Modeling Local Interest Points for Semantic Detection and Video Search at TRECVID 2006

Yu-Gang Jiang, Xiaoyong Wei, Chong-Wah Ngo, Hung-Khoon Tan, Wanlei Zhao, Xiao Wu
Department of Computer Science
City University of Hong Kong
Email: {yjiang,xiaoyong,cwngo,hktan,wzhao2,wuxiao}@cs.cityu.edu.hk

Abstract

Local interest points (LIPs) and their features have been shown to obtain surprisingly good results in object detection and recognition. Its effectiveness and scalability, however, have not been seriously addressed in large-scale multimedia database, for instance TRECVID benchmark. The goal of our works is to investigate the role and performance of LIPs, when coupling with multi-modality features, for high-level feature extraction and automatic video search.

In high-level feature extraction, we explore LIPs with both local description and spatial distribution for characterizing and sketching semantic concepts respectively. Two visual dictionaries, based upon universal visual keywords and concept-based visual keywords, are generated for experiments. The 39 concepts are learnt by SVM in vector space model with the support of both dictionaries. In addition, the distribution of LIPs is also exploited for detection with the multi-resolution and embedded Earth Mover's Distance settings. We submit six runs by incorporating the two properties of LIPs with other modalities including grid-based color moment and wavelet texture.

- CityU-HK1: average fusion of 4 SVM classifiers using universal visual keywords, distribution of LIPs, grid based color moment, and wavelet texture.
- CityU-HK2: average fusion of 3 SVM classifiers using universal visual keywords, grid based color moment, and wavelet texture.
- CityU-HK3: average fusion of 3 SVM classifiers using distribution of LIPs, grid based color moment, and wavelet texture.
- CityU-HK4: grid based *apriori* mining method.
- CityU-HK5: average fusion of 3 SVM classifiers using concept-based visual keywords, grid based color moment, and wavelet texture.
- CityU-HK6: baseline method by average fusion of 2 SVM classifiers using grid based color moment, and wavelet texture.

Results show that the LIP-based features could generate comparable results with traditional color/texture features. By incorporating the LIP-based features upon color moment and wavelet texture, an improvement of 51.4% is reported.

In automatic search, we study the performance of *query-by-example* (QBE) and mini-ontology (39 concepts) on top of baseline text search. In QBE, the properties of LIPs are utilized as one of features for retrieval. In mini-ontology, we measure the similarity of query terms to 39 concepts and adopt various heuristic settings (Detailed in Section 3.4) to test its significance for search. We submit six runs, for all queries, to show the advantage of search with the mini-ontology as semantic filters, and compare its performance to the classical text search.

- CityU-HK1: multimodal automatic run using text search, mini-ontology with setting 1.
- CityU-HK2: multimodal automatic run using text search, mini-ontology with setting 3.
- CityU-HK3: multimodal automatic run using text search, QBE, and mini-ontology with setting 1.
- CityU-HK4: multimodal automatic run using text search, mini-ontology with setting 4.
- CityU-HK5: multimodal automatic run using text search, mini-ontology with setting 3.
- CityU-HK6: required baseline run using ASR/MT transcripts only.

1 Introduction

We participated in two TRECVID tasks in 2006 – high-level feature extraction and automatic search. Our aim at TRECVID-2006 is to investigate the role of local invariant features, specifically the local interest point (LIP) and the related descriptors, in boosting the performance of the two tasks from the view of feature-level analysis.

Figure 1 illustrates the basic idea of our work. The middle column shows a group of LIPs overlaid on two keyframes with a concept *mountain*, and the right column sketches the basic outline of *mountain* in the keyframes with LIPs. Intuitively, both examples indicate the expressive and delineative power of LIPs respectively in locating key parts and describing the shape of a concept. In high-level feature extraction task, we explore the potential of LIPs in these two aspects: 1) generate LIPs as visual keywords to describe high-level features, 2) model the location distribution of LIPs to sketch high-level features.

In automatic search task, we utilize the two aforementioned LIP-based features in *query by example* (QBE) process. We also use a mini-ontology to measure similarity of query terms to 39 high-level concepts and adopt various heuristic settings to test its significance for search.

2 High-level Feature Extraction

2.1 Approach Overview

Our approach is based on our previous work in [1]. Figure 2 depicts the flow of framework which is composed of a group of classifiers based on various descriptors including the proposed local

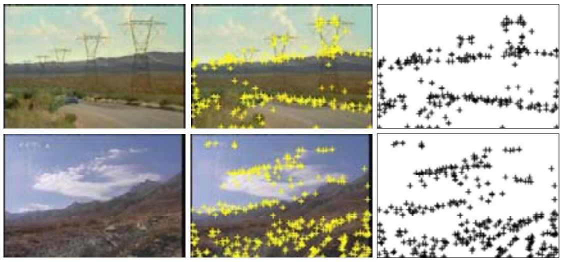

Figure 1: Keyframes with detected LIPs. Both the description and spatial location of the LIPs are utilized for high-level feature extraction and automatic search.

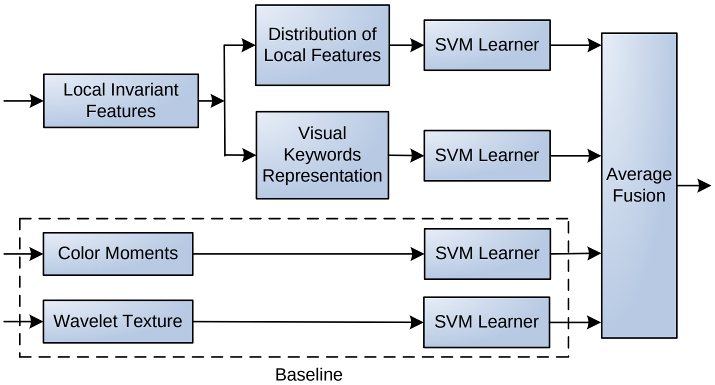

Figure 2: High-level feature extraction framework.

invariant features. The color moment and wavelet texture serve as the baseline to judge the improvement of local features formed by LIPs.

In this approach, LIPs are located by the Difference-of-Gaussian (DoG) detector [2] over different scales. The detector is scale invariant and can tolerate certain amount of affine transformation. Each LIP is characterized by a 36-dimensional PCA-SIFT feature descriptor. The descriptor has been demonstrated to be distinctive and robust to color, geometric and photometric changes [3]. Generally the number of LIPs in a keyframe can range from several hundreds to few thousands and thus prohibit the efficient matching of LIPs with PCA-SIFT across large amount of keyframes. We generate a visual dictionary as in [4] by offline quantization of LIPs. Subsequently each keyframe is described as a vector of visual keywords that facilitate direct keyframe comparison without point-to-point LIP matching. The local distribution of LIPs, on the other hand, is represented as shape-like features in the multi-resolution grids. The features are then embedded in a space where distance is evaluated with the e-EMD measure.

For each concept, an ensemble of classifiers as in Figure 2 is learnt. The extracted unimodal features are attached respectively to support vector machines (SVM) for discriminative classification in their own feature space. The margin output of SVM could be converted to posterior probability by Platt's method [5], and the probability outputs of various SVM learners are then re-ranked with average fusion. Since our aim is to investigate the role of LIPs from the feature-level point of view, we do not pay particular attention to the aspects of machine learning and multi-modality fusion. The framework we adopt is one of the commonly used platform for learning and fusion.

2.2 Generating Visual Keywords

2.2.1 Universal Visual Keywords

We generate a visual dictionary of LIPs based on [4]. We select approximately $1,500$ keyframes from TRECVID-2005 development set, with about 70% of them containing the 39 high-level concepts in LSCOM-lite lexicon. In total, there are about 850,000 LIPs extracted. Empirically we quantize these local points into 5,000 clusters, and each cluster represents a visual keyword. With this visual dictionary, the classical *tf-idf* is used to weight the importance of keywords. A keyframe is then represented as a vector of keywords, analogous to the traditional text-based vector space model.

2.2.2 Concept-based Visual Keywords

Different from universal visual keywords, we cluster LIPs of each high-level concept separately to generate concept-based visual keywords. Then the visual keywords of all concepts are concatenated to form the visual dictionary. Intuitively, this method could generate more representative visual keywords for each concept, due to the fact that clustering on larger dataset is easier affected by outliers, so as to make the generated clusters suboptimal.

However, since clustering are conducted on concept-level separately, one key problem is to remove the duplicate/similar visual keywords generated from different semantic concepts. This process is analogous to "stop word removal" in text retrieval, i.e. the keywords appear in most of the concepts are meaningless. We use entropy to evaluate the distinctiveness of each visual keywords. For visual keyword i, its *term frequency vector* is modeled as $T_i = \{t_{i1}, t_{i2}, t_{i3}, ..., t_{ic}\}$, in which c is the total number of concepts, and t_{ij} counts the term frequency of keyword i in concept j. Then the entropy of visual keyword i is defined as:

$$Entropy(T_i) = \frac{-1}{\log(c)} \sum_{j=1}^{c} \frac{t_{ij}}{\sum_{j=1}^{c} t_{ij}} \log \frac{t_{ij}}{\sum_{j=1}^{c} t_{ij}}. \quad (1)$$

Eqn 1 measures the distinctiveness of a visual keyword, i.e. smaller entropy value obtains for more distinctive visual keywords. Note that if keyword i only appears in one concept, its entropy value is 0. Finally, for each concept, the top k visual keywords with lower entropy value are retained. Truncating the visual keywords with higher entropy is reasonable even when the visual keywords frequently appear in one concept, since this kind of visual keywords are less discriminative for classification.

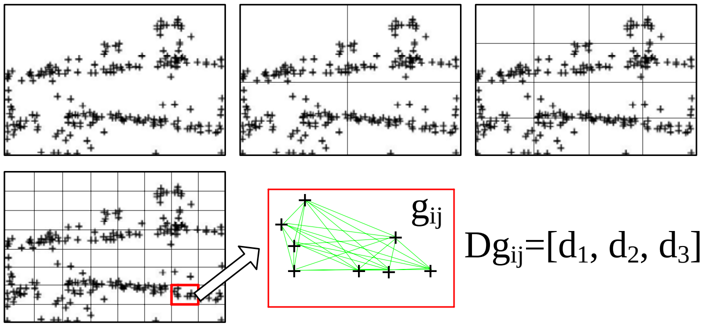

Figure 3: Modeling the distribution of local features. Grids with different resolution are imposed on the LIPs and a descriptor Dg_{ij} composed of three moments (d_1, d_2, d_3) are computed for each grid g_{ij}.

2.3 Modeling Location Distribution

We describe the distribution of LIPs with multi-resolution grid representation as illustrated in Figure 3. The size of grids varies at different resolutions and thus the granularity of shape information formed by LIP distribution changes according to the scale being considered. We compute the first three moments of grids to describe the shape-like information of LIPs across resolutions. Each grid is physically viewed as a point characterized by moments and weighted according to its level of resolution. With this representation, basically a keyframe is treated as a bag of grid points. The similarity between keyframes is based upon the matching of grid points within and across resolutions depending to their feature distance and transmitted weights that can be evaluated with Earth Mover's Distance (EMD). The complexity of EMD, nevertheless, is expensive and has an exponential worst case with the number of points.

For speed reason, we adopt embedded EMD which provides a way to map the weighted point sets from the metric space into the normed space [6] with low distortion.

The basic idea of the EMD embedding is as follows: Let two point sets $\mathbf{P}$ and $\mathbf{Q}$ with equal cardinality s, each in $\Re^k$ and $\mathbf{V} = \mathbf{P} \cup \mathbf{Q}$. Imposing grids on the space $\Re^k$ of side length 2^i, $-1 < i < \log(\Delta)$, where Δ is the diameter of $\mathbf{V}$. Let G_i be grid of side 2^i, in order to embed a point set $\mathbf{P}$, a vector $\mathbf{v}_i$ is constructed with one coordinate per cell, where each coordinate counts the number of points in the corresponding cell. Ultimately, by concatenating all $\mathbf{v}_i$ scaled by the side lengths, we can obtain the embedding of $\mathbf{P}$:

$$f(\mathbf{P}) = [\mathbf{v}_{-1}(\mathbf{P})/2, \mathbf{v}_0(\mathbf{P}), 2\mathbf{v}_1(\mathbf{P}) \ldots 2^i\mathbf{v}_i(\mathbf{P}) \ldots]. \tag{2}$$

In the embedded space, the normed distance between $f(\mathbf{P})$ and $f(\mathbf{Q})$ is an estimation of the exact EMD distance. The EMD embedding has a provable upper bound of distortion of $O(\log \Delta)$. Because the dimension of embedded vector is high, the locality sensitive hashing (LSH) technique is frequently used for nearest neighbor search [6, 7].

In our approach, each LIP is indexed with its spatial location location (x, y) in the keyframe. To keep the length of feature vector in an acceptable level, we only impose grids with four side

Table 1: Feature components of 6 runs for high-level feature extraction.

Run ID	*Components*
Run 1	CM+WT+uVK+LIP-D
Run 2	CM+WT+uVK
Run 3	CM+WT+LIP-D
Run 4	grid-based *apriori* mining
Run 5	CM+WT+cVK
Run 6	CM+WT

lengths, i.e., $\frac{1}{8}\Delta$, $\frac{1}{4}\Delta$, $\frac{1}{2}\Delta$ and Δ in this $2D$ space. Then, for each grid, the three moments of LIPs are computed to describe their distribution. The first moment counts the number of LIPs, while the second and third moments are the mean and variance of distances between all possible LIP pairs in the grid, as illustrated in Figure 3. Note that under the e-EMD setting, all grid points in the resolution i are grouped as a vector $\mathbf{v}_i(\mathbf{P})$ weighted by 2^i in the subspace. In our case for semantic concept retrieval, instead of using LSH for fast searching, we adopt machine learning approach which is proved to have better performance than direct searching in a metric space. The SVM is expected to learn the decision boundary that discriminates the embedded vectors of a semantic concept from others in the one-against-all strategy.

2.4 Experiments

We submit 6 runs for high-level feature extraction. Table 1 shows the feature components of the 6 runs. We use grid-based color moment (CM) and wavelet texture (WT) as baseline features. Both CM and WT have been shown as quite useful features in TRECVID-2005 corpus, which may partially because the time span of last year's news videos is only one month and there exist a lot of near-duplicate keyframes. In CM, three color moments (i.e., mean, standard deviation and skewness) are computed. Basically each keyframe is divided into 5×5 grids, and the color moments are computed for each grid in *Lab* color space. In WT, we use 3×3 grids and each grid is represented by the variances in 9 DB-4 wavelet sub-bands. While VK and LIP-D describe the LIPs (around corners and edges) that are robust to various transformations over different scale spaces, WT accounts for the statistical distribution of edge points in multi-resolution space.

Firstly, we study the improvement while incorporating single LIP-based feature on top of the baseline, i.e. universal/concept-based visual keywords (denoted as u-VK and c-VK respectively) and LIP distribution (LIP-D). Table 2 shows the performance of each run. Our baseline system (Run 6), by using average fusion of CM and WT, obtains a mean average precision of 0.070. While incorporating u-VK or c-VK upon the baseline (Run 2 and Run 5), improvements of 38.6% and 32.9% are obtained respectively. The results indicate that VK is indeed useful for most of the high-level concepts, e.g. *Mountain*, *Water*, *US flag*, *Car*, and *Charts*. This is due to the fact that these concepts mainly belong to objects or scenes, which can appear anywhere in the keyframes

Table 2: Experimental results of 6 runs in high-level feature extraction. The best results are given in bold.

High-Level Concepts	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
Sports	**0.312**	0.282	0.272	0.064	0.259	0.214
Weather	**0.305**	0.304	0.291	0.016	0.294	0.290
Office	**0.019**	0.017	0.011	0.005	0.013	0.007
Meeting	**0.192**	0.165	0.160	0.045	0.162	0.121
Desert	**0.027**	0.021	0.021	0.001	0.023	0.014
Mountain	**0.118**	0.110	0.055	0.001	0.096	0.044
Water	**0.069**	0.067	0.035	0.002	0.055	0.033
Corporate leader	0.000	0.000	0.000	0.000	0.000	**0.001**
Police/security	0.022	0.021	0.021	0.009	**0.023**	0.022
Military	0.058	**0.066**	0.042	0.002	0.063	0.051
Animal	0.010	0.009	0.010	0.001	**0.011**	0.010
Computer/TV screen	**0.140**	0.124	0.130	0.004	0.114	0.112
US-flag	0.093	0.095	0.036	0.020	**0.111**	0.025
Airplane	**0.034**	0.019	0.028	0.002	0.018	0.012
Car	**0.117**	0.120	0.073	0.006	0.100	0.066
Truck	**0.066**	0.054	0.062	0.004	0.046	0.040
People marching	**0.040**	0.033	0.033	0.000	0.034	0.021
Explosion/fire	**0.089**	0.057	0.060	0.004	0.052	0.028
Maps	0.224	0.202	0.215	0.152	**0.230**	0.193
Charts	**0.177**	0.167	0.138	0.014	0.156	0.095
Mean Average Precision	**0.106**	0.097	0.085	0.018	0.093	0.070

with different scales and viewpoints. VK, without any spatial information, is indeed good to model these concepts. c-VK, which is supposed to be better at the beginning, is a bit worse than u-VK on average. This mainly because we only select 200 out of 500 visual keywords from each concept according to the entropy value, which may miss some useful information. Even so, it is still better than u-VK for *US flag* and *Map*. Our recent experiments show that, if carefully engineered, c-VK is a bit better than u-VK, and both of them can get comparable results with the CM. LIP-D, on the other hand, gets an improvement of 21.4% (Run 3). LIP-D is useful for the concepts *Meeting*, *Mountain*, *Desert*, *Computer/TV screen* and *Explosion/fire*, because these concepts exist with somewhat uniform background or contour pattern (e.g. *Mountain* in Figure 1 that can be delineated with the location distribution of LIPs). On the contrary, the improvement of LIP-D is relatively smaller for concepts like *US flag*, *Military*, and *Car*, which can appear anywhere in the keyframes and LIP-D cannot effectively capture their LIP distribution under the presence of background clutter.

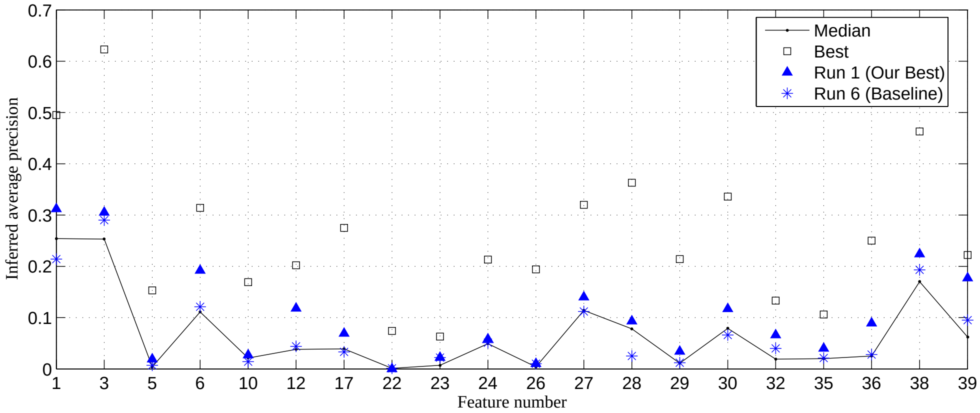

Figure 4: Performance comparison of our high-level feature extraction results with Best and Median of TRECVID-2006.

Because the two LIP-based features are designed on two different attributes of LIPs, i.e. the description and spatial location respectively, we also use average fusion to incorporate both of them upon the baseline (Run 1, results visualized in Fig. 4). Obvious improvement is noticed for most of the high-level concepts. Based on the experiments, we can conclude that the two LIP-based features, VK and LIP-D, indeed complement to each other. Moreover, the LIP-based features also complement with the traditional color and texture features, in terms of mean average precision, an improvement of 51.4% is reported upon the baseline of Run 6.

3 Automatic Video Search

3.1 Approach Overview

The overview of our automatic search system is visualized in Fig. 5. The system is composed of three portions: user interface, query analysis and search, and fusion. The core of our system, i.e. the query analysis and search portion, could be further divided into three modalities: 1) Query by text modality based on the speech transcripts; 2) Query by image example (QBE) modality; 3) Mini-ontology based re-ranking modality using the 39 high-level semantic concepts. The three modalities are detailed in the following sections.

3.2 Query by Text

The text only search is the required baseline run, using query sentence only against the ASR/MT transcript. The query by text framework we used is a very common one for text search. Firstly, the stop words in query sentences are removed. Then, instead of traditional *tf-idf* model, we use Okpai [8] to index the transcript. Lastly, the Lemur system [9] is employed to conduct text search.

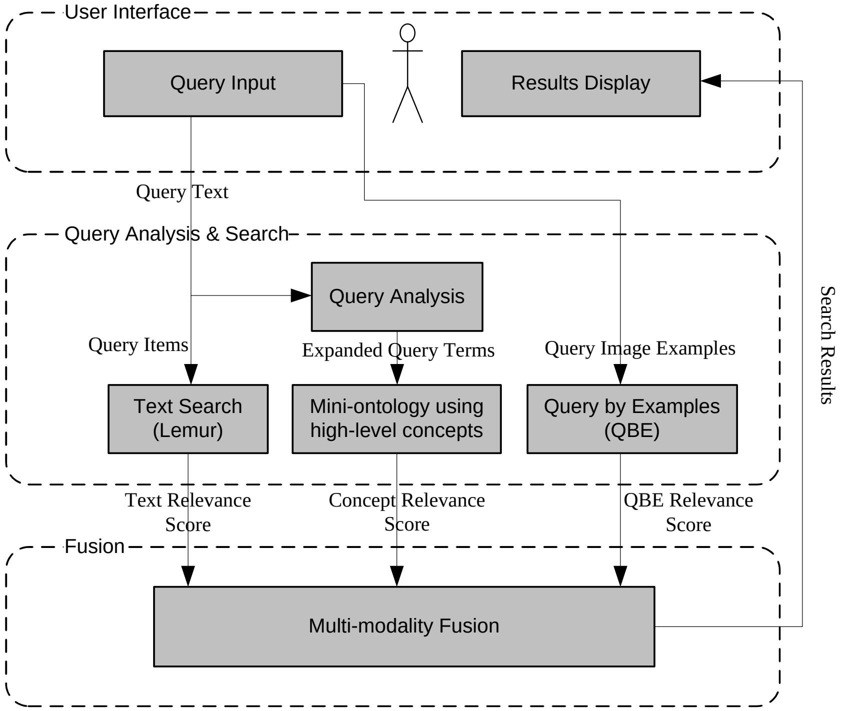

Figure 5: Automatic search system.

3.3 Query by Example

We use the same features in high-level feature extraction task, i.e. LIP-based features and grid-based color/texture. For QBE, it is hard to use supervised learning, since we only have limited number of positive query samples. Under this circumstance, the cosine similarity is employed. Actually there exists some sophisticated methods to generate pseudo-negative samples to solve the aforementioned problem [10]. It is not included in our study due to the implementation complexity and limited time. Finally, for one query with multiple query examples, we use the average similarity value to generate the final QBE results.

3.4 Mini-ontology

In this modality, the results of high-level feature extraction are used to support automatic search. We measure the similarity of query terms to 39 concepts and adopt various heuristic settings to test its significance for search.

A number of natural language processing (NLP) techniques are employed to analyze and expand the queries. Firstly, we adopt a dictionary-based method to extract name entities from the query sentences. Then, by searching in WordNet [11], the detected name entities are replaced with their shortest form (e.g. replace “People’s Republic of China” with “China”). Secondly, we employ a maximum entropy-based part-of-speech (POS) tagging method [12] to tag query sentences with replaced name entities. Then we simply discard the query items whose POS is not noun, and employ an automatic sense disambiguation algorithm “Lesk” [13] to get the

Table 3: Components and number of queries improved over baseline for automatic search runs.

Run ID	*Components*	*No. of Queries Improved*
Run 1	Text Search+Mini-ontology (Setting 1)	13
Run 2	Text Search+Mini-ontology (Setting 2)	15
Run 3	Text Search+Mini-ontology (Setting 1)+QBE	13
Run 4	Text Search+Mini-ontology (Setting 4)	12
Run 5	Text Search+Mini-ontology (Setting 3)	13
Run 6	Text Search	–

actual sense of each query term. Finally, by using the WordNet, the synonyms of the query terms are obtained to expand the query.

We propose to use a number of heuristic settings to test the significance of each high-level concept for search:

1. *Max query-concept similarity selection*: The similarity between all query terms and high-level concepts are calculated. Then the semantic concept with the highest similarity value is selected to re-rank the results. Note that under this setting, only one semantic concept is used for each query.

2. *Mean query-concept similarity selection*: Similar with setting 1, but for each high-level concept, we use the average similarity with all query terms to measure its significance. So, all of the concepts are used to re-rank the results, and the weight of each concept in the re-ranking process are determined according to the similarity values with query terms.

3. *Mean query-concept similarity selection with exclusion*: Same with setting 2, but the concepts with very low similarity values are used as negative samples, i.e. the keyframes are removed from the ranked list if they belong to those irrelevant (low similarity value) high-level concepts. For example, if we are search *soccer*, the concept *studio* maybe used to exclude the wrongly retrieved keyframes by text search and/or QBE.

4. *Mean query-concept similarity selection with concept reliability constraint*: Same with setting 2, except that the weights of concepts are determined not only by the similarity values, but also the confidences of the high-level concepts. The confidences of the high-level concepts are obtained by cross validation during training classifiers.

3.5 Experiments

We submit six runs, for all queries, to show the advantage of search with the mini-ontology as semantic filters, and compare its performance to the classical text search. For each run, the components and number of queries improved over baseline (Run 6) are given in Table 3.

Experimental results show that our mini-ontology could improve performance for more than half of the queries (12-15 out of 24), while QBE basically has no noticeable contribution. The

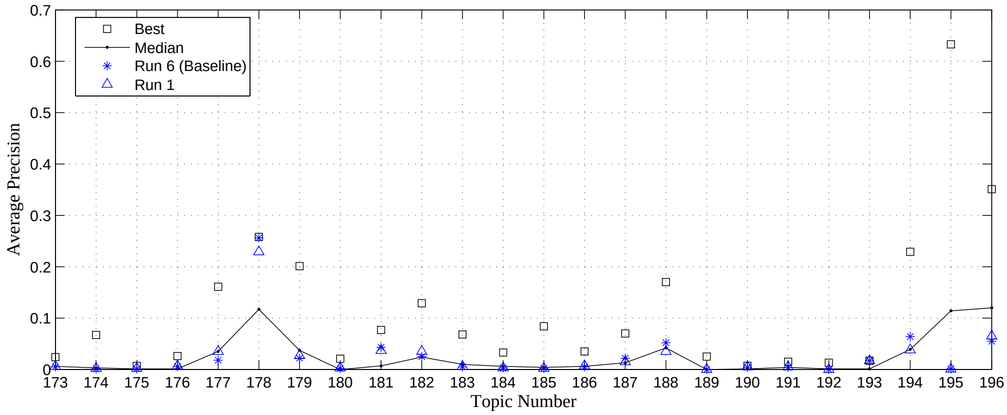

Figure 6: Performance comparison of our automatic search results with Best and Median of TRECVID-2006.

improvement due to the use of mini-ontology depends on the accurate association between terms and concepts. The performances of few queries degrade significantly, which results in lower mean average precision, partly because of incorrect association. This may due to the fact that the similarity obtained through WordNet may not be reasonable sometimes. For instance, *cars* and *airplanes* may have high similarity value in WordNet since both of them are vehicles, but return *airplanes* to searchers who want to find *cars* is definitely incorrect. In addition, while fusing different modalities, i.e. text search, QBE and mini-ontology, we simply use linear fusion with heuristical weights. This probably leads to unsatisfactory performance, since the three modalities are using different techniques and the outputs are in different scales. The linear fusion could easily make one of them dominating the others. Fig. 6 shows the detailed results of our Run 1 and Run 6. By fusing with mini-ontology modality, the results of query 177 (*Find shots of a daytime demonstration or protest with at least part of one building visible*) and 196 (*Find shots of scenes with snow*) are noticeably improved. On the contrary, the performances of few queries degrade obviously. This may because, except the WordNet reason as aforementioned, our mini-ontology only has 39 high-level concepts and does not support all terms (e.g. *Dick Cheney* and *Condoleeza Rice* in query 178 and 194 respectively). The performance, on average, delivers the usefulness of the mini-ontology to some extent when the query terms are properly associated with high-level features. We believe that by improving the similarity evaluation between queries and concepts, the results could be improved significantly.

4 Conclusions

We have conducted experiments for two tasks in TRECVID-2006. This year, our aim is to investigate the role and performance of LIP-based features for high-level feature extraction and automatic search. For high-level feature extraction, based on the results, we can conclude that the LIP-based features are indeed good, and they are complement to the traditional color/texture features. In automatic search, we study the performance of QBE and mini-ontology (39 concepts) on top of baseline text search. The properties of LIPs are utilized as one of the features for QBE. While QBE is not performing satisfactorily, we notice the potential of using mini-ontology for retrieval. The key here is how to unambiguously associate query terms and high-level features, while minimizing heuristics to generalize correct association.

Acknowledgment

The work described in this paper was fully supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 118905 and CityU 118906).

References

[1] Y.-G. Jiang, W.-L. Zhao, and C.-W. Ngo, "Exploring semantic concept using local invariant features," in *Asia-Pacific Wrokshop on Visual Information Processing*, 2006.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal on Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition*, 2004.

[4] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, 2003.

[5] J. C. Platt, "Probabilities for sv machines," *Advances in Large Margin Classifiers*, pp. 61–74, 2000.

[6] P. Indyk and N. Thaper, "Fast image retrieval via embeddings," in *3rd Int. Workshop on Statistical and Computational Theories of Vision*, 2003.

[7] K. Grauman and T. Darrell, "Efficient image matching with distribution of local invariant features," in *Computer Vision and Pattern Recognition*, 2005.

[8] S. Robertson and S. Walker, "Okapi/keenbow at trec-8," in *The Eighth Text REtrieval Conference (TREC-8)*, 2005, pp. 151–163.

[9] The Lemur Toolkit for Language Modeling and Information Retrieval, in *http://www.lemurproject.org/*.

[10] A. Natsev, M. Naphade, and J. Tesic, "Learning the semantics of multimedia queries and concepts from a small number of examples," in *ACM Multimedia Conference*, 2005, pp. 598–607.

[11] G. Miller, "Wordnet: An on-line lexical database," *International Journal of Lexicography*, 1995.

[12] A. Ratnaparkhi, "A simple introduction to maximum entropy models for natural language processing," in *Technical Report 97-08, Institute for Research in Cognitive Science, University of Pennsylvania*, 1997.

[13] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine code from an ice cream cone," in *Proceedings of the 5th annual international conference on Systems documentation, ACM Press*, 1986, pp. 24–26.