THU and ICRC at TRECVID 2008

Yingyu Liang, Xiaobing Liu, Zhikun Wang, Jianmin Li
Binbin Cao, Zhichao Cao, Zhenlong Dai, Zhishan Guo, Wen Li, Leigang Luo, Zhaoshi Meng, Yinfeng Qin, Shi Qiu, Aibo Tian, Dong Wang, Qiuping Wang, Chenguang Zhu
Xiaolin Hu, Jinhui Yuan, Peijiang Yuan, Bo Zhang
Intelligent multimedia group,
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Computer Science and Technology, Tsinghua University, Beijing, P. R. China

Shi Chen, JianGuo Li, Tao Wang, Yimin Zhang
Scalable Statistical Computing Group in Application Research Lab, MTL
Intel China Research Center, Beijing, P. R. China

Abstract

High level feature extraction

ID	MAP	Training set	Testing set	Brief description
run1	0.116	LIG & CAS	1frame/shot & 3frame/shot	baseline+keypoint
run2	0.123	LIG & CAS	3frame/shot	baseline
run3	0.057	CAS & Flickr & Peekaboom	1frame/shot & 3frame/shot	trecvid+flickr+peekaboom
run4	0.103	CAS & Flickr & Peekaboom	1frame/shot & 3frame/shot	borda fusion for run3 and run2
run5	0.080	CAS	1frame/shot	keypoint features
run6	0.090	CAS	1frame/shot	light version using only a few features

Search

ID	MAP	Brief description
run1	0.0274	example-based search result, and Flickr pictures are involved
run2	0.0305	combine all 3 modalities
run3	0.0277	only example-based search result, without extra pictures from Flickr
run6	0.0029	only text-based search result

Content-based copy detection pilot

ID	Brief description
run1	Baseline: SURF + 2 level descriptor-pairing + 2 level parameter estimation + post processing
run2	Baseline with different parameters in parameter estimation
run4	Baseline with different parameters in descriptor-pairing

Introduction

This year, Intelligent Multimedia Group in Department of Computer Science and Technology, Tsinghua University

and Scalable Statistical Computing Group in Application Research Lab, MTL, Intel China Research Center took part in TRECVID 2008 as a joint team and submitted the results for high level feature extraction, search, content-based copy detection pilot and rushes summarization. In this paper, the methods of the former 3 tasks are presented.

1. High Level Feature Extraction

Our video indexing system follows the same framework as our Video Diver system in TRECVID 2006 and 2007 [Wang 07][Tsinghua 06][Tsinghua 07]. Based on the framework, attempts on feature representation and dataset transfer have been made. (1) Several new local descriptors were experimented. Besides the color, texture, edge features and detector-based keypoint features used in our previous systems, several local descriptors including Color Coherence Vector, Edge histogram, Gabor Texture and Geometric-Blur were adopted, which shows considerable performance. (2) 28 types of features were evaluated on TRECVID 2007 and 2008 dataset. (3) A subset of images from Flickr [Flickr] and Peekaboom [Peekaboom] were used as a complementary to deal with the sparsity of the positive samples in the TRECVID dataset.

1.1 Annotation and keyframe extraction

Two versions of the annotation data on the TRECVID 2007 dataset were used in our systems, including (1) the collaborative annotation organized by the LIG (Laboratoire d'Informatique de Grenoble) and LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information) [LIG&LIRIS], and (2) the annotation provided by MCG-ICT-CAS [CAS 08].

Keyframes were extracted to represent the visual content for each shot. On the TRECVID 2007 dataset, the keyframes we used are provided by LIG-LIRIS and MCG-ICT-CAS. Since these two versions of the annotations are labeled on different keyframes, each version of keyframes and their annotation was treated as an independent dataset for training.

The testing keyframes were extracted by each participant, on the TRECVID 2008 dataset. The keyframes in our system were extracted by temporal equally sampling. Two versions of the keyframes sets were built with 1 keyframe per shot and 3 keyframes per shot separately.

1.2 Feature representation

To recognize objects, scenes, people and events, feature representation is crucial, for both human and artificial vision systems. From past experience, we know that various features can provide diverse discriminative information in video concept detection task. In this year, 35 types of features were adopted to represent the visual content of the videos. On TRECVID dataset, which is of large scale and without manual selection or simplification, the comparison results of these features are described in Section 1.7 to show how they performs and which of them may be good alternatives visual representation for real world application.

The features can be loosely categorized by feature descriptor type or by representation granularity. Thus, each feature is considered from two aspects: (1) which kind of feature (e.g. color, texture, or edge) to extract for representing the visual content, and (2) where to extract (e.g. on entire image, on local patches, or on the regions extracted by segmentation or fixed grid partition) and how to organize them to build a feature vector for each image.

1.2.1 Features descriptor type

Color. There are five types of color feature descriptors in our system, including Color Moment (CM, 9 dimensions),

Color Auto-Correlograms (CAC, 64 dimensions and 166 dimensions), Color Coherence Vector (CCV, 72 dimensions), Color Histogram Haar Correlogram (CHC, 314 dimensions), and Color Histogram on HSV space (HSV, 36 dimensions).

Texture. For texture features, our system used Gabor feature [Lee 96] (48 dimensions), GLCM feature (48 dimensions), CTN feature (48 dimensions), Haar Wavelet Moment Feature (HM, 10 dimensions) and Color Texture Moment feature on HSV space (CTMHSV, 48 dimensions), Moment Haar Band (MHB, 60 dimensions), Moment LUV Band (MLUVB, 36 dimensions).

Edge. In our previous experiments, edge features often perform better than color or texture features. Also, edge features show their power when serving as local descriptors in the famous Geometric-Blur [Berg 01] and Shape Context [Belongie 02]. We implemented Geometric-Blur [Berg 01], Shape Context [Belongie 02] (270 dimensions and 72 dimensions) and Canny Edge Histogram (32 dimensions and 64 dimensions). Additionally, we remained the Edge Coherence Vector (ECV, 32 dimensions) feature which borrows idea from the Color Coherence Vector (please refer to [Tsinghua 07] for details).

Gradient. The gradient feature is widely adopted in local descriptors (e.g. SIFT [Lowe 04]) and global feature (e.g. HOG) and is verified to get good results in image matching, object categorization, scene classification, pedestrian detection and video retrieval. Two types of popular gradient feature, SIFT [Lowe 04][Mikolajczyk05] and Histogram of Gradient (HOG), are used in our system with various spatial layout schemes and codebook sizes.

Markov Chain Stationary Feature. Markov Chain Stationary (MCS) Feature [Jianguo Li 2008] is a general framework to extend histogram representation by incorporating spatial co-occurrence information. In our system, three histogram features were extended to their MCS versions as MCS Histogram of Gradient (MCSHOG, 632 dimensions), MCS Local Binary Pattern [Ojala02] (MCSLBP, 594 dimensions) and MCS Color Histogram on HSV space (MCSHSV, 432 dimensions).

1.2.2 Representation Granularity

Spatial layout partition schemes. To add spatial information to the basic features, using spatial layout partition on fixed grids is a simple but very helpful method with significant performance improvement, especially for scene detection. These kinds of methods are based on the assumptions that: (1) videos are of similar capturing customs and styles, (2) most of the view points are with the horizontal line, and (3) most of the target objects are captured in the center of the screen.

Several schemes were designed and applied in our systems, according to those previous mentioned assumptions to capture different types of spatial constrains. These different schemes can improve the discriminative ability of our basic global features. Some of the implementations of the spatial layout partition schemes are shown in the following figure.

Keypoint representation. Local representations have been verified in both computer vision and multimedia area. Three kinds of the local feature detectors were adopted: (1) edge detector which is for Geometric-Blur [Berg 01], (2) DoG detector which is for SIFT [Lowe 04], and (3) non-overlap 20x20-pixel grid patches. For grid patches, beside the popular SIFT descriptors, 3 traditional color, texture and edge features were applied as our local descriptors, including

Canny Edge Histogram (64 dimensions), Color Coherence Vector (CCV, 72 dimensions), Gabor feature [Lee 96] (48 dimensions). In summary, 6 types of local features had been used in the system (Geometric-Blur, DoG-SIFT, patch-SIFT, patch-CEH64, patch-CCV72, patch-Gabor48). Codebooks were built for these local features by K-Means. And then keypoints were quantized by the codebook to generate a histogram for the region or image.

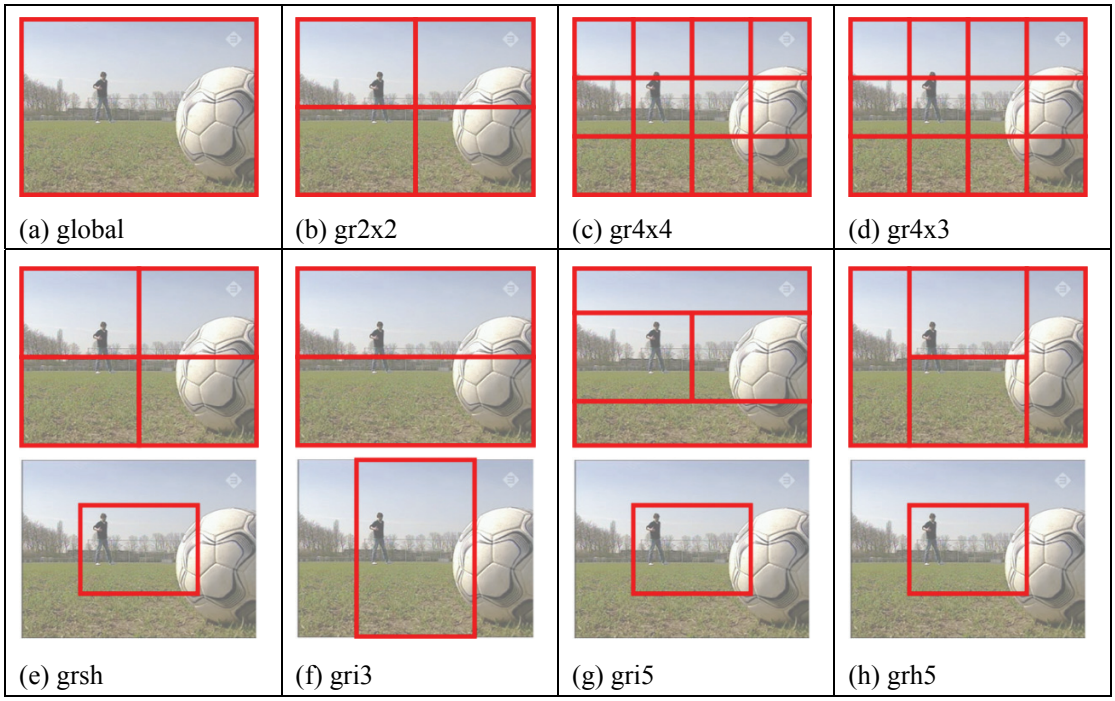

Fig 1 Spatial layout partition schemes

Feature and Spatial Covariant Kernel. Adding spatial information for the keypoint histogram by fixed spatial layout has been study in several resent works [Lazebnik 06][Liu07a]. We implement our Feature and Spatial Covariant (FESCO) kernel in [Liu07a] to suggest the combinations of codebook size and spatial layout partition. This approach augments is motivated to augment the holistic histogram representation with implicit spatial constrains. This solution is both accurate and fast. Partly adapted from the spatial pyramid match kernel (SPM) [Lazebnik06], this scheme achieves better match accuracy than SPM [Lazebnik06] and PM [Grauman06] (Please refer to [Liu07a] for details). In short, fine spatial layout partition is suggested to work with small size codebook. The three combinations of codebook size and spatial layout partition are (1) 36-bin codebook with 4x4 grid partition, (2) 144-bin codebook with 2x2 partition, and (3) 576 codebook with global partition.

1.3 Learning and fusion

In this year, we still followed the Relay Boost Method for learning and fusion as last two years. Additionally, cross validation strategy was applied on TRECVID 2007 dataset. And the models from different training set were combined together through a bagging strategy to predict on TRECVID 2008 testing dataset.

SVM modeling. SVM served as the basic classifier in our system. The χ^2 kernel was applied for the keypoint histogram features with the RBF kernel for the other features. In the training phase, the parameters of SVM (C, i.e. complexity; γ, i.e. the coefficient in RBF and χ^2 kernel) are determined in a coarse-to-fine grid-search manner by the cross validation results.

Video Diver Framework. Since the sparsity of the positive samples in video retrieval related topics, strategies for

countering imbalanced data are required. The Relay Boost Algorithm was implemented to as we did in previous works. Given a training dataset with its annotation, a rankboost model, which has 5 SVM models as its week classifiers, was trained for each feature in our systems. Please refer to [Wang 07][Tsinghua 06][Tsinghua 07] for details.

Cross validation and bagging. We adopted the cross validation strategy in the training phase. There are two advantages: (1) the models can be evaluated to know how they perform on this dataset, and (2) splitting the training dataset can be considered as using bagging strategy to gain robustness. In our system, the shots of TRECVID 2007 dataset were split by the order of the video-ID to make a 3-fold cross validation to train models for each feature. Additional SVM models were also trained on the whole TRECVID 2007. So that for each feature, 4 training set were used, including three sub-dataset generated by cross validation containing 2/3 of the TRECVID 2007 keyframes, and a full-dataset of TRECVID 2007. Thus, each feature had 20 SVM models.

Fusion. All the SVM models were fused by sum their probability outputs on each keyframe. This simple sum score late fusion were applied in (1) 5 SVM models for each feature on each training set, and (2) models from 3 fold cross validation subset and the whole TRECVID 2007 dataset. From keyframes to shots, the shot score is the maximum of the scores from the keyframes in this shot.

1.4 Specific detector for face

Sparse feature based boosting face detector [Huang05], which is robust with high precision, was used to detect faces. To rank keyframes for the "Two-People" concept after face detection, simple manual rules were designed. (1) All the keyframes with two faces detected are listed on the top part of the result ranking list. (2) The keyframes with large size faces are given high rank score. (3) The keyframes with two faces which have similar size are given high rank score. (4) After that, the keyframes with only one detected face were ranked by the size of the face, and listed after the two-faces-detected keyframes.

1.5 Dataset Adaptation

We had made some attempts to add new training data from other corpus in this year's TRECVID for three reasons. First, the positive samples on TRECVID 2007 dataset are very sparse. For the 43616 keyframes provided by LIG-LIRIS, 17 of 20 concepts have less than 500 positive samples (less than 1.15% of the dataset), while 6 of all concepts have less than 150 positive samples (less than 0.35% of the dataset). Second, we want to build more robust models by introducing new data to achieve better generality, not only on TRECVID data, but also on other sort of images and videos. Third, we want to find a way to use the web scale images and raw labels on the internet.

Flickr Dataset. Flickr is a photo sharing website [Flickr]. About 4 million images, together with their tags, were crawled from Flickr website. And then the images, whose tags contain the concepts of the 20 concepts for evaluation in this year, were selected to build a small subset with about 20k images. We simply use the tags provided by web users as the annotations for each image. Actually, this kind of annotation has a lot of noises (e.g. users will tag "airplane" on the sky photos captured from the window of airplane, although no airplane can be found from the images).

Peekaboom Dataset. Peekaboom is an online game to collect object level labeling for images [Peekaboom]. About a thousand images, which contain the evaluated concepts of this year, were selected in our experiments. The annotations on Peekaboom dataset are of high quality. But the problem is that the size of the dataset is too small to

cover the variance within class.

Classification. We implemented the multi-label boosting algorithm for classification [Wang 07], which can provide efficient multi-label prediction by sharing weak classifiers among different classes. We trained models for TRECVID, Flickr and Peekaboom separately. And the final result is the linear combination of the output of these three models. In our experiments, the models trained on Flickr and Peekaboom cannot help a lot. Maybe the reason is that images from other sources are too different from the TRECVID dataset.

1.6 Results

Feature representation is essential for video concept detection. In this year, 28 types of features were evaluated on TRECVID 2007 and 2008 dataset to figure out which of them are more discriminative. These features were trained on TRECVID 2007 dataset using the annotations from MCG-ICT-CAS [CAS 08]. 3-fold cross validation has been used to provide an evaluation result on TRECVID 2007 dataset.

Table 1 shows the Mean Average Precision (MAP) of the 28 types of features ranked by MAP on the TRECVID 2008 dataset. The feature with best MAP on TRECVID2008 is 32-bin Canny Edge Histogram with grh5 layout. Canny Edge Histogram, Edge Coherence Vector, Patch CEH, Shape Context and Markov-Chain Stationary LBP are top features in our experiments. In average, local features perform better than the global or grid layout color, edge or texture features. It is an interesting result that the traditional features (CCV, CEH and Gabor) perform equal to or better than SIFT as the descriptor for patch.

Table 1 MAP of features

Name	Description	Dim.	MAP@07	MAP@08
grh5_CEH32	Canny Edge Histogram with grh5 layout	160	0.0800	0.0302
grsh_CEH64	Canny Edge Histogram with grsh layout	320	0.0849	0.0300
gri5_ECV32	Edge Coherence Vector with gri5 layout	160	0.0816	0.0291
G_ShapeContext270	Shape Context	270	0.0678	0.0248
G_MCSLBP594	Markov Chain Stationary Local Binary Pattern	594	0.0561	0.0243
Gr4x4_PatchCEH_q36	Canny Edgy Histogram on Patches	576	0.0773	0.0241
gri5_CTN48	Grey Level Co-occurrence Matrix (energy entropy contrast homogeneity) with gri5 layout	240	0.0671	0.0241
G_MCSHOG632	Markov Chain Stationary Histogram of Gradient	632	0.0621	0.0240
Gr4x4_SIFT_q36	SIFT on Patches	576	0.0768	0.0223
g_MCSHSV432	Markov Chain Sationary Color Histogram on HSV space	432	0.0704	0.0217
Gr2x2_SIFT_q144	Dog-SIFT of 144 visual-words with 2x2 grid layout	576	0.0803	0.0209
Gr4x3_HM10	Haar Wavelet Moment Feature with gr4x3 layout	120	0.0577	0.0202
Gr4x4_PatchGabor_q36	Gabor filter on Patches with 4x4 grid layout	576	0.0695	0.0201
SPM5_ShapeContext72	Shape Context with 2-level spatial pyramid layout	360	0.061	0.0198
gri3_HSV36	Color histogram on HSV space with gri3 layout	108	0.0531	0.0196
Gr4x4_GBlur_q36	Geometric Blur with 4x4 grid layout	576	0.0659	0.0193
gri3_CCV72	Color Coherence Vector with gri3 layout	216	0.0628	0.0192
Gr4x3_CM9	Color Moment with 4x3 grid layout	108	0.0491	0.0189

grh5_Gabor48	Gabor filter with grh5 layout	240	0.0600	0.0187
gri3_GLCM48	Grey Level Co-occurrence Matrix (energy entropy contrast correlation) with gri3 layout	144	0.0595	0.0185
Gr4x4_PatchCCV_q36	Color Coherence Vector on Patches with 4x4 grid layout	576	0.0626	0.0182
gri3_CTMHSV48	Color Texture Moment Feature on HSV space with gri3 layout	144	0.0655	0.0181
G_MHB60	Moment Haar Band	60	0.0614	0.0179
gri3_MLUVB36	Moment Band on LUV space with gri3 layout	108	0.0618	0.0161
Gr4x4_PatchSIFT_q36	SIFT on Patches with 4x4 grid layout	576	0.0580	0.0152
G_CAC166	Color Auto-Correlograms	166	0.0520	0.0118
G_CAC64	Color Auto-Correlograms	64	0.0481	0.0113
G_HSV166	Color histogram on HSV space	166	0.0352	0.0108

Table 2 shows all the 6 submission runs.

- **Run1: All feature.** All features were used in Video Diver framework, with training on both LIG and CAS annotations and testing on both 1frame per shot (1f/shot) and 3 frames per shot (3f/shot) testing dataset on TRECVID 2008.
- **Run2: Baseline.** All non-keypoint features were used in Video Diver framework, with training on both LIG and CAS annotations and testing on 3f/shot testing dataset.
- **Run3: Dataset adaptation.** TRECVID, Flickr and Peekaboom were used as training dataset to build multi-label boosting models for dataset adaptation.
- **Run4: Borda fusion.** Borda fusion was used to combine Run3 and Run2.
- **Run5: Keypoint features.** All keypoint features were used in Video Diver framework, with training on CAS annotations and testing on 1f/shot testing dataset.
- **Run6: Light version.** Only 4 features were used in Video Diver framework, with training on CAS annotations and testing on 1f/shot testing dataset. The 4 features are grsh_CEH64, gri3_CCV72, grh5_Gabor48 and gr4x4_GBlur_q36.

Table 2 submission runs

ID	MAP	Training set	Testing set	Brief description
run1	0.116	LIG & CAS	1frame/shot & 3frame/shot	baseline+keypoint
run2	0.123	LIG & CAS	3frame/shot	baseline
run3	0.057	CAS & Flickr & Peekaboom	1frame/shot & 3frame/shot	trecvid+flickr+peekaboom
run4	0.103	CAS & Flickr & Peekaboom	1frame/shot & 3frame/shot	borda fusion for run3 and run2
run5	0.080	CAS	1frame/shot	keypoint features
run6	0.090	CAS	1frame/shot	light version using only a few features

The baseline run2 is our best submission run with MAP 0.123. The unexpected performance drop from run2 to run1 may be due to the low performance on the only 1 keyframe per shot testing dataset on TRECVID 2008. According to run3 and run4, using the dissimilar dataset of Flickr and Peekaboom with multi-label boosting cannot give help for baseline. For run6, only using 4 features has achieved about 75% MAP of that when using about 30 features, which shows a light alternative for efficiency-required tasks.

1.7 Conclusion

From the experiments in this year, we found that:

- Local representation is better than grid representation, when using the same descriptor.
- Various local descriptors perform well.
- Spatial Layout is of additional value.
- Dataset adaptation remains difficult, especially for the images from other sources.

2. Search

We took part in both automatic and interactive retrieval tasks this year. Our system framework was mostly inherited from previous work, but many new methods were adopted.

2.1 Automatic Search

2.1.1 Text-based Search

The text-based search system is exactly the same as previous system. Keywords in a query are first extracted before query expansion, and additional words are appended by WordNet. We use Lucene to index those translated speech transcripts and calculate the scores of each shots. A temporal spread method is adopted. In this method, the score of a certain shot will spread and affect its temporal neighbors. This method was proven to be useful in experiments on the training dataset. Linear combination was used for fusion of the results from three levels, while the weights of each level were derived from the training data.

2.1.2 Example-based Search

This year we are armed with a richer visual feature set. For each feature, SVM models are built in order to classify images corresponding to each topic. Experiments on Trecvid07 dataset are conducted to select features.

Because a much larger feature set is available, we have to choose appropriate features for a certain topic. Since the example images and videos are very sparse, although a lot of useful features have been prepared, we can only involve a few features for each topic. We introduced a measure for each feature on each topic: the ratio of average distance under this feature space among image and video examples (from this topic's description) and average distance among TRECVID 2007 video documents. The intuition is that, for a certain topic, feature with smaller ratio will describe the topic better. Besides this theoretic measure, we also consider a practical measure, by setting up an experiment to evaluate the performance of each feature in the training dataset. We believe it is most convincing to take both measures into account. Simply, we take the product of the above two measure as the final criterion of performance of features, for certain topic, and choose the features which rank top 4 on this criterion. In TRECVID 2007 search topics, the most frequently selected features were gr6x5_semangist_r2600, gri5_eccv32, grsh_ceh64, and gr4x4_patch20r20_ceh64_kmlocal_q36, they are outstanding for both measures in most cases.

Fusion of results from different features is also important. We test several fusion methods and finally chose linear combination due to its simplicity and consistence performance. The weight of each feature is obtained from experiments on training dataset.

Another work is motivated by the insufficiency of query examples. We seek help from Internet. This year we prepare a picture collection crawled from Flickr, where tags for each picture are available as clues. Then we use a two-step approach to extract pictures from the collection and use them as extra query examples: 1. Use query keywords to

search for some pictures and 2. Eliminate irrelevant pictures according to existing query examples. For the first step, we use the BM25 ranking function and treat the top-N pictures as candidates. According to previous experiments on Google Image Search, tags on Flickr are more reliable. We simply choose top-50 pictures to supplement query examples and achieve a 10% improvement upon using query examples only. (Since too many additional pictures could drown out provided examples, we assign weights to each query example and make the sum weight of provided examples equal to the additional ones.) However, the top-50 pictures have many noises. Possible reasons include ambiguity of keywords and personalized tags. So we need the second step to purify extra pictures and maintain diversity as well. We try to use clustering methods but preliminary experiments show negative results.

2.1.3 Concept-based Search

Two set of lexicons in this year are adopted by our system. The first one is the sets of concept annotations made publicly available as part of LSCOM. Concepts in these lexicons were chosen based on extensive analysis of video archive query logs and related to program categories, setting, people, objects, activities, events, and graphics. We filtered out concepts with less than 20 positive examples in the training set and get a number of 374 concepts left. Given the LSCOM concept annotation on a training set, we follow the state-of-the-art concept detection system to build the semantic concept indices. The second set of lexicons consists of concepts in high-level feature extraction task. These two set of lexicons are combined and duplication is reduced.

We find concepts related to a topic in two ways, namely text-concept mapping and image-concept mapping. We collect text description for the concepts and then built an index. The text-concept mapping is using a text retrieval method to calculate scores that present the relation between concept and topic. On the other hand, the image-concept mapping follows the algorithm introduced in [Li07]. The topic-concept mapping is a fusion of two ways mentioned above.

The topic-concept mapping result for each topic and the concept detection result for each shot are both kept as a concept vector. Retrieval in this modality is simply calculating the dot product of two concept vector.

2.1.4 Multi-modality fusion

The fusion methods are chosen empirically and experimentally. The 4 runs are as follows:

run1: use example-based search result, and Flickr pictures are involved.
run2: combine all 3 modalities.
run3: use only example-based search result, without extra pictures from Flickr.
run6: use only text-based search result.

2.1.5 Result

Using Flickr pictures

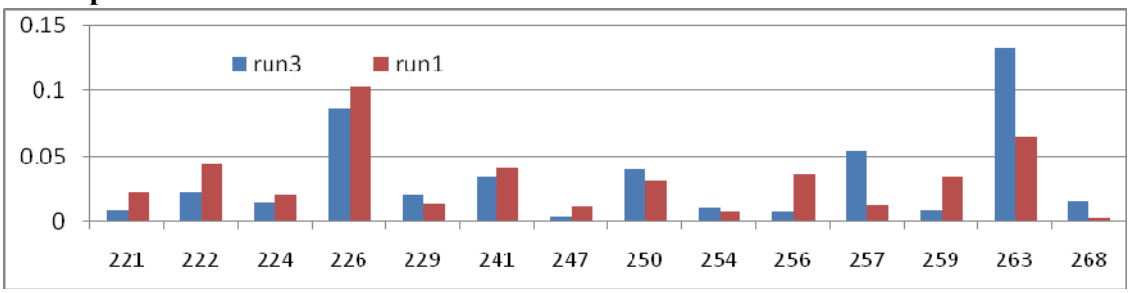

Fig 2. Comparison of run1 and run3 on some queries

Comparing run1 with run3, we find that additional training images from Flickr bring improvement in most topics. Some representative topics are shown in the above chart. According to our observation, there are many factors that influence the results.

Determinacy and Imageability of query keywords: We selected candidates by matching query keywords and tags. Determinacy and imageability of keywords are critical traits that decide the quality of mapping between topics and pictures. Queries like "bridge", "sitting, table" and "map" result in high-quality candidates. Whereas "Find shots of a road taken from a moving vehicle, looking to the side" doesn't reach many useful candidates.

Precision of tags: During experiments, we found that tags were not as reliable as we had expected. For example, some people label photos in a same album with the same tags. And another difficulty is that tags are not specific enough to capture the query need. Topic 0250 has two keywords: "airplane" and "exterior"; unfortunately "Exterior" is seldom used as tags, and many personal photos with the tag "airplane" are taken inside airplanes.

Although those extra pictures bring noises and didn't help much in the evaluation, preliminary experiments show promising results. Further researches on this track may include improving the mapping and reducing noises.

Using Concept Indices. In the evaluation, using concept indices causes relatively less improvement than it did before. For example, in topics 0241, 0247 and 0250, concepts "Food", "Animal" and "Airplane" respectively should have improved the performance of run4.

The main reason is that our LSCOM concept detectors are trained on news videos. They are quite different from the videos in Trecvid08. Therefore in some topics where related concepts can be found in LSCOM, like "boat/ship", "interview" and "computer", run5 doesn't take advantages of concept indices. Here transfer learning is a possible solution to adjust existing detectors to new data.

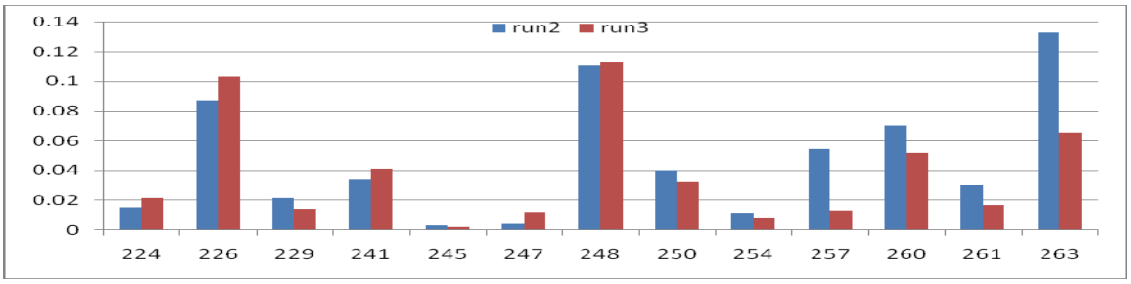

Fig 3. Comparison of run2 and run3 on some queries

3. Content-based copy detection pilot

3.1 System Overview

The formal definition of video copy detection can be described as follows. Note each video V as a frame sequence $\{V_i, 0 \le i < N\}$. Given a video database $\{R^j\}$ and a query video $Q = \{Q_i, 0 \le i < N\}$, the task of video copy detection is to determine for each video $R^j = \{R_i^j, 0 \le i < M^j\}$, whether there exist $0 \le u < v < N$ and $0 \le x < y < M^j$ satisfying $\{Q_i, u \le i < v\}$ is a copy of $\{R_i^j, x \le i < y\}$.

Each descriptor d is associated with a spatio-temporal location $d.loc = (x, y, t)^T$. When copy is performed, descriptor rd in the database and its corresponding one qd in the copied segment satisfy the following equation

$$qd.loc = (xScale, yScale, tScale)rd.loc + (xOffset, yOffset, tOffset).$$

We pair the descriptors in the query with the corresponding ones in the database and estimate the parameters (*xScale*, *yScale*, *tScale*) and(*xOffset*, *yOffsete*, *tOffset*). According to the pairs that satisfy the estimated parameters, we can judge whether there are copy segments in the query.

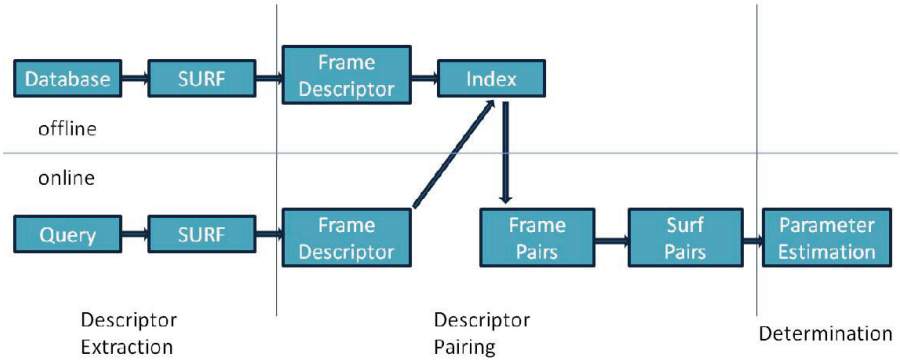

Fig4. System overview

As illustrated in Figure 4, the system consists of offline and online parts, dealing with the database and the query respectively. The system can be divided into three steps: descriptor extraction, descriptor pairing and determination.

3.1.1 Descriptor Extraction

The various kinds of descriptors used in video copy detection fall into two categories: global and local. Global descriptors show poor performance when used to detect copies with local transformations while local descriptors are robust to both global and local transformations.

The descriptor we employed is Speeded Up Robust Features (SURF) [Bay06]. The speed of SURF extraction is relatively fast, and SURF is scale-invariant and robust, which makes it the suitable descriptor for video copy detection. We select one keyframe every Pd=20 frames, and extract at most M=512 SURF descriptors from the keyframe.

3.1.2 Descriptor pairing

There are huge numbers of SURF descriptors, and the pairing is space and time consuming even if index is built on the descriptors. Therefore we use a two-level pairing approach which significantly reduces the data needed to be indexed and the time spent in search.

First we compress SURF descriptors in one frame into a frame descriptor and build ANN index [Arya98] on frame descriptors in the database. A frame's descriptor is a histogram of its SURF descriptors on a chosen set of D seed vectors, which are chosen points in the descriptor space. Details can be found in [Liu07b] where the seed vector histogram method has been used for generating video descriptors from frame descriptors.

Then similar frames are searched for each query keyframe. Thus we get frame pairs between the query and the database. SURF descriptors of each frame pair are matched to form descriptor pairs.

3.1.3 Determination

Given the scale parameters (*xScale*, *yScale*, *tScale*), we estimate the possible offset parameters between two videos. First we estimate the parameters Fp.offset for each frame pair Fp. Each descriptor pair of the frame pair can derive an offset and a score indicating their similarity. Fp.offset can be defined as the weighted average of descriptors' offset with their score as the weight. We eliminate pairs that have offset too far from Fp.offset and compute it again. Iterate several (for example, 3) times to generate a fairly accurate Fp.offset, and calculate the sum of the remaining pairs' score as Fp.score.

Then we estimate offset parameters between two videos in a similar approach, but use Fp.offset and Fp.score instead of the descriptor pairs' offset and score. The approach generates final offset parameters for the scale parameters and a score indicating their credibility. To further improve the result, algorithm in [Gengembre08] is used to estimate for each video in the database the probability of being the corresponding copy source for the query. If the similar frame for a query frame is from a video with higher probability, the frame pair's score Fp.score is adjusted to be higher; if from a video with lower probability, the score is adjusted to be lower.

After the offset parameters are estimated for each (*xScale*, *yScale*, *tScale*), we choose those ones with the highest score as the final parameters for the two videos. For evaluation, scores of different videos for the same query are normalized into [0, 100].

3.2 Experiments and Results

We perform 3 runs on the 10 transformations of TRECVID2008 copy detection benchmark. Run 1 is the baseline for comparison. Run 2 uses different parameters in the determination step to test the robustness of the determination. Run 3 returns 3 similar frames while Run 1 returns 1 similar frame. The NDCR [trecvid] and total online time is represented below.

3.3 Result Analysis

First we notice the high NDCR for transformation 2, which is picture-in-picture type-1, the transformation that scales the frames from an original video into a small picture and inserted into another background video. The reason why our system show poor perform for this transformation is that the frame descriptors generated in the pairing step are interfered severely by the background, and thus few corresponding frames in the database are returned for the query frames. Searching directly for similar SURF descriptors may solve this problem, but also introduces high complexity.

Comparison of Run 1 and Run 2 show little variation in NDCR, which indicates the determination step is robust to the variation in parameters. However, the time of Run 2 is higher than that of Run 1, for Run 2 performs more iteration in parameter estimation.

Comparison of Run 3 and Run 2 also show little change in NDCR. A detail inspect of the result indicates that the number of true similar frames returned show almost no increase while returning more neighbors in similarity search. More robust descriptor is need for an improvement in similar frame search.

In conclusion, the frame descriptor serves as the bottleneck for the system, but the two-level paring also reduce the complexity significantly. SURF and the parameter estimation approach are suitable for video copy detection for their effectiveness and efficiency.

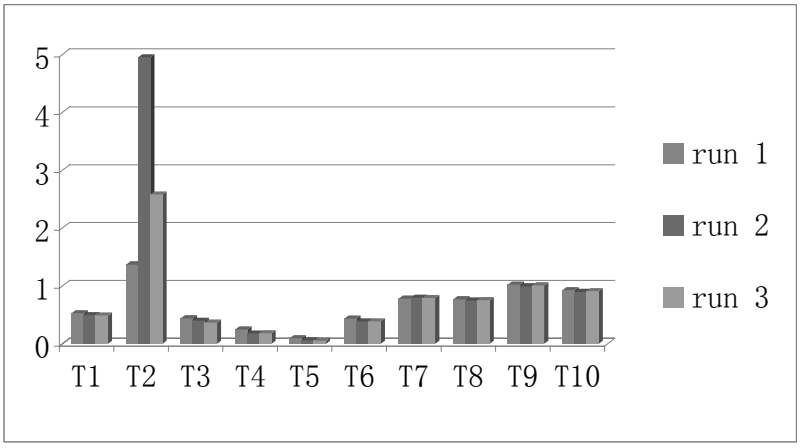

Figure 5. NDCR result

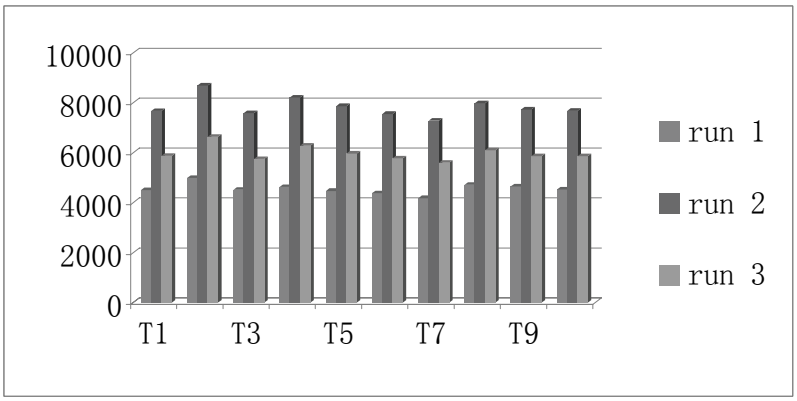

Figure 6. Total time (in second)

Acknowledgement

This work is supported by the National Natural Science Foundation of China under the grant No. 60621062 and 60605003, the National Key Foundation R&D Projects under the grant No. 2003CB317007, 2004CB318108 and 2007CB311003, and the Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList). The authors would also like to acknowledge NLIST for usage of THPCC.

Reference

[Wang 06] Dong Wang, Jianmin Li and Bo Zhang. Relay Boost Fusion for Learning Rare Concepts in Multimedia. CIVR 2006

[Wang07] D. Wang, X. Liu, L. Luo, J. Li, B. Zhang. Video Diver: Generic Video Indexing with Diverse Features. MIR workshop at ACM Multimedia, 2007.

[Wang 08] Dong Wang, Xiaobing Liu, Duanpeng Wang, Jianmin Li, and Bo Zhang. SemanGist: a Local Semantic Image Representation. PCM 2008

[Tsinghua06] Jie Cao, et al. Intelligent Multimedia Group of Tsinghua University at TRECVID 2006. In Proceedings of TRECVID 2006 workshop.

[Tsinghua 07] Jinhui Yuan, et al. THU and ICRC at TRECVID 2007. In Proceedings of TRECVID 2007 workshop.

[Liu07a] X. Liu, D. Wang, J. Li, and B. Zhang. The feature and spatial covariant kernel: Adding implicit spatial constraints to histogram. In Proceeding of CIVR 2007.

[Flickr] http://www.flickr.com/
[Peekaboom] Luis von Ahn, Ruoran Liu and Manuel Blum. Peekaboom: A Game for Locating Objects in Images. ACM Conference on Human Factors in Computing Systems, CHI 2006 [LIG&LIRIS] Stephane Ayache, Georges Quenot, Video Corpus Annotation Using Active Learning, ECIR 2008
[CAS 08] Sheng Tang, etal., TRECVID 2008 High-Level Feature Extraction By MCG-ICT-CAS, Proc. TRECVID 2008 Workshop, Gaithesburg, USA , Nov. 2008.
[Jianguo Li 08] Jianguo Li, Weixin Wu, Tao Wang, Yimin Zhang，One Step Beyond Histograms: Image Representation using Markov Stationary Features
[Lowe04] David G. Lowe: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2): 91-110 (2004).
[Mikolajczyk05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. IEEE Trans. PAMI,27(10):1615-1630, 2005.
[Belongie 02] S. Belongie, J. Malik and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. IEEE Trans. PAMI, 24(4), pps.509–522, 2002.
[Berg 01] A. C. Berg and J. Malik. Geometric Blur for Template Matching. In Proceeding of CVPR 2001.
[Ojala02] Timo Ojala, Matti Pietikainen, Topi Maenpaa, Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, IEEE Trans. PAMI, July 2002 (Vol. 24, No. 7) pp. 971-987
[Huang05] Chang HUANG, Haizhou AI, Yuan LI, Shihong LAO, Vector Boosting for Rotation Invariant Multi-View Face Detection, The IEEE International Conference on Computer Vision (ICCV-05), pp.446-453, Beijing, China, Oct 17-20, 2005.
[Torrabala04] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In Proc. CVPR, 2004.
[Lazebnik06] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proc. of CVPR 2006.
[Grauman06] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image features (version 2). Technical Report CSAIL-TR-2006-020, MIT, 2006.
[Lee96] Lee, T. S., “Image representation using 2D Gabor wavelets, IEEE Trans. PAMI, 18(10), pps. 959–971, 1996.
[Li07] Xirong Li, Dong Wang, Jianmin Li and Bo Zhang. Video Search in Concept Subspace: A Text-Like Paradigm. CIVR 2007: 603-610
[Bay06] H. Bay, T. Tuytelaars, L. Gool. Surf: Speeded up robust features. ECCV, May 2006.
[Liu07b] Lu Liu, Wei Lai, Xian-Sheng Hua, Shi-Qiang Yang. Video histogram: A novel video signature for efficient web video duplicate detection. International MultiMedia Modeling Conference, 2007.
[Arya98] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. J. ACM, 45:891–923, 1998.
[Gengembre08] N. Gengembre, S. Brrani. A probabilistic framework for fusing frame-based searches within a video copy detection system. ACM International Conference on Image and Video Retrieval, 2008.
[trecvid] http://www-nlpir.nist.gov/projects/trecvid/