23

Machine Learning for Video Repeat Mining

Xianfeng Yang[1] and Qi Tian[2]
[1]Academy of Broadcasting Science,
[2]Institute for Infocomm Research,
[1]China
[2]Singapore

1. Introduction

An incommensurable amount of digital audiovisual information is becoming available in digital archives, on the World Wide Web, in broadcast data streams and in personal and professional databases, and this amount is only growing. Mining short video repeats is just providing an effective way to extract useful information from fast growing video data and exploit the reuse value of video archives. Short video clips, ranging from a few seconds to a few minutes, are widely used in many types of video production, including news video, sports video, education program where they serve as program logo, station logo, sports reply flying logo or as commercials. These short video clips are repeated many times in these video programs because of the nature of their usages in video production. Identifying these short video repeats based on their contents has great value in many media applications. First it can be used to monitor commercials and detect infringement of copyrighted content in broadcast videos or web videos (Agnihotri et al., 2003; Cheung et al., 2005; Lienhart et al., 1997; Snchez et al., 2002; Kashino et al., 2003; Yuan et al., 2004), and this content based video identification approach is an important complementary method to other media copyright protection techniques such as watermarking. Second, short video repeat mining is very useful in video structure analysis task. By detecting video repeats in unlabeled raw video data, we can discover correlation of different video parts and structural video elements used for syntactic segmentation purpose, thus video structure model can be effectively constructed and applied to video syntactical segmentation (Yang & Tian et al., 2007). Structure analysis by finding repeat objects has also been extensively employed in DNA data mining domain (Kurtz & Schleiermacher, 1999; Bao & Eddy, 2002). Video repeat mining also has many other applications, such as content summary, personalization as well as lossless video compression (Pua & Gauch, 2004).

Video repeats are defined as those video clips having the same video contents, but low level signal distortions are allowed, such as noises, image quality reduction, frame size change et al. Video repeat mining problem can be classified into two categories: one problem is to identify known video repeats. In this case we have a model of a sample video in advance, then use this model to detect its repeated instances in video archives, which is treated as a pattern recognition problem; The second problem is to identify unknown video repeats. In this case we do not have prior knowledge about the video repeats including their frame

Source: Theory and Novel Applications of Machine Learning, Book edited by: Meng Joo Er and Yi Zhou,
ISBN 978-3-902613-55-4, pp. 376, February 2009, I-Tech, Vienna, Austria

content, length and location. What we do is to define some rules to discover these video repeats through machine learning techniques, which is treated as a data mining problem.

To identify known short video repeats, we propose a novel representation scheme for short video clips based on composite HMM. HMM is a powerful statistical model for time-series data and composite HMM is a network of single HMM. Composite HMM has been successfully applied to speech recognition (Rabiner, 1989; Young et al., 2000), and we will show it also has good performance on video sequence recognition. Compared to single HMM approach proposed by Kulesh et al. (2002), composite HMM approach is more robust to temporal editing of video clip (Yang et al., 2003).

To solve unknown video repeat mining problem, we propose a novel approach which combines with video segmentation, self-similarity analysis, cascaded detection, LSH indexing and reinforcement learning (Yang et al., 2005; Yang & Xue et al., 2007; Yang & Tian et al., 2007). Video self-similarity was used by Cooper & Foote (2001) to detect scene boundary, and we applied this approach to repeat sequence detection. Compared to other media repeat pattern identification methods (Cheung et al., 2005; Herley, 2006; Pua & Gauch, 2004), our approach can detect very short repeats (e.g. those less than 1 second) along with long ones, and high accuracy has been achieved in our experiments. Methods by Cheung et al. (2005) and Herley (2006) both use a fixed time window to do feature extraction and comparison, so those repeats significantly shorter than the window are very likely be missed. The method by Pua et al. (2004) is able to identify repeated shots but can not identify partially repeated shots, while our approach can identify even small portion of a shot or clip by adopting segmentation with granularity smaller than the shot. Another novelty of our approach is that a reinforcement learning approach is adopted to train the video repeat detectors, and this approach demonstrates effectiveness and efficiency in parameter learning, which makes the repeat mining system manageable and easy to train.

The remainder of this chapter is arranged as follows: In section 2, we present the HMM based video repeat identification approach. In section 3, we present the unknown video repeat mining method, and concluding remarks are presented in section 4.

2. Short video clip recognition by composite HMM

In this section we propose a composite HMM approach to represent known short video clips and recognize their repeated instances. This approach is not only robust to video feature distorion but also robust to temporal editing. Details of this approach are presented in the following.

2.1 Short video clip modeling

The modeling process is divided into two levels. The first level is the clip level modeling based on video shots, and the second level is shot modeling using HMM. A video clip is composed of a series of shots, and the reproduction of the clip is often based on shot editing, so it is meaningful to exploit the clip structure based on shots. In this approach, a video clip is firstly segmented into shots, and a directed graph *G* is used to represent the temporal constraints of the shots. Each shot is a node of the graph *G*. If shot 1 precedes shot 2, then shot 2 is a child node of shot 1, and there is a directed edge from shot 1 to shot 2. It is assumed that a shot can reach any of its child nodes. Besides all the shots, two other null nodes are added into the graph, namely entry node and exit node. Entry node is the parent node of any other node, and exit node is the child node of any other node. These two nodes will not occupy any time. The clip graph with 4 shot nodes is shown as Fig. 1 in which shot 1 precedes shot2, shot 2 precedes shot3, and shot3 precedes shot 4. In the clip graph, not only the temporal constraints of the shots of sample video clip are preserved, but also the possible structure variations of the clip are also included. For example, because the entry node can reach any child node, the sequence can start from any shot, and end at any shot. This graph representation is suitable for short video clips because the number of shots in them is generally within 20, so the graph will not be complex.

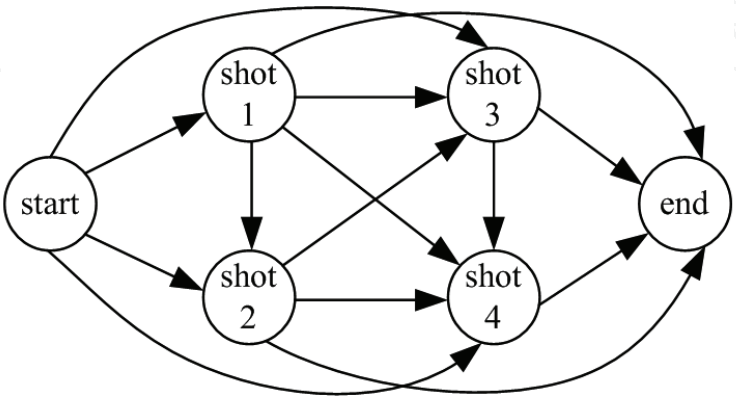


Fig. 1. Clip graph with 4 shots

For each shot node, one HMM is used to represent the color feature distribution within the shot, so the clip graph *G* becomes as a composite HMM. The RGB histogram of a frame is chosen as the color feature vector, and all the frame features constitute the feature sequence. Each color channel is quantized into 8 bins, resulting in a 512 dimensional feature vector. To reduce the dimensionality of the feature, the feature vectors are mapped to symbols by vector quantization, so the color feature sequence is transformed to a symbol sequence which is used for model training and testing. The symbol values do not represent real feature values, and adjacent symbol values do not mean that their underlying real feature values are close to each other.

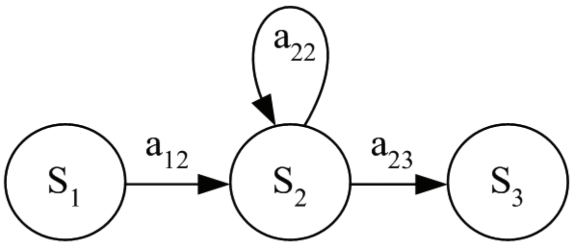


Fig. 2. One state HMM for a shot

Because in many cases the color within one shot is homogeneous, so we choose one state HMM to represent the color statistics. In order to connect to other models in recognition networks, one entry and one exit state are added to the model, as shown in Fig.2. S_1 and S_3 are entry and exit state of the model. The self transition probabilities of S_1 and S_3 are both zeros, and a_{12} equals to one. S_2 is the state that actually captures the color statistics of the shot. The output probability distribution, as well as the state transition probabilities a_{22} and a_{23}, is estimated by Bum-Welch algorithm (Rabiner, 1989). Each model is trained by symbol sequences of sample videos. The learning results show that parameter a_{22} is generally a large value close to 1, while a_{23} is a small value close to 0. The output probability of the state is the discrete probability of the symbols.

2.2 Video repeat recognition using composite HMM

After the video clip model was built, recognition of a test video sequence is to compute the likelihood of the test sequence generated from this model. Given a composite HMM *G* and the test video's symbol sequence $o_1 o_2 \cdots o_n$, the log probability of $o_1 o_2 \cdots o_n$ generated from *G* is computed by maximizing the following conditional probability:

$$\begin{aligned} &\log P(O_1 O_2 \cdots O_n \,|G) \approx \\ &\max \sum_{i=1}^{m} \log P(O_{starti} O_{starti+1} \cdots O_{endi} \,\big| N_i \in G) \end{aligned} \quad (1)$$

where *m* is the total number of nodes the test sequence passes through, N_i represents one node of *G*, and N_i is a child node of N_{i-1}, for 2≤ *i* ≤m. $\log P(O|N_i)$ is the logarithm probability of sequence ***O*** belonging to shot model N_i. o_{starti} is the first observed feature assigned to node N_i, and o_{endi} is the last observed feature assigned to node N_i. The maximum score can be efficiently computed by Viterbi algorithm.

Because the testing sequence may have variable length, the score should be normalized by the sequence length. The average log probability per frame is used to make decision. If there are multiple video models, the test sequence will be match against each model. If the highest score is above a threshold, then the title of the model is determined as the title of the test video.

2.3 Results

In the experiment, 100 different advertisement clips ranging from 10 to 60 seconds are collected for model building. The video format is: MPEG-1, CIF, frame rate 30fps, bit-rate 242 kbits/s. The average number of the shot per clip is 11. Several clips have more than 20 shots, but some have less than 5 shots. The color features of all the shots are extracted and used to construct the codebook of vector quantization. The codebook size is set to 128.

To test this method's robustness against feature distortion, the clips for modeling are transformed by the following operations: Frame rate is reduced to 8fps; bit-rate is reduced to 121 kbits/s. To test the recognition performance on shorter version of the video clips, we remove nearly half of the shots from each clip and join the rest shots together. The shots are removed by this way: shot 1,3,5,… are retained, while shot 2,4,6,… are removed. The transcoding and cut version clips are considered as true positive samples. In addition, another 100 clips which do not belong to any of the 100 advertisement clips are used as negative test samples. So there are totally 300 test samples.

As a performance comparison, each clip is also modeld by with single HMM approach proposed by Kulesh et al. (2002). The number of the states of HMM is set as the number of shots of the clip. The ROC curves of our approach and single HMM is shown in Fig.3. As is shown the ROC curve of composite HMM is closer to the origin than single HMM approach. The performance comparison of two approaches is also shown in Table 1.

From Table 1 we can see that the composite HMM outperforms single HMM when recognizing clip's cut versions. Even though nearly half of the shots are missing in the cut version, composite HMM approach can correctly recognize them and the error is just about 1%. This result coincides with our expectation that composite HMM would more accurately model the video clip and its variations. The results also show that composite HMM is robust to transcoding operations which severely change the quality of the image and the frame rate.

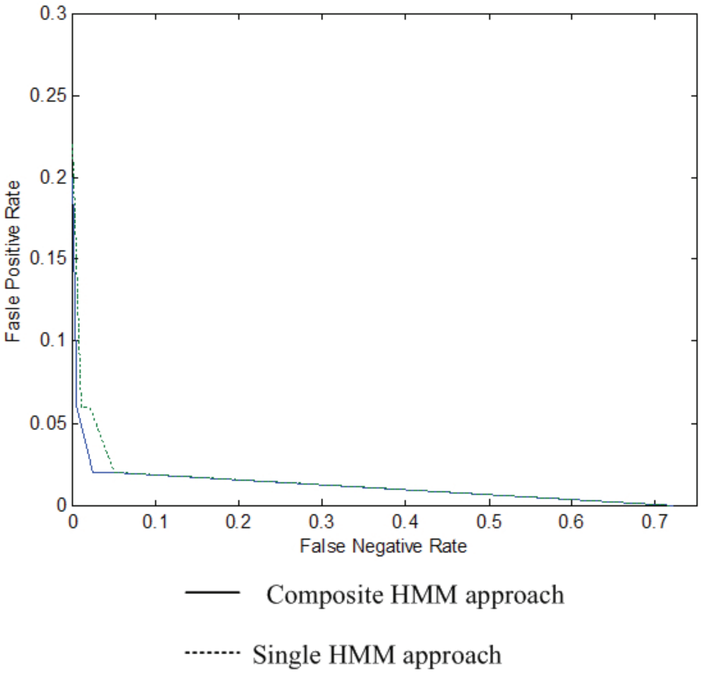


Fig. 3. Receiver Operating Curve: FNR vs FPR

Accuracy	Transcoded version	Cut version
Composite HMM	98.0%	99.0%
single HMM	98.0%	96.0%

Table 1. Recognition performance comparison

3. Unknown video repeat mining

This section we propose a novel approach for unknown repeat repeats mining. Two detectors in a cascade structure are employed to achieve fast and accurate detection, and a reinforcement learning approach is adopted to efficiently maximize detection accuracy. In this approach very short video repeats (< 1s) and long ones can be detected by a single process, while overall accuracy remains high. Since video segmentation is essential for repeat detection, performance analysis is also conducted for several segmentation methods.

3.1 Method overview

The proposed repeat mining framework is shown as Fig. 4. It contains the following main components: 1) video abstraction and feature extraction function; 2) two stage detectors in cascade. The first stage detector performs similarity analysis on abstracted form of video to efficiently detect most true repeats with a reasonable size of errors, while the second stage detector performs more accurate classification on candidate repeat clips based on their full frames; 3) two detection modes: stand-alone and reference mode. The stand-alone mode is detecting unknown repeats from one input video through self-similarity analysis, while reference mode is detecting repeats between two videos by cross-similarity analysis, and this mode can be used to search known video clips in video stream; 4) locality-sensitive hashing (LSH) of video unit features used to reduce repeat searching complexity; 5) repeat instance extraction and labeling.

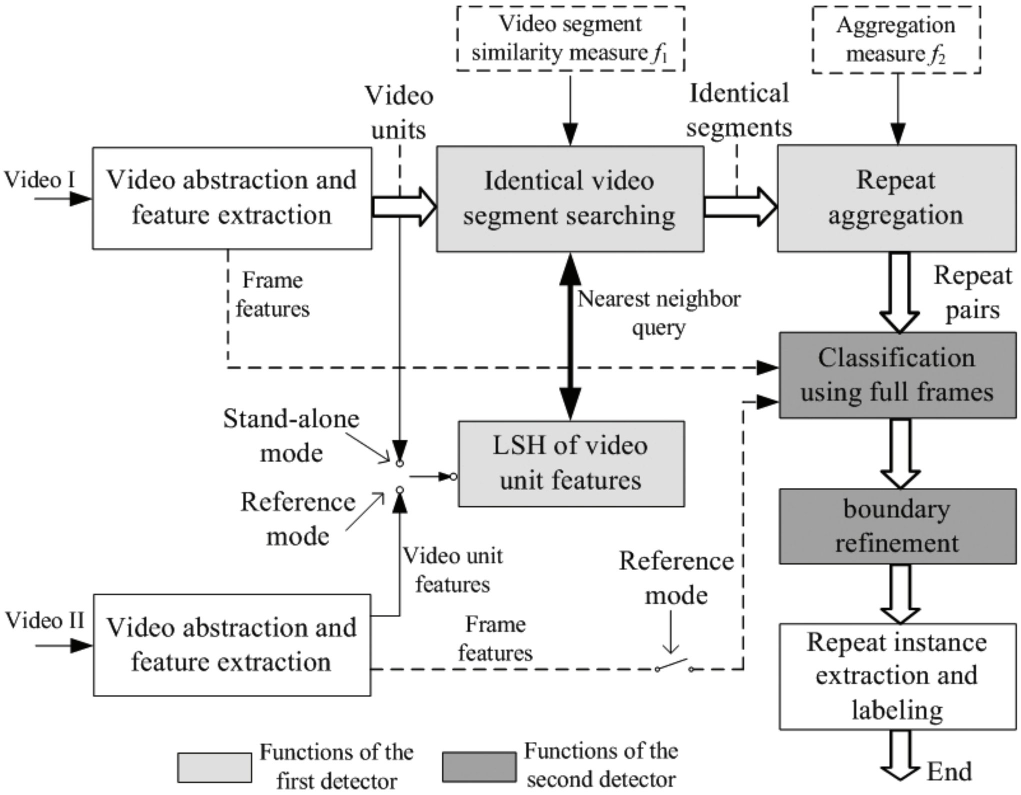

Video segment similarity measure f_1
Aggregation measure f_2
Video units
Identical segments
Video I
Video abstraction and feature extraction
Identical video segment searching
Repeat aggregation
Frame features
Nearest neighbor query
Repeat pairs
Classification using full frames
Stand-alone mode
LSH of video unit features
Reference mode
boundary refinement
Video unit features
Reference mode
Video II
Video abstraction and feature extraction
Frame features
Repeat instance extraction and labeling
Functions of the first detector
Functions of the second detector
End

Fig. 4. Framework of repeat clip mining

The first step in this framework is video data reduction and feature extraction. Raw video data is sampled or segmented into a sequence of basic video units (VU) and their features are calculated to feed into the first detector. Meanwhile feature of each frame is computed for the second detector.

The first stage detector discovers unknown repeats through two steps. Firstly *W* neighbor video units are bunched into bigger size video segments (VS), e.g. two units as one group. Then time-indexed segments are compared with each other using a binary similarity measure f_1 to find identical segment pairs. The set of identical segment pairs is denoted as $\omega \; : \{(VS_{i_1}, VS'_{j_1}), (VS_{i_2}, VS'_{j_2}), \cdots, (VS_{i_n}, VS'_{j_n})\}$, where $i_1, \ldots, i_n$ and $j_1, \ldots, j_n$ are time indexes, and VS_i, VS'_j represents the *i*th and *j*th video segment. If in stand-alone mode VS_i and VS'_j will belong to the same video, but in reference mode they will lie in two input videos respectively. Secondly identical segment pairs are aggregated into repeat clips by aggregation measure $f_2 : \omega' \subset \omega \mapsto (VC, VC')$, where (VC, VC') is a repeat clip pair, and in reference mode VC, VC' will lie in two input videos respectively. If binary similarity values between time-indexed video segments are embedded into a binary matrix S in which '1' represents one identical segment pair, then repeat clips will correspond to diagonal tracks of '1'. Under stand-alone mode S is a self-similarity matrix, so diagonal tracks should offset from the main diagonal by a certain time.

The first detector also contains an LSH function to reduce complexity of identical segment searching. Suppose similarity measure f_1 is based on distance in feature space, each video segment only needs to compare with those within a distance to find its identical ones, which would be a small portion of the total segments. LSH is such a method that can efficiently perform approximate nearest neighbor search even in high dimensional space. In stand-alone mode all video units of the input video will both be indexed by LSH and used as queries, but in reference mode only one video's units are indexed, while the other's act as queries.

To improve detection accuracy the second cascaded detector employs full frames matching to further classify candidate repeat pairs obtained from the first detector. Once a repeat pair is verified a boundary refinement is followed to extend their boundaries to the maximum ones.

The last step of this framework is to extract repeat instances from repeat pairs and group them into multiple categories each of which represents an independent repeat pattern.

3.2 Video representation

The two stage detectors adopt different video representations. The first stage detector performs video abstraction to reduce detection complexity. However, data loss by abstraction will downgrade the detection accuracy. Therefore it is important to choose suitable abstraction strategy to balance detection efficiency and accuracy, and we are especially interested in that for short video repeats detection.

3.2.1 Video abstraction strategy

Video abstraction methods usually depend on frame sampling. Uniform sampling is simple, but it disregards video characteristic, so content redundancy can not be well removed. Moreover, this method is not robust to boundary shift between two repeat objects. Content-based keyframe selection is more effective for video abstraction. It can well remove video redundancy while capturing significance of video sequence (Zhang et al., 1997). Generally it can achieve more than ten times frame reduction. Shot based video representation is also widely used in video analysis. However, its granularity is probably too big to effectively detect many short video repeats.

Content-based keyframes is the first choice for video abstraction in our approach. Keyframe selection can be based on color or motion criterions. The interval between two consecutive keyframes is treated as the basic video unit (VU), as is shown in Fig. 5, and its average granularity should be smaller than that of shots. This video unit representation can also compensate temporal information loss of simple keyframes sampling because temporal and motion features can be extracted from the unit. The second level video representation (VS) is formed by grouping two neighbor units, e.g. the i*th* unit and (i+1)*th* unit forming the i*th* video segment. Compared to the first level, the second level has almost the same number of video samples, but its discriminative ability will improve a lot, thus providing a less noisy platform to build higher repeat clip level.

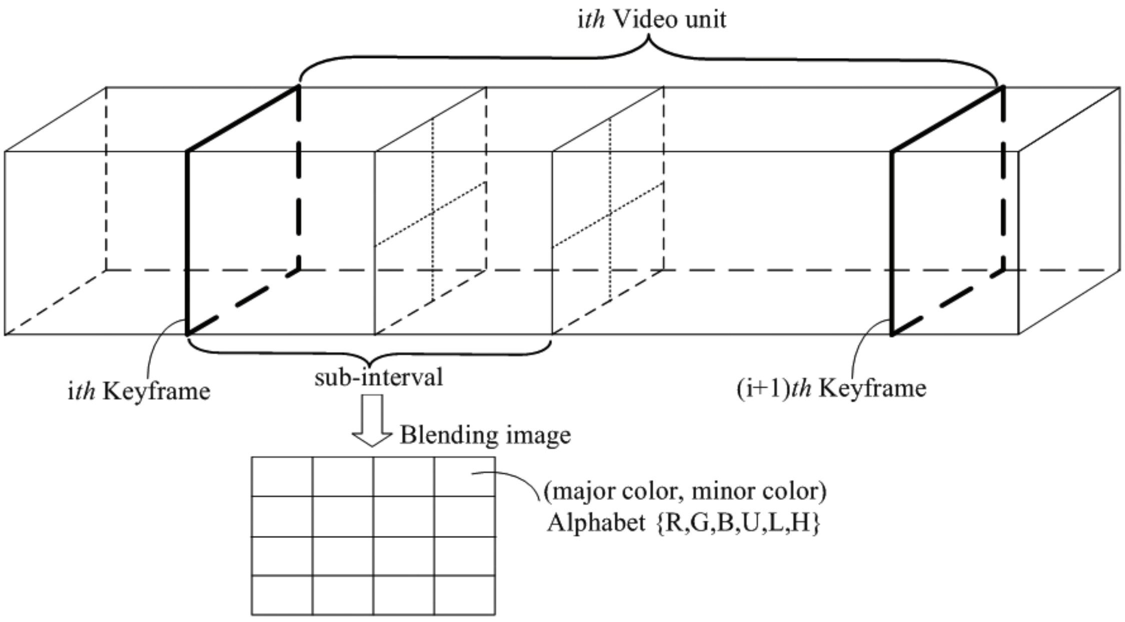


Fig. 5. Illustration of video segmentation and feature extraction

3.2.2 Video features

Two types of video features are extracted. The first one is video unit (VU) feature used by the first stage detector, and the second one is frame feature for the second stage detector.

1. Video unit feature

Video unit feature includes interval length and color fingerprint proposed by Yang et. al (2004). A video unit is partitioned into *K* sub-intervals, and represented by *K* blending images formed by averaging frames within each sub-interval along time direction. Each blending image is then divided into *M*×*N* equal size blocks each of which is represented by the major and minor color components among RGB, as illustrated in Fig. 5. Color fingerprint is the ordered catenation of these block features. If $\overline{R}$, $\overline{G}$, $\overline{B}$ are the average color values of a block, and their descending order is (V_1, V_2, V_3), then the major color and minor color are determined by the following rules:

Rule 1: if $V_1 > V_3$,

$$\text{Major Color} = \begin{cases} \arg\max(\overline{R}, \overline{G}, \overline{B}) & \text{if } (V_1 - V_3) > \tau \\ \text{Uncertain} & \text{if } (V_1 - V_3) \le \tau \end{cases}$$

$$\text{Minor Color} = \begin{cases} \arg\min(\overline{R}, \overline{G}, \overline{B}) & \text{if } (V_2 - V_3) > \tau \\ \text{Uncertain} & \text{if } (V_2 - V_3) \le \tau \end{cases}$$

Where τ is the parameter that controls the robustness to color distortion and discriminative ability of this feature.

Rule 2: if $V_1 = V_3$ (gray image),

$$\text{Major Color} = \text{Minor Color} = \begin{cases} \text{bright} & \text{if } V_1 > \tau_1 \\ \text{dark} & \text{if } V_1 \le \tau_1 \end{cases}$$

Major and minor color patterns have six possible symbol values from alphabet {R, G, B, U, L, H}, where U, L and H stand for uncertain, dark and bright respectively. In this work one blending image (*K*=1) is used for each unit, and divided into 8x8 blocks (*M*=*N*=8), thus the color feature is a 128 dimensional symbol vector. Parameter τ is set to 3.0 to achieve good balance between robustness and discriminative ability, while τ_1 is set to 127.

2. Frame feature

Each frame is divided into 4 sub-frames, and RGB color histogram(8x8x8 bins) of each sub-frame is quantized to a symbol by VQ, so each frame is represented by 4 symbols.

3.3 Detector functions

3.3.1 Video segment similarity measure

Given two video units vu_i and vu_j, their distance $D(vu_i, vu_j)$ is defined as

$$D(vu_i, vu_j) = \sqrt{d^2(F_i, F_j) + \left[len(vu_i) - len(vu_j)\right]^2} \tag{2}$$

where F_i, F_j are color fingerprint vectors of vu_i and vu_j, $d(F_i, F_j)$ is color fingerprint distance function (Yang et. al, 2004), *len*(·) is length feature. Similarity measure f_1 between the *i*th segment and *j*th segment VS_j : $\{vu_j, vu_{j+1}\}$ is defined as:

$$f_1(VS_i, VS_j) = \begin{cases} 1 & \text{if } D(vu_i, vu_j) < \varepsilon_1, D(vu_{i+1}, vu_{j+1}) < \varepsilon_1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where ε_1 is distance threshold.

3.3.2 Complexity reduction by LSH indexing

We apply LSH indexing on the color fingerprint to efficiently retrieve nearest neighbors of a video unit. LSH can perform ($\lambda;\varepsilon$)-NN similarity search in sub-linear time (Indyk and Motwani, 1998). For a query point **q** in *d*-dimensional space, if we want to retrieve its nearest neighbors in distance ε, LSH can return indexed points within a distance $(1+\lambda)\cdot\varepsilon$ ($\lambda>0$) with high probability, while those beyond this distance with low probability, and its efficiency is related to λ. Similarity search is realized by designing a set of hash functions that can make hash collision probability for two points be related to their distance. If two points are close, their hash collision probability should be high; otherwise it should be low. By using multiple such hash functions with different parameters in parallel, LSH can reduce the false negative rates.

LSH algorithm proposed by Gionis *et al.*(1999) transforms a feature vector into a bit string, and selects a subset of bits that satisfies locality-sensitive property. The color fingerprint can be easily converted to bit vector without incurring extra errors. The hashing algorithm is described in Table 2. According to this hash function, the hash collision probability of two points with *p* percent bit difference can be estimated as $1-\left[1-(1-p)^k\right]^l$. The two parameters *k* and *l* should be tuned to balance NN search accuracy and efficiency. Increasing *k* can improve recall but result in more non-nearest neighbors at mean time.

```
Select parameters $k$ , $l$, and a hash table size M;
For $i$=1 to $l$ ,
Produce $k$ different random integers $\{ I_1^i,...,I_k^i \}$ within [1, $n$]
End
Produce $k$ random integer hash coefficients $\{ h_1,...,h_k \}$
Given a binary vector $B$=$\{ b_1,...,b_n \}$
For $i$=1 to $l$,
Select subset of bits of $B$ $\{ b_{I_1^i},...,b_{I_k^i} \}$,
$sum = b_{I_1^i} \cdot h_1 + \cdots + b_{I_k^i} \cdot h_k$
$res$ = $sum$ mod M;
Insert time index of $B$ and partition number $i$ into bucket $res$.
End
```

Table 2. LSH algorithm

3.3.3 Repeat aggregation algorithm

Repeat clips will appear as diagonals in similarity matrix. However, due to segmentation errors, the line will not be the integrated one. Moreover those line fragments will not be collinear if non-uniform partition is used. Fig. 6 shows part of a similarity matrix computed in our experiment. As we can see, diagonal tracks are fragmented and contaminated by noises. To get the whole repeat clip correctly we design a hierarchical aggregation algorithm purely based on temporal boundaries of repeat segments.

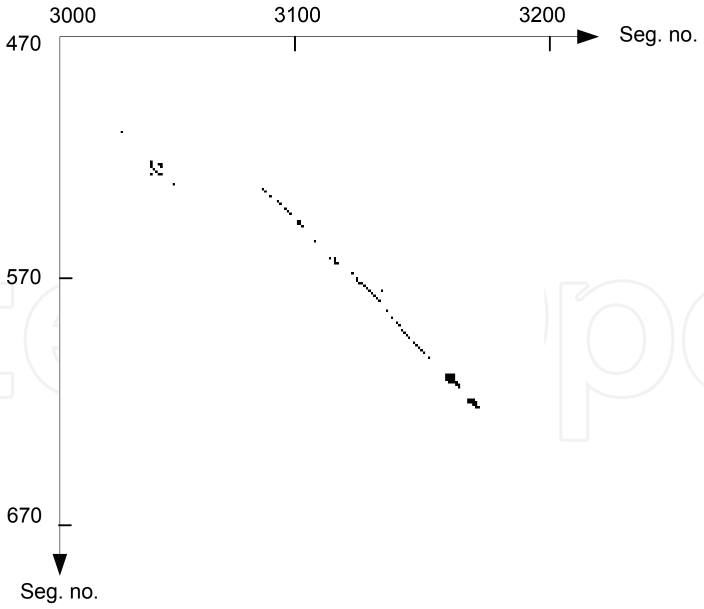


Fig. 6. Example of diagonal tracks for repeat sequences

This algorithm is described as follows:

Step 1. First link strong diagonal tracks whose length exceeds one. The start and end time of two pairs of repeat sequences (I,I′) and (II,II′) corresponding to two diagonal lines are represented by ($T1_{start}$,$T1_{end}$), ($T1'_{start}$,$T1'_{end}$) and ($T2_{start}$,$T2_{end}$), ($T2'_{start}$,$T2'_{end}$) respectively, which is illustrated in Fig. 7.

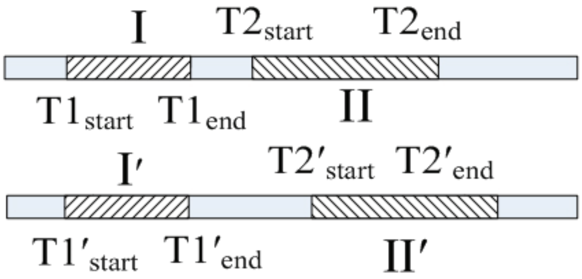


Fig. 7. Illustration of two pairs of adjacent repeat segments

If one of the two conditions in (4) is satisfied, (I,I′) and (II,II′) will be merged into one repeat pair.

a. Overlap: $T1_{start} \leq T2_{start} \leq T1_{end}$, $T1'_{start} \leq T2'_{start} \leq T1'_{end}$

b. Adjacency: $|T2_{start} - T1_{end}| < \mu_1$, $|T2'_{start} - T1'_{end}| < \mu_1$, $|(T2_{start} - T1_{end}) - (T2'_{start} - T1'_{end})| < \varepsilon_2$ (4)

where μ_1 defines neighborhood distance, ε_2 is displacement allowed for neighbor repeat segments, thus controls temporal variations of the whole repeat clip.

Boundaries of merged repeat pair are computed as:

$$T_{start}=\min(T1_{start},T2_{start}),\ \ T_{end}=\max(T1_{end},T2_{end})\ ;$$

$$T'_{start}=\min(T1'_{start},T2'_{start}),\ T'_{end}=\max(T1'_{end}\ ,\ T2'_{end}).$$

This new repeat pair will be put into the repeats list to replace originals, and the above process is iterated till no change of the list.

Step 2. Connecting single dots based on results of step 1 with the same merging criterion as step 1.

Step 3. The connected sequences after above two steps are further connected and merged until there is no change.

By the above aggregation algorithm the whole image of repeat clips can be well constructed from their local repeat segments, thus providing good foundation for further similarity analysis and boundary refinement. Moreover, this algorithm only needs to store boundaries of repeat segments but not similarity matrix, which can have efficient implementation for even large video data mining.

3.3.4 Second stage matching

In this stage the repeat pairs obtained from the first stage detection are further matched using their full frames in order to remove false positives. If a repeat pair only has minor length variation, simple frame by frame matching method can be used; otherwise, dynamic programming should be used to align two symbol sequences with different length. The total number of identical symbols of aligned frames is normalized by the average sequence length to get the similarity score. A repeat pair is classified as true one if the following condition is satisfied,

$$score > (1+e^{-L})\varepsilon_3 \tag{5}$$

where *score* is the similarity value, *L* is the minimum length of the two clips in seconds, and ε_3 is threshold. This decision rule uses soft thresholds for different length sequences. Since shorter sequences are assumed less reliable ones, they should satisfy more stringent condition to pass through verification.

For those repeat pairs verified as true ones, their boundaries are to be extended frame by frame as follows: if the external neighbor frames of current left or right boundaries are similar, corresponding boundaries will be expanded by one frame. This process continues until dissimilar frames are encountered.

3.3.5 Repeat labeling

In this step different repeat pairs belonging to the same type of repeat pattern are to be unwrapped to get instance representation and grouped together. The basic labeling method is based on transitivity between repeat pairs. If instance *A* is paring with instance *B*, and *B* is paring with *C*, then clip *A*, *B*, *C* will have the same label. However, problem is not that simple, because one repeat instance may have length and boundary polymorphisms in different repeat pairs, which can create ambiguity for repeat instance extraction and labeling. The first type of polymorphism comes from partial repeats of one repeat pattern or from detection inaccuracy; the second type comes from independent sub-repeated patterns joined together. As shown in Fig. 8, repeat instances I, I´ contains two independent sub-repeated patterns II and III. II and III don't have sub-repeated patterns, but III have a partial repeat instance III´. In this case I and II should not be grouped together, as they most likely carry different syntactic or semantic information, but III and III´ should be grouped together. Although it is hard to tell which type the polymorphism is if without really understanding underlying contents, the two polymorphisms can be well classified using boundary overlap information between candidate repeat instances.

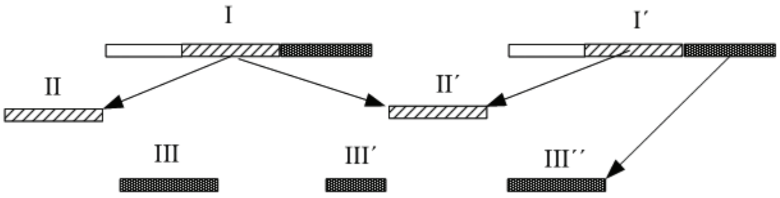


Fig. 8. Video repeats polymorphism

The instance extraction and labeling algorithm is as follows: Given a repeat pair containing candidate instance *A* and *B*, compute their overlapping with labeled instances respectively. Overlapping between instance *A* and labeled instance *C* is computed by (6)

$$overlap = \frac{\min(right(A), right(C)) - \max(left(A), left(C))}{\max(len(A), len(C))} \tag{6}$$

where *left*(.), *right*(.) means left and right boundary, *len*(.) means length of the sequence. If *overlap* is greater than a value, *A* and *C* are classified as the first type repeat polymorphism, and will be merged into a new candidate repeat instance *D* which has the leftmost and rightmost boundary of *A* and *C*. If only *A* finds its matched one from labeled instances, *B* will join the group too, or vice visa. If both *A* and *B* find respective matched instances but with different labels, then one group will be removed and merged into the other group. If neither *A* nor *B* finds matched instances, they will be put in a new group.

3.4 Detector parameter learning

The two cascade detectors contain several parameters, like distance thresholds, LSH parameters etc., but the intrinsic and crucial ones that affect detection accuracy are ε_1, $\varepsilon_2$and ε_3 in (3)(4)(5) respectively. Tuning these three parameters can significantly change detection results. The three parameters have clear physical meanings. ε_1 reflects feature distortion of identical video units for certain video data and feature extraction; ε_2 that defines maximum temporal displacement between neighbor repeat segment pairs in clip aggregation function is related to video unit granularity and temporal variation allowed for the whole repeat clips. ε_3 in the second detector balances recall and precision. Parameter μ_1 in (4) defining neighborhood of repeat segments is not crucial for final results as long as it is in a range, e.g. 10s ~ 20s. Segmentation related parameter is important for final results, but it is not intrinsic to the detector. Different segmentation methods may have different types of parameters or no parameter at all.

LSH parameters generally affect detection speed but not the accuracy, and they can be empirically chosen to achieve high efficiency, which will be shown later in experimental part. Threshold in repeat labeling stage just affects labeling errors, but not recall and precision of repeats. Moreover, since human interaction is indispensable in labeling stage, labeling errors can always be corrected manually or by adjusting the threshold.

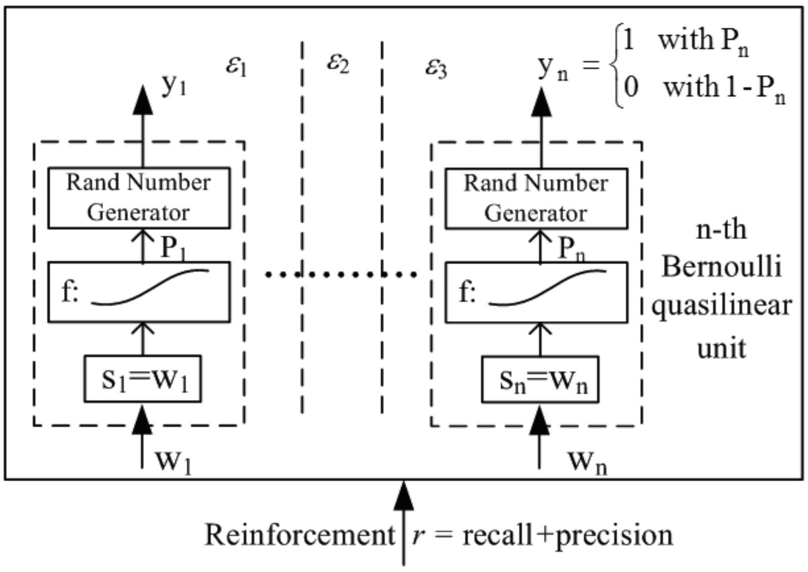

ε_1
ε_2
ε_3
y_1
$y_n = \begin{cases} 1 & \text{with } P_n \\ 0 & \text{with } 1-P_n \end{cases}$
Rand Number Generator
P_1
f:
$s_1 = w_1$
w_1
Rand Number Generator
P_n
f:
$s_n = w_n$
w_n
n-th Bernoulli quasilinear unit
Reinforcement r = recall+precision

Fig. 9. Connectionist reinforcement learning network

In the following we will propose a method to learn appropriate values of ε_1, ε_2 and ε_3 in order to achieve optimal performance on selected video data. Given certain segmentation and feature extraction, the three parameters in the two detectors are trained together by reinforcement learning in a non-associative paradigm. Given an input, the learning network produces the three parameters, then a scalar indicating "goodness" of detection results under these parameters is immediately used as a reinforcement for the learning network. In our approach the sum of recall and precision is taken as the reinforcement factor. We also adopt the connectionist REINFORCE algorithm (Williams, 1992) in which the units of network are Bernoulli quasilinear units whose output is 0 or 1, statistically determined by Bernoulli distribution with parameter $p = f(s) = 1/(1+\exp(-s))$, which is shown in Fig. 9. Each Bernoulli quasilinear unit has one input weight, and the three parameters are encoded by gray codes corresponding to the outputs of *n* Bernoulli quasilinear units. After receiving a reinforcement *r*, the weights of Bernoulli quasilinear units are updated by

$$\Delta w_i = \alpha(r-b)(y_i - p_i) \tag{7}$$

where α is a positive learning rate, *b* serves as a reinforcement baseline, y_i is the output of the *i*th Bernoulli quasilinear unit, and p_i is the Bernoulli distribution parameter. It has been shown by Williams (1992) that this learning algorithm statistically climbs the gradient of expected reinforcement in weight space, which means that the detector parameters will change in the direction along which the sum of recall and precision increases.

3.5 Results

This section presents experimental results of repeat detection accuracy and efficiency, performance comparison for different segmentation methods, and news video structure analysis results. Experiments are mainly conducted on news videos, but several hour movies and commercials are also included. CNN and ABC news videos from TRECVID data are chosen to form two video collections each of which contains 12 day programs with 6 hours around. The CNN news videos contain a lot of repeated commercials and program logos, such as headline news logo, health program and sports program logo etc, with length ranging from less than 1s to 60s. ABC news videos contain few program logos, but still contain many repeated commercials. It is also observed that there exist non-trivial distortions between many repeat clips especially for those short program logos, and distortions include color distortion, caption overlay, length truncation etc. So the two video collections provide a very good platform to test the robustness of a detection approach on variable length short video repeats.

A . Ground truth setting

To calculate recall of repeat clips, we manually inspect the two collections to find short repeats. Since manual identification of unknown repeats is quite difficult, we first boost a repeat detector with empirical parameters to get initial results based on which we further search interesting repeated program logos and commercials, but miss still scenes, such as anchor shots, black frames. By this approach we found 34 kinds of repeated clips with totally 186 repeat instances from CNN collection. Among the 34 repeat types 14 are program logos, while the rest are commercials. From ABC collection we found 35 kinds of repeated clips that have totally 116 instances, but only two types are program lead-in and lead-out, while the rest are all commercials.

B. Detector training

Parameters of the two detectors are learned by the approach presented in section 3.4. Three hour CNN news videos are randomly chosen for training. Videos are segmented by content based keyframes which are selected by the following criterion,

$$\left|1-\mathrm{inter}(H_1, H_0)\right| > \eta \tag{8}$$

where H_1 and H_0 are color histograms(RGB 8x8x8bins) of current decoded frame and the last keyframe respectively, $\mathrm{inter}(\cdot,\cdot)$ is histogram intersection, η is threshold set to 0.15. Video unit feature is 128 dimensional color fingerprint plus unit length. Frame feature is 4 symbols, and codebook size of LBG VQ is 128. The vector quantizer is trained by histogram features of 5000 frames randomly chosen from a collection of commercials.

The reinforcement learning rate α in (7) is set to 0.01 and reinforcement baseline *b* set to 0.7. Parameters ε_1, ε_2 and ε_3 are each encoded by 5 bit gray code, so there are totally 15 Bernoulli units in this network. Parameter value range is set to [0,1]. Initial parameters are set to empirical values, and initial weights are all zeros. During each learning round we manually check the detection results to compute recall and precision, then feed their sum as reinforcement of the learning network. Recall and precision are calculated as (9).

$$
\begin{aligned}
\text{recall} &= \frac{\text{number of correct repeat instances}}{\text{number of all true repeat instances}} \\
\text{precision} &= \frac{\text{number of correct repeat instances}}{\text{number of all detected instances}}
\end{aligned}
\tag{9}
$$

In experiment recall and precision in the first round learning are 74% and 100%, but after ten rounds of learning, recall and precision already climb to 94.2% and 96% respectively. Since the next several rounds of learning do not lead to reinforcement increase, we then stop the learning.

C. Testing results

The trained detectors are tested on the rest 3 hour CNN videos and 6 hour ABC videos. Recall and precision on CNN videos are 92.3% and 96%, while 90.1% and 90% those for ABC videos. This accuracy is obtained without setting a minimum sequence length to filter errors, so most of the errors come from those very short clips. The shortest correct repeat detected is just 0.26s (partial of "play of the day" logo in CNN video), while the longest one is 75 seconds long. Our results also show that partial repeats can be effectively detected. In some repeat categories the shortest instance is less than half of the longest one.

Fig. 10 shows temporal distribution of short video repeats identified from CNN news videos of six days. Those repeat instances linked by curves are chosen as marker instances. From this map we can clearly see that the whole program is segmented into several layers each of which contains certain topics, such as health program, top stories, financial news, sports news, commercials et al.

We also measure how accurately the detected boundaries of repeat pairs approach their maximum boundaries which are manually chosen. We selected 300 repeated pairs that cover almost all repeat patterns and checked their boundary shift before boundary refinement. The smallest shift is 0 s, while the largest one is 16.4s. The average shift is 0.47s. Around 80% of the shifts are within 0.2 seconds. After frame by frame boundary refinement those large shifts can be effectively reduced to 0~1 second.

D. Performance analysis of segmentation methods

Video segmentation is essential for this approach, so experiments are conducted to compare performances by proposed keyframe based segmentation, uniform segmentation and shot segmentation. The video data are 3 hour CNN videos used in Section 3.5-*B*. Two keyframe based segmentations are implemented with η=0.15, 0.30 respectively. Uniform segmentation utilizes I frames (every 12 frames). Shot detection includes cuts, fades in-outs and dissolves. Video unit features for all segmentations are color fingerprint and length. The video segment (VS) size *W* for shot segmentation is set to 1, and the minimum number of diagonal points in repeat aggregation is also set to 1. Thus this method can not only detect single repeat shots, but also repeat clips beyond shots. Detectors are separately trained for each segmentation to achieve their nearly optimal performance, and training results are shown in Table 3.

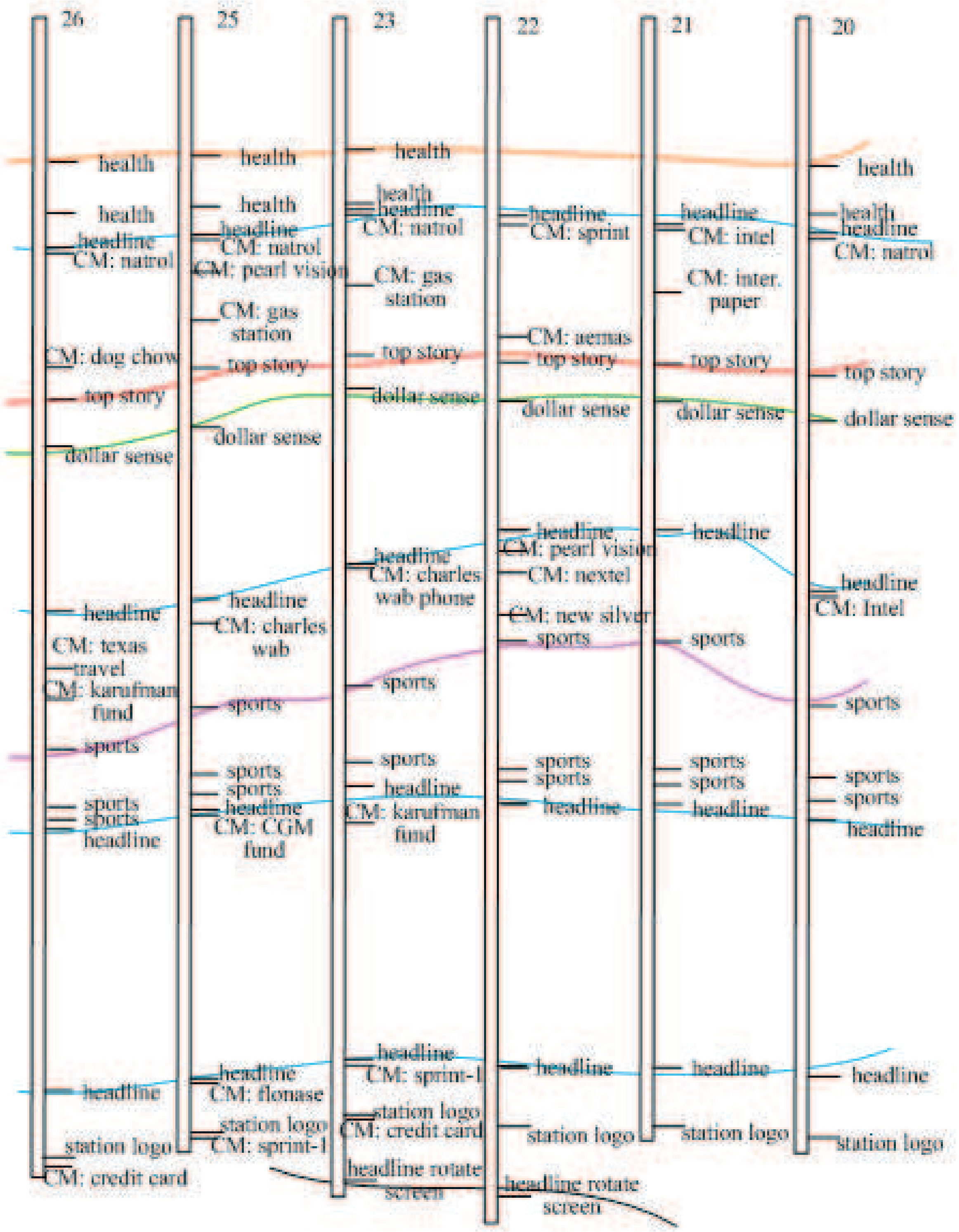


Fig. 10. CNN news video structure analysis by video repeats

	Uniform sampling	Keyframe (η =0.15)	Keyframe (η =0.30)	Shot based
recall	87.8%	94.2%	90.7%	66.7%
precision	95.9%	96.0%	86.0%	84.7%
Video units	26344	14872	6316	1911

Table 3. Performance comparison of video segmentation methods

From Table 3 we know that keyframe based segmentation achieves best performance. The uniform segmentation results in several times more video units than keyframes, but still gets lower recall on program logos and commercials. Uniform segmentation also detects quite many stationary scenes, such as anchor shots and black frames which occupy nearly 74% of the whole detected repeat clips pool thus overwhelm other interesting repeat patterns like program logos and commercials. Under keyframe based segmentation these still scenes are all filtered, program logos and commercials are main body of detected repeats. Shot based segmentation results in much fewer video units, but its total accuracy is much lower and many fast changing program logos are missed. When granularity of keyframe based segmentation becomes bigger, its performance will also drop because of heavier data loss.

E. Efficiency evaluation

1) **Speed**

In experiment LSH parameters are set as: *k*=50 and *l*=30, thereby the average number of retrieved units for a query unit of CNN collection (totally 629,380frames and 31496 units) is 320, and the number of color feature comparisons are further reduced to 20 by pre-filtering one dimension length feature at trained distance threshold ε_1 =0.1, thus speedup factor is about 1575 compared to pair-wise searching. For ABC collections (totally 616,780 frames and 29838 units), the average number of retrieved units for a query unit is 1026, and further reduced to 56 by length filtering, thus speedup factor is 533. On PC with Pentium-4 2.5GHz processor the two stage detections on 6 hour CNN videos can be finished in 22 seconds, while 40 seconds for ABC videos.

It is found that adjusting LSH parameters to shrink nearest neighbor distance would significantly reduce feature comparison complexity, but final recall almost does not change. For instance, when *k* =50 and *l*=30, recall of video units at distance threshold 0.1 for a query unit is roughly 37.4%, which is estimated from the hash collision probability. When *k* = 70 and *l* = 30, recall of video units at distance threshold 0.1 decreases to 8.4%. The average number of retrieved units for a query decreases from 320 to 186 for CNN collection, and from 1026 to 773 for ABC collection, but final recall of repeats does not drop, while precision even increases a bit. The main reason includes robust color fingerprint and repeat aggregation algorithm that is quite robust to loss of identical video segments on diagonal tracks. This property gives this approach much space to pursue speed without compromising accuracy.

When this approach works on reference mode to search one-minute video clip against 12 hour video database, the whole detection can be finished in 1 second after video features are extracted.

2) **Storage requirement**

In our implementation the color fingerprint is a 128-byte string, and length feature is a 4-byte float number. Frame feature is represented by four bytes. In our approach similarity matrix is not needed to be created, but only temporal boundaries of repeat segments should be stored, which usually occupy small memory and can be ignored. The storage requirements for 24 hours video database are as follows:

Color fingerprint + length feature: 15.5 Mega-bytes
Frame feature: 9.9 M; Hash table: 150 M
Total: 175 M

4. Conclusion

In this chapter we have proposed two methods to identify known and unknown video repeats respectively, and different machine learning approaches are used in the two methods. Composite HMM approach used for known short video repeat identification models the probabilistic temporal relation of the clip's shots as well as their frame feature distributions, and the model is trained by statistical learning approach. Although the model building process appears somewhat complex, the reward is that this approach can resist severe video content tamper, such as random shots removal. The color histogram used as observed feature in HMM shows robustness to feature distortion induced by transcoding.

The proposed unknown video repeat mining method achieves high accuracy on arbitrary length video repeats detection by cascaded detectors that employ different features and similarity measures. The detectors' performance can be efficiently optimized in a few rounds of reinforcement learning, which makes our approach easily adapt to different video data. Repeat searching complexity is largely reduced through video abstraction and LSH indexing. Moreover, the robust color fingerprint feature and repeat aggregation measure enable much space to pursue speed without compromising final detection accuracy. Video segmentation utilizing content-based keyframes achieves best balance between detection accuracy and efficiency on short video repeats compared to uniform and shot based segmentation. Results also show that short video repeats mining is an effective way to discover syntactic structure of news videos.

5. References

Agnihotri, L; Dimitrova, N; McGee, T; Jeannin, S; Schaffer, D& Nesvadba, J (2003). Evolvable visual commercial detector. *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2003.

Bao, Z; Eddy, S.R. Automated deno vo identification of repeat sequence families in sequenced genomes. Genome Research, 2002, 12 (8): 1269-1276.

Cheung, S-C. & Ngueyen, T P. (2005). Mining Arbitrary-length Repeated Patterns in Television Broadcast. *Proc. IEEE Int. Conf. on Image Processing*, 2005.

Cooper, M & Foote, J (2001). Scene Boundary Detection via Video Self-Similarity Analysis. *Proc. IEEE Int. Conf. Image Processing, 2001.*

Gionis, A; Indyky, P & Motwaniz, R (1999). Similarity Search in High Dimensions via Hashing. *Proc. Int. Conf. Very Large Data Bases*, pp. 518—529, 1999.

Herley, C (2006). ARGOS: Automatically Extracting Repeating Objects From Multimedia Streams. *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 113-129, Feb. 2006.

Indyk, P & Motwani, R (1998). Approximate Nearest Neighbor - Towards Removing the Curse of Dimensionality. in *Proc. the 30th Symposium on Theory of Computing*, 1998, pp. 604-613.

Kashino, K; Kurozumi, T & Murase H (2003). A quick search method for audio and video signals based on histogram pruning. *IEEE Trans. Multimedi*a, vol. 5, no. 3, pp. 348–357, Sep. 2003.

Kulesh, V; Petrushin, V & Sethi, I (2002). Video Clip Recognition Using Joint Audio-Visual Processing Model. *Proc. Int. Conf. on Pattern Recognition* , August 2002, Quebec City, Canada.

Lienhart, R; Kuhmunch, C & Effelsberg, W (1997). On the detection and Recognition of Television Commercials. *Proc. IEEE Int. Conf. Multimedia Computing and Systems,* 1997.

Oostveen, J.C; Kalker, A.A.C & Haitsma, J.A (2001). Visual hashing of digital Video: applications and techniques. *Proc. SPIE applications of digital image processing XXIV*, July/August 2001, San Diego, USA.

Pua, K.M & Gauch, J.M (2004). Real time repeated video sequence identification. *Computer Vision and Image Understanding* 93 (2004) pp.310-327.

Rabiner, L.R (1989). A tutorial on hidden Markov model and selected application in speech recognition. *Proceedings of the IEEE*, 77 (2), 257-286, Feb. 1989.

Snchez, J.M; Binefa, X & Vitri, J (2002). Shot partitioning based recognition of TV commercials. *Multimedia Tools and Applications* 18 (2002), pp. 233–247.

Kurtz, S; Schleiermacher, C. REPuter: Fast computation of maximal repeats in complete genomes. Bioinformatics, 1999, 15 (5): 426-427.

Williams, R.J (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learnin*g, vol. 8, pp. 229-256, 1992.

Yang, X; Tian, Q & Gao, S (2003). Video clip representation and recognition using composite shot models. *Proc. IEEE Pacific Rim Conf. Multimedia 2003*, Singapore, Dec. 2003.

Yuan, J; Duan, L.-Y; Tian, Q & Xu, C (2004). Fast and robust short video clip search using an index structure. *Proc. ACM Multimedia's Multimedia Information Retrieval Workshop*, 2004.

Yang, X; Tian, Q & Chang, E.C (2004). A Color Fingerprint of Video Shot for Content Identification. *Proc. ACM Multimedia* 2004, NY, USA, 2004.

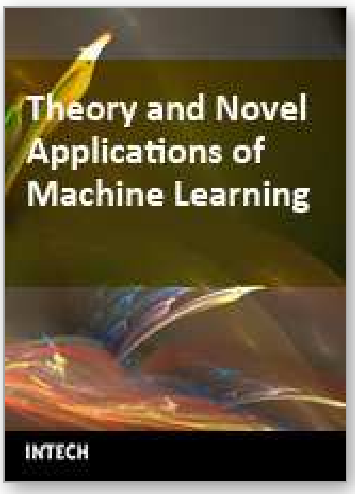
Yang, X; Xue, P & Tian, Q (2007). Automatically Discovering Unknown Short Video Repeats. *Proc. Inter. Conf. on Accoustic, Speech and Signal Processing (ICASSP)*, Hawaii, USA, 2007.

Yang, X; Xue, P & Tian, Q (2005). A repeated video clip identification system. in *Proc. ACM Multimedia*, Singapore, 2005.

Yang, X; Tian, Q & Xue, P (2007). Efficient Short Video Repeat Identification With Application to News Video Structure Analysis. *IEEE Trans. Multimedia*, vol.9, pp: 600 – 609, April 2007.

Young, S et al. (2000). "The HTK Book ver. 3.0", Cambridge University 2000.

Zhang, H. J; Wu, J; Zhong, D & Somaliar, S.W (1997). An integrated system for content-based video retrieval and browsing. *Pattern Recognition* 30 (1997), no. 4, pp. 643–658.



Theory and Novel Applications of Machine Learning
Edited by Meng Joo Er and Yi Zhou

ISBN 978-953-7619-55-4
Hard cover, 376 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821