \sum_{i=1}^n \sum_{j=1}^n (1 - S_{ij}) (A^T x_i - A^T x_j) (A^T x_i - A^T x_j)^T \end{aligned}$$

$$\begin{aligned}
&= A^T (S^T - S^N) A \\
&= A^T S_N A,
\end{aligned} \tag{19}$$

where $S_T = \sum_{i=1}^n \sum_{j=1}^n (x_i - x_j)(x_i - x_j)^T$, $S_N = S_T - S_L$.

Lastly, the objective function of the improved LPP can be expressed as follows:

$$\begin{aligned}
\arg \max_A J(A) &= \arg \max A^T ((1 - \alpha) S_T - \alpha S_L) A \\
\text{s.t.} \quad A^T A &= I,
\end{aligned} \tag{20}$$

where α is an adjustable factor.

So we can find that A consists of the eigenvectors associated with d top eigenvalues of the following eigen-equation:

$$((1 - \alpha) S_T - \alpha S_L) a = \lambda a. \tag{21}$$

The algorithmic procedure of video feature extraction is stated below.

- (1) Extract original video feature including the color, shape and texture feature.
- (2) Perform PCA projection. In order to make the matrix XX^T become nonsingular, we project the dataset into a PCA subspace with a transformation matrix.
- (3) Define the similarity matrix. For each point x_i , compute the similarity $W_{ij} = \exp(-d^2(x_i, x_j)/\beta_1)$, if x_j is the adaptive neighbors of x_i and $c_i = c_j$, and compute the similarity $W_{ij} = \exp(-d^2(x_i, x_j)/\beta_1) \exp(-d^2(x_i, x_j)/\beta_2)$, if x_j is the adaptive neighbors of x_i and $c_i \neq c_j$.
- (4) Compute the diagonal matrix M and Laplacian matrix L and then compute the top d eigenvalues and its corresponding eigenvectors based on (20).
- (5) Perform the ALPP transformation. Let A be an optimal projection matrix; we can project the new data into low dimensionality with

$$Y = A^T X. \tag{22}$$

4. Shot Boundary Recognition

The process of shot boundary detection includes two steps. Firstly, we extract the video features using ALPP method. Secondly, we detect shot boundary using an optimized MKSVM.

4.1. Multiple Kernel SVM. Support vector machines are a family of pattern classification algorithms which is based on the idea of structural risk minimization rather than empirical risk minimization [27]. However, it is often unclear what the most suitable kernel for the task at hand is. Recently, the multiple kernel learning theory has been used for training different kernels by jointly optimizing both the coefficients

of the classifiers and the weights of the kernels which have a more excellent effectiveness for object recognition than SVM [28]. In this paper, we combine several possible kernels to improve the precision of shot boundary recognition.

Let $d = [d_1, d_2, \dots, d_M]^T$ be a vector of weights for the mixture of kernels. A multiple kernel is the combination of the M basis kernels K_m

$$\begin{aligned}
K(x_i, x_j) &= \sum_{m=1}^M d_m K_m(x_i, x_j) \\
\text{s.t.} \quad d_m &\geq 0
\end{aligned} \tag{23}$$

$$\sum_{m=1}^M d_m = 1.$$

Assume there are a data set $D = \{x_i, z_i\}_{i=1}^m$ of labeled examples, where $x_i \in X$ is the input vector and $z_i \in \{-1, 1\}$.

According to paper [29], the primal form of multiple kernel support vector machine (MK SVM) is thus formulated as the following optimization problem:

$$\begin{aligned}
\min_{d, b, \xi} J(d) &= \frac{1}{2} \sum_{m=1}^M \frac{1}{d_m} \|f_m\|^2 + C \sum_{i=1}^l \xi_i \\
\text{s.t.} \quad y_i \sum_{m=1}^M f_m(x_i) + y_i b &\geq 1 - \xi_i \\
\xi_i &\geq 0, \quad \forall i \\
\sum_{m=1}^M d_m &= 1, \quad d_m \geq 0, \quad \forall m.
\end{aligned} \tag{24}$$

Similar to the SVM, with the constraint on d_m , the above minimization problem can thus be transformed into the following dual problem:

$$\begin{aligned}
\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \sum_{m=1}^M d_m k(x_i, x_j) \\
\text{s.t.} \quad \sum_{i=1}^n y_i \alpha_i = 0, \quad \forall i = 1, \dots, n \\
0 \leq \alpha_i \leq C, \quad \sum_{m=1}^M d_m = 1, \quad d_m \geq 0.
\end{aligned} \tag{25}$$

For the test input x , the decision function of MK SVM can be computed as

$$F_{\text{MKL}}(x) = \text{sign} \left(\sum_{i=1}^n \sum_{m=1}^M d_m K_m(x_i, x) a_i + b \right). \tag{26}$$

4.2. Ant Colony Optimization Method. Ant colony optimization method (ACO) is an optimizing method inspired by the foraging behavior of ant colonies [30]. When ants walk between their nest and a food source, they mark the paths with special kind of chemical termed pheromone, and

the shorter paths can attract more and more pheromone [31]. In the method, an ant determines its transfer direction according to the amount of pheromone in each path. Firstly, every ant constructs an edge from a start vertex to an end vertex. Then when all ants reach the end vertex, the edges are marked with a pheromone quantity. Thus the colony can converge to the shortest path [32]. In this paper, we apply the ant colony optimization method to solve the weights optimized problem in MKSVM.

Let $P_{ij}^k(t)$ be the proximity which an ant k transfer from element i to j at iteration t . $P_{ij}^k(t)$ can be defined as

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(i-1)\eta_{ij}^\beta}{\sum_{z \in V_k(x)} \tau_{ij}^\alpha(i-1)\eta_{ij}^\beta} & \text{if } z \in V_k(x) \\ 0 & \text{else,} \end{cases} \quad (27)$$

where $V_k(x)$ is the element which has not been unvisited. τ_{ij} is the pheromone quantity of path (i, j) , η_{ij} is a heuristic measure of moving element i to element j , α , and β are two parameters that control the relative importance of the information heuristics and exception heuristics factor, respectively.

The amount of p all m ants pheromone trail on a path deposited step by step. After time m , the pheromone quantity $\tau_{ij}(t)$ associated with an edge joining element i and j is updated according to the following formula:

$$\tau_{ij}(t+m) = (1-\rho)\tau_{ij}(t-1) + \sum_{k=1}^n \Delta\tau_{ij}^k(t), \quad (28)$$

where ρ is a pheromone evaporation loss coefficient and $\Delta\tau_{ij}^k(t)$ is the pheromone quantity deposited at iteration i by ant k on an edge joining element i and j . The $\Delta\tau_{ij}^k(t)$ is usually defined as

$$\Delta\tau_{ij}^k(i) = \begin{cases} \frac{Q}{L_k} & \text{if } w_{ij} \in T_k(I) \\ 0 & \text{else,} \end{cases} \quad (29)$$

where Q is a constant and L_k is the cost function of k th ant.

For the process of ant colony optimization method please refer to paper [33].

4.3. Shot Boundary Recognition. The process of shot boundary recognition includes the following stages. Firstly, we extract the original feature. Then we reduce the video feature with improved LPP. Lastly, we classify the frame into boundary frame with the optimized MKSVM classifier based on ACO method.

The algorithm of video boundary recognition is described as follows.

- (1) Extract original video feature which includes colors, shape, and texture feature.
- (2) Reduce video feature with improved LPP.

- (3) Define input vector $[(y_{i-k} - y_i), \dots, (y_{i+k} - y_i)]$, where y_i is the video feature vector and i is i th frame of video.
- (4) Label the boundary frames (+1) and nonboundary frame (-1) and build training sample set.
- (5) Initialize the parameters of ACO. Let time $t = 0$, interactive times $I_t = 0$. Set the maximum iterative times $I_t(\max)$ and let initial pheromone $\tau_{xy} = c$ and $\Delta\tau_{xy}(0) = 0$.
- (6) Update I_t by $I_t = I_{t+1}$ and choose the ant element j , as its transform direction according to function (27).
- (7) Update the taboo table pointer. Move the ant to the selected new element and add the element into ant taboo table.
- (8) If all elements of the set have been fully traversed, go to step 7 or else go to the next step.
- (9) Recalculate the pheromone of each path, if $I_t < I_t(\max)$, go to step 6, or else save the weights $d = [d_1, d_2, \dots, d_M]$.
- (10) Classify the video frame into boundary frame based on the following rule:

$$F_{\text{MKL}}(x) = \text{sign} \left(\sum_{i=1}^n \sum_{m=1}^M d_m K_m(x_i, x) a_i + b \right). \quad (30)$$

5. Experiments and Analysis

In this section we present some experiments to validate the proposed approach. Firstly, we investigate the performance of the proposed ALPP method for video feature extraction experiment. Then we recognize shot boundary with the proposed method. The video database includes movie, news, sports, documentary, and MTV. The reason of selecting this type video is that news videos have many long abrupt shots, MTV has fast changes of scenes, sport have fast camera movements and zooming in, movie includes many gradual shots. Some shot samples are shown in Figure 3. For evaluation, we use the common figures of merit of the algorithm standard precision and recall [34]

$$\begin{aligned} \text{Precision} &= \frac{\#(\text{Boundaries correctly detected})}{\#(\text{Total boundaries detected})}, \\ \text{Recall} &= \frac{\#(\text{Boundaries correctly detected})}{\#(\text{Total ground truth boundaries})}. \end{aligned} \quad (31)$$

5.1. Video Feature Exaction Experiment. In order to testify the effectiveness of the adaptive LPP (ALPP), we extract video feature with ODP, MODP, and ALPP method. For convenience of comparison, we use the same method in paper [35] to detect shot boundary. In the experiment, original video feature is built on color feature, shape feature, and texture feature. Then ODP, MODP, and ALPP method, are used to extract video feature with $\alpha = 0.5$ and $\gamma = 0.6$. In the ODP and MODP methods we adopt k nearest neighbor



FIGURE 3: Some shot samples.

TABLE 1: Comparison results between ODP, MODP, and ALPP.

Methods	OPP	MODP	ALPP
Precision	83.7	86.1	89.6
Recall	85.5	87.8	90.4

criterion to define the adjacency matrix, in which the k set to 10. The results of performance comparison by using ODP, MODP, and ALPP are shown in Table 1.

From Table 1, we can find that ALPP obtains comparable recognition performance to LPP and ODP with the same shot boundary detect method. In ALPP method, the improved weight definition combines the label information and model information with adaptive nearest neighbor select strategy, which is very important to reflect the data information truthfully. The experiment shows that ALPP has more useful and discriminating ability to extract video feature than others.

5.2. Shot Boundary Recognition Experiment. In order to investigate the performance of the proposed method, we recognize different video shots, especially for gradual shots. The system performance is compared with the multilayer perception network method (MPN method) [36]. We use the original video feature and set the parameters α and γ to be the same as before. In the proposed method, we use polynomial kernel, radial basis kernel and linear kernel to build MKSVM and set $d = 3$, $\sigma^2 = 0.005$, $\alpha = 1$, $\beta = 5$, $\rho = 0.5$, and $Q = 100$. The experimental results are summarized in Figure 4.

From Figure 4, we found that the above methods can detect not only abrupt cuts but also gradual shots very well, but the proposed method achieves more desired performance than the MPN method for shot boundary recognition. The average precision and recall of the proposed method is up to 94.1% and 91.7%, which is higher by 3.5% and 3.1% than the MPN method, respectively. These results demonstrate that the proposed method is a good tool for shot boundary recognition by using the optimized MKSVM.

5.3. Discussion. Two experiments for different type video have been systematically performed, and so now we can conclude the following.

- (1) We improve the effect of shot boundary detection in two stages. In the feature extract stage, we use ALPP to extract more useful and discriminating video

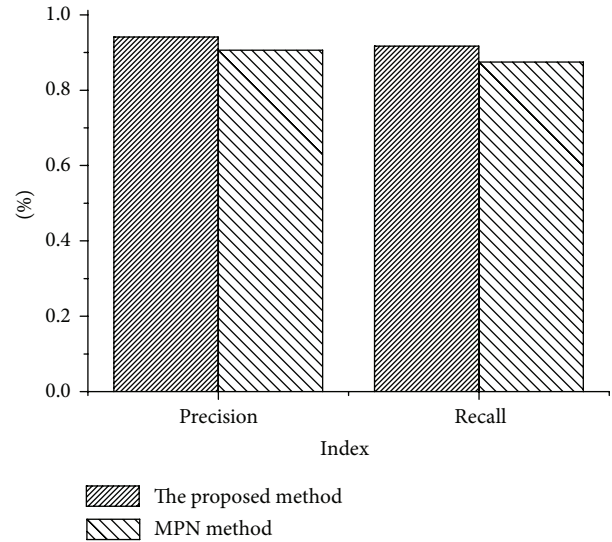


FIGURE 4: Comparison of two recognition methods.

feature. In the shot boundary detection stage, we use optimized MKSVM obtaining more remarkably boundary detection accuracy.

- (2) For feature extraction, the proposed ALPP performed better than LPP and ODP. This is because the former makes use of the label information and model information with adaptive nearest neighbor select strategy. At the same time, the improved weight definition can guarantee that two near points from different modes have a smaller similarity.
- (3) Compared with the MPN method, the proposed method can yield better performance on shot boundary recognition. It owes much to the optimized MKSVM, in which the parameters are optimized by the ant colony method. It should be noted that there are some false detection results in the above methods, which may be due to the existence of irregular object movement and the small content change between consecutive frames.

6. Conclusion

In this paper, we present a new video shot boundary recognition method, which focuses on two key problems: extracting more useful and discriminating feature and improving

the accuracy of shot boundary classifier. The major contributions of the paper are to propose an optimized locality preserving present method with model detection and optimized neighbor selection strategy. Meanwhile, an optimized shot boundary classifier based on MKSVM is designed with the ant colony optimization method. Experiments demonstrate that the proposed method is outstanding. The future work is to optimize the other parameters of MKSVM to achieve more desired result.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This project is supported by the Postdoctoral Science Foundation of Central South University, Scientific Research Fund of Hunan University of Finance and Economics under Grant (no. K201205), the Construct Program of the Key Discipline in Hunan Province, Hunan Province Education and Science Issue “Performance Evaluation for College Teacher Based on Adaptive Learning” (no. XJK013CGD083), and the Research Foundation of Science & Technology Office of Hunan Province under Grant (no. 2012GK3064).

References

- [1] G. Y. Wang, H. Z. Kou, and T. Li, “News video shot boundary detection by automated threshold,” *Computer Engineering and Design*, vol. 32, no. 5, pp. 1734–1737, 2011.
- [2] M. B. A. Vila, Q. Xu, M. Feixas, and M. Sbert, “Tsallis entropy-based information measures for shot boundary detection and keyframe selection,” *Signal, Image and Video Processing*, vol. 7, no. 3, pp. 507–520, 2013.
- [3] F. Liu, Y. W. Guo, and K. Y. Yin, “Shot boundary detection using adaptive threshold and fourier fitting,” *Journal of Software*, vol. 23, no. 2, pp. 105–114, 2012.
- [4] K. K. Warhade, S. N. Merchant, and U. B. Desai, “Shot boundary detection in the presence of fire flicker and explosion using stationary wavelet transform,” *Signal, Image and Video Processing*, vol. 5, no. 4, pp. 507–515, 2011.
- [5] V. B. Thakar and S. K. Hadia, “An adaptive novel feature based approach for automatic video shot boundary detection,” in *Proceedings of the International Conference on Intelligent Systems and Signal Processing*, pp. 145–149, Gujarat, India, March 2013.
- [6] Y. Huo, P. Z. Zhang, and Y. F. Wang, “Adaptive threshold based video shot boundary detection framework,” in *Proceedings of the International Conference on Image Analysis and Signal Processing*, pp. 1–5, Hangzhou, China, November 2012.
- [7] Y. Feng, S.-L. Luo, L.-P. Wang, and L.-M. Pan, “A new method of adaptive shot boundary detection,” *Transaction of Beijing Institute of Technology*, vol. 30, no. 1, pp. 100–104, 2010.
- [8] K. K. Warhade, S. N. Merchant, and U. B. Desai, “Shot boundary detection in the presence of illumination and motion,” *Signal, Image and Video Processing*, vol. 7, no. 3, pp. 581–592, 2013.
- [9] G. G. Gao and H. D. Ma, “To accelerate shot boundary detection by reducing detection region and scope,” *Multimedia Tools and Applications*, 2012.
- [10] H. Fang, J. Jiang, and Y. Feng, “A fuzzy logic approach for detection of video shot boundaries,” *Pattern Recognition*, vol. 39, no. 11, pp. 2092–2100, 2006.
- [11] L. Qun, W. Jiang, and Y. Wu, “Approach of shot-boundary detection based on multi-feature fusion,” *Computer Engineering and Applications*, vol. 46, no. 13, pp. 171–174, 2010.
- [12] Z.-W. Huang, W. W. Y. Ng, P. P. K. Chan, J. Li, and D. S. Yeung, “Video shot boundary detection using RBFNN minimizing the L-GEM,” in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '10)*, pp. 2156–2160, Qingdao, China, July 2010.
- [13] P. P. Mohanta, S. K. Saha, and B. Chanda, “A model-based shot boundary detection technique using frame transition parameters,” *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 223–233, 2012.
- [14] J. Li, Y. Ding, Y. Shi, and W. Li, “A divide-and-rule scheme for shot boundary detection based on SIFT,” *International Journal of Digital Content Technology and Its Applications*, vol. 4, no. 3, pp. 202–214, 2010.
- [15] L. Zhao, X. M. Sun, B. J. Shang, and J. G. Wu, “A shot boundary detection method based on context vector and Tabu-SVM,” *Journal of Computational Information Systems*, vol. 8, no. 16, pp. 6817–6824, 2012.
- [16] C. Zhang and W. Q. Wang, “A robust and efficient shot boundary detection approach based on fisher criterion,” in *Proceedings of the 20th ACM International Conference on Multimedia*, pp. 701–704, Nara, Japan, October 2012.
- [17] L. Zhao, X. Sun, and M. Zhang, “A shot boundary detection method based on PSO-SVM,” *Applied Mechanics and Materials*, vol. 130–134, pp. 3821–3825, 2012.
- [18] J. Lankinen and J. K. Kamarainen, “Video shot boundary detection using visual bag-of-words,” in *Proceedings of the International Conference on Computer Vision Theory and Applications*, pp. 788–791, Barcelona, Spain, February 2013.
- [19] A. Donate and X. Liu, “Shot boundary detection in videos using robust three-dimensional tracking,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '10)*, pp. 64–69, San Francisco, Calif, USA, June 2010.
- [20] X. J. Li and P. Chen, “The application of fast and effective shot boundary detection algorithm in mechanical engineering,” *Advanced Materials Research*, vol. 681, no. 1, pp. 121–125, 2013.
- [21] X. F. He and P. Niyogi, “Locality preserving projections,” in *Proceedings of the Neural Information Processing Systems*, pp. 153–160, Vancouver, Canada, 2003.
- [22] B. Li, C. Wang, and D.-S. Huang, “Supervised feature extraction based on orthogonal discriminant projection,” *Neurocomputing*, vol. 73, no. 1–3, pp. 191–196, 2009.
- [23] S. W. Zhang, Y. K. Lei, Y. H. Wu, and J. A. Yang, “Modified orthogonal discriminant projection for classification,” *Neurocomputing*, vol. 74, no. 17, pp. 3690–3694, 2011.
- [24] L. Shapira, S. Avidan, and A. Shamir, “Mode-detection via median-shift,” in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 1909–1916, Kyoto, Japan, October 2009.
- [25] X. Y. Pan and H. Chen, “Multi-agent evolutionary clustering algorithm based on manifold distance,” in *Proceedings of the 8th International Conference on Computational Intelligence and Security*, pp. 123–127, Guangzhou, China, November 2012.
- [26] W. K. Wong and H. T. Zhao, “Supervised optimal locality preserving projection,” *Pattern Recognition*, vol. 45, no. 1, pp. 186–197, 2012.

- [27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [28] D. Tuia, G. Camps-Valls, G. Matasci, and M. Kanevski, "Learning relevant image features with multiple-kernel classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 10, pp. 3780–3791, 2010.
- [29] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1531–1565, 2006.
- [30] A. Hogenboom, F. Frasinca, and U. Kaymak, "Ant colony optimization for RDF chain queries for decision support," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1555–1563, 2013.
- [31] M. Shen, W. N. Chen, J. Zhang, H. S. H. Chung, and O. Kaynak, "Optimal selection of parameters for nonuniform embedding of chaotic time series using ant colony optimization," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 790–802, 2013.
- [32] K. S. Yoo and S. Y. Han, "A modified ant colony optimization algorithm for dynamic topology optimization," *Computers and Structures*, vol. 123, no. 2, pp. 68–78, 2013.
- [33] L. Lei, "Text categorization using SVM with exponent weighted ACO," in *Proceedings of the 31st Chinese Control Conference*, pp. 3763–3768, Hefei, China, July 2012.
- [34] J. Yuan, H. Wang, L. Xiao et al., "A formal study of shot boundary detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 2, pp. 168–186, 2007.
- [35] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, vol. 1, no. 1, pp. 10–28, 1993.
- [36] P. P. Mohanta, S. K. Saha, and B. Chanda, "A model-based shot boundary detection technique using frame transition parameters," *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 223–233, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

