Key-frame Extraction and Shot Retrieval Using Nearest Feature Line (NFL)

Li Zhao[†♣], Wei Qi[*], Stan Z. Li[*], Shi-Qiang Yang[†♣], H. J. Zhang[*]

[*] Microsoft Research China
Beijing 100080, China

{weiqi,szli,hjzhang}@microsoft.com

[†] Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China

lizhao99@stumail.tsinghua.edu.cn

ABSTRACT

Query by key frame or video example is a convenient and often effective way to search in video database. This paper proposes a new approach to support such searches. The main contribution of the proposed approach is the consideration of both feature extraction and distance computation as a whole process. With a video shot represented by key-frames corresponding to feature points in a feature space, a new metric is defined to measure the distance between a query image and a shot based on the concept of Nearest Feature Line (NFL). We propose to use the ``breakpoints'' of feature trajectory of a video shot as the key frames and use the lines passing through these points to represent the shot. When combined with the NFL method, it helps to achieve a better performance, as evidenced by experiments.

Keywords

Content-based retrieval, nearest feature line (NFL), key-frame extraction, color histogram.

1. INTRODUCTION

For content-based video retrieval, currently there are two widely accepted query modes: keyword-based and example-based. The example-based approach is necessary in the situation that users could not describe clearly what they exactly want by only using text. The example queries could be a video clip, a frame, an object or some low-leveled features such as color, texture and motion [1][2][3][4][5][6]. The system should search in the video database and cluster the relevant video segments (such as video shots) according to the content of the query example. The most important issues for such example-based approaches lie in two aspects: First, how to select features to represent the content of a video segment. Second, how to define a distance metric in the feature space to measure the similarity of two video segments. This paper focused on how to combine these two issues together for consideration.

Most of the presented methods work on the key-frames of the shots to measure the similarity. Dimitrova et.al [1] regarded the average distance of corresponding frames between two videos as the similarity measure. Zhang et al.[3] defined another similarity according to the sum of the most similar pairs of key-frames. These methods could all be classified as the Nearest Center (NC) or Nearest Neighbor (NN) approaches. The drawback for these kinds of methods lies in that they all leave out the temporal variations and correlation between key-frames within an individual shot.

Li proposed a new pattern classification method called Nearest Feature Line (NFL), which has been shown to yield good results in face recognition and audio classification and retrieval[7][8].

Zhao et. al. [9] extended the NFL method to video retrieval. Unlike conventional methods such as NN and NC, the NFL method takes into consideration of temporal variations and correlations between key-frames in a shot. The main idea is to use the lines passing through the consecutive feature points in the feature space to approximate the trajectory of feature points.

In this paper, we further develop the previous work by proposing a feature (key-frame) extraction scheme for the NFL based shot classification and retrieval. Considering the way that the NFL works in shot classification, it is important to select or extract proper key-frames based on which the NFL does the classification. The paper is organized as follows. In section 2, the main algorithm is produced. Section 3 is experimental results and analysis. Conclusion is drawn in section 4.

2. VIDEO SHOT RETRIEVAL USING KEY-FRAMES COMBINED NEATEST FEATURE LINE (NFL) METHOD

NFL method assumes that there are at least two sample points (feature point) in each class and through these known samples linear extrapolation or interpolation could be made to generate the feature line.

We regard the key-frames in a shot as the known sample points in the feature space. Since in this paper we focus on the clustering issue, we simply select the color histogram space as the feature space that we discussed.

In section 2.1 we describe how to use the NFL method in shot retrieval. In section 2.2 we introduce how to combine key-frame extraction with NFL method.

2.1 NFL Method used for shot retrieval

In a shot we think of that the distance in color histogram

♣ This work was performed at Microsoft Research China.

space between two adjacent key-frames mainly caused by the object motion or camera manipulation. So we use the line passing through the feature points of two key-frames to approximate the trajectory of the continuously frames between the two end key-frames.

We consider two frames F_i and F_j in video space mapping to the points f_i and f_j in the feature space. Let

$$f_k = \{f_{1k}, f_{2k}, \cdots f_{mk}\} \quad (1)$$

where m is the dimension of the feature space. The difference between frames F_i and F_j can be measured as Euclidean distance $\Delta f = \|f_i - f_j\|$ in their feature space. The straight line passing through f_i and f_j of the same class, denoted by $\overline{f_i f_j}$, is called a feature line (FL) of that class (See figure 1).

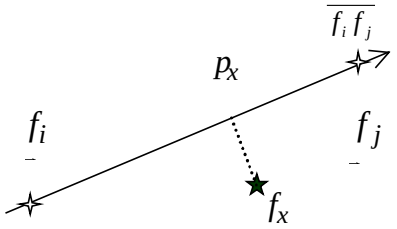

Figure 1: Feature points f_i and f_j, feature line $\overline{f_i f_j}$, and query feature point f_x.

Let $F^c = \{f_i \mid 0 < i \le N_c\}$ be the set of the N_c prototypical feature points belonging to class *c*. A number of $K_c = \frac{N_c(N_c - 1)}{2}$ lines can be constructed to represent the class. The FL space for class *c* is composed of the K_c feature lines: $S = \{\overline{f_i \ f_j} \mid 0 < i, j \le N_c, i \ne j\}$, which is a subset of the entire feature space. When there are ***Q*** classes in the database, ***Q*** such FL spaces can be constructed, with a total number of N_{total} FL's where $N_{total} = \sum_{c=1}^{Q} K_c$.

Letting f_x be the query feature point, supposing the query-by-frame situation. We define the distance between the query point f_x and the feature line $\overline{f_i f_j}$ as $Dist(f_x, \overline{f_i f_j})$. Let p_x denote the projection of f_x on the feature line $\overline{f_i f_j}$ (see figure 1)**:**

$$p_x = f_i + \mu(f_j - f_i) \quad (2)$$

where

$$\mu = \frac{(f_x - f_i) \bullet (f_j - f_i)}{(f_j - f_i) \bullet (f_j - f_i)} \quad (3)$$

so that

$$Dist(f_x, \overline{f_i f_j}) = \|f_x - p_x\| \quad (4)$$

The distances between the query point f_x and each line in the whole feature line space could be calculated, and these distances are then sorted in ascending order. The best matching shot in the video database should be the one containing the key-frames that form the feature line of the smallest distance with the query frame. The sorted list of shots gives the retrieval result in order.

2.2 Key-frame Extraction for NFL-based Retrieval

Let $f_1, f_2, \cdots f_N$ be the feature points corresponding to the key frame sequences of shot ***C***, where ***N*** is the number of key frames in the shot. The trajectory through $f_1, f_2, \cdots f_N$ forms a curve in the feature space. As mentioned above, we use the feature line $\overline{f_k f_{k+1}}$ to approximate the curve segment $\overbrace{f_k f_{k+1}}$ between feature points f_k, $f_{(k+1)}$. The FL space for shot ***C*** is composed of an order of $N-1$ feature lines: $S^c = \{\overline{f_k \ f_{k+1}} \mid 0 < k \le N-1\}$, which is a subset of the entire feature space. It is an approximate of the trajectory.

To achieve better performance based on NFL, it is desirable that the actual manifold between two successive feature points is close to the FL, i.e. the manifold bounded by two success feature points should be as linear as possible between the two points. A better approximate may be obtained by break the entire curve at "Sharp Corners" and use the corners as the key frames, calls break point (BP) key frame.

A classic curve splitting algorithm, is available at [10] (See Figure 2).

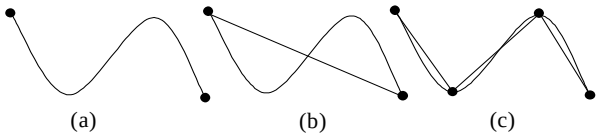

Figure 2: Stages of classic curve splitting algorithm.

However, that algorithm is very time-consuming in computing the breakpoint on the curve.

Here we propose a simple but efficient key frame extraction approximate method, calls SBP (Simplified Breakpoint) key frame extraction. (See figure 3)

[Algorithm 1]

1. Set the number of frames in shot ***C***: $N = frames\_in\_shot\_C$, and initialize the number of key frames: $M = 0$.
2. Let $k = 1$.
   - If $Dist(f_k, f_{k+1}) > \delta$ or $Dist(f_{k-1}, f_{k+1}) > C_1 \cdot \delta$, then a new key frame generated: frame f_{k+1} is a new key frame and $M = M + 1$.
   - If $Dist(f_k, f_{k+1}) < \delta$ and $Dist(f_{k-1}, f_{k+1}) < C_1 \cdot \delta$, no key frame generated.
   - $k = k + 1$.
3. If $k <= N - 1$, then go to step 2, else exit.

In this algorithm, δ is a tolerance predefined by user; C_1 a constant between 1~2. We have chosen in our experiments, C_1 =1.8.

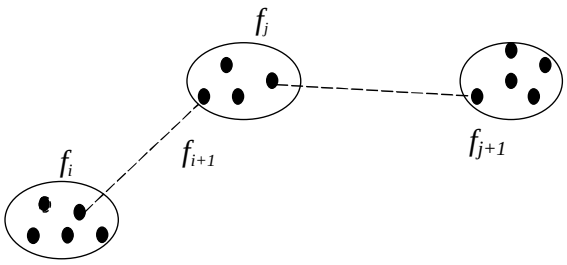

Figure 3: The approximate algorithm of key frame extraction.

After extract key frames, the number of key frames M is an important parameter that reflect the character of the shot. If the M is very small or $M = 0$, we could draw the conclusion that there is very little change in the shot, so we can regard the shot as one feature point in feature space. If the M is very large, we could draw conclusion that the feature in the shot change greatly or the tolerance δ selected before is too small, the δ can be adjust then recomputed again.

Obviously, it is a bad instance, if the feature points mapping to the key frames extracted out are on a line. Here we use the dot product of vectors of $\overrightarrow{f_{k-1}f_k}$ and $\overrightarrow{f_k f_{k+1}}$ to remove the redundant feature points. Of course, if not taking this step, it will take no effect on result, but increase the complexity of computing.

3. EXPERIMENTAL RESULT

In our experiment we use color histogram as the feature. The color feature is defined according to the 1976 CIE u'v' perceptually uniform color space. The generative model is a histogram: space u'v' is divided into 256 square bins (16 bins on a side). The model is that a bin is chosen with probability proportional to the stored histogram count and the color of a pixel is the color of the center of the chosen bin. There are 16 bins in u' that span from 0.16 to 0.2883. There are 16 bins in v' that span from 0.4361 to 0.5361. Colors outside of these spans are clipped to the nearest bin.

To evaluate the performance of the proposed improved NFL method, we build a video database of 160 shots. These shots are taken from forty minutes Sports News of CCTV, including track and field, swimming, soccer, gig and advertisements etc. Figure 4 shows the user interface of our experimental program. Up-row is the browsing area for showing the first key-frame of each shot. And user can select the image that he wants to query among these frames. Down-row is the result area for showing the query results in ascending order of distance.

We use precision and weight score to measure the performance. The following measure [8] will be used in performance evaluation:

$$\eta(q,m) = \sum_{k=1}^{m} w_k Match(q, r_k) \qquad (5)$$

$$Match(q, r_k) = \begin{cases} =1 & \text{If } r_k \text{ and } \boldsymbol{q} \text{ is correlative} \\ =0 & \text{If } r_k \text{ and } \boldsymbol{q} \text{ is not correlative} \end{cases}$$

where, $\boldsymbol{q}$ represents the query shot and $r_1, r_2, \cdots, r_m$ are the m top ranked matches for the query shot $\boldsymbol{q}$. The judgments for the value of $Match(q, r_k)$ are given by subjective evaluation in our experiments.

$w_k = W \cdot \frac{1}{k}$ is a decreasing sequence of weights $(k = 1,2,\cdots)$ where $W = 1 \Big/ \sum_{k=1}^{N_q} (1/k)$, here N_q is the number of queried shots that match the query shot $\boldsymbol{q}$. Because the weights W_k are decreasing with the rank position $\boldsymbol{k}$, a higher ranked correct match contributes more to $\eta(q,m)$. The weights are normalized by the factor $\boldsymbol{W}$ in the following sense: When the top N_q matches are all correct, $\eta(q,m)$ reaches the highest possible value of 1.

We designed five experiments for evaluation:

1. SBP (Simplified Breakpoint) key frame extraction with NFL method (total key frame number: 622);
2. BP (Breakpoint) key frame extraction with NFL method (total key frame number: 597);
3. EI (Equal Interval) key frame extraction with NFL method (total key frame number:700);
4. EI (Equal Interval) key frame extraction with NN method (total key frame number:700);
5. EI (Equal Interval) key frame extraction with NC (total key frame number:700).

Experiment 1 and 2 are based on our approach. The difference between these two is that in the Experiment 2 the key frame extraction method is classic curve splitting algorithm [10], while in experiment 1 the key frame extraction method is our simplified breakpoint search algorithm (Algorithm 1 in 2.2). The experiments 3, 4 and 5 are designed for comparing traditional classification method with ours. In our experiments, we keep the number of key frames extracted in experiment 1 and 2 less than that experiment 3, 4 and 5.

Figure 5 gives the result of our experiments. The results show that the 'breakpoints-based key frame extraction' + NFL methods produce better performance than NC, NN and traditional NFL method.

4. CONCLUSIONS

In this paper, we present a new approach for query-by-example video retrieval. The breakpoint based key-frame extraction and NFL classification method are combined and considered as a whole process. In this way, the NFL method could achieve its best performance.

The experimental results show that the proposed combined method performs not only better than the traditional classification methods such as NN and NC, but also better than the traditional NFL method without considering key-frame extraction.

Now in our experiment we only use color histogram as the feature for classification. In the future, we will add more

features (texture, shape, etc) for classification. We expect better performance with adding these new features.

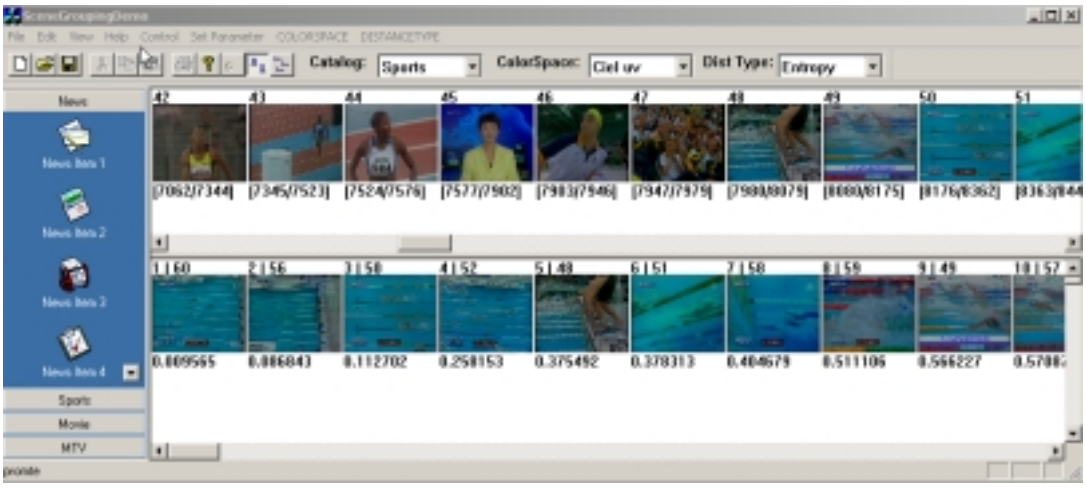

Figure 4: Interface for the retrieval program

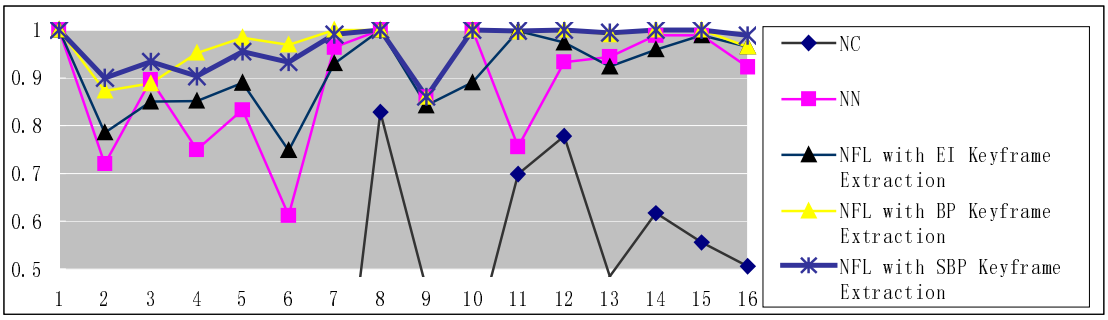

Figure 5: The score of different methods

5. ACKNOWLEDGEMENTS

This work is supported by Microsoft-Tsinghua Multimedia Lab.

6. REFERENCES

[1] N. Dimitrova and M. Abdel-Mottaled, "Content-based video retrieval by example video clip", SPIE Vol. 3022, 1998.

[2] M.M. Yeung and B.Liu, "Efficient matching and clustering of video shots", IEEE International Conference on Image Processing 1995 Vol.1 pp.338-341

[3] H.J.Zhang, D.Zhong and S.W.Smoliar, "An Integrated System for Content-Based Video Retrieval and Browsing," Pattern Recognition, Vol.30, No.4, pp.643-658, 1997.

[4] D.Zhong, S.F.Chang, "Spatio-Temporal Video Search using the Object-Based Video Representation", IEEE International Conference on Image Processing 1997 Vol 1, pp.21-24

[5] M.-K. Shan, S.-Y. Lee, "Content-based Video Retrieval based on Similarity of Frame Sequence", *Proc. IEEE Conf. on Multimedia Computing and Systems*, pp.90-97, 1998

[6] Y.Deng and B.S.Manjunath, "Content based search of video using color, texture and motion", IEEE International Conference on Image Processing 1997 Vol 2, pp.13-16

[7] S. Z. Li and J. Lu, "Face recognition based on nearest linear combinations", *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp.439-443, March 1999.

[8] S. Z. Li, "Content-based Classification and Retrieval of Audio Using the Nearest Feature Line Method", IEEE Transactions on Speech and Audio Processing. September 2000.

[9] Li Zhao, Wei Qi, S.Z. Li, etc., "Content-based retrieval of video shots using the nearest feature line method", submitted to IEEE WACV 2000.

[10] D.H. Ballard, C.M. Brown, "Computer vision", Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1982.