University of Central Florida at TRECVID 2006 High-Level Feature Extraction and Video Search

Jingen Liu, Yun Zhai, Arslan Basharat, Bilal Orhan
Saad M. Khan, Humera Noor, Phillip Berkowitz, Mubarak Shah

School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, Florida 32816, USA

ABSTRACT

In this paper, we describe our experiments in high-level features extraction and interactive topic search tasks of TRECVID 2006. We designed a unified high-level features extraction framework for the 39 high-level features. Various low-level visual features were extracted from the key-frames of the shots. Then the SVM classifiers were trained fore. The final classification results were produced by fusing and combining these classifiers. The experiment results show that the combined classifiers substantially improved the performance over the individual feature based classifier. In topic search task, we improved our PEGASUS news video retrieval system, which has friendly user interface, fast indexing and various relevance feedback mechanisms. Based on the evaluation results, this year's topic search results are better compared to last year.

1. INTRODUCTION

This year, the Computer Vision Lab team at University of Central Florida participated in the high-level features extraction and topic search tasks. We submitted six runs for high-level features extraction and two runs for interactive topic search. The returned evaluation results show that almost all our results from best run are above the median value and some of them hit the best.

1.1. High-Level Feature Extraction

In the highe-level feature extraction task, we designed a unified framework for all 39 high-level features. For each high-level feature we extracted various low-level features and trained SVM classifiers on them. The final classified results were produced by fusing classifiers trained on different low-level features. Our experimental results show that the fusion based approaches substantially improved the performance over the individual feature based approach. For every high-level feature our main steps are as follows:

- Extract low-level features.
- Train a classifier using color moments, color correlogram and edge histogram respectively.
- Combine the classifiers using training-based and non-training based approach.
- Test the fused classifiers on this year's testing data.

In the low-level feature extraction phase, we computed color and edge features. These features are able to capture most of the visual information in the videos, and have been successfully applied to high-level features extraction in previous TRECVID.[6,7] Support Vector Machine[1](SVM) with a Radial Basis Function (RBF) kernel, is choosen as classifier to learn the concept model for each high-level feature separately. When combining the classifiers separately trained from color and edge features, we used simple fusion like "average fusion" and "product fusion", but also we adopted the training-based fusion approach. In this approach, we learnt the conditional probability density function (pdf) of score given positive sample and score given negative sample. From these learnt pdfs we are able to estimate the posterior probability $p(positive|score)$.

This year we submitted the following six runs in the high-level features extraction task:

- **A_UCF.CE.PROD**: product fusion of two classifiers using color moments and edge histogram.
- **A_UCF.CE.PROB**: training-based fusion of two classifiers using color moments and edge histogram. The final decision is made using the product of probabilities.
- **A_UCF.CEC.PROD**: product fusion of three classifiers using color moments, color correlogram and edge histogram.
- **A_UCF.MIX**: re-rank the results of the run **A_UCF.CE.PROD** using simple concept dependency.
- **A_UCF.CM**: the output of the classifier using color moments.
- **A_UCF.EDGE**: the output of the classifier using edge histogram.

Based on the evaluation results, these runs which used various fusion approaches obtained good performance. The average precision (AP) of most high-level features is above the median performance, and the rest are close to median performance. Compared to run **A_UCF.CM**, all the fusion approaches are able to achieve 37% to 66% improvment in average precision. The best result is obtained by the fusion using three visual features (color moments, edge histogram and color correlogram), which is a little better than the fusion using color moment and color correlogram. The results show that the combination of the classifiers trained with individual visual feature is very useful to enhance the classification performance.

1.2. Interactive Topic Search

The Computer Vision lab at the University of Central Florida has also participated in the topic search task. Our experiments were performed on the PEGASUS system,[2] an online video retrieval system with a highly efficient user interface. The proposed system has five searching mechanisms: (1) searching by the automatic speech recognition (ASR) transcript, (2) searching by video or image examples, (3) searching by matching the visual statistics of the key-frames, like color moment, edge histogram and color correlogram, (4) searching by matching the region visual features, and (5) video shot browsing via Video on Demand. There are several features of the PEGASUS system: (a) ability to combine any number of the four searching mechanisms; (b) ability to evaluate the logical expressions of the search queries; (c) ability to perform the relevance feedback iterations.

We submitted two runs for the interactive topic search. Run **A_1_UCFVISION1** is purely based on the ASR information with word-histogram refinement, while **A_2_UCFVISION2** involves the interactive search using text and visual information. In each run, the temporal "K-nearest neighbor" method is used in the final step of the relevance feedback. In run **A_2_UCFVISION2**, we launched the search by example images or video keyframes for some topics which are hard to get initial results using query on text. As we expected, the run using both text and visual information to refine performs better than the one using text information only.

2. HIGH-LEVEL FEATURE EXTRACTION

In TRECVID 2006, we developed a unified high-level features extraction framwork. The entire work flow has been displayed in Figure 1. There are three main steps involved.

- Low-level feature extraction. We computed three simple visual features: color moments (CM), color correlogram (CC) and edge histogram (EDGE).
- Model training and selection. We adopted SVM as our classification method. First, we trained the individual SVM classifiers for each low-level feature. Then, non-training based model fusion and training-based model fusion were performed to combine the models learnt using single low-level feature.
- Apply the combined SVM classifiers to the TRECVID 2006 testing dataset.

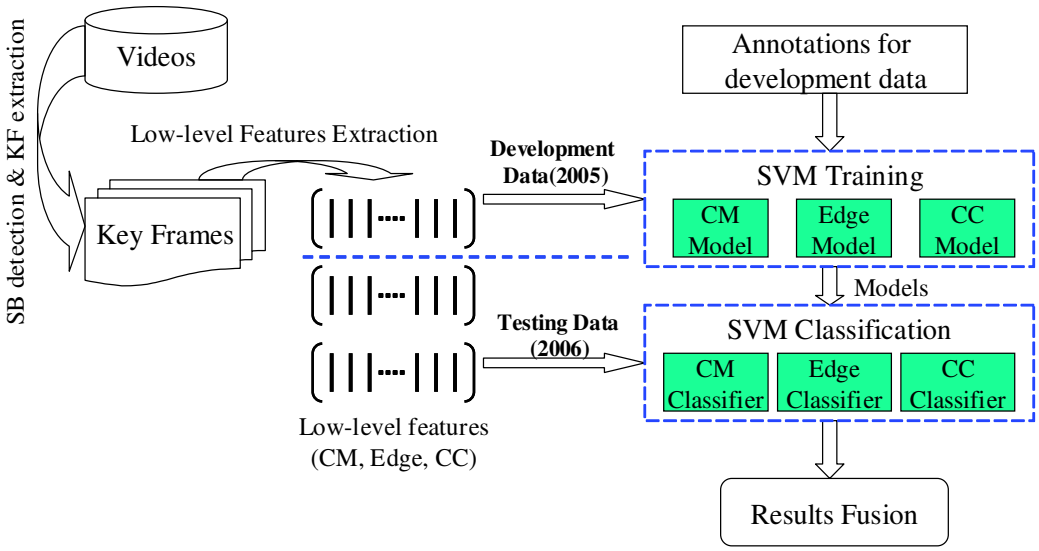


Figure 1. The framework of high-level features extraction for TRECVID 2006.

2.1. Low-level feature extraction

In order to efficiently describe a video, we created the color and edge visual feature descriptors for the representative key-frames of the video shots. These features have been widely used in video/image analysis and content-based image/video retrieval,[5, 9, 10] and have shown very impressive performance. In our experiments, we tried the following visual features: global color histogram, grid-based color histogram, grid-based color moments, color correlogram, grid-based edge histogram and Gabor wavelet. However, we noticed that some of these features are redundant, so we eventually limited our visual features in this system to the following three:

- Color Moments (CM). Color moment is a very useful color statistics of an image. We divided a keyframe into 4x4 grid, and then in HSV color space, the Mean, Standard Deviation and Skewness value of each sub-block in each color channel were computed. Therefore, we got a feature vector of 4x4x9 dimensions. Compared to the image representation whose color moments are computed on the whole image, this grid-based color moments descriptor also includes the spatial color information like "blue" color normally showing up on the top of a "sky" image.
- Color Correlogram[5] (CC). Both color histogram and color moments are not able to describe the spatial distribution of the color, as they are orderless information. Color correlogram is one color statistics which contains the local spatial distribution information of the colors. We computed the auto-correlogram for 64 colors in RGB color space with 5 radii depths, and then obtained a 320 dimensional image descriptor.
- Edge Histogram[9](EDGE). Like color feature, edge feature is also significant for the human being to recognize a scene or object. Based on Sobel filter, we created an edge histogram with eight bins in eight directions, and four bins in gradient magnitude for each block of a 5 by 5 grid. The dimension is 800 (8 by 4 by 25).

2.2. SVM-based Training and Model Selection

Since Vapnik proposed the Support Vector Machine (SVM) in [1], it has been widely used for pattern classification, and has been shown to achieve impressive results. Given a vector space, this method is able to find the decision surface which can separate the data points of one class from the other. It is a kernel-based method for classification, which means with proper kernel selected, the SVM can map the original feature vector into another feature space. For instance, with the nonlinear kernels, the SVM is able to map the original feature points to a higher dimensional feature space where an optimal separating hyperplane helps to separate the classes. Suppose

we have a dataset containing two classes which can be linearly separated. Then, the decision surface is the hyperplane which maximizes the margin between the two classes, like

$$\langle \mathbf{w}, \mathbf{x} \rangle - b = 0$$

where $\mathbf{x}$ is a data vector and the vector $\mathbf{w}$ and the constant b are learnt from the training set. Let $y_i \in \{1, -1\}$ be the classification label for input vector x_i. The optimal hyperplane can be obtained by minimizing the following formula,

$$\|\mathbf{w}\|^2 = (\mathbf{w} \cdot \mathbf{w})$$

under the the following constraints

$$y_i \left[\langle \mathbf{w}, \mathbf{x} \rangle + b \right] \geq 1 \tag{1}$$

In our high-level feature extraction experiments, each key-frame is represented with three low-level feature vectors, like color moments vector V_{cm}, color correlogram vector V_{cc} and edge histogram vector V_{edge}. Then, we got three feature spaces, say space S_{cm}, space S_{cc} and space S_{edge} consisting of feature vectors V_{cm}, V_{cc} and V_{edge} respectively. We used SVMs with nonlinear kernel to separate these three feature spaces.

In the training procedure, we have two phases. First phase, we trained three SVM models for each concepts in the three feature spaces. In this phase, the development dataset was divided into two parts with two thirds for training and one third for validation. Second phase, we fused the three models for each concept. We have two ways, fusion with training and fusion without training. For fusion with training we further divided the validation dataset into two equal parts, which are used to train and validate in the fusion phase.

When training the classifiers in the three visual feature space, SVMs with a Radial Basis Function(RBF) kernel are used. We noticed that the classification performance of SVMs varies with different parameters. In our experiments, we used "grid-search"[12] method to find out the proper parameter γ and C for RBF kernel. Since the dataset is very unbalanced between the number of positive and negative key-frames, we also tuned the "weight" parameter, which represents the relative significance of positive samples to negative samples. In our experiments, we set this parameter to be the ratio of negative to positive samples in the dataset.

2.3. Score Normalization and Fusion

The human recognition system is better adapted to recognize objects or scenes when there are both color and contrast (edges). However, the SVM models are separately trained from color and edge features. We noticed that models built using color and edge features had different performances for different high-level features. For instance classifiers that use color statistics achieve better performance for "sky" and "sports", while classifiers trained on edge features work better for "building" and "crowds". Therefore, it might be helpful to combine the output of individual classifiers.

Because the classifiers are independently trained in three feature sapces, there arises a problem of incomparable classifier output scores. Two well-known nomalization techniques used in our experiments are listed below,

- Max-Min: $S_{new} = \frac{S - min}{max - min}$
- Z-score: $S_{new} = \frac{S - mean}{standard\ deviation}$

where S_{new} denotes the normalized score and S is the classifier output score.

We can loosely group the fusion methods into training-based and non-training-based. In our experiments we used both of them. Suppose S_i be the classification score from the classifier C_i, $P(pos|S_i)$ be the posteriori probability of S_i being positive sample. The fusion approaches are as follows,

- Average Score: $S_{new} = \frac{\sum_{i=1}^{N} S_i}{N}$,
- Maximum/Minimum Score: $S_{new} = max(S_i)$ or $S_{new} = min(S_i), i = 1, 2, ..., N$,

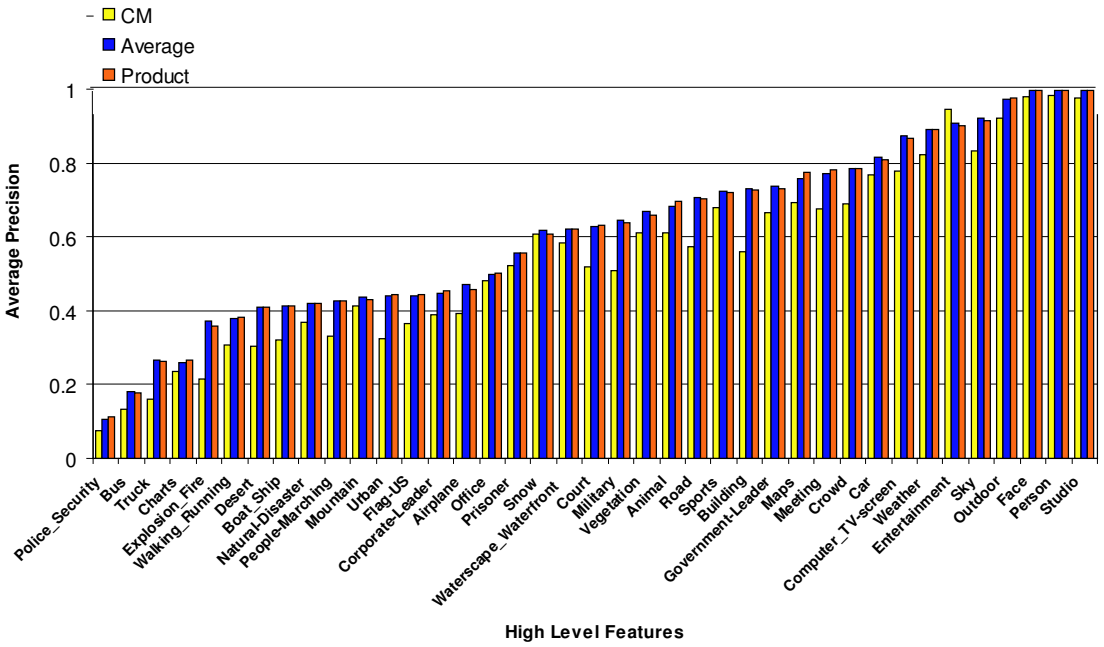


Figure 2. This figure compares the performance of three approaches: CM-model classification, Average Fusion classification and Product Fusion classification. These classifiers were tested on the validation dataset for all 39 high level features

- Product Score: $S_{new} = \prod_{i=1}^{N} S_i$,
- Product Probabilities: $S_{new} = \prod_{i=1}^{N} P(pos|S_i)$.

For the training-based fusion, the posteriori probability $P(pos|S)$ can be infered from the following formula according to Bayes Rule,

$$P(pos|S) = \frac{P(S|pos)P(pos)}{P(S|pos)P(pos) + P(S|neg)P(neg)} \tag{2}$$

The probability density function (pdf) $p(pos|S)$ also can be computed by substituting the probability with their corresponding *pdf*. These conditional *pdf* like $p(S|pos)$ and $p(S|neg)$ can be estimated from the training dataset. One straightforward way is to assume they are normal distribution. However, normal distribution might not be the actual distribution. Hence, we used Kernel Density Estimation (KDE)[11] technique to obtain the actual distribution. This method has been popularly used to generate the statistical modeling for a dataset, because it does not impose any prior assumption on the data. We can model $p(S|pos)$ from the positive training data by

$$p(S|pos) = \frac{1}{N} \sum_{i=1}^{N} K(S - S_i), \tag{3}$$

where K is a stochastic kernel. Generally, we adopt standard Gaussian function with zero mean and variance σ^2 as the kernel,

$$K(S - S_i) = \frac{1}{\sigma\sqrt{2\pi}} exp(-(S - S_i)^2/2\sigma_2), \tag{4}$$

2.4. Results and Discussion

In the model training and validation phase, we noticed that the model trained using CM (CM-model) over performed the CC-model and EDGE-model. We limited the performance comparison among the CM-model

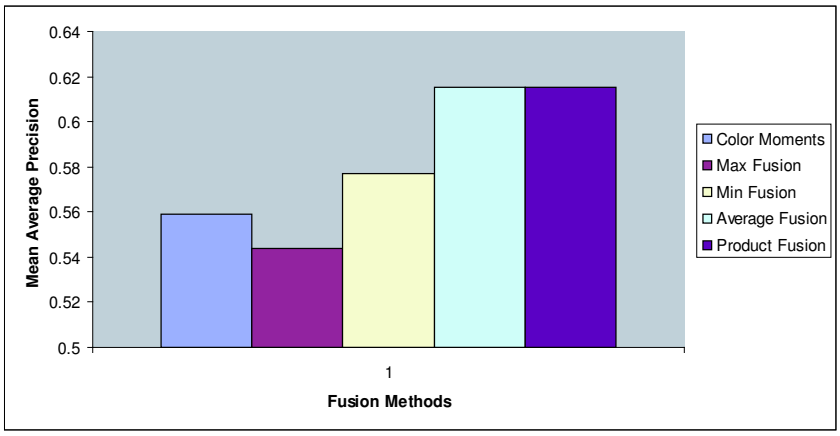


Figure 3. The performance comparison (in term of mean average precision) of different fusion approaches to CM-model. The dataset is the validation dataset.

and the fused models. Figure 2 shows the performance improvement for the 39 high-level features after the fusions were applied. In figure 3, we plot the graphical comparison of Mean Average Precision (MAP) across all the high-level features. The results show that Average Fusion and Product Fusion are competitive, and the performance improvement is impressive.

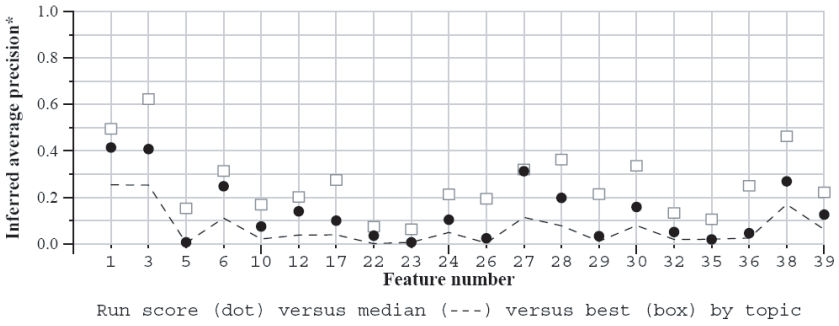
We submitted the following six runs to this year's TRECVID:

- **A_UCF.CE.PROD**: product fusion of two classifiers using color moments and edge histogram.
- **A_UCF.CE.PROB**: training-based fusion of two classifiers using color moments and edge histogram. The final decision is made using the product of probabilities.
- **A_UCF.CEC.PROD**: product fusion of three classifiers using color moments, color correlogram and edge histogram.
- **A_UCF.MIX**: re-rank the results of the run **A_UCF.CE.PROD** using simple concept dependency.
- **A_UCF.CM**: the output of the classifier using color moments.
- **A_UCF.EDGE**: the output of the classifier using edge histogram.

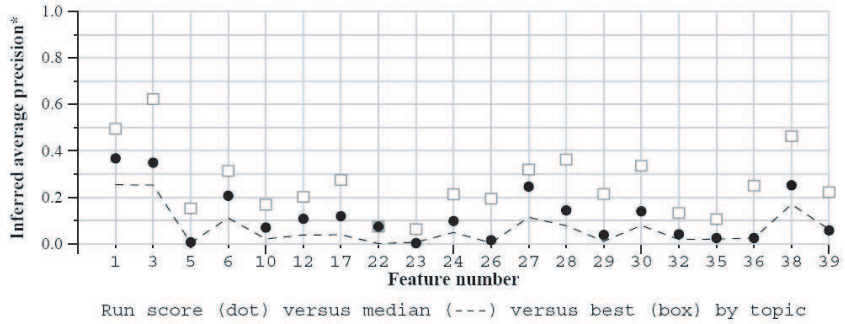
Figure 4 displays the performance of each run compared to all the runs in the TRECVID 2006. Overall, our performance is above the median value, and some features hit or approach the best results like feature 27 "computer or TV screen" in run **A_UCF.CE.PROD** and feature 22 "corporation leader" in run **A_UCF.CE.PROB**. The comparison in term of Mean of inferred AP among all the runs is shown in Figure 5. The result is similar to that we got in our validation phase. Compared to the result produced by CM-model, all fusion approaches can get 37% to 66% improvment. The best result is obtained from the fusion using three visual features (CM, EDGE and CC), which is a little better than the fusion using CM and EDGE. In our experiment, we also noticed that color correlogram worked good for validation dataset, but performed poorly for TRECVID 2006 testing dataset. The reason might be that color correlogram is sensitive to similiarity between the training and testing dataset.

3. INTERACTIVE TOPIC SEARCH

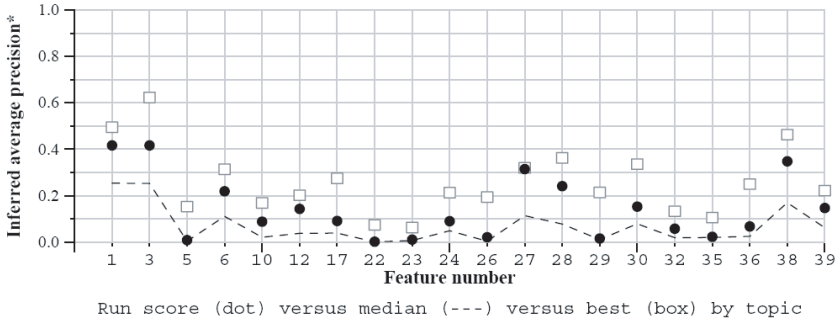
We performed the topic search task through the PEGASUS system, an online video retrieval system with a highly efficient user interface. It was developed for the topic search task in TRECVID 2005.[4] This year, we improved the system with new approaches and more visual features. The PEGASUS system comprises of three components mainly user interface,search engine server and feature index system as shown in Figure 6. Through the web-based



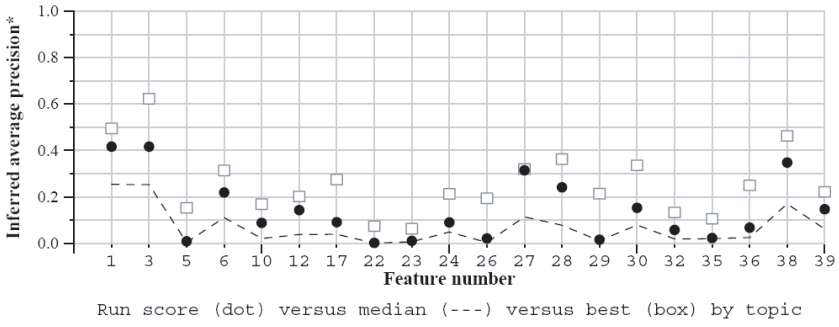
(a) run A_UCF.CE.PROB



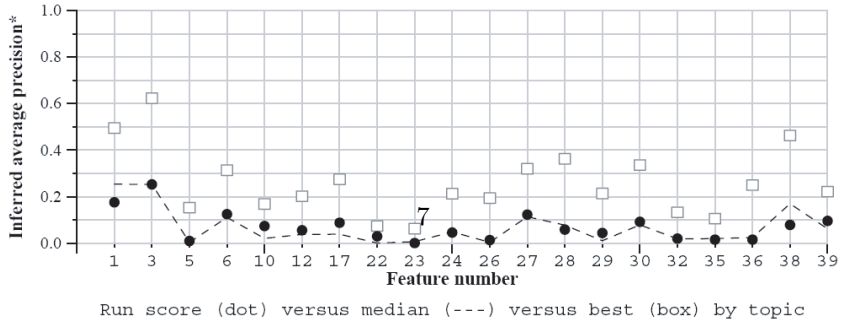
(b) run A_UCF.CE.PROB



(c) run A_UCF.CEC.PROD



(d) run A_UCF.CM



(e) run A_UCF.EDGE

Figure 4. Performance of our four runs compared to all the TRECVID 2006 runs. Dot, box and dotted line represent our result, the best result and the median result respectively.

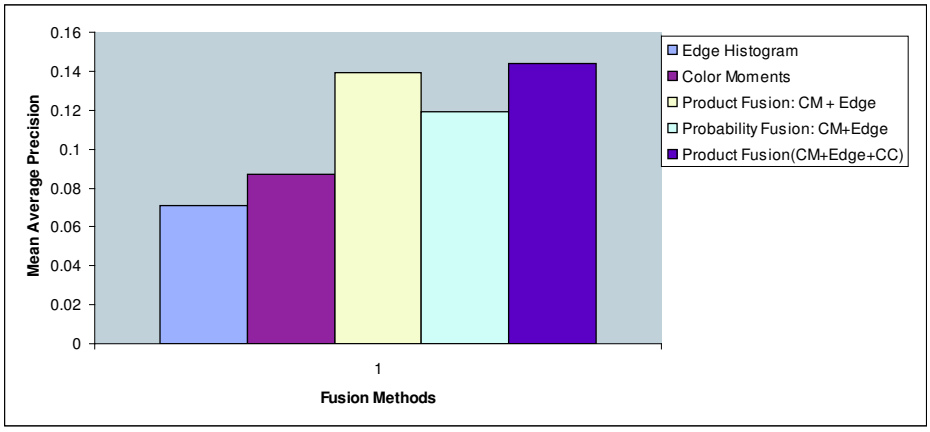


Figure 5. The performance comparison among five runs.

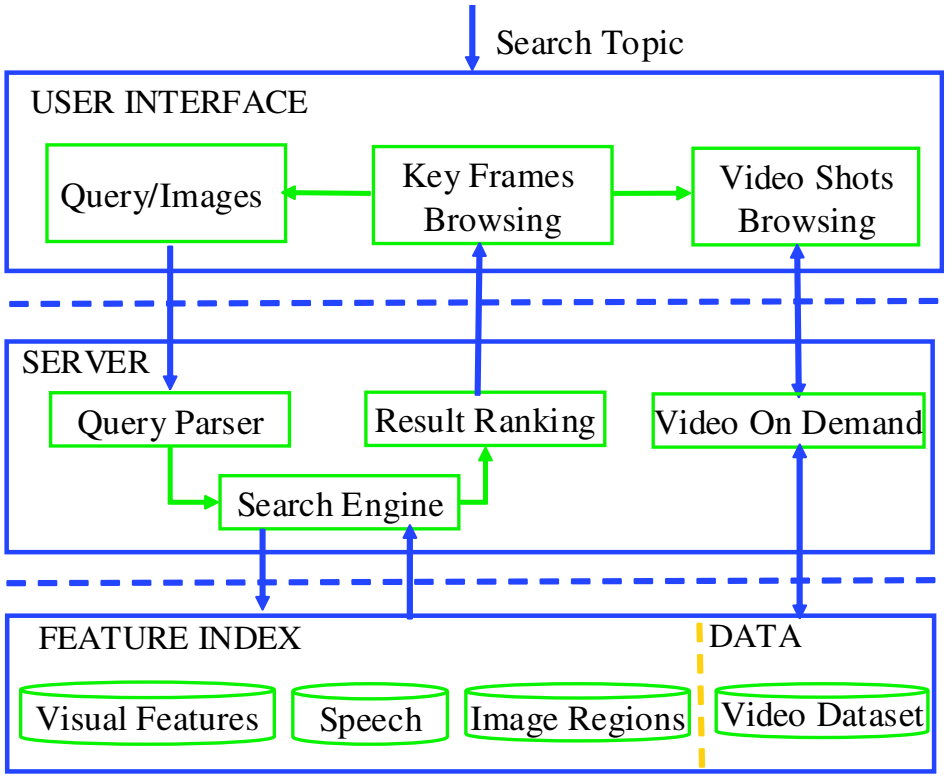


Figure 6. The overview of the PEGASUS news video online search system

interactive interface (as shown in figure 7), the user can launch a new search in two ways: formulate the text query according to the search topic using any known words, or search by the known image/video examples. Once the query is submitted, the server engine is able to retrieve the relevant shots from the feature index system. From the initial results, further refinement could be conducted based on the text and visual features such as words-histogram, region-based refinement.[3] This system also provides a Video on Demand (VoD) scheme to make it convenient for the user to browse the video shot without being limited to the key frames.

3.1. Indexing

Indexing is crucial for fast video retrieval. In the indexing part, it contains three components based on which the indices were created: ASR, global visual information and region information. The Lucene full-text index is adopted to index the ASR information. For global visual information, we computed the color moments, color correlogam and edge histogram. The indexing system for global visual features is implemented in the linked-list form. Each key-frame is accompanied with a ranked-list of its top N similar shots. SR-tree, a fast efficient high dimensional data indexing technique, has been used to index the region features.

3.2. Query Expansion

Text information can be useful for video retrieval, especially for finding person X, sports and some events. Generally, there exists multiple ways to describe the same event and object, which is called "Synonymy". However, the initial manually formulated query is not quite relevant to what the user really wants to retrieve. Thus, the user will either miss some relevant video shot with different speech expression from the query, or return false video shots that have speech expression with similar semantic meaning. For example, the user wants to find the shots that are related to Condoleezza Rice. If the query is simply "Rice", we might miss the shots containing "secretary of state" or "national security advisor" in the ASR transcript. On the other hand, we may find shots on the food "rice", that are irrelevant. We believe there exists a statistic co-occurrence relation between the key words. For instance, "Rice" has high co-occurrence with "secretary", and rice has high co-occurrence with "food" or "production".

Latent semantic analysis[13] can discover the covariance between the keywords and video shots. However it needs training phase, which is not much helpful for the wildly distributed in video contents. Normally, a single user is limited by specific knowledge making it really hard in the first attempt to formulate a perfect query that can retrieve all the relevant video shots. Thus, we need to provide a way for the user to expand the query based on the returned video shots, such that the refined query is able to find more relevant video shots. From the observation, we notice that the relations between keywords in different datasets should be varying. For example, in the news data corpus, the topic "soccer" is strongly correlated with keywords "tournament", "cup" and "game". On the other hand, in the instructional videos of soccer, "soccer" is more correlated with "Forward-Foot Pass", "Flick Pass" or "Far Forward". Thus, the relationships between keywords should be discovered based on the target data corpus rather than from a neutral source, like wordnet.

We used a query expansion technique based on words-histogram, which enriches the search query to cover more relevant shots. The expansion is performed using the speech (ASR) information of the videos, expressed in the text format. From the shots returned by the first round of search with query Q_{i-1}, the user can select a set of shots, which is considered relevant, A keyword histogram $WH = \{(a_1^+, W_1^+), (a_2^+, W_2^+), \cdots, (a_m^+, W_m^+)\}$ is computed based on the ASR of positive set, where W_i^+ is the extracted keyword accompanied by its normalized frequency a_i^+ in the positive set. The system returns a specified number of keywords with highest a_i value. The user is able to formulate a new query based on this significant information obtained.

3.3. Relevance Feedback Using Visual Features

3.3.1. Global Matching

We extracted color moments, color correlogram and edge histogram from the keyframes. When computing color moments and edge histogram, the image is divided into 5 by 5 grid, and then for each sub-block the visual features are extracted. When computing the similarity of two images, L_1 distance measure is used due to its robust. Equation 5 gives the "relative" distance measure for two images. It has been justified theoretically in [8].

$$|I_a - I_b| = \sum_{i=1}^{D} \frac{|f_i(I_a) - f_i(I_b)|}{\epsilon + f_i(I_a) + f_i(I_b)}, \tag{5}$$

where I_a and I_b are two images, $f_i(I)$ denotes the i-th dimension of the feature vector of image I, ϵ is a constant used to prevent the denominator from being zero. It was set to 1 in our system.

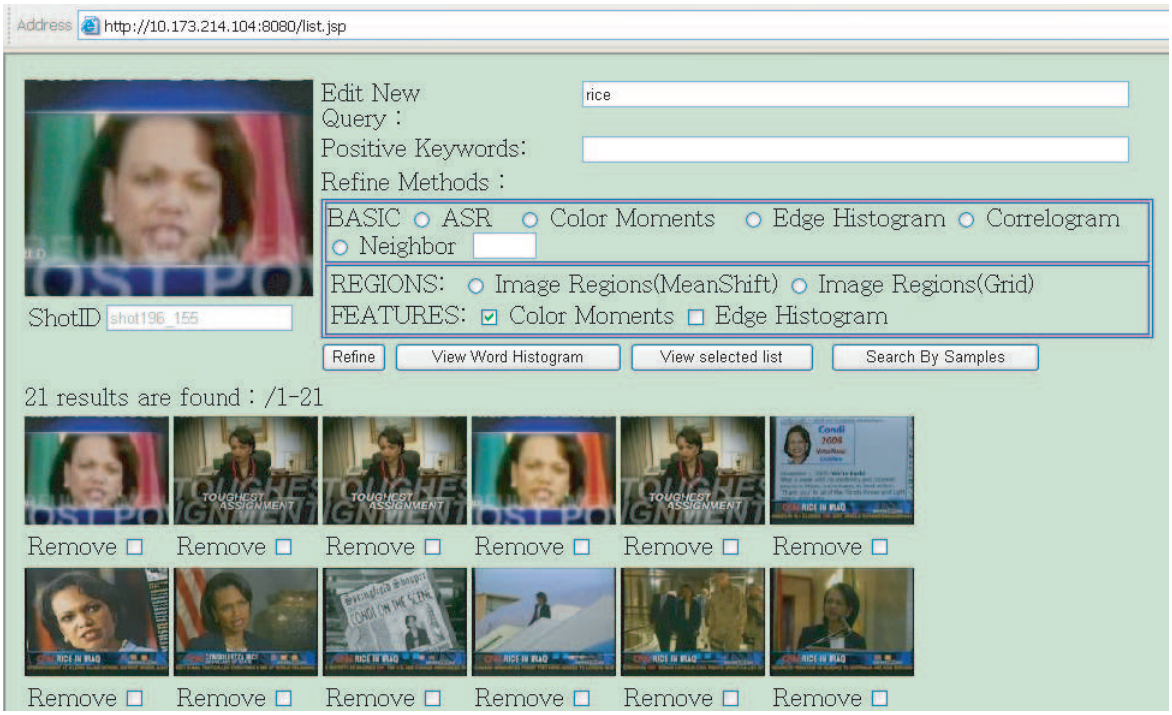


Figure 7. A snapshot of the interface.

3.3.2. Region-based Refinement

We also developed an image-based refinement scheme, that analyzes the similarities between the shot key-frames. We use the mean shift color segmentation algorithm[14] to segment the images, then extract the local color feature and local edge histogram from each regions. We eliminate the weak regions that are either with $\frac{A_I^i}{max(A_I^j)} \leq A_h$, where A_I^i is the area of the region i from image I, A_h is a threshold on area, or $d(C_I^i, O) \leq C_h$,where C_I^i is the centroid of the region, $d(C_I^i, O)$ denotes the distance between region I^i and the centroid of image I, and C_h is the threshold.

The image-based refinement is performed in a relevance feedback process. The user selects a set of relevant shots from the results returned by the previous search round. The key-frame regions of the selected shots are treated as the new visual queries for the next round. The search is based on individual regions. The returned results contain the key-frames which have the similar regions to the query regions. For example, given a query image I with multiple regions$\{I^1, I^2, I^3, \cdots\}$, the region-based search result is $\{(I_i^1, I_j^1, I_k^1, \cdots)$,$(I_i^2, I_j^2, \cdots)$, $(I_m^3, I_i^3, I_k^3, \cdots)$, $\cdots\}$. In this case, $\{(I_i^1, I_j^1, I_k^1, \cdots)$ are the images that have the similar regions to the first query region of I, I^1,,$(I_i^2, I_j^2, \cdots)$, are the images that have the similar regions to the second region of I,I^2, and so on. Suppose $d(I_i^Z, I_j^K)$ denotes the distance between the region Z of key frame I_i and the region K of key frame I_j. All the returned regions should satisfy $d(I_i^Z, I_j^K) \leq D$, where D is the threshold of the distance between any regions. Therefore, all the images returned through the filter are taken as candidates for similar images. To further rank the relevance of returned images, we incorporate the Earth Mover's Distance (EMD)[15] in the image-to-image similarity computation. We model the regions in the image by the nodes in the bipartite graph, and regions from the same image are the nodes in the same partite. Here, node N_i in the graph is used interchangeably with region I^i in the image. Thus, given two images I_X and I_Y, their EMD is computed as follows,

$$EMD(I_X, I_Y) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d(I_X^i, I_Y^j) f_{ij}}{\sum_{j=1}^{m} f_{ij}}, \tag{6}$$

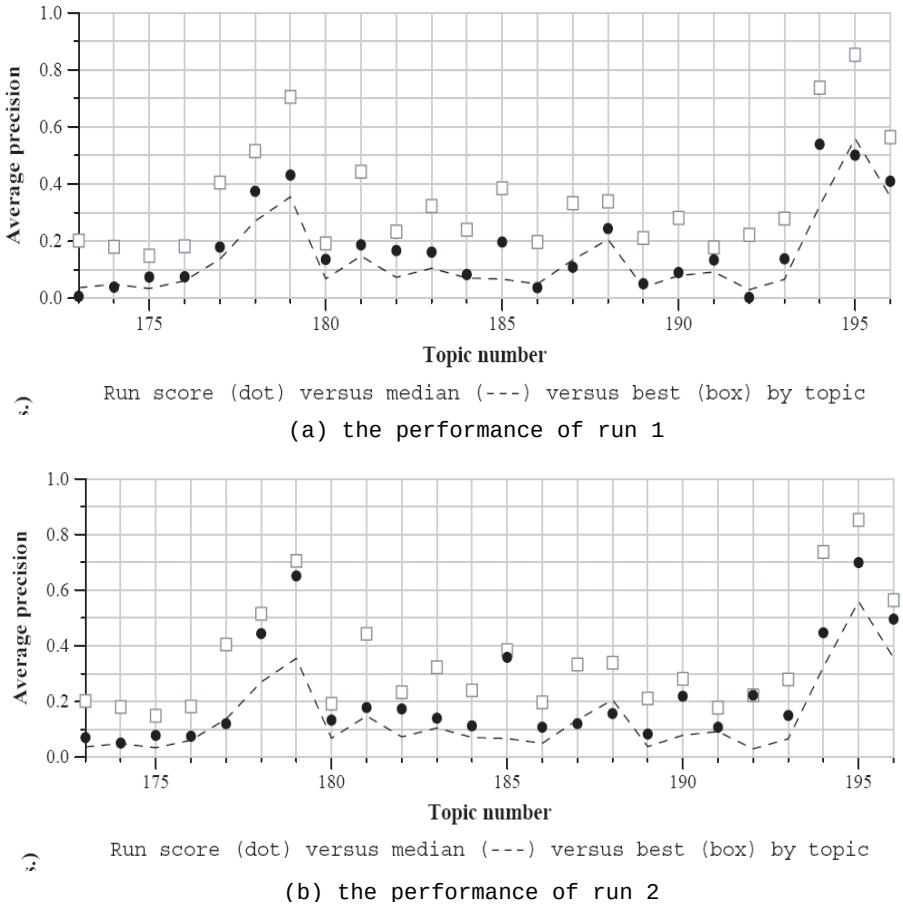


Figure 8. The performance comparison of Run A_1_UCFVISION1 and Run A_2_UCFVISION1 to all the TRECVID 2006 runs. Dot, box and dotted line represent our result, the best result and the median result respectively.

where $d(I_X^i, I_Y^j)$ is the distance between node N_i to node N_j, f_{ij} is the flow amount from node N_i to node N_j, and m and n are the numbers of regions in images I_X and I_Y, respectively. In our system, we use the Euclidean distance between the feature vectors of the regions. The flow amount f_{ij} is computed by solving the maximum flow problem for the bipartite graph. In this graph formulation, the area A_i is used as the weight of each node N_i.

3.4. Results and Discussion

We submitted two runs for the interactive topic search. Run "A_1_UCFVISION1" is purely based on the ASR information with word-histogram refinement, while "A_2_UCFVISION2" involves the interactive search using text and visual information. In each run, the temporal "K-nearest neighbor" method is used as the final step in the relevant feedback. In run "A_2_UCFVISION2", we launched the search by example images or video keyframes for some topics, which is hard to get initial results using text. As we expected, the run using both text and visual information to refine performs better than the one only using text information. Figure **??** plots the performance comparison of our runs to all the TRECVID 2006 runs in term of average precision. The whole performance is a little better than the median performance among all the submitted runs.

REFERENCES

1. B. E. Boster, I. Guyon and V. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In COLT, pp. 144-152, 1992.

2. J. Liu, Y. Zhai and M. Shah, PAGASUS: An Information Mining System for TV News Videos, SPIE Defense and Security Symposium, April 17-21, Orlando, 2005.
3. Y. Zhai, J. Liu and M. Shah, Automatic Query Expansion in News Video Retrieval, International Conference on Multimedia and Expo (ICME), July 7-12, 2006. Toronto, Canada.
4. Y. Zhai, J. Liu, X. Cao, A. Basharat, A. Hakeem, S. Ali, M. Shah, C. Grana and R. Cucchiarra, Video Understanding and Content-Based Retrieval, TREC Video Retrieval Evaluation Workshop (TRECVID 2005), Gaithersbury, Maryland, November 14-15, 2005.
5. J. Huang, S. Kumar, M. Mitra, W. Zhu and R. Zabih, Image Indexing Using Color Correlogram, Proceeding of Computer Vision and Pattern Recognition, pp.762-768, 1997.
6. A. Amir, J. Argillander, et.al., IBM Research TRECVID-2005 video Retrieval System, TREC Video Retrieval Evaluation Workshop (TRECVID 2005), Gaithersbury, Maryland, November 14-15, 2005.
7. S. Chang, W. Hsu, et.al. Columbia University TRECVID-2005 Video Retrieval and High Level Feature Extraction, TREC Video Retrieval Evaluation Workshop (TRECVID 2005), Gaithersbury, Maryland, November 14-15, 2005.
8. D. Haussler, Decision Theoretic Genelization of the PAC model for neural net and other learning applications. Information and Computation, 100:78-150, 1992.
9. S. Brandt, J. Laaksonen and E. Oja, Statistical Shape Features for Centent-Based Image Retrieval, Journal of Mathematical Imaging and Vision, pp187-198, 2002.
10. R. Yong and T. Huang, A Novel Relevance Feedback Technique in Image Retrieval, ACM Multimedia, 1999.
11. Scott, D.W., Multivariate Density Estimation : Theory, Practice, and Visualization. Wiley-Interscience, 1992.
12. C. Hsu, C. Chang, and C. Lin. A Practical Guide to Support Vector Classification. http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf.
13. Thomas Hofmann, "Unsupervised Learning by Probabilistic Latent Semantic Analysis", Machine Learning,Vol.42, pp. 177-196, 2001.
14. D. Comaniciu and P. Meer. Mean shift: A Robust Approach Toward Feature Space Analysis, PAMI, 24, pp603-619, 2002.
15. Y. Rubner, C. Tomasi and L. Guibas, A Metric for Distributions with Applications to Image Databases, ICCV, 1998.