ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st&22ndMarch, Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Feature Fusion Based Cartoon Character Retrieval Using Semi-Msl

J.Ida princess, V.Nithya, P.Arjun

UG Student, University College of Engineering, Villupuram, India.

UG Student, University College of Engineering, Villupuram, India

Assistant Professor, University College of Engineering, Villupuram, India.

Abstract— Cartoon is popular and successful media in our life, its creation is usually of high cost and labor intensive. The computer assisted systems are designed to reduce the time cost of the cartoon production and to develop novel systems that can synthesize new cartoons by reusing existing cartoon materials. For that purpose, we propose a method for cartoon character retrieval with two applications, namely content based cartoon image retrieval and cartoon clip synthesis. In order to retrieve similar cartoon characters two issues are considered. The first issue is how to represent the cartoon characters. The color histogram, hausdorff edge feature, skeleton features and curvature scale space are used to represent the color, shape and gesture of the cartoon characters. The second issue is to combine these features using semi-supervised multi-view subspace learning algorithm. This algorithm uses patch alignment framework to construct and align local patches. Aligned patches have high dimension, it is not efficient to retrieve similar character. In order to improve the efficiency, alternating optimization is utilized for reducing high to low dimensional subspace. Within that space similar characters can be retrieved. This method is useful for animators and cartoon enthusiasts to effectively create new animations.

Keywords— Cartoon character retrieval, Multiview subspace learning, clip synthesis, patch alignment framework.

I.INTRODUCTION

Cartoon plays important roles in many domains, e.g., entertainment, education, and advertisement. It has attracted much research attention in the field of multimedia and computer graphics. To produce cartoon movie clips, key-frames drawing, in-betweening, and painting require a large amount of human labor. Therefore, it is valuable to develop a computer-aided system to reduce the labor in cartoon clip production.

There is a large scale cartoon library which consists of various animation materials, including character design, story board, scenes, and episodes. It is useful for animators and cartoon enthusiasts to effectively create new animations by reusing and synthesizing. But this is too complicated for non-professional users who may want to produce cartoon clips either for fun or other purposes**.**

In the cartoon industry, there is existing software to reduce the human labor in producing new cartoon clips. For example, Toon Boom [15] is well-known commercial software for cartoon creation, which covers the entire animation workflow from sketching to publishing. By using this software, the cartoonists can draw a few cartoon images and reuse them to produce new cartoon clips. However a limitation is that Toon Boom does not support cartoon image retrieval.

In this paper, to perform cartoon image retrieval and clip synthesis four features are considered. i.e., color histogram (CH), edge feature (EF), skeleton feature(SF) and curvature scale space(CSS) .These features reveal different characteristics of a cartoon character. But the simple concatenation of these features results in curse of dimensionality problem.

For this purpose semi supervised Multiview subspace learning algorithms have been developed. Semi-MSL uses "patch alignment" framework (PAF) [2] that has two stages: local patch construction and whole alignment. Each patch is constructed by one sample and its related ones according to both the characteristics of the data set and the objective of the algorithm. In the whole alignment stage, all part optimizations are integrated together to form a consistent global coordinate for all independent patches. After alignment, it obtains a high dimensional subspace which is not efficient to retrieve the similar cartoon characters. Thus Semi-MSL adopts an alternating-optimization [16] which is used to combine multiple features from different views for dimensionality reduction. Finally, a compact subspace is achieved for subsequent retrieval and synthesis.

The following is the organisation of the remainder of this paper.In Section II related works are investigated. Section III, Cartoon character retrieval is analyzed. The results of experiments on cartoon character retrieval and clip synthesis are given in Section IV. Finally, Section V concludes this paper.

II. RELATED WORKS

In this section, we briefly review related works (content based image retrieval and clip synthesis) and discuss the difference between them.

A. Content Based Image Retrieval

Content-based image retrieval (CBIR) is a long standing research topic in the field of multimedia. In the early years, CBIR algorithms mainly use low-level visual features, such as color feature and texture feature to compute image similarity. However, despite the intensive research effort, the performance of CBIR systems is not satisfactory due to the well-known semantic gap[11]. In [12] and [13] Haseyama and Matsumura have proposed a cartoon image retrieval system, where they define Regions and Aspects to compute cartoon image similarities. However, the partial features are not suitable for cartoon character gesture recognition. It can only be used to retrieve face image of cartoon characters.

B. Cartoon Clip Synthesis by Recycling

Recently, a few systems have been proposed to synthesize new multimedia data by reusing existing resources. For example, a system is proposed in [10] to synthesize new video from a series of existing video clips by reordering the original video frames. In [3] Juan and Bodenheimer developed Graph based cartoon clip synthesis (GCCS) that combines similar characters into user-directed sequence based on dissimilarity measure in edges. It performs well for simple cartoon characters. GCCS fails to generate smooth clips because edges can encode neither the color information nor the gesture pattern. Yang *et al.* [6] recently proposed a Retrieval-based Cartoon Clip Synthesis (RCCS) that combines similar characters based on dissimilarity measure in cartoon characters motion and edges. Based on these two features, RCCS uses an unsupervised bidistance metric learning (UBDML) algorithm to measure dissimilarity. Drawback of RCCS is that, they fail to retrieve dissimilarity in color and gesture. Furthermore they developed Cartoon gesture space (CGS) [7] which represents the characters based on color, edge and motion, cannot accurately characterize the gestures of character.

C. Feature Fusion

There are a growing number of scenarios in pattern recognition where different features are used. For feature fusion three algorithms are used i.e., multiple kernel learning (MKL), Gaussian process (GP), and features concatenation. These algorithm support only labeled information. In [17] Many researchers have introduced various spectral methods in the field of machine learning and data mining. Most of them directly concatenate different feature vectors to produce a high-dimensional vector and then perform dimension reduction. However the above mentioned method does not support unlabeled data. This problem can be overcome by using a method like semi-MSL which supports both labeled and unlabeled data.

III.CARTOON CHARACTER RETRIEVAL

The similar cartoon characters has to be retrieved from cartoon database, which is prepared by extracting the characters from cartoon videos as shown in fig 1using LOG (Laplacian of Gaussian) filter , then color histogram, hausdorf edge feature, skeleton features and curvature scale space are extracted from cartoon characters. Fusion of these features is done by using semi-MSL algorithm. Alternating optimization are utilized by semi-MSL to obtained the low dimensional subspace. Based on each low dimensional subspace, a standard retrieval procedure is conducted for all characters in the database. For example, one cartoon character is selected as a query, and then all the other characters in the database are ranked according to the Euclidean distance to the query computed in the low-dimensional subspace. For each query cartoon character, the top *10* characters are obtained. The detailed work of the project is explained in the following sections (cartoon image feature extraction and semi-MSL).

A. Cartoon Image Feature Extraction

The data comes from 2D cartoon videos, which contain a character as shown in Fig.1. When the animator makes the cartoon, a sketch is painted before colorization. Then the edge around the character can be easily detected by using Laplacian of Gaussian (LOG) filter. After edge detection, the flood fill algorithm is used to fill the region surrounded by the closed edge. Finally, the cartoon character can be extracted by background subtraction. After obtaining the cartoon characters $X = [x_1, . . . , x_N]$, we perform feature extraction CH x_i^{CH} , HEF x_i^{HEF} , SF x_i^{SF} , *CSS* x_i^{CSS} which is given in the following sections.

Fig .1Character extraction from cartoon videos

1) Color Histogram

CH is an effective representation of the color information. Color is one of the most reliable visual features that are also easier to implement in image retrieval systems. Color histogram is the most common technique for extracting the color features of colored images. It tells the global distribution of colors in the images. It is defined as the distribution of cartoon characters.

In this paper each component of RGB color spaces are quantized into four bins (0–63, 64–127, 128–191, and 192–255). Therefore, CH comprises 64 (4 × 4 × 4) bins. In this case, each component of CH for cartoon character *xi* is defined as $x_i^{CH}(j) = c_j/C_i$ where j=1, 2,...,64 where c_j is the number of pixels with value *j* and C_i is the total number of pixels in x_i. The distance between the characters cannot be measured accurately by using CH alone. Thus, it is necessary to consider the cartoon character's shape and gesture information.

2) Hausdorf Edge Feature

Shape is also important image content used in retrieval. The primary mechanisms used for shape retrieval include identification of features such as edges, lines, regions and boundaries. Edge features, which contain information about the shapes of objects, are frequently used to measure the dissimilarity between objects. To efficiently represent the character's shape, we use a modified Hausdorff distance to calculate the edge distance. Let *Ei* be the edge point set of the cartoon character x_i. Given two cartoon characters x_i and x_j , *h*(*Ei, Ej*) is defined as,

$$h(E_i, E_j) = \frac{\left(\sum_{p_a \in E_i} \min_{p_b \in E_j} (\| p_a - p_b \|)\right)}{| E_i |}$$

Where $\| p_a - p_b \|$ is the Euclidean distance between the coordinates of the two edge points p_a and p_b. This $|Ei|$ represents the total number of edge points on E_i. Thus, the Hausdorff edge distance is

$$Haus\_Dist_{ij} = \begin{cases} \max(h(E_i, E_j), h(E_j, E_i)), i \neq j \\ o, i = j \end{cases}$$

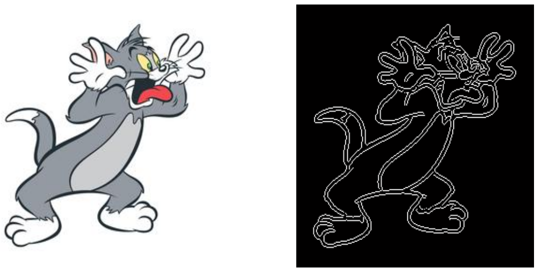

Fig.2 (a) Original image (b) Detection of edge from the image.

By calculating the Hausdorff distance in the whole data set, we obtain the edge dissimilarity matrix. Fig 2 shows the contours of cartoon character, it can be detected by using canny edge detector.

3) Skeleton Feature

The skeleton is important for object representation and recognition in different areas, such as image retrieval and computer graphics, and analysis of biomedical images. Skeleton, which integrates both geometrical and topological features of an object, is an important descriptor for object representation [1]. Therefore, we use the skeleton to describe the cartoon characters gesture. First, we adopt discrete curve evolution to extract skeletons of cartoon characters. Then, we calculate relative angles between every two successive endpoints of the skeletons and utilize these angles to describe the cartoon characters' gestures. For instance, we adopt two successive endpoints (v_i^k,v_i^{k+1}) and the centroid c_i of character x_i to construct the relative angle $A_i{}^k$.

$$A_i^k = \cos^{-1}\left(\frac{\overrightarrow{c_i v_i^k} \bullet \overrightarrow{c_j v_i^{k+1}}}{|\overrightarrow{c_i v_i{}^k}||\overrightarrow{c_i v_i{}^{k+1}}|}\right)$$

Where c_iv_i is a vector and | $c_iv_i^k$ | represents the length of the vector. The relative angle can be used to effectively represent the geometric structures of the character. Based on the constructed correspondence, the corresponding relative angle of A_i in character x_j is,

$$M(A_i{}^k) = \cos^{-1}\left(\frac{\overrightarrow{c_j M(v_i{}^k)} \bullet \overrightarrow{c_j M(v_i^{k+1})}}{|\overrightarrow{c_j M(v_i^k)}||\overrightarrow{c_j M(v_i^{k+1})}|}\right)$$

By using these angles, the character *xi* can be represented as (A_i^1,A_i^2, . . . A_i^8), and the character x_j can be represented as ($M(A_i^1)$,$M(A_i^2)$, . . . , $M(A_i^8)$). Thus, the skeleton dissimilarity can be measured by,

$$Dist(x_i, x_j) = \frac{1}{N} \sum_{k \in N} \| A_i^k - M(A_i^k) \|$$

Where *N* is the number of the corresponding points. After determining the distance between the characters the SF vectors for all cartoon characters can be obtained as $[x_1^{SF}$, , . $x_N^{SF}]^T$.

4) curvature scale space

Curvature scale space is concerned with the problem of shape similarity retrieval. The CSS image consists of several arch-shape contours representing the inflection points of the shape as it is smoothed. The maxima of these contours are used to represent a shape. The formula for computing the curvature function can be expressed as

$$K(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{\left(\dot{x}^2(u) + \dot{y}^2(u)\right)^{3/2}}$$

Where u represent the arbitrary parameter. In order to find the curvature scale space of an image, first the Shrinkage and smoothing of the curve is performed by decreasing the number of curvature zero-crossings. First, the curve is smoothed by a Gaussian function. The smoothed curve is called Ʈ. The locations of curvature zero-crossings on $Ʈ_\sigma$ are determined at different levels of scale ie., (σ). The process starts with $\sigma = 1$, and at each level, σ is increased by σ, chosen as 0*:*1 in our experiments. As σ increases, $Ʈ_\sigma$ shrinks and becomes smoother, and the number of curvature zero crossing points on it decreases. Finally, when σ is sufficiently high, $Ʈ_\sigma$ will be a convex curve with no curvature zero-crossings. The process of creating ordered sequences of curves is referred to as the *evolution* of Ʈ. The result of this process can be represented as a binary image called the *CSS image* of the curve. The second step is to extract maxima of CSS contours. After the maxima is extracted from the image, matching procedure is carried out, which compares two sets of maxima and assigns a matching value to them. The matching value represents the similarity measure between the actual boundaries of objects. The advantage of using CSS representation is that it is robust with respect to scale, noise, rotation and change in orientation.

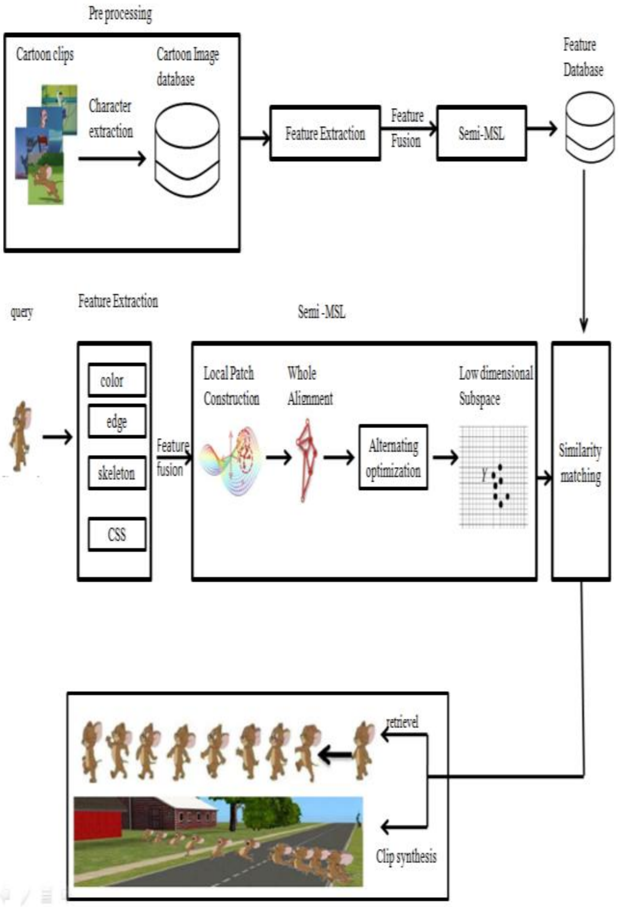

Fig.3 Frame work of this paper

B. *Semi-MSL*

In this section, we introduce semi-MSL that finds a unified low-dimensional and sufficiently smooth subspace over all views simultaneously. To clearly explain the technique details of the proposed semi-MSL, we present important notations used in the rest of this paper. Capital letters, e.g., X, represent the cartoon database. Lower case letters, e.g., x, represent cartoon characters, and x_i is the ith cartoon character of X. Superscript (i), e.g., $X^{(i)}$ and $x^{(i)}$, represents the cartoon character's feature from the ith view. Based on these notations, semi-MSL can be described under the umbrella of the PAF [2].

Fig.3 shows the framework of this paper for cartoon character retrieval and clip synthesis. First, cartoon characters are represented by the four visual features. According to PAF [2], semi MSL builds the local patches. There are two types of local patches: local patches for labeled characters and local patches for unlabeled characters. Based on these patches, the part optimization can be performed to get the low-dimensional subspace for each patch. Afterward, all low-dimensional subspaces from different patches are unified by whole alignment to obtain the subspace in each view. Then, we derive an alternating optimization [16]-based iterative algorithm to obtain the optimal low-dimensional subspace from multiple views. Finally, in this subspace, the Euclidean distance can be directly used to measure the dissimilarity between two cartoon characters, and the cartoon character retrieval and clip synthesis can be conducted accordingly.

1) *Patch Alignment Framework*

PAF consists of two parts: local patch construction and whole alignment. PAF shows that, the subspace learning algorithms are different in the local patch construction stage and share an almost identical whole alignment stage. This general framework is a flexible tool for character retrieval by constructing special patches. Given the data matrix X which consists of N sample points $x_i(1 \leq i \leq N)$ in the high-dimensional space R^m, i.e., $X = [x_1, . . . , x_N] \in R^{m\times N}$, PAF constructs the corresponding low-dimensional representation $Y = [y1, . . . , yN] \in R^{d\times N}$, wherein $d < m$. In the local patch construction stage, different algorithms have different optimization criteria. Local patch is formed by combining the sample x_i and its k related samples x_1^i, . . . , x_i^k (e.g., its k nearest neighbors) as $X_i = [x_i, x_{i1}, . . . , x_{ik}] \in R^{m\times(k+1)}$). For each Xi, we find the mapping $f_i : X_i \rightarrow Y_i$, wherein $Y_i = [y_i , y_{i1}, . . . , y_{ik}] € R^{d\times(k+1)}$. This optimization for the local patch is defined by,

$$\arg\min_{Y_i} tr\left(Y_i L_i Y_i^T\right)$$

Where tr(・) is the trace operator and L_i describes the local geometry for the ith patch. Thus L_i varies with different algorithms. In the whole alignment stage, all achieved local patches will be aligned together to form a global optimization. For each patch Xi, there is a low-dimensional representation Y_i. All Y_i's can be unified together as a whole one by assuming that the coordinate for the i^{th} patch $Y_i = [y_i, y_{i1}, . . . , y_{iK}]$ is selected from the global coordinate $Y = [y_1, . . . , y_N]$, such that,

$Y_i = YS_i$, where S_i is the selection matrix. Summing all the patch optimizations, the following can be obtained:

$$\begin{aligned}&\arg\min_{Y} \sum_{i=1}^{N} tr\left(YS_i L_i S_i^T Y^T\right)\\ &= \arg\min_{Y} tr\left(Y\left(\sum_{i=1}^{N} S_i L_i S_i^T\right)Y^T\right)\\ &= \arg\min_{Y} tr\left(YLY^T\right)\end{aligned}$$

Where L is the alignment matrix obtained by an iterative procedure. The patch optimization can be solved by using the Lagrangian multiplier method, and the solution can be obtained by adopting the eigenvalue decomposition on L. The optimal solution is the d eigenvectors associated with d smallest eigenvalues.

a) *Local Patch Construction*

The local patch for labeled characters in semi-MSL is built by its neighbors who include the characters from not only the same class but also different classes. . In semi-MSL, the local patches represent the local geometry of the characters and the discriminative information to separate characters with different labels. Then local patch construction for unlabeled characters is used to reveal the manifold structure of data in the original high dimensional space. To preserve the local geometry of the patch, the nearby characters should stay nearby in the low-dimensional subspace.

b) Whole Alignment

For each patch $X_j^{(i)}$, there is a low-dimensional subspace $Y_j^{(i)}$. All $Y_j^{(i)}$ can be unified together as a whole one by assuming that the coordinate for $Y_j^{(i)} = [y_j^{(i)} , y_{j1}^{(i)} , . . . , y_{jk}^{(i)}]$ is selected from the global coordinate $Y = [y_1, . . . , y_N]$, i.e., $Y_j^{(i)} = Y_j^{S(i)}$, wherein $S_j^{(i)} \in R^{N\times(k+1)}$ is the selection matrix.

c) Alternating Optimization

Because of the complementary characteristics of multiple views to each other, different views definitely have different contributions to the final low-dimensional subspace. In oder to well explore the complementary property of different views, a set of nonnegative weights $\sigma = [\sigma_1, . . . , \sigma_t]$ is imposed on different views independently. The σ with a larger value means that the view $X^{(i)}$ plays a more important role in obtaining the low-dimensional subspace. $\sigma_l = 1$ corresponding to the minimum tr($Y L^{S(i)} Y^T$) over different views, and σ_l =0 otherwise. It means that only one view is finally selected.

This solution cannot explore the complementary characteristics of different views. In order to avoid this problem, we set $\sigma_i \leftarrow \sigma_i^r$ with $r > 1$. The multiview subspace learning is realized by summing all views,

$$\arg\min_{Y,\sigma} \sum_{i=1}^{t} \sigma_i^r tr\left(YL^{S(i)}Y^T\right)$$

In this paper, iterative algorithm can be derived by using the alternating optimization to obtain a local optimal solution. The optimization alternatively updates low-dimensional subspace Y and weights $\sigma = [\sigma_1, . . . , \sigma_t]$. Initially, we update σ with fixed Y. Afterward, we update Y with fixed σ. The optimization problem can resolve by using the eigenvalue decomposition on L. The optimal subspace Y is constructed by the eigenvectors associated with the smallest d eigenvalues of the matrix L [4].

IV. EXPERIMENTS

To test the performance of the proposed method, we collected Tom and Jerry cartoon images from the Disney cartoon series. There are 1000 cartoon images in the database which is used for experiment. Feature are extracted as explained in section 3. After that, a semi-MSL is constructed for each cartoon character. We test the proposed method both in content-based cartoon image retrieval and cartoon clip synthesis.

TABLE I
Precision Rates (in percent) Of Cartoon Character Retrieval

Dims	Semi-MSL	MSL	UMSL	UBDML	DMSL	FCSL
d=10	63.8	60.2	51.3	50.5	48.3	53.3
d=20	61.4	58.7	50.4	49.6	46.5	55.7
d=30	59.5	57.3	48.5	47.7	45.5	54.5
d=40	62.6	56.4	55.6	48.8	46.7	56.4
d=50	61.8	57.5	54.7	49.9	45.4	55.7

A. Retrieval Evaluation

First, the low dimensional subspaces of the cartoon database are learned by the subspace learning algorithms, including semi-MSL, MSL, UMSL, UBDML, CGS, FCSL, DMSL, and SSL, and second, based on each low-dimensional subspace, a standard retrieval procedure is conducted for all characters in the database. In detail, for each category, one cartoon character is selected as a query, and thus, 100 cartoon characters are used as queries, respectively. Then, all the other characters in the database (including other categories) are ranked according to the Euclidean distance to the query computed in the low-dimensional subspace. For each query cartoon character, the top N characters are obtained, among which the number of characters from the same class as the query shape is recorded. The precision rate, shown in table 1 is the ratio of the recorded number to N.

We first compare semi-MSL against other methods with different dimensionalities d on the database of Tom. Here, ASSL means the average performance of SSL with different features. In Table I, the dimensionality increases from 10 to 50, and we observe that the precision rates of semi-MSL are significantly better than those of the other methods.

Based on these results shown in Table I, we observe that semi-MSL achieves satisfactory performance when the dimensionality is around ten. The experiments conducted on the databases of Jerry also suggested similar results.

B. Clip Synthesis

SSL considers only one feature that is hausdorf edge feature for performing cartoon clip synthesis, hence the clip synthesized by using SSL is not smooth. In case of RCCS it uses two features, edge and motion direction feature. The clips synthesized by semi-MSL is smoother than SSL and RCCS, because RCCS neglects the color information in the distance metric learning, the characters in the synthesized results may have different colors and SSL uses only edge feature, hence it is not efficient. Hence in this section, we show that semi-MSL outperforms the SSL used in GCCS and the UBDML used in RCCS for cartoon clip synthesis.

V. CONCLUSIONS

In this paper, we have defined Four visual features, i.e., CH, HEF, and SF, CSS to represent the color, shape, and gesture information of a cartoon character. However HEF feature is used to extract the shape of the cartoon character, it is not efficient. Because of, affine transformation. So we proposed CSS feature for extracting shape with affine transformation. To efficiently and effectively combine these visual features for subsequent retrieval and synthesis tasks, we developed a new multiview learning method named semi-MSL. It encodes different features from different views in a physically meaningful subspace. In this unified subspace, the Euclidean distance can be directly used to measure the distance between two cartoon characters. After ranking similar cartoon characters can be retrieved. Experimental evaluations on cartoon character retrieval and clip synthesis suggest the effectiveness of the visual features and semi-MSL.

REFERENCES

[1] X. Bai, L. J. Latecki, and W.-Y. Liu, "Skeleton pruning by contour partitioning with discrete curve evolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 449–462, Mar. 2007.

[2] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Non-negative patch alignment framework," *IEEE Trans. Neural Netw.*, vol. 22, no. 8, pp. 1218–1230, Aug. 2011.

[3] C. D. Juan and B. Bodenheimer, "Cartoon textures," in *Proc.Eurographics Symp. Comput. Animation*, Grenoble, France,2004, pp. 267–276.

[4] I. T. Jolliffe, Principal Component Analysis, 2nd ed. New York: Springer-Verlag, 2002.

[5] F.wang luo, "Semi-supervised metric learning by maximizing constraint margin," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 4, pp. 931–939, Aug. 2011.

[6] Y. Yang, Y. T. Zhuang, D. Xu, Y. H. Pan, D. C. Tao, and S. Maybank, "Retrieval based interactive cartoon synthesis via unsupervised bi-distance metric learning," in *Proc. ACM Int. Conf. Multim.*, Beijing, China, 2009, pp. 311–320.

[7] Y. Yang, Y. Zhuang, D. Tao, D. Xu, J. Yu, and J. Luo, "Recognizing cartoon image gestures for retrieval and interactive cartoon clip synthesis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1745–1756, Dec. 2010.

[8] T.Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," *IEEE Trans. Known. Data Eng.*, vol. 21, no. 9, pp. 1299–1313, Sep. 2009.

[9] W. Bian and D. Tao, "Biased discriminated Euclideanembedding for content based image retrieval," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 545–554, Feb. 2010.

[10] A. Scholl, R. Szeliski, D. H. Salesin, and I. Essa, "Video textures," in *Proc. SIGGRAPH*, 2000, pp. 489–498.

[11] Y. Liu, D. Zhang, G. Lu, and W. Ma, "A survey of content-based image retrieval with high-level semantics," *Patt. Recog.*, vol. 40, no. 1, pp. 262–282, 2007.

[12] M. Haseyama and A. Matsumura, "A trainable retrieval system for cartoon character images," in *Proc. ICME*, Jul. 2003, pp. 393–396.

[13] M. Haseyama and A. Matsumura, "A cartoon character retrieval system including trainable scheme," in *Proc. ICIP*, Sep. 2003, pp. 37–40.

[14] T. Zhang, D. Tao, and J.Yang, "Discriminative locality alignment," in *Proc. ECCV*, 2008, pp. 725–738.

[15] J. Fekete, E. Bizouarn, E. Cournarie, T. Galas, and F. Taillefer, "TicTacToon: A paperless system for professional 2D animation," in *Proc.ACM Int. Conf.* CA, 1995, pp. 79–90.

[16] J.C.Bezdek and R. J. Hathaway, "Some notes on alternating optimization," in *Proc. Int. Conf. Fuzzy Syst.*, London, U.K., 2002, pp. 288–300.

[17] M. Belkin and P. Niyogi, "Laplacian Eigen maps and spectral techniques for embedding and clustering," in *Proc. NIPS*, 2001, pp. 585–591.