PKU-IDM @ TRECVid 2010: Pair-Wise Event Detection in Surveillance Video

Kaihua Jiang [b1], Zhipeng Hu [a*], Zhongwei Chen [c*], Guochen Jia [a], Teng Xu [a], Qiong Hu [c], Guangcheng Zhang [b]
Yaowei Wang [a], Lei Qin [c], Yonghong Tian [a+], Xihong Wu [b], Wen Gao [a]

[a] National Engineering Laboratory for Video Technology, Peking University

[b] Speech and Hearing Research Center, Peking University

[c] Key Lab of Intel. Inf. Proc., Institute of Computing Technology, Chinese Academy of Sciences

[+] Corresponding author: Phn: +86-10-62758116, E-mail: yhtian@pku.edu.cn

Abstract

In this paper, we describe our system for the surveillance events detection task in TRECVid 2010. We focused on pair-wise events (e.g., PeopleMeet, PeopleSplitUp, Embrace) that need to explore the relationship between two active persons. For our team had participated in the TRECVid SED task in 2009, we developed the system based on the old one. The improvements are three-fold. First, we refined the background subtraction method of last year. Some better background frames are automatically selected to train and update the background model and the background reconstruction is performed at pixel level instead of frame level. Second, we employed a MPL (Multi-Pose Learning) based method for head-shoulder detection, which can effectively improve the detection recall. Third, a structural SVM (SVM-HMM) classifier is employed for pair-wise events detection. According to the comparative results in the TRECVid SED formal evaluation, our experimental results are promising.

1. Introduction

This year we chose four events and focused on pair-wise events (e.g., PeopleMeet, PeopleSplitUp, Embrace) that need to explore the relationship between two active persons. As our team had participated in the TRECVid SED task in 2009, we developed this year's system based on the old one (eSur). The improvements are three-fold.

First, we refined the background subtraction method of last year. Some better background frames with fewer foreground objects are automatically selected as training samples to train and update the background model by comparing video frame with a Gaussian background model, and the background reconstruction is performed at the pixel level instead of the frame level. Experimental results show that the method can detect the foreground objects sensitively with much lower false alarms than the classic background modeling methods.

Second, within the extracted foreground region, we used the cascaded HoG(Histograms of Oriented Gradients) [8] for head-shoulder image reorientation, and apply Multiple Pose Learning-based RealBoost for classifier learning. The online Boosting method is then used for tracking each detection part. Intermediate experimental results show that our human detection and tracking technique, together with background modeling, obtains better performance than last year.

Third, a structural SVM classifier is employed for pair-wise events detection. As the events videos are inherently sequential data, we introduced the Hidden Markov Support Vector Machine (SVM-HMM) to model and classify the interactive events with consideration of the statistical dependencies over adjacent frames. Features like distance between two persons are extracted from every frame. Instead of simply concatenating the features into a vector, we treat them as sequential data to exploit not only the discrete information from individual frames, but also the sequence and correlation information among frames. The final detections are parsed from raw sequential results generated by SVM-HMM.

The remainder of this paper is organized as follows. In section 2, we present our system framework briefly. Background subtraction is described in section 3. In section 4, we describe our head-shoulder detection and tracking approach. In section

---

[1] These persons are equally important in the contest. This work is partially supported by grants from the Chinese National Natural Science Foundation under contract No. 61035001, No. 60973055, No. 61072095 and No. 61003165, National Basic Research Program of China under contract No. 2009CB320906, and Fok Ying Dong Education Foundation under contract No. 122008.

5, we present our approach for detecting different events in given surveillance video sequences. Experimental results and analysis are given out in section 6. Finally, we conclude this technical report in section 7.

2. The eSur System Framework

The diagram of our eSur system is shown in Fig.1. The whole system is similar to the one we developed last year and the main difference lies in the event detection module. Last year we used classic linear SVM classifiers and automata to classify and identify different events in this module. However, this module is completed replaced by SVM-HMM and outlier classifier this year. Besides, some significant improvements are achieved in the background subtraction module and the human detection and tracking module.

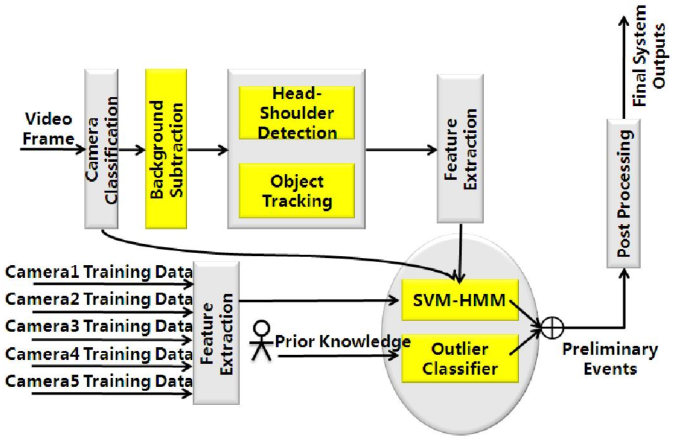

Fig.1 Diagram of our system, eSur

3. Background Subtraction

In our framework, background subtraction is used to extract foreground regions to accelerate the head-shoulder detection and the tracking process. At the same time, the detection and tracking false alarms are decreased effectively.

We proposed a selective eigenbackground method, which is a reformation of the method we used last year. In the training stage, the dimensionality of the training samples is reduced to build a Gaussian model G_m. Then those training samples containing fewer foregrounds are selected to compute the initial eigenbackgrounds according to their similarities to the Gaussian model.

In the subtraction stage, the dimensionality of the input frame vector is reduced to update the Gaussian model G_m in a running average style as in GMM[1]. If the similarity of the frame to the Gaussian model is sufficiently high, incremental PCA is performed to update the eigenbackgrounds. Then the most descriptive eigenbackground is selected for each pixel to reconstruct the background, according to the minimum absolute value of the eigenbackground element. This process is formulated in Equ. (1) – (3), where B(i) is the reconstructed background value of the ith pixel, $\psi_{\kappa i}$ is the reconstructed background frame, $u_{\kappa i}$ is the selected eigenbackground for the ith pixel to reconstruct the background, x is the input frame vector and $u_j(i)$ is the ith element of the jth eigenbackground.

$$B(i) = \psi_{\kappa i}(i) \tag{1}$$

$$\psi_{\kappa i} = u_{\kappa i} u_{\kappa i}^T x \tag{2}$$

$$u_{\kappa i} = \min_j \{|u_j(i)|\} \tag{3}$$

At last, adaptively thresholding is applied to the absolute difference image between the input frame and the reconstructed background image to get the foreground mask image.

4. Detection and Tracking

4.1 Head-Shoulder detection

Pedestrian Detection is an important step in this system. As there are many occlusions in the TRECVid corpus, parts or even the whole body of the pedestrians are frequently unseen. For this reason, we apply head-shoulder detection instead of human body detection.

In [2], Dalal and Triggs proved that Histograms of Oriented Gradients are powerful for pedestrian detection. In order to speed up, Zhu et al. [3] combined the cascaded rejection approach with HOG feature. They used AdaBoost to select the best features and constructed the rejection-based cascade.

In our system, we use HOG feature to represent head-shoulder samples, piece-wise function to construct weak classifiers, and apply Multiple Pose Learning-based RealBoost for classifier learning. Multiple Pose Learning [4] is used to deal with large intra-class variety within the pedestrian's samples of the TRECVid corpus. The framework is presented in Fig. 2.

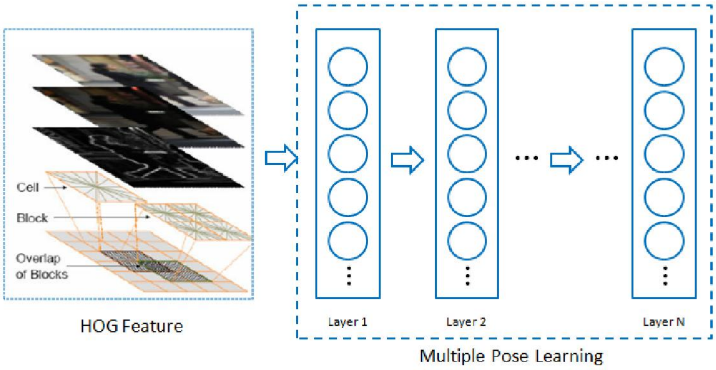

Fig.2 Detection module

The Multiple Pose Learning-based boosting used in this work is described as below.

Given n samples $x_i \in X$ and n corresponding labels $y_i \in \{-1, +1\}$, we assume, however, that there are K latent variables $y_i^k \in \{-1, +1\}$ associated with each sample. Each latent variable defines membership to one of the K groups. A sample is considered positive if it belongs to at least one of these groups, which can be expressed as $y_i = \max_k\{ y_i^k \}$. Our goal is to simultaneously split the positive data into K groups and train K classifiers $H^1, H^2, \ldots, H^K$, one per group, so that $\max_k(H^k(x_i)) = y_i$.The algorithm is summarized as below:

FOR t= 1 TO DO

FOR k=1 TO K DO

Compute weights $\omega_i^k = -\dfrac{\partial \varsigma}{\partial H^k(x_i)}$

Train the best weak classifier with the current weights

$$h_t^k = \arg\min_h \sum_i 1(h(x_i) \neq y_i)\left|\omega_i^k\right|$$

Find α_t^k via line search to minimize cost

$$\alpha_t^k = \arg\min_\alpha \varsigma(\ldots, H^k(x) + \alpha h_t^k, \ldots)$$

Update strong classifier

$$H^k(x) \leftarrow H^k(x) + \alpha_t^k h_t^k$$

END FOR

END FOR

Fig.3 Flowchart of Multiple Pose Learning algorithm.

Some other cues are used for making the detection process more efficient. With the coarse foreground regions extracted by the background subtraction module, candidate sub-windows with sparse foreground can be neglected immediately. We can also estimate the reasonable sub-window size of head-shoulder appeared in all positions for each scene. In addition, regions those have low possibility of events are pruned in the searching process.

In practice, we labeled about 5000 head-shoulders as positive training samples, and collected hundreds of images without head-shoulders as the source to extract negative training samples.

4.2 Tracking

In the TRECVid corpus, target appearance always changes significantly. The same as last year, we use an adaptive Online

Boosting framework for tracking process as described by Helmut Grabner [5].

In camera 3 and 5, the head-shoulder of pedestrians are mostly small and blur, so we extend the head-shoulder detection result proportionally down to use the whole body for tracking instead. Another method is applied to deal with drifting. Dominant color similarity between corresponding object in two frames give a score to evaluate the matching. And Online Boosting tracking also provides a matching score. We combine these two scores to get the final tracked position of an object in each next frame.

5. Pair-Wise Events detection

To detect the pair-wise events in this year's SED task, the interactive events, such as PeopleMeet, PeopleSplitUp, and Embrace, are considered as a time-variant holistic pattern, and proper sequential model and structural classifier are introduced to serve the detection task.

It is comprehensible that the discriminative patterns for these three events in video sequences are inherently time sequential. However, most pervious activity recognition methods did not handle this properly with only modeling the patterns in single frames or simply concatenating them together. In our solution, the event is considered as a whole sequence and described by the stochastic sequential model and classified using support vector machines. Specifically, we employ the Markov Support Vector Machine proposed in [6]. This method handles dependencies between neighboring frames using Viterbi-like decoding and the learning procedure is based on a maximum margin criterion. With the sequential learning method, the temporal correlations between different stages of the event are properly considered, and decisions based on integrated event sequences are reliable and semantically reasonable.

As shown in Fig.4, features are extracted based on the motion trajectories generated by human detecting and tracking module mentioned in previous sections. According to the locations of every person in a frame, we calculate the absolute velocity, the acceleration, the distance between each pair of people and the angular separation of moving directions as the raw features. Then the extracted raw features from the same video clips (ground truth event samples for training and test samples for detecting) are transformed to structural sequence feature. Some statistics of raw features are also included into the reformed features to explicitly employ the information of the temporal dependencies over adjacent frames.

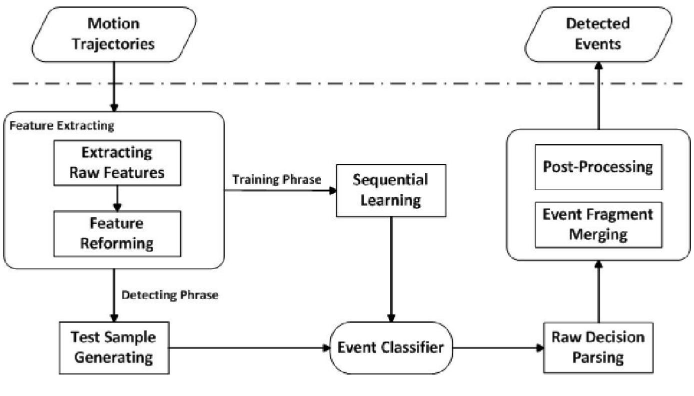

Fig.4 Flowchart of sequential learning based event detection

With the structural features, an appropriate implementation of Hidden Markov Support Vector Machine, SVM-HMM [7], is applied to train events classifiers and make decisions. It learns a hidden Markov model from training samples for each event category and makes sequence decisions for testing samples. As the raw decision is a sequence of binary decisions for each frame in a testing sample, we need to parse it into a single decision for the testing sample with the strategy like voting. As the detection task is actually transformed to a classification problem by using sliding window method to generate testing samples, the original results would be fragmental. So in the post-processing phrase, we merge the preliminary detections and introduce some prior knowledge based rules to filter out incredible detections. These rules are usually empirical restrictions such as a distance threshold between persons before "PeopleSplitUp" or after "PeopleMeet".

6. Experiment and results

Our team submitted four versions of results, which are obtained by using different human detection, tracking and events detection modules.

(a) (b (c)

Fig.5 Background subtraction results. (a) video frame (b)result with classic eigenbackground (c)result with proposed method

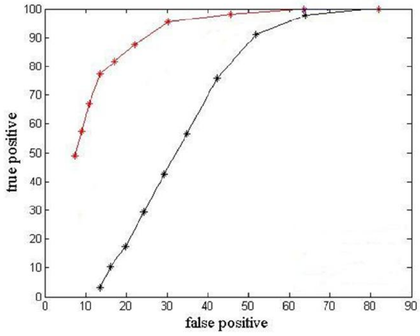

Fig.6 ROC analysis. Black line: classic eigenbackground; Red line: proposed method

Figure. 5 and 6 give the comparison results between the classic eigenbackground method and our proposed method for background subtraction. It can be observed the false alarms and the miss detections are significantly lowered by our selective eigenbackground method.

Table 1 Head-shoulder detection results of this year and last year

Camera1	**Recall**	**Precision**	**F-score**	**Camera2**	**Recall**	**Precision**	**F-score**
Last Year	0.335	0.888	0.4734	Last Year	0.243	0.816	0.3745
This Year	0.539	0.796	0.6429	This Year	0.560	0.773	0.6495
Camera3	**Recall**	**Precision**	**F-score**	**Camera5**	**Recall**	**Precision**	**F-score**
Last Year	0.305	0.728	0.4299	Last Year	0.385	0.662	0.4869
This Year	0.429	0.667	0.5222	This Year	0.468	0.757	0.5783

Table 2 Tracking results of this year and last year

Camera1	**MOTA**	**MOTP**	**Miss**	**FA**	**ID Switch**
Last Year	0.09	0.55	0.571	0.322	0.017
This Year	0.321	0.591	0.51	0.134	0.035
Camera3	**MOTA**	**MOTP**	**Miss**	**FA**	**ID Switch**
Last Year	-0.152	0.552	0.632	0.505	0.016
This Year	0.022	0.571	0.652	0.293	0.033
Camera5	**MOTA**	**MOTP**	**Miss**	**FA**	**ID Switch**
Last Year	-0.866	0.587	0.498	1.339	0.029
This Year	-0.002	0.602	0.537	0.44	0.025

Table 1 and 2 show the comparison detection and tracking results between the best outputs of our system this year and those of last year. It can be seen from the tables that detection result is improved greatly in recall with low or no decrease in the precision. Here we introduce Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP)[8], metrics used in PETS 2009, to evaluate overall performance. These ID switches used in MOTA are calculated from the number of identity mismatches in a frame, from the mapped objects in its preceding frame. The MOTP is calculated from the spatiotemporal overlap between the ground truth tracks and the algorithm's output tracks. Conclusion can be drawn from table 2 that our performance is improved greatly.

Table 3 shows the comparison results between the best outputs of our system this year and those of last year. It can be seen from the table that our eSur system is greatly improved by detecting more correct events. The number of correctly detected PeopleMeet and PeopleSplitUp events is two times more than last year and that of Embrace are raised dramatically. Meanwhile, the false alarms do not rise too much and event dramatically decreased for PeopleSplitUp. This year we did not use any prior knowledge like last year, so it is believed that when prior knowledge is used, the performance can be further improved. It should be noticed that for the events PeopleSplitUp and Embrace, the NDCRs last year of our system are higher than 1.0 but we lowered them below zero this year, which verifies the effectiveness of our improvement methods.

According to the comparative results in the TRECVid SED formal evaluation, our experimental results are promising this year, especially for the events PeopleMeet and PeopleSplitUp where the NDCRs are the lowest among all the participants.

Table 3 Comparsion results between the best ourputs of eSur this year and last year

PeopleMeet	**#Ref**	**#Sys**	**#CorDet**	**#FA**	**#Miss**	**#F-score**	**Act.DCR**
eSur last year	449	125	7	118	442	0.9031	1.023
eSur this year	449	156	12	144	437	0.8570	1.02
PeopleSplitUp	**#Ref**	**#Sys**	**#CorDet**	**#FA**	**#Miss**	**#F-score**	**Act.DCR**
eSur last year	187	198	7	191	180	0.5864	1.025
eSur this year	187	167	16	136	171	0.6505	0.959
Embrace	**#Ref**	**#Sys**	**#CorDet**	**#FA**	**#Miss**	**#F-score**	**Act.DCR**
eSur last year	175	80	1	79	174	0.7932	1.02
eSur this year	175	925	6	71	169	0.8024	0.989

7. Conclusion

This year we improved our system significantly in background subtraction where selective eigenbackground method is proposed, head-shoulder detection where multi-pose learning based method is employed and event detection where SVM-HMM classifier is used for pair-wise events detection and a distance-based outlier detection method is employed to the single-actor event detection. The promising results of our system this year verify the effectiveness of these improvements. However, we believe there are still large improvement spaces for our system in exploring more effective and descriptive event models.

Reference

[1] Stauffer C, Grimson W. E. L. Adaptive background mixture models for real-time tracking. in Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). IEEE Comput. Soc. Part Vol. 2, 1999.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. Coference on Computer Vision and Pattern Recognition (CVPR), 2005.

[3] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, Shai Avidan: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. CVPR (2) 2006: 1491-1498

[4] Boris Babenko, Piotr Dollar, Zhuowen Tu, Serge Belongie. Simultaneous Learning and Alignment: Multi-Instance and Multi-Pose Learning, Euro. Conference of Computer Vision, 2008.

[5] H. Grabner, T.T. Nguyen, B. Gruber, H. Bischof. On-line boosting-based car detection from arial images. ISPRS Journal of Photogrammetry & Remote Sencing, 2007,63(3),pp.382-396.

[6] Yasemin Altun, Ioannis Tsochantaridis and Thomas Hofmann. Hidden Markov Support Vector Machines. International Conference on Machine Learning (ICML), 2003.

[7] Thorsten Joachims. SVMHMM tool package available at http://www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html

[8] A. Ellis, A. Shahrokni, J. Ferryman. Overall Evaluation of the PETS 2009 Results. Performance Evaluation of Tracking and Surveillance Online Proceedings, pp: 117-123, 2009.

PKU-IDM @ TRECVid 2010: Copy Detection with Visual-Audio Feature Fusion and Sequential Pyramid Matching*

Yuanning Li, Luntian Mou, Menglin Jiang, Chi Su, Xiaoyu Fang, Mengren Qian,
Yonghong Tian[+], Yaowei Wang, Tiejun Huang, Wen Gao
National Engineering Laboratory for Video Technology, Peking University
[+] Corresponding author: Phn: +86-10-62758116, E-mail: yhTian@pku.edu.cn

Abstract

Content-based copy detection (CBCD) over large corpus with complex transformations is important but challenging for video content analysis. To accomplish the TRECVid 2010 CBCD task, we've proposed a copy detection approach which exploits complementary visual/audio features and sequential pyramid matching (SPM). Several independent detectors first match visual key frames or audio clips using individual features, and then aggregate the frame level results into video level results with SPM, which works by partitioning videos into increasingly finer segments and calculating video similarities at multiple granularities. Finally, detection results from basic detectors are fused and further filtered to generate the final result. We have submitted four runs (i.e., "PKU-IDM.m.balanced.kraken", "PKU-IDM.m.nofa.kraken", "PKU-IDM.m.balanced.perseus" and "PKU-IDM.m.nofa.perseus") and achieved excellent NDCR performance along with competitive F1 measures.

1. Introduction

Along with the exponential growth of digital videos and the development of video delivering techniques, content-based video copy detection has shown great value in many video applications such as copyright control, illegal content monitoring and so on. However, copy detection is pretty challenging due to the following factors. First, a copy video can be produced by different kinds of visual and/or audio transformations. However, one certain kind of feature is robust only to several kinds of modifications. Second, for frame-based methods without proper temporal voting mechanism, copies are not likely to be accurately detected or precisely located. Last but not least, compact feature representation and efficient index are required for a practical copy detection system.

Therefore, we propose a copy detection approach with multimodal feature fusion and sequential pyramid matching (SPM), which is shown in Figure 1. Complementary visual/audio features are employed to achieve the goal of total robustness to various transformations through later result fusion. And SPM is adopted to aggregate frame level results into video level results as well as aligning two sequences of a copy and its original reference video.

The remainder of this paper is organized as follows. Sec. 2 describes the proposed approach. Sec. 3 presents the experimental results. Sec. 4 concludes this paper.

2. Proposed approach

This section presents the modules of our copy detection approach, namely preprocessing, basic detectors, SPM as a component of each detector, and fusion & verification.

2.1. Preprocessing

During preprocessing, reference/query videos are first split into video and audio components. Then, visual key frames are obtained by uniform sampling at a rate of 3 frames per second. Audio frames are obtained by dividing the audio signal into segments of 60ms with a 40ms overlap between consecutive frames, and 4-second-long audio clips are constructed by every 198 audio frames with a 3.8 seconds overlap between adjacent clips. Visual key frames where intensity of each pixel is below a predefined threshold are dropped as black frames. Finally, additional preprocessing is dedicated to handle the Picture-in-Picture (PiP) and Flip transformations. Hough transform that detects two pairs of parallel lines is employed to

* This work is partially supported by grants from the Chinese National Natural Science Foundation under contract No. 90820003 and No. 60973055, and the CADAL project.

detect and locate the inserted foreground videos. For those queries with PiP transformation, our system will process the foreground, background and the original key frames respectively. Also those queries asserted as non-copies will be flipped and matched again to deal with potential flip transformation.

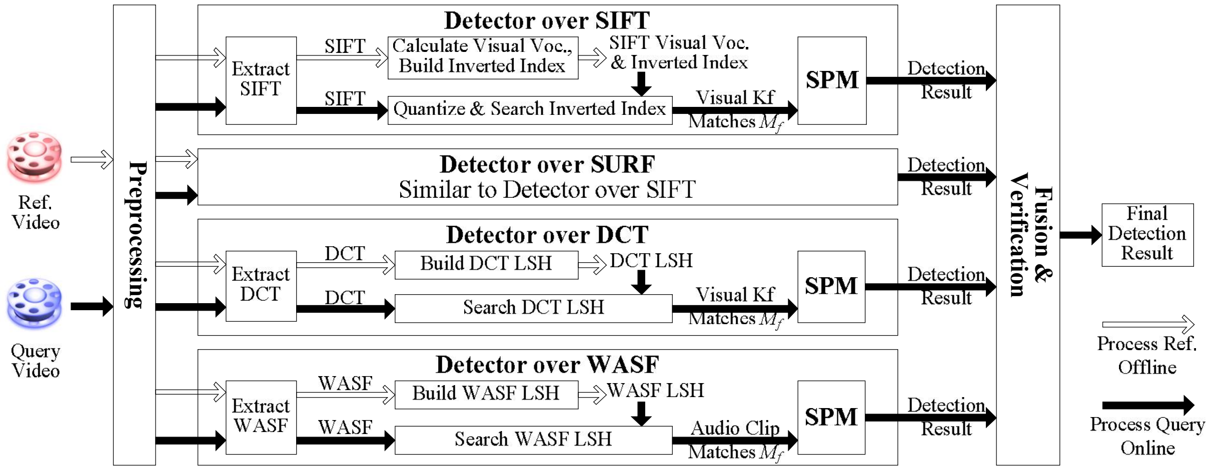

Figure 1. Overview of our video copy detection approach

2.2. Basic detectors

Four detectors are constructed respectively upon two local visual features, one global visual feature and an audio feature. Each detector is briefly described as follows, leaving SPM to be presented in the next section.

Detectors over local visual features: two similar detectors over local visual features employ the bag-of-words (BoW) framework [1] for SIFT [2] and SURF [3] respectively. Take the detector over SIFT feature as example. During offline process, it first extracts SIFT features from all the reference videos' key frames, here a local feature refinement proposed in [4] is utilized to keep the most stable features. After that, K-means algorithm (K=400) is implemented on a random subset (2M) of the features to calculate a visual vocabulary. Then all the reference features are quantized as visual words and stored in an inverted index. To further improve the performance of local feature matching, position, orientation and scale of SIFT features are also used so that only features belonging to the same visual word with similar position, orientation and scale are regarded as matches. In particular, the space of key frames is divided into 1×1, 2×2 and 4×4 cells and the position of each local feature is quantized into three integers ranging from 0 to 20. Orientation and scale of each local feature are also quantized into 8 and 2 bins respectively. Accordingly, such quantized information is integrated within the inverted index. During query process, SIFT BoW along with the additional position, orientation and scale information is obtained from each query key frame through the same feature extraction and quantization method. By searching the inverted index, reference key frames that have similar appearance and spatial layout can be found efficiently. Figure 2 illustrates the key frame retrieval process using the inverted index of SIFT visual words and spatial information.

Detector over global visual feature: inspired by [5], we propose a global image feature based on the relationship between the discrete cosine transform (DCT) coefficients of adjacent image blocks. It has been shown that the DCT feature is robust to simple transformations such as T3 (Pattern Insertion), T4 (Re-encoding) and T5 (Gamma Change). DCT also works well on several complex transformations such as T2 (Picture-in-Picture) with the help of preprocessing. In particular, a key frame is firstly normalized to 64×64 pixels and converted to YUV color space, keeping the Y channel only. Then the Y-channel image is divided into 64 blocks (numbered from 0 to 63) with the size of 8×8 pixels, and a 2-D DCT is applied over each block to obtain a coefficient matrix with the same size. After that, energies of the first four subbands of each block (c.f. Figure 3) are computed by summing up the absolute values of DCT coefficients belonging to each subband. Finally, a 256-bit DCT feature D_{256} can be obtained by computing the relative magnitudes of the energies:

$$d_{i,j} = \begin{cases} 1, if\ e_{i,j} \geq e_{i,(j+1)\%64} \\ 0, otherwise \end{cases} \quad 0 \leq i \leq 3, 0 \leq j \leq 63 \tag{1}$$

$$D_{256} =< d_{0,0},\cdots,d_{0,63},\cdots,d_{3,0},\cdots,d_{3,63} > \qquad (2)$$

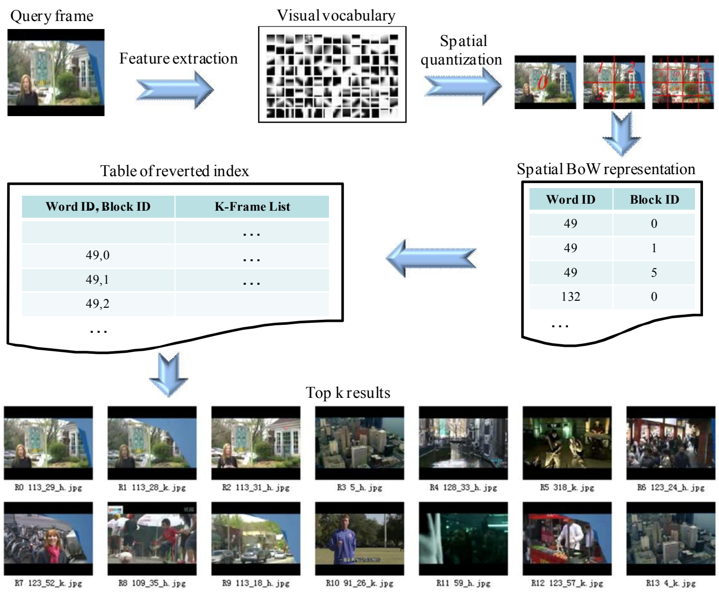

Figure 2. Key frame retrieval using the inverted index of SIFT visual words and spatial information

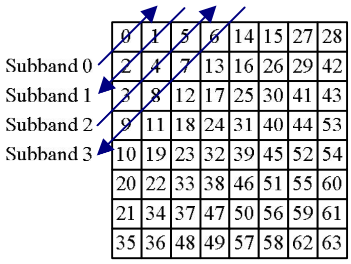

Figure 3. DCT subband allocation

where $e_{i,j}$ is the energy of the i-th band of the j-th image block. Hamming Distance is used as the distance metric. To speed up feature matching, all the reference videos' DCT features are indexed by locality sensitive hashing (LSH) [6].

Detector over audio feature: Our system utilizes the Weighted ASF (WASF) [7] as audio feature, which extends the MPEG-7 descriptor - Audio Spectrum Flatness (ASF) by introducing Human Auditory System (HAS) functions to weight audio data. This feature is proven to be robust to several audio transformations such as mp3 compression, noise addition, speed change and so on. In particular, a 14-D single WASF feature is first extracted from each 60ms audio frame. Then, each audio clip's 198 single WASF features are aggregated and reduced to a 126-D integrated WASF feature. Euclidean Distance is adopted to measure the dissimilarity between two 126-D integrated WASF features, and all the reference videos' integrated WASF features are stored in LSH for efficient feature matching.

Given a query video, a detector picks up the top K_1 (K_1=20) similar reference key frames (audio clips) for each query key frame (audio clip), resulting in a collection M_f which contains a series of frame level matches m_f:

$$m_f =< t_q, r, t_r, s_f > \qquad (3)$$

Where t_q and t_r are timestamps of the query and reference key frames (audio clips), r identifies the reference video, and s_f is the similarity of the key frame (audio clip) pair. Since s_f computed through different features are not consistent, histogram equalization is applied in each detector to make these scores more evenly distributed and comparable. Distribution of s_f for each feature is learned on the training data set.

2.3. Sequential Pyramid Matching

Given the frame matches M_f, copies are detected through the following three steps. First, a 2-D Hough transform like [8] is conducted on M_f to vote in K_2 hypotheses $<r,\delta t>$ (K_2=10), where $\delta t = t_q - t_r$ specifies the temporal offset between queryand reference video. Second, for each hypothesis, the begin and end of copy are identified by picking up the first and last matches m_f in M_f that accord with this hypothesis. Finally, SPM is performed on each potential video match to calculate its similarity, getting:

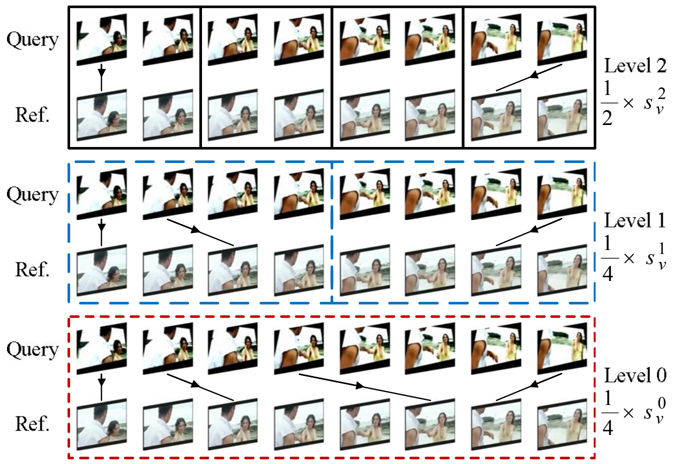

Figure 4. Toy example for a L=2 SPM

$$m_v = <q, t_{q,b}, t_{q,e}, r, t_{r,b}, t_{r,e}, s_v> \quad (4)$$

Which means the sequence [$t_{q,b}$, $t_{q,e}$] of query q is likely to be a copy from the sequence [$t_{r,b}$, $t_{r,e}$] of reference r with a similarity s_v. Only if s_v is above a threshold T_1, will m_v be accepted as a video match. When several m_v for query q exceed T_1, only the one with the highest s_v is reserved.

Inspired by spatial pyramid matching [9] which conducts pyramid match kernel [10] in 2-D image space, we adapt the kernel to 1-D video temporal space, resulting in the SPM which works by partitioning videos into increasingly finer segments and computing video similarities at each resolution. As shown in Figure 4, in level 0, video similarity s_v^0 is evaluated over the entire sequence. In level 1, sequences of key frames are divided into 2^1=2 segments, and only key frames within corresponding segments can be matched across two sequences. In level 2, sequences are divided into 2^2=4 segments, and so on (in practice we use four levels 0~3). The final similarity s_v is calculated by accumulating the weighted similarities from multiple levels. Since SPM only needs a set of frame level matches as input, it is suitable for all kinds of visual/audio features and is computationally efficient.

2.4. Fusion and verification

A result level fusion is utilized to fuse the detection results from different detectors. Besides, considering that the BoW representation inevitably causes decrease in feature's discriminability, a verification module is added to calculate the similarities of certain video matches again with original (vectorial) SIFT and SURF features. More specifically, if a query is asserted as a copy by any two detectors, i.e. there're two tuples like (5) satisfying (6), it is confirmed as a copy represented by (7):

$$\bar{m}_v = <q, \bar{t}_{q,b}, \bar{t}_{q,e}, r, \bar{t}_{r,b}, \bar{t}_{r,e}, \bar{s}_v>, \hat{m}_v = <q, \hat{t}_{q,b}, \hat{t}_{q,e}, r, \hat{t}_{r,b}, \hat{t}_{r,e}, \hat{s}_v> \quad (5)$$

$$[\bar{t}_{q,b}, \bar{t}_{q,e}] \cap [\hat{t}_{q,b}, \hat{t}_{q,e}] \neq \phi, [\bar{t}_{r,b}, \bar{t}_{r,e}] \cap [\hat{t}_{r,b}, \hat{t}_{r,e}] \neq \phi \quad (6)$$

$$<q, \max(\bar{t}_{q,b}, \hat{t}_{q,b}), \min(\bar{t}_{q,e}, \hat{t}_{q,e}), r, \max(\bar{t}_{r,b}, \hat{t}_{r,b}), \min(\bar{t}_{r,e}, \hat{t}_{r,e}), \max(\bar{s}_v, \hat{s}_v)> \quad (7)$$

Query asserted as a copy by only one detector is passed to the verification module. Only if the new calculated similarity for the video match is above a threshold T_2, will it be accepted as a copy.

3. Experimental results

We've submitted four runs, the first pair "balanced.perseus" & "nofa.perseus" follows the exact scenario presented above, while the second pair "balanced.kraken" & "nofa.kraken" omits the verification module and instead uses higher threshold T_1 in SPM to prevent false positives. Official evaluation results are summarized below.

NDCR: Normalized Detection Cost Rate measures the detection effectiveness of a CBCD system, i.e. how many queries it finds the reference video for or correctly tells users there is none to find. Compared with other participants, our system achieves excellent NDCR performance: for BALANCED profile, our system gets 39 top 1 among 56 "Actual NDCR" and 51 top 1 among 56 "Optimal NDCR"; for NOFA profile, it gets 52 top 1 among 56 "Actual NDCR" and 50 top 1 among 56 "Optimal NDCR". The detailed analysis on Actual NDCR for NOFA profile is shown in Table 1, tables on the other three NDCRs are not listed due to space limitation.

As to our NDCR for each transformation, results indicate that NDCRs for "simple" transformations are relatively better (lower) than those for "complex" transformations, which accords with people's intuitive sense. For instance, our NDCRs for video transformation T5 merged with audio transformations T1~T4 are all below 0.01 while the NDCRs for video transformation T10 merged with audio transformation T5~T7 are all above 0.17, as is shown in Table 1.

The NDCR measure also verifies our fusion strategy. Compared with BALANCED profile, submissions tuned for NOFA profile (using higher T_2 threshold) have fewer false positives at a cost of small decrease in true positives, and both profiles have achieved good NDCRs. Besides, the "balanced.perseus" & "nofa.perseus" pair with additional verification module achieves a little better NDCRs than the "balanced.kraken" & "nofa.kraken" pair.

Table 1. Actual NDCR performance for NOFA profile. The "V+A=M" column identifies Video Trans. ID, Audio Trans. ID and Video+Audio Trans. ID. The "perseus" and "kraken" columns correspond to the Act. NDCR of "PKU-IDM.m.nofa.perseus" and "PKU-IDM.m.nofa.kraken". The "best" column is the best NDCR obtained by all the other participants (excluding our results), and the "median" column indicates the median NDCR of all the participants (including our results). Note that the items in bold mean these are the best (lowest) NDCRs among all the participants.

V+A=M	perseus	kraken	best	median	V+A=M	perseus	kraken	best	median
1+1=T1	**0.046**	0.054	0.246	108.048	5+1=T29	**0.008**	0.046	0.046	535.411
1+2=T2	**0.046**	0.054	0.246	108.071	5+2=T30	**0.008**	0.046	0.038	535.657
1+3=T3	**0.046**	0.054	0.262	214.566	5+3=T31	**0.008**	0.046	0.054	535.634
1+4=T4	**0.046**	0.054	0.277	108.064	5+4=T32	**0.008**	0.046	0.054	535.611
1+5=T5	**0.123**	0.169	0.285	108.033	5+5=T33	**0.008**	0.062	0.054	321.537
1+6=T6	**0.138**	0.162	0.285	107.525	5+6=T34	**0.031**	0.092	0.054	321.222
1+7=T7	**0.108**	0.138	0.323	107.541	5+7=T35	**0.015**	0.069	0.069	321.222
2+1=T8	**0.023**	0.038	0.185	428.516	6+1=T36	**0.046**	0.054	0.100	535.403
2+2=T9	**0.023**	0.038	0.185	321.576	6+2=T37	**0.046**	0.054	0.092	535.657
2+3=T10	**0.023**	0.038	0.200	321.576	6+3=T38	**0.046**	0.054	0.108	535.634
2+4=T11	**0.023**	0.038	0.215	321.576	6+4=T39	**0.046**	0.054	0.123	535.611
2+5=T12	**0.062**	0.100	0.223	108.071	6+5=T40	**0.100**	0.200	0.123	214.851
2+6=T13	**0.046**	0.092	0.223	107.641	6+6=T41	0.123	0.185	**0.100**	214.512
2+7=T14	**0.062**	0.108	0.254	214.666	6+7=T42	0.115	0.185	**0.077**	214.489
3+1=T15	**0.023**	0.038	0.069	428.516	8+1=T50	**0.046**	0.054	0.138	321.737
3+2=T16	**0.023**	0.038	0.062	535.411	8+2=T51	**0.046**	0.054	0.131	535.411
3+3=T17	**0.023**	0.038	0.077	535.411	8+3=T52	**0.046**	0.054	0.146	535.411
3+4=T18	**0.023**	0.038	0.085	535.411	8+4=T53	**0.046**	0.054	0.162	321.737
3+5=T19	**0.031**	0.069	0.085	321.507	8+5=T54	**0.146**	0.169	0.169	321.514
3+6=T20	**0.031**	0.077	0.085	214.274	8+6=T55	**0.115**	0.138	0.162	215.089
3+7=T21	**0.031**	0.069	0.100	214.381	8+7=T56	**0.138**	0.162	0.185	215.02
4+1=T22	**0.054**	0.069	0.062	428.686	10+1=T64	**0.054**	**0.054**	0.123	428.516
4+2=T23	**0.054**	0.069	0.054	535.411	10+2=T65	**0.054**	**0.054**	0.123	535.411
4+3=T24	**0.054**	0.069	0.077	535.411	10+3=T66	**0.054**	**0.054**	0.138	322.168
4+4=T25	**0.054**	0.069	0.077	535.411	10+4=T67	**0.054**	**0.054**	0.154	322.176
4+5=T26	**0.077**	0.215	0.077	214.281	10+5=T68	0.192	0.215	**0.162**	108.048

4+6=T27	**0.085**	0.200	0.085	214.312	10+6=T69	0.185	0.223	**0.154**	214.697
4+7=T28	**0.062**	0.177	0.092	108.056	10+7=T70	**0.177**	0.200	0.185	108.018

Mean F1: F1 measures the accuracy of localization for true positives, i.e. when a copy is detected, how accurately the system locates the copy video in the reference data set. Our system achieves competitive F1 performance. For both BALANCED and NOFA profiles and all the transformations, our F1 measures are all around 0.9 with a few percent of deviation. Besides, our F1 measures for different transformations are at the same level even though the NDCRs vary. This demonstrates that once the correct reference video is found, our SPM strategy generally locates the copy position precisely.

Mean Processing Time: Processing Time measures the efficiency of a CBCD system, i.e. how much elapsed time is required to process a query. When using all the detectors and strategies discussed above, our system requires comparatively long processing time. However, it is worth to mention that our prototype system did not use any parallel programming techniques in the competition. In fact, currently, **processing time has decreased at least by an order of magnitude only by optimization with multi-threading and multi-processing** (c.f. Figure 5). Besides, our system is configurable. With fewer detectors used, it could obtain a slightly less excellent result with a small fraction of current processing time.

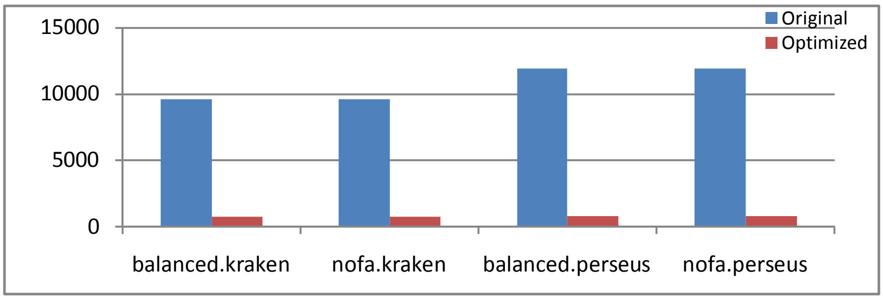

Figure 5. Mean Processing Time over original and optimized system

4. Conclusion

Official evaluation results show that our system outperforms other systems at most transformations in terms of NDCR and F1. It demonstrates the effectiveness of the adopted strategies: multi-feature extraction, multi-granularity sequence matching and fusion at the result level. Although our system is effective, endeavors will be devoted to the improvements on efficiency by parallelizing the algorithms and optimizing the implementation.

References

[1] J. Sivic, and A. Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos”, *ICCV’03*, Nice, France, pp. 1470-1477, October 13-16, 2003.

[2] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, *IJCV’04*, Vol. 60, No. 2, pp. 91-110, 2004.

[3] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded Up Robust Features”, *ECCV’06*, Vol. 3951, pp. 404-417, May 2006.

[4] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian, "Building Contextual Visual Vocabulary for Large-scale Image Applications", *ACM MM'10*, pp. 501-510, October 2010.

[5] C. Lin, and S. Chang, “A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation”, *IEEE TCSVT*, Vol. 11, No. 2, pp. 153-168, February 2001.

[6] A. Gionis, P. Indyk, and R. Motwani, “Similarity Search in High Dimensions via Hashing”, *VLDB’99*, Edinburgh, Scotland, pp. 518-529, 1999.

[7] J. Chen, and T. Huang, “A Robust Feature Extraction Algorithm for Audio Fingerprinting”, *PCM’08*, Tainan, Taiwan, pp. 887-890, December 9-13, 2008.

[8] Y. Liu, W. Zhao, C. Ngo, C. Xu, and H. Lu, “Coherent Bag-of Audio Words Model For Efficient Large-Scale Video Copy Detection”, *ACM CIVR’10*, Xi’an, China, pp. 89-96, July 5-7, 2010.

[9] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories”, *CVPR’06*, Vol. 2, New York, NY, USA, pp. 2169-2178, June 17-22, 2006.

[10] K. Grauman, and T. Darrell, “The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features”, *IEEE ICCV’05*, Beijing, China, pp. 1458-1465, October 17-21, 2005.