FXPAL at TRECVID 2005

Matthew Cooper[1], John Adcock[1], Robert Chen[2], and Hanning Zhou[1]

[1]FX Palo Alto Laboratory
Palo Alto, CA 94304
{last name}@fxpal.com

[2]Carnegie Melon University
Pittsburgh, PA 15213
mychen@cs.cmu.edu

In 2005 FXPAL submitted results for 3 tasks: shot boundary detection (section 1, page 1), high-level feature extraction (section 2, page 5), and interactive search (section 3, page 7).

1 Shot boundary detection

1.1 Summary of submitted runs

The shot boundary detection system we are using for 2005 builds on the framework and system developed in 2004 which combines pairwise similarity analysis and supervised classification. Using primitive low-level image features, we build secondary features based on inter-frame dissimilarity. These secondary features are used as input to an efficient k-Nearest-Neighbor (kNN) classifier. The classifier labels each frame as a shot boundary or non-boundary, and the classifier outputs are minimally processed to determine the final segmentation.

This year we added information-theoretic feature selection to determine two secondary feature subsets to improve cut transition detection and gradual transition detection, respectively. These systems appeared as runs sys10M_0X in the run table. This indeed improved performance over our baseline runs (sys05_0X), and improved on the performance of a similar system using random projection for dimension reduction (sys10R_0X). Our performance was worse than anticipated, as our training data was not an accurate reflection of the test data in the case of video from LBC and CCTV. On the remaining videos, our performance was very good, and consistent with our training experiments. The realizability of this approach remains an open question.

1.2 Overview

Our systems are based on the use of pairwise inter-frame similarity features in combination with a fast exact k-nearest-neighbor (kNN) classifier to label frames as members of the boundary or non-boundary class. The basic system is documented in [1, 2] and has three main components:

Low-level feature extraction For each frame we extract three channel color histograms in the YUV colorspace. We extract both global image histograms and block histograms using a uniform 4×4 spatial grid. Denote the frame indexed feature vectors $\mathbf{X} = \{X(n) : n = 1, \cdots, N\}$ for N frames.

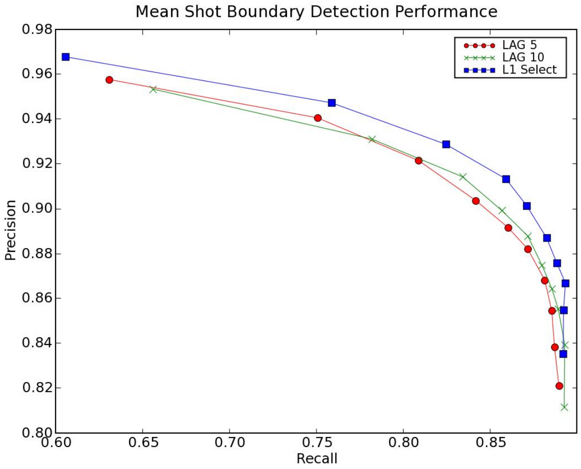


Figure 1: Mean performance of 2005 shot boundary systems on SB04 data.

Inter-frame similarity features For a maximal lag L which is either 5 or 10, we compute two lag domain (partial) similarity matrix of the form:

$$S(i,l) = D_{\chi}(X_i, X_{i-l}) = \frac{1}{2}\sum_b \frac{(X_i(b) - X_{i-l}(b))^2}{X_i(b) + X_{i-l}(b)} \quad . \tag{1}$$

The first matrix S_G contains the chi-square similarity between frames' global histograms. The matrix S_B contains the chi-square similarity between frames' block histograms. Intermediate-level features are constructed from these matrices by concatenating elements within a local neighborhood. For frame n, the inter-frame similarities $S_G(n+k, n+l) : k, l = -L, \cdots, L$ and $S_B(n+k, n+l) : k, l = -L, \cdots, L$. Because D_{χ} is symmetric and $S(n,n) = 0$, the intermediate-level features have dimensionality 90 for $L = 5$ and 380 for $L = 10$.

Frame classification Given the intermediate level feature vector for each frame and a labeled training set, we use the efficient kNN classifier of [3] to classify each frame as either a non-boundary, cut boundary, or gradual boundary. This implementation has been tested in a similar context and provided speedups of more than a factor of 10 over naive implementations.

Our runs were generated by the three following systems:

LAG05 This system used intermediate features corresponding to maximum lag $L = 5$. It included no dimension reduction and was used in 2004. These are the sys05_0X runs.

LAG10 This system used intermediate features corresponding to maximum lag $L = 10$. These features, and the training data, were then randomly projected [4] to a 90 dimensional subspace. The 90 dimensional features were then input to the kNN. This system was also used in 2004. These are the sys10R_0X runs.

LAG10MI90 This system used intermediate features corresponding to maximum lag $L = 10$. These features, and the training data, were then projected via information-theoretic feature selection as described in [5]. A 90 dimensional subset of the original 380 features were then input to the kNN. These are the sys10M_0X runs.

In 2004, LAG05 outperformed LAG10 in cut boundary detection. This is not surprising since abrupt boundaries will be most prominent in the inter-frame similarities in a very small local neighborhood around the boundary frame. These similarities are being smoothed in the random projection, and resolution is lost. The opposite is true in the gradual boundary case in which LAG10 outperforms LAG05. Although the feature dimensionality is essentially the same for the two systems, the features in LAG10 include longer range inter-frame comparisons which improve gradual boundary detection performance. Smoothing these similarities does not hurt performance since the transitions occur over a series of frames.

For 2005, we added **LAG10MI90** to select feature subsets for detecting cut transitions versus gradual transitions. The approach has a few drawbacks however. First, it does neglect high order dependencies which may be important here. Secondly, evaluating the first order terms involves repeated density estimation. In our experience, estimating these densities, and in turn the feature subsets, has been unstable. The choice of the number of bins for histograms, and the use of different training files has produced very different results which is unsettling. Nonetheless, results in the training experiments for this year are promising.

Shot boundary detection results									
	Mean			Cut			Gradual		
SYS	R	P	F	R	P	F	R	P	F
LAG05-4	0.8490	0.8080	0.8280	0.8690	0.8470	0.8579	0.7880	0.7010	0.7420
LAG05-5	0.8390	0.8210	0.8299	0.8610	0.8550	0.8580	0.7770	0.7280	0.7517
LAG05-6	0.8270	0.8370	0.8320	0.8510	0.8620	0.8565	0.7560	0.7640	0.7600
LAG10-5	0.8430	0.8050	0.8236	0.8600	0.8470	0.8535	0.7910	0.6970	0.7410
LAG10-6	0.8360	0.8200	0.8279	0.8540	0.8530	0.8535	0.7840	0.7280	0.7550
LAG10-7	0.8220	0.8320	0.8270	0.8410	0.8590	0.8499	0.7650	0.7560	0.7605
LAG10MI90-4	0.8550	0.8060	0.8298	0.8780	0.8540	0.8658	0.7850	0.6810	0.7293
LAG10MI90-5	0.8470	0.8240	0.8353	0.8700	0.8630	0.8665	0.7790	0.7190	0.7478
LAG10MI90-6	0.8380	0.8410	0.8395	0.8620	0.8730	0.8675	0.7690	0.7480	0.7584
LAG10MI90-7	0.8180	0.8580	0.8375	0.8450	0.8800	0.8621	0.7410	0.7900	0.7647
TV AVG.	0.7881	0.7328	0.7594	0.8508	0.78561	0.8169	0.6045	0.5656	0.5844
FXPAL AVG.	0.8374	0.8252	0.8313	0.8591	0.8593	0.8592	0.7735	0.7312	0.7518

Table 1: Table summarizing shot boundary detection results in terms of recall (R), precision (P), and the f-score (F). In the system description column, the final digit after the dash corresponds to the value of κ used.

1.3 Training Experiments

For SB05 we are submitting runs using the three systems whose performance on the 2004 test data set appears in Figures 1. As was the case last year, the mean performance of the systems LAG05 (red) and LAG10 (green) are similar. The new system, LAG10MI90 (blue) outperforms both consistently. In these experiments, and in our test systems, we use the 2003 test set to generate our training data. To generate the training data, we use the manually generated reference segmentation and randomly discard 90% of the non-transition frames.

Examining the separate results for cut and gradual boundary detection, we see that in both cases LAG10MI90 is able to outperform both 2004 systems. In the cut boundary case, LAG10MI90 performs best of the three systems. This is consistent with the idea that most of the information relevant to cut boundary detection is contained in a small number of local features. In the gradual boundary case at high recall levels, LAG10 is superior. Again, the use of more data, even if smoothed together in projection, is beneficial for high recall gradual boundary detection. My conjecture is that any feature selection (rather than projection) technique will exhibit this behavior. The highest F-scores for gradual boundary detection are achieved by LAG10MI90.

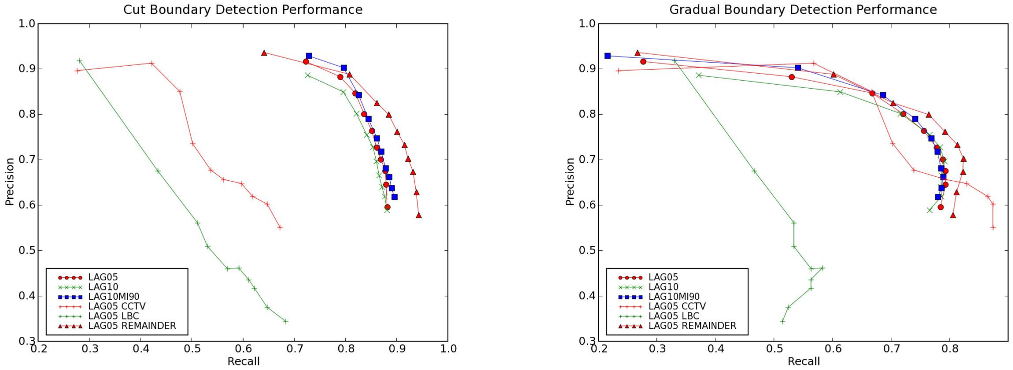


Figure 2: Performance of 2005 shot boundary systems. Additional curves show performance on LBC data, CCTV data, and remainder. The panels show cut boundary detection (top) and gradual boundary detection (bottom). The bottom columns show average TRECVID and average FXPAL results.

1.4 Submitted runs and results

For our submission, we select ten runs for the TRECVID evaluation conducted by NIST. The systems include the variants described above, and tradeoff precision and recall. For this, we use a parameter $\kappa : 1 \leq \kappa \leq k$ where k is the number of neighbors considered by the kNN classifier. For all our systems, $k = 11$. A test frame is traditionally assigned the majority label of its k nearest neighbors in the training set. To study the tradeoff between precision and recall, we assign a test frame the boundary label if at least κ of its nearest neighbors are boundaries. We then vary κ between 0 and k.

Our performance was disappointing relative to our results in 2004 and the training experiments above. Post-evaluation analysis revealed that two of the videos in the test set[1] exhibited hundreds of two frame dissolves. Apparently, such dissolves were present neither in the other videos in the test set, nor in the 2003 test set we used for training data. Figure 2 shows our results for evaluation, the results on the two aforementioned videos, and the results on the remaining videos.

It seems difficult to draw a clear conclusion from the discrepancy in performance between the training experiments and test results. One could argue that the approach generalizes poorly, and perhaps this is an inherent limitation of supervised approaches generally, and nearest-neighbor methods in particular. The fault could alternately lie with the intermediate similarity features, which may not flexibly capture characteristics of shot boundaries that would permit such generalization. On the other hand, the approach did generalize successfully to the data from MSNBC, NBC, NTDTV, and NASA. It's also worth noting that the training data is six years older than the test data. Because the TRECVID data

[1] "20041102_160001_CCTV_DAILY_NEWS_CHN.mpg" and "20041119_140000_LBC_LBCNAHAR_ARB.mpg"

from NIST is provided on a two year cycle, we will probably need to wait until next year for manually segmented training data from the same content providers as the test data to start to resolve these issues.

2 High-level feature detection

2.1 Summary of submitted runs

Unlike shot boundary detection, this was our first year of participation in the high-level feature detection task. As a result, our chief goal was to develop infrastructure and submit some reasonable baseline systems. Two of our runs were based on simple single modality SVM classifiers (A_AL-Run1_4 and A_AL-Run3_5). An additional run was based on combining these classifiers via logistic regression (A_AL-Run3_3). A final system combined the classifier outputs for the current shots with those of its immediate neighbors via logistic regression (A_AL-Run4_2). This system produced our best runs. The run A_AL-Run1_1 integrated all of this information using a random field modeling framework.

Feature detection results											
SYS	38	39	40	41	42	43	44	45	46	47	**AVG**
A_AL-Run1_4	0.099	0.017	0.117	0.053	0.254	0.168	0.17	0	0.207	0.145	0.12
A_AL-Run2_5	0.094	0.013	0.136	0.084	0.254	0.161	0.157	0.001	0.132	0.137	0.12
A_AL-Run3_3	0.154	0.028	0.151	0.071	0.39	0.239	0.184	0	0.312	0.237	0.18
A_AL-Run4_2	0.161	0.03	0.161	0	0.399	0.246	0.225	0.001	0.357	0.259	0.18
A_AL-Run1_1	0.137	0.041	0.122	0.052	0.224	0.196	0.173	0.024	0.17	0.264	0.14
MED	0.106	0.031	0.171	0.061	0.225	0.165	0.128	0.001	0.206	0.158	0.139

Table 2: Table summarizing the high-level feature detection results in terms of mean-averaged precision (MAP). The bottom column shows median TRECVID results.

2.2 Per shot pre-processing

Low-level feature were extracted from each shot in the reference segmentation, mostly from processing of the keyframes. The informedia team at CMU generously shared their low-level features with us this year:

- three channel color histograms: hsv, hvc, rgb
- color correlogram
- Gabor texture features
- Canny edge feature
- three Audio feature: MFCC, SFFT and FFT
- face detector output

For each feature in the development set, we trained a SVM classifier using each of the above low-level features. This results in 400 individual SVMs corresponding to the 40 features and ten modalities.

2.3 Multi-modality fusion via SVMs

Our systems thus used ten different SVM outputs per feature which were fused according to various strategies.

A_AL-Run1_4 In this run, only the best performing single modality SVM from the training data was used.

A_AL-Run2_5 In this run, only the second best performing single modality SVM from the training data was used.

A_AL-Run3_3 In this run, the results of the ten SVMs were combined via logistic regression.

A_AL-Run4_2 In this run, the results of the ten SVMs on the previous, current, and subsequent shot were combined via logistic regression.

A_AL-Run1_1 In this run, the results of the ten SVMs on the previous, current, and subsequent shot were combined using a random field model to capture inter-concept interaction.

2.4 Information fusion via random fields

The final run was based on the discriminative random field (DRF) model of [11]. This model combines discriminative classifiers with pairwise interaction representing contextual information. In the original DRF work, the goal is to perform binary classification of pixel blocks. The random field incorporates spatial dependencies into the inference. More specifically, the probability of the vector of binary class labels Y given the image block X as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i \in S} A_i(y_i, \mathbf{x}) + \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} I_{ij}(y_i, y_j, \mathbf{x}) \right) . \tag{2}$$

The terms A_i are the association potentials which are local discriminative classifiers for the i^{th} spatial location in the set of locations $\mathcal{S}$. I_{ij} is the term representing the interaction between the i^{th} and j^{th} spatial locations, for the set of neighbors $\mathcal{N}_i$. [11] details learning and inference methods for this class of random fields.

We adapted this approach to the semantic labeling context. As before, we use logistic classifiers for the association terms in (2). Thus we need to identify which concepts are related, i.e. which concepts are neighbors (connected by an edge in our graph). For this, we performed a chi-squared test using the ground truth labeling of our training set. We (arbitrarily) selected the five most statistically significant inter-label relationships to form a fully connected concept graph for each cluster. These graphs define the neighborhoods $\mathcal{N}_i$ for each label y_i. This approach allows us to jointly infer *all labels in* $\mathcal{N}_i$ *simultaneously* in hopes of exploiting inter-label dependencies.

In contrast to [11], we wanted to include case by case interaction terms (rather than the smoothing effect provided in the formulation of (2)). Similarly,

$$I_{ij}(y_i, y_j, \mathbf{x}) = \sum_{k,l \in \pm 1} \delta(y_i = k)\delta(y_j = l) V_{kl}^T \mathbf{x} \quad . \tag{3}$$

Here we use Kronecker delta functions so that we learn different weights for each of the four permutations of (y_i, y_j). To learn the regression weights, we use gradient descent with the following equations:

$$\frac{dl(\theta)}{dW_i} = \sum_{m=1}^{M} \left\{ \sum_{i \in C} \left(y_i^m - \sum_{Y} P(Y|\mathbf{x}^m, \theta) y_i \right) \mathbf{x}^m \right\} \quad , \tag{4}$$

and

$$\frac{dl(\theta)}{dV_{ij}} = \sum_{m=1}^{M} \left\{ \sum_{i \in C} \sum_{j \in N_i} \left(1 - \sum_{Y(i=k,j=l)} P(Y|\mathbf{x}^m, \theta) \right) \; k \; l \; \mathbf{x}^m \right\} \quad . \tag{5}$$

The performance of this system was relatively disappointing, and we believe this is because we did not approach model induction with sufficient care. Additionally, the use of multiple modalities and multiple time samples obscured how informative various single modality concept associations and inter-concept interactions were. In other words, the model was too complicated to properly assess DRFs for high-level feature extraction. Both the groups of concepts that are collectively detected as well as the association terms used for a specific concept group need to be constructed and selected with great care. We hope to apply alternate model induction and feature selection strategies to improve our results in the future.

2.5 Evaluation results

Generally speaking, A_AL-Run3_3 and A_AL-Run4_2 did fairly well. For most features, they were above the median. A_AL-Run1_4 and A_AL-Run2_5 were generally below the median, but this was not unexpected. A_AL-Run4_2 did improve on A_AL-Run3_3, although it was more costly computationally. All systems were straightforward in design, and we look forward to incorporating enhancements to the system in the future. Results are summarized in Table 2

3 Interactive Search

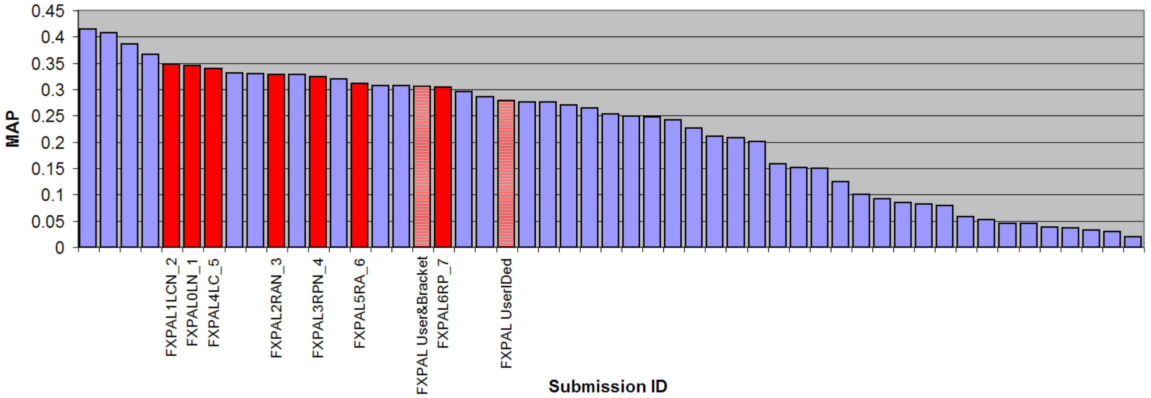


Figure 3: MAP performance of all interactive search submissions with FXPAL submissions shown in red. The striped-bars are not actual submissions made to NIST, but indicate the MAP performance of just the user-identified shots and just the user-identified shots with neighboring shots included.

3.1 Summary of submitted runs

The interactive search system we used this year is incrementaly different from our 2004 system. The most noteable change for 2005 is the incorporation of high-level feature detection results within the

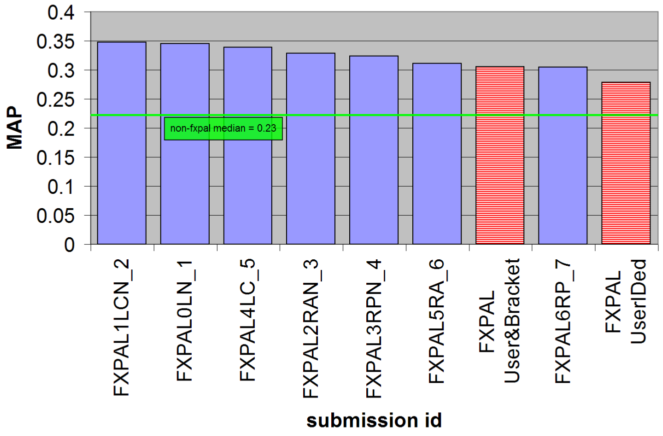


Figure 4: MAP performance of FXPAL interactive search submissions. The striped-bars at the right are not actual submissions made to NIST, but indicate the MAP performance of just the user-identified shots and just the user-identified shots with neighboring shots included.

search system. We submitted 7 runs for the interactive search task. Each of the 2 searchers answered his set of 12 topics once, and the system used 7 different methods to augment the list of explicitly-identified relevant shots at the end of the 15 minute interactive session. The submitted runs differ only in this final automated query step. The complete set of submitted runs in priority order with system names and brief descriptions:

1. FXPAL0LN: LSA text query with bracketing - identical to our TRECVID2004 system denoted *LSA1*[2]
2. FXPAL1LCN: LSA and concept similarity scores averaged, with bracketing
3. FXPAL2RAN: Ranked combination of LSA and positive and negative concept similarity, with bracketing
4. FXPAL3RPN: Ranked combination of LSA and positive concept similarity only, with bracketing
5. FXPAL4LC: LSA and concept similarity scores averaged, no bracketing
6. FXPAL5RA: Ranked combination of LSA and positive and negative concept similarity, no bracketing
7. FXPAL6RP: Ranked combination of LSA and positive concept similarity only, no bracketing

All runs were fully interactive, type A, condition 1 and in the summary results are abbreviated as: I_A_1_AL0LN, I_A_1_AL1LCN, I_A_1_AL2RAN, I_A_1_AL3RPN, I_A_1_AL4LC, I_A_1_5RA, I_A_1_AL6RP. The system names are loosely acronymed with the following decoding:

L Latent Semantic Analysis (LSA) based text query/similarity scoring used
N Shots neighboring the relevant shots included (bracketing)
C Concepts (high level features) similarity scoring used
R Combination of text and concept scores by average rank rather than average retrieval score
A Both positive and negative reinforcement from concept similarity used

P Only positive reinforcement from concept similarity used

See section 3.7 for more details about the nature of the concept and text (LSA) scores.

System	MAP score	Notes
FXPAL 1 LCN	0.348	
FXPAL 0 LN	0.345	Post-processing identical to 2004 text-only system LSA1
FXPAL 4 LC	0.339	
FXPAL 2 RAN	0.329	
FXPAL 3 RPN	0.323	
FXPAL 5 RA	0.311	
User ID-ed shots with Neighbors	0.306	
FXPAL 6 RP	0.304	
User ID-ed shots	0.279	

Table 3: MAP scores for the 7 systems in performance order

Our performance was strong overall, submitting the run with the 5th highest MAP score. This is similar to our performance in 2004, in fact the the baseline system from 2004 is nearly tied as our best scoring run. Figure 3 shows the MAP performance of the FXPAL runs against the entire set of interactive search submissions and Figure 4 the FXPAL submissions alone. This year we had only 1 searcher session for each topic (in 2004 we had 3 unique users answer each topic) so all variation in our runs is from system differences. The MAP performance between different systems is not overwhelming, ranging from 0.348 to 0.304. Table 3.1 lists the runs in performance order. From this ordering a few observations can be made:

- The text-only baseline system from 2004 is out-performed by 1 concept-enhanced system by a small margin
- For every strategy, including a shot bracketing step improves performance
- The rank-based methods of combining similarity measures are outperformed by the score-averaging methods

Using the randomization script provided by NIST, an analysis of the statistical significance between our results yielded a probability of only 0.00034 that the LCN>LN result is due to chance, despite the small margin in performance. Figure 5 illustrates the performance in a slightly different way. For each topic the 7 systems were ordered by average precision and a histogram formed of the ranks over the 24 topics. In this we can see that the LCN system did not garner the highest MAP score through exceptional performance on a small handful of topics, but rather was most likely to perform best on any given topic.

3.2 Overview

The interactive search interface was designed for efficient browsing and rich visualization of search results, and is largely unchanged from our 2004 system [2].

Query results are displayed as a list of story thumbnails, sized in proportion to their query relevance. The story-level graphical summaries (thumbnails) use query relevance to build a queryrelated montage of the underlying shot thumbnails. Visual cues are widely used throughout the application to represent query-relevance and navigation history as well as shots included and excluded from the results list. Keyboard shortcuts are used throughout to reduce the amount of mousing required to sift through results

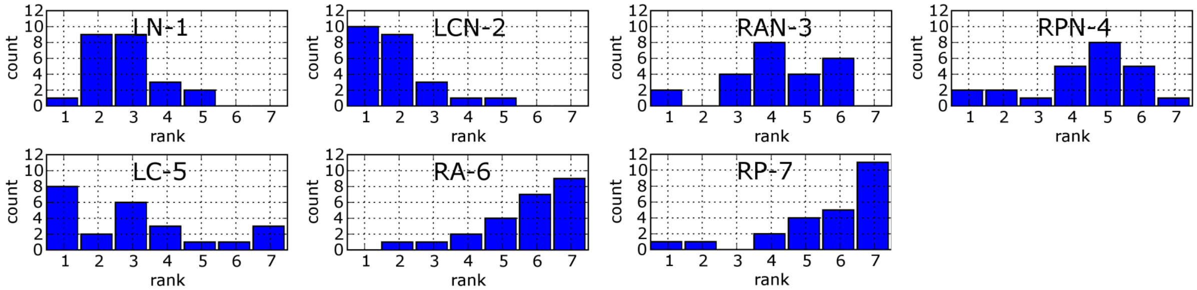


Figure 5: For each system, the histogram of the ranks achieved when the performance on each topic is individually compared with average precision. LCN stands out as the system most likely to be in first place, coming in first or second in 19 of the 24 topics.

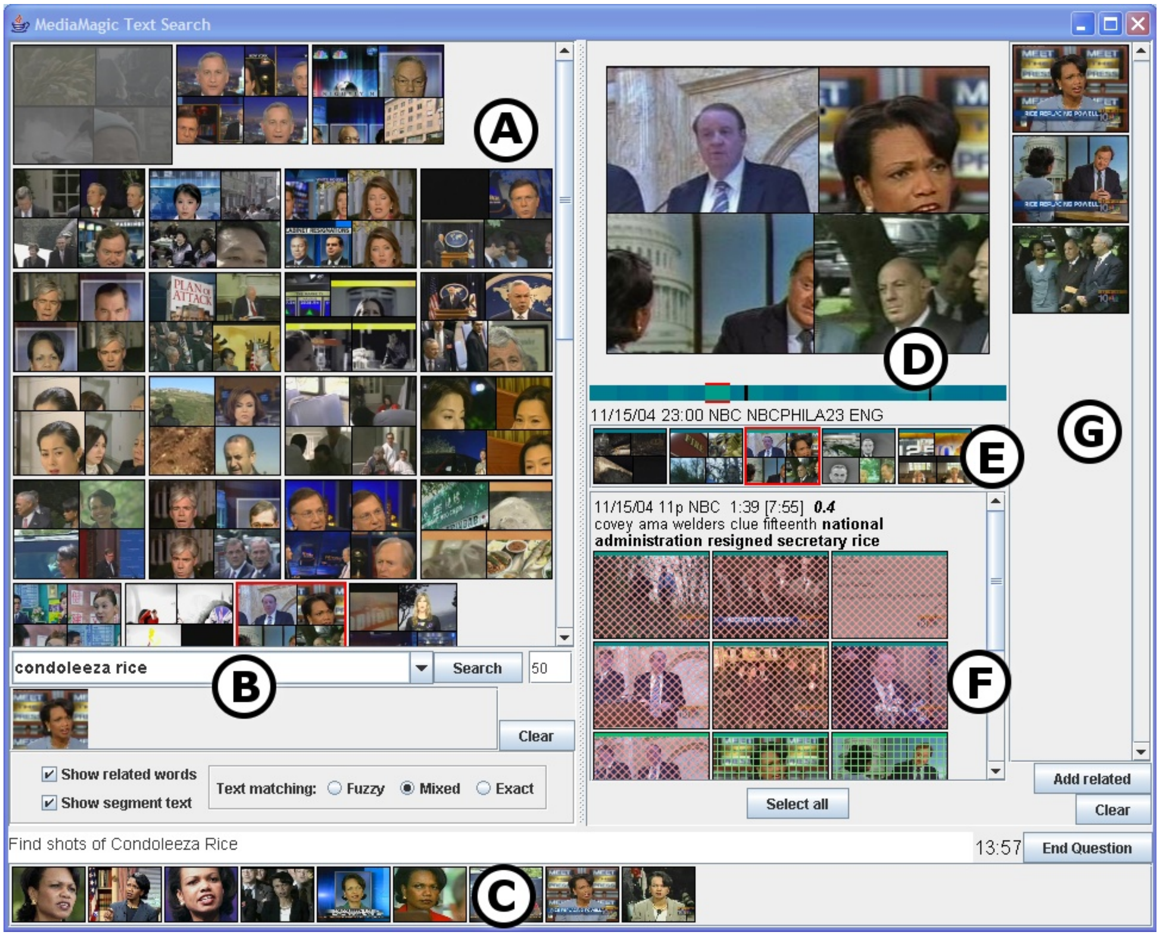
lists. We use a 2 level video segmentation with an automatically generated story-level segmentation supplementing the reference shot segmentation[6]. Text-based latent semantic analysis (LSA) of the transcripts is used to build the story-level segmentation. User text searches are performed with exact and/or LSA-based text search and optionally combined with a still-image query-by-example capability. The LSI of story segments is also leveraged in the UI to allow the user 2 different ways to search for “similar” stories or shots. In 2005 this function has incorporated similarity of high-level features as well.

3.3 Data Pre-processing

We perform a completely automatic pre-processing step to identify topic or story units to augment the reference shot boundaries [6]. These story segments provide the basic unit of retrieval during queries. To accomplish this segmentation we use the reference shot boundaries and the ASR transcripts. We build a latent semantic space (LSS) treating the stopped and stemmed [8] text tokens for each shot in the testing corpus as a separate document, adding words from adjacent shots to maintain a minimal number of tokens in each document. We then project the text for each shot into this shot-based LSS and compute a similarity matrix for each video using cosine similarity on the reduced-order vectors (one vector per shot). A checkerboard kernel is passed over the similarity matrix and points of highest novelty are chosen as story boundaries, as in [1]. A post-processing step assures the sanity of the boundary sizes and finds new boundaries in overly large segments. In preparation for interactive operation text indices are built for both shot-level and story-level segmentations using Lucene [9] (for keyword search) and our latent semantic indexing system (for fuzzy text search) Color correlograms [10] are pre-computed for each shot thumbnail image.

3.4 Search Engine

Queries are specified by a combination of text and images. The searcher can opt to perform a textonly or image-only search by leaving the image or text query area empty. The searcher can choose an exact keyword text search, a latent semantic analysis (LSA) based text search, or a combination of the two whereby the keyword and LSA-based retrieval scores are averaged together to form a combined score. We use only the provided ASR transcript to provide text for story and shot segments. The exact text search is based on a Lucene [9] back end and ranks each story based on the tfidf values of the specified



condoleeza rice

50

☑ Show related words

☑ Show segment text

Text matching: ○ Fuzzy ● Mixed ○ Exact

Find shots of Condoleeza Rice

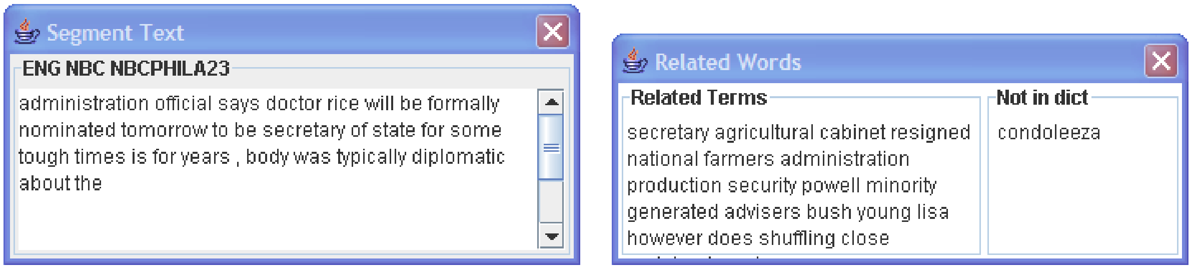
13:57

11/15/04 23:00 NBC NBCPHILA23 ENG

11/15/04 11p NBC 1:39 [7:55] ***0.4***
covey ama welders clue fifteenth **national administration resigned secretary rice**

Figure 6: Interactive system interface. (A) Story keyframe summaries in the search results (B) Search text and image entry (C) TRECVID topic display (D) Media player and keyframe zoom (E) Story timeline (F) Shot keyframes (G) Relevant shot list

keywords. In this mode the story relevance, used for results sorting and thumbnail scaling and color coding as described in following sections, is determined by the Lucene retrieval score. When the LSA based search is used [7], the query terms are projected into a latent semantic space (LSS) of dimension 100 and scored in the reduced dimension space against the text for each story and each shot using a cosine similarity function. In this mode, the cosine similarity value determines the query relevance score. In our application the LSS was built treating the text from each story segment (determined as described in Section 3.3 as a single document. When determining text-query relevance for shots, each shot gets the average of the retrieval score based on the actual shot text and the retrieval score for its parent story. That is, the shots inherit text relevance from their stories. An image similarity matching capability is provided based on color correlograms [10]. Any shot thumbnail in the interface can be dragged into the query bar (Figure 6 B)and used as part of the query. For each shot thumbnail the color correlogram is compared to the correlogram for every shot thumbnail in the corpus. To generate an image-similarity



Segment Text

ENG NBC NBCPHILA23

administration official says doctor rice will be formally nominated tomorrow to be secretary of state for some tough times is for years , body was typically diplomatic about the

Related Words

Related Terms

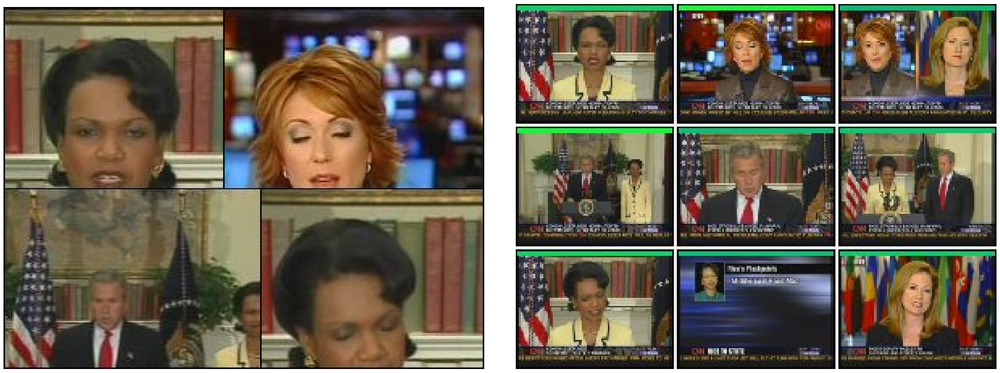
secretary agricultural cabinet resigned
national farmers administration
production security powell minority
generated advisers bush young lisa
however does shuffling close

Not in dict

condoleeza

(a) (b)

Figure 7: Transcript feedback dialog windows. (a) shows transcript text for the selected shot or story and (b) shows terms related to the query and terms not in the dictionary. Note that 'Condoleeza' does not appear in the dictionary because it is never correctly transcribed in the data.



(a) (b)

Figure 8: A story summary quad (a) and the complete set of 9 shots contained in the story. The search target was condi Rice (the text query was "lisa rice")

relevance score at the story level, the maximum score from the component shots is propagated to the story level. The document scores from the text search are combined with document scores from the image similarity to form a final overall score by which the query results are sorted. A query returns a ranked list of stories.

3.5 Concept Similarity

As part of our high-level feature detection effort (see section 2) we had available 'concept' (high-level feature) probability vectors for 29 of the concepts in the light weight ontology for every shot in the testing set. That is, each shot has an associated 29 element vector describing the probability of occurrence of 29 of the high-level features. Table 4 shows which features were included in this vector. These were incorporated into the interactive portion and the post-processing portion of the system.

During interactive operation the user can choose to "find similar" shots based on a set of selected shots. This action uses the same components that are used at the end of the interactive session. Two

animal boat_ship building car charts crowd explosion_fire face flag_us government_leader maps military mountain natural_disaster outdoor people_marching person police_security prisoner road sky sports studio urban vegetation walking_running waterscape_waterfront weather

Table 4: The high-level features included in the concept vector for each shot.

similarity measures are combined; one between the text of the selected segment(s) and those of candidate stories, and one between the concept vectors the selected segments and those of candidate stories. The text-similarity is the cosine distance between the text of the selected segment(s) and the text of each candidate segment. The concept distance is the minimum euclidean distance between the concept vectors of the example shots and the concept vectors of each candidate segment. The two similarity scores are averaged together to create a similarity score for each candidate segment.

3.6 Interface Elements

The interactive search system is pictured in Figure 6. The TRECVID test question and supporting images are shown in section C. Text and image search elements are entered by the searcher in section B. Search results are presented as a list of story visualizations in section A. A selected story is shown in the context of the video from which it comes in section E and expanded into shot thumbnails in section F. When a story or shot icon is moused-over an enlarged image is shown in section D. When a video clip is played it is also shown in section D. User selected shot thumbnails are displayed in section G.

3.6.1 Thumbnails

Shots are visualized with thumbnails made from the primary keyframe drawn from the reference shot segmentation. Story thumbnails are built in a query-dependent way. The 4 shot thumbnails that score highest against the current query are combined in a grid. The size allotted to each portion in this 4-image montage is determined by the shots score relative to the query. Figure 8 shows an example of this where the query was "Lisa Rice" ("Lisa" was a common mis-recognition of the name "Condoleeza") and the shots most relevant to the query are allocated more room in the story thumbnail.

3.6.2 Overlays

Semi-transparent overlays are used to provide 3 cues. A gray overlay on a story icon indicates that it has been previously visited (see Figure 6 A and E). A red overlay on a shot icon indicates that it has been explicitly excluded from the relevant shot set (see Figure 6 F). A green overlay on a shot icon indicates that it has been included in the results set (see Figure 6 F). Horizontal colored bars are used along the top of stories and shots to indicate the degree of query-relevance, varying from black to bright green. The same color scheme is used in the timeline depicted in Figure 6 D.

3.6.3 Transcript Dialogs

An addition to the interface in 2005 year are two optional dialogs pictured in Figure 7 intended to provide information about the underlying transcript and text query operation. One dialog shows the transcript from the selected shot or story and the other shows terms related to the query (determined from the latent semantic space) and query terms that are not contained in the dictionary.A

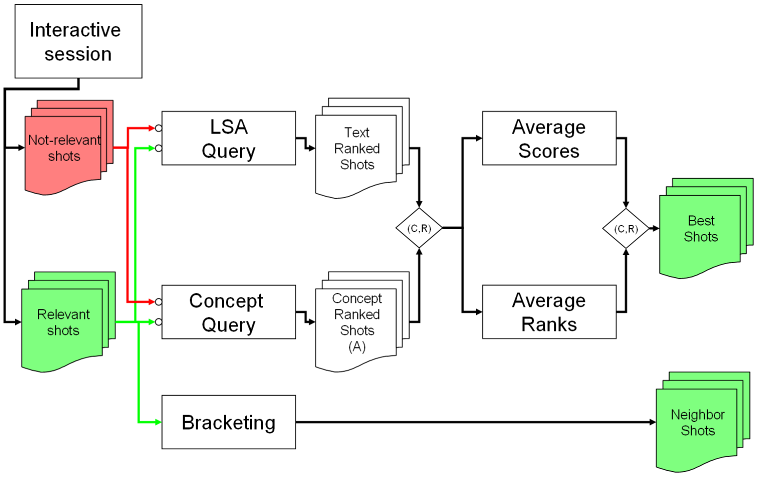


Figure 9: Process flow for the various methods of performing queries to augment the shots identified by the user during the interactive session.

3.7 Post-Interactive Processing

When the searcher decides to end the task by pressing the `end question` button or when the 15 minute allotted time expires, the search system uses 7 different methods to perform an automated search process to fill out the remaining slots in the 1000 shot result list. This process is illustrated in the flow diagram of Figure 9. The MAP performance of the user-identified shots with no automated query to fill out the results is shown in Figures 3 and 4.

Three methods are used to identify and rank candidate shots for the post-interactive portion of the system operation.

- The transcript text from the shots marked relevant is used to form a text query which is passed to the LSA-based text query.
- The concept vectors from the shots marked relevant are used to rank the remaining candidate shots.
- The shots neighboring (or bracketing) the user-identified relevant shots are added to the result list even if they were marked as not-relevant by the user. The MAP performance contribution of this step can be seen in Figures 3 and 4 and is roughly one third of the total difference between the user-identified shots alone and the best performing system (LCN).

3.7.1 LSA-based Similarity

In this method the text from the shots that have been judged by the searcher to be relevant is combined to form a single LSA-based text query. This query is applied to the unjudged shots and the highest scoring ones retained for the result list.

3.7.2 Concept Query

In this method the concept vector of a shot is compared against the concept vectors of the marked relevant and not-relevant shots. For each group (relevant, not-relevant) the minimum euclidean distance is computed, yielding a positive and negative similarity measure for each candidate shot.

3.7.3 Combining Measures

The concept similarity measure(s) and the text similarity measures were either averaged with equal weighting (systems LCN LN LC) or the candidate shots were sorted by each measure (text, positive concepts, negative concepts) and their ranks in each sorted list averaged together (systems RAN RPN RA RP) to form a final ordering from which to select likely shots. In all cases, if bracketing is used (systems LCN LN RAN RPN) the bracketed shots (those shots immediately adjacent to all shots marked relevant by the searcher) are included in the results immediately following the user-selected shots.

3.8 Future Work

In 2005 we integrated the high-level features into the search system in a fairly ad-hoc manner. Moving forward we'll apply more principled methods to determine the best measures and weightings for using concept-vector distance in the retrieval framework. Also in 2006 we hope to increase the number of test subjects and submit complete topic answers from more than one set of users.

References

[1] M. Cooper. Video Segmentation Combining Similarity Analysis and Classification. *Proc. ACM Multimedia*, 2004.

[2] J. Adcock, A. Girgensohn, M. Cooper, T. Liu, E. Rieffel, and L. Wilcox. FXPAL Experiments for TRECVID 2004. *Proceedings of TRECVID 2004*, 2004.

[3] T.Liu, A. W. Moore, A. Gray. Efficient Exact k-NN and Nonparametric Classification in High Dimensions. *Proc. of Neural Information Processing Systems(NIPS 2003)*, 2003.

[4] D. Fradkin and D. Madigan. Experiments with random projections for machine learning. *Proc. ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2003.

[5] N. Vasconcelos and M. Vasconcelos. Scalable Discriminant Feature Selection for Image Retrieval and Recognition. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2004

[6] C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot Boundary Detection System *Proceedings of TRECVID 2004*, 2004.

[7] Michael W. Berry, Susan T. Dumais and Gavin W. O'Brien Using linear algebra for intelligent information retrieval *SIAM Review, v.37 n.4, p.573-595*, Dec. 1995

[8] M. F. Porter An algorithm for suffix stripping *Program*, 14(3):130–137, 1980.

[9] Jakarta Lucene. http://jakarta.apache.org/lucene/docs/index.html.

[10] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms *In Proc. IEEE Comp. Soc. Conf. Comp. Vis. and Patt. Rec.*, pages 762–768, 1997.

[11] S. Kumar and M. Hebert. Discriminative Fields for Modeling Spatial Dependencies in Natural Images. *Advances in Neural Information Processing Systems, NIPS 16*, 2004.