# FXPAL at TRECVID 2006

Matthew Cooper, John Adcock, and Francine Chen
FX Palo Alto Laboratory
Palo Alto, CA 94304
{last name}@fxpal.com

In 2006 FXPAL submitted results for 3 tasks: shot boundary detection (section 1, page 1), high-level feature extraction (section 2, page 4), 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 2006 builds on the framework and system developed in 2004 and 2005 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.

Last year our performance was worse than anticipated, as our training data was not an accurate reflection of the test data for the videos from LBC and CCTV. On the remaining videos, our performance was very good, and consistent with our training experiments. As a result, this year we submitted experimental runs using machine generated output from the master shot reference of the development set to label the training data used for classification. Our results were reasonably good, but remained below expectations and previous performance using manually generated training sets. Further analysis revealed that errors in our frame decoding software were responsible for most of this performance loss.

### 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 $4 \times 4$ spatial grid. Denote the frame indexed feature vectors $\mathbf{X} = \{X(n) : n = 1, \cdots, N\}$ for $N$ frames.

**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 $D_\chi$ 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 and 2005. 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 and 2005. 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. This system was also used in 2005. These are the sys10M_0X runs.

The only difference for the 2006 submissions was the construction of training sets combining machine-generated and manually labeled examples. For all runs, the training set was generated as follows. We include all cut and gradual frames from the 2005 shot boundary test set. We also include a random subset of the non-transition frames by randomly selecting ten percent of those frames. Additionally, we extract boundary frames and one percent of the non-boundary frames from videos by LBC and CCTV in the development data set per the master shot reference. We then cluster these frames using k-means clustering ($k = 9$), and retain the clusters whose labels are at least 90% boundaries. These are added to the training set as cut frames.

In limited experiments on the 2005 test set, we used this approach to combine manually labeled training data from the 2004 test set with clustered data from the 2005 development set. Table 1 shows the results averaged across the 2006 submitted systems in these experiments.

| Shot boundary detection results | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Mean | | | Cut | | | Gradual | | |
| **SYS** | R | P | F | R | P | F | R | P | F |
| FXPAL MEAN | 0.8385 | 0.8507 | 0.8441 | 0.8623 | 0.8651 | 0.8634 | 0.7688 | 0.8077 | 0.7864 |

Table 1: Table summarizing shot boundary detection results in training using the 2005 test set in terms of recall (R), precision (P), and the f-score (F).

| Shot boundary detection results | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Mean | | | Cut | | | Gradual | | |
| **SYS** | R | P | F | R | P | F | R | P | F |
| LAG10-6 | 0.735 | 0.736 | 0.7355 | 0.771 | 0.754 | 0.7624 | 0.639 | 0.681 | 0.6593 |
| LAG05-7 | 0.742 | 0.798 | 0.7690 | 0.789 | 0.803 | 0.7960 | 0.615 | 0.78 | 0.6877 |
| LAG05-8 | 0.721 | 0.822 | 0.7682 | 0.772 | 0.82 | 0.7953 | 0.584 | 0.828 | 0.6849 |
| LAG05-6 | 0.751 | 0.778 | 0.7643 | 0.798 | 0.79 | 0.7940 | 0.626 | 0.738 | 0.6774 |
| LAG10MI90-7 | 0.721 | 0.78 | 0.7493 | 0.765 | 0.784 | 0.7744 | 0.601 | 0.765 | 0.6732 |
| LAG10MI90-8 | 0.691 | 0.806 | 0.7441 | 0.739 | 0.804 | 0.7701 | 0.56 | 0.812 | 0.6629 |
| LAG05-5 | 0.76 | 0.753 | 0.7565 | 0.804 | 0.779 | 0.7913 | 0.643 | 0.679 | 0.6605 |
| LAG10-8 | 0.706 | 0.795 | 0.7479 | 0.734 | 0.797 | 0.7642 | 0.631 | 0.79 | 0.7016 |
| LAG10-7 | 0.728 | 0.764 | 0.7456 | 0.756 | 0.774 | 0.7649 | 0.651 | 0.734 | 0.6900 |
| LAG10MI90-6 | 0.732 | 0.752 | 0.7419 | 0.779 | 0.761 | 0.7699 | 0.606 | 0.723 | 0.6593 |
| FXPAL AVG. | 0.7287 | 0.7784 | 0.7527 | 0.7707 | 0.7866 | 0.7786 | 0.6156 | 0.753 | 0.6774 |
| TV MEAN | 0.6675 | 0.6880 | 0.6776 | 0.7281 | 0.7205 | 0.7243 | 0.5030 | 0.5893 | 0.5428 |
| FXPAL REVISED | 0.8182 | 0.8634 | 0.8398 | 0.8723 | 0.9112 | 0.8910 | 0.6723 | 0.7322 | 0.6989 |

Table 2: 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 $\kappa$ used. The bottom row shows revised results using corrected video decoding software.

## 1.3 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 $\kappa$ of its nearest neighbors are boundaries. We then vary $\kappa$ between 0 and $k$.

Unfortunately, the results on the 2006 test set, shown in Table 2, again show significant discrepancies with the results of our training experiments. We determined in post-evaluation analysis that our feature extraction code was dropping frames while decoding the input videos[1]. We present updated results with corrected decoding software below.

## 1.4 Updated Results

Figure 1 shows performance curves for cut and gradual boundary detection for the SB06 test data using training data from the SB05 test set. It is remarkable that the LAG10RP system produces the best cut boundary detection performance. This is a reflection of the high number of 3 frame cut transitions in the 2005 and 2006 test data. As a result, the LAG10MI90 system optimized on the 2003 test data containing mostly 2 frame cut transitions performs worse. These two data sets are largely from the same content providers, although the 2005 data includes NASA documentaries absent in 2006. Table 3 includes revised results on the evaluation data.

Table 3 demonstrates that the decoding problems degraded boundary detection performance in 2005 and 2006. Additionally, it is clear that the feature subsets of the LAG10MI90 system trained for the content used in 2003 and 2004 do not produce the same performance improvements on the 2005 and 2006 data.

[1]Curiously, the erroneous software correctly processes the 2003 and 2004 test data.

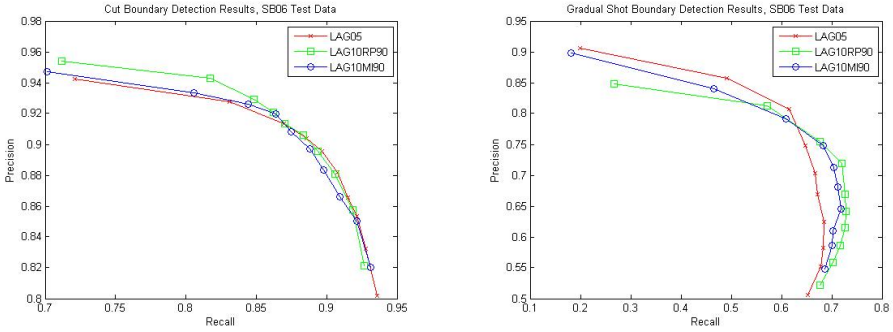


Figure 1: Performance of shot boundary systems on SB06 data: cut boundary detection (left) and gradual boundary detection (right).

---

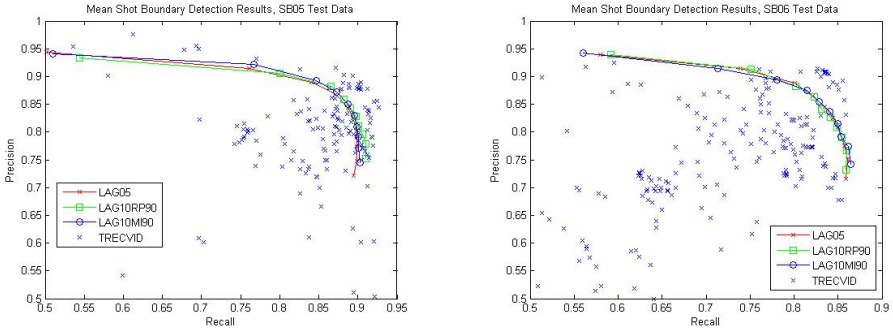


Figure 2: Visualizing the performance of FXPAL shot boundary systems relative to other TRECVID participants: mean performance for SB05 (left) and SB06 (right).

---

# 2 High-level feature detection

## 2.1 Summary of submitted runs

We continued our work on high level feature extraction using probabilistic graphical models [6, 7]. We submitted a baseline run (B_AL-06Beta-4) using the single concept SVM models provided by the MediaMill team [8]. We also submitted five runs based on different variations on conditional random field models for collective concept detection. These collective model runs did worse than the baseline, and we are continuing our post-evaluation analysis to better understand the causes for this drop in performance.

## 2.2 Per shot pre-processing

This year, we relied on the MediaMill data for a pre-processed low-level feature data and for SVM outputs for each concept using both visual and textual features. To create our baseline run, we used the SVM output for the modality with the best performance on a held-out data set for each concept

| FXPAL-SB05 SUBMISSIONS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | MEAN | | | CUT | | | GRADUAL | | |
| | R | P | F | R | P | F | R | P | F |
| AVG. | 0.8374 | 0.8252 | 0.8313 | 0.8591 | 0.8593 | 0.8592 | 0.7735 | 0.7312 | 0.7518 |
| FXPAL-SB05 REVISED | | | | | | | | | |
| | MEAN | | | CUT | | | GRADUAL | | |
| | R | P | F | R | P | F | R | P | F |
| AVG. | 0.8791 | 0.8547 | 0.8662 | 0.9115 | 0.9222 | 0.9166 | 0.7842 | 0.6864 | 0.7303 |
| FXPAL-SB06 SUBMISSIONS | | | | | | | | | |
| | MEAN | | | CUT | | | GRADUAL | | |
| | R | P | F | R | P | F | R | P | F |
| AVG. | 0.7287 | 0.7784 | 0.7527 | 0.7707 | 0.7866 | 0.7786 | 0.6156 | 0.753 | 0.6774 |
| FXPAL-SB06 REVISED | | | | | | | | | |
| | MEAN | | | CUT | | | GRADUAL | | |
| | R | P | F | R | P | F | R | P | F |
| AVG. | 0.8182 | 0.8634 | 0.8398 | 0.8723 | 0.9112 | 0.8910 | 0.6723 | 0.7322 | 0.6989 |

Table 3: Table comparing submitted and revised results for TRECVID SB05 and SB06. The "R", "P", and "F" columns list values for recall, precision, and F-score, respectively.

| Feature detection results | | |
|---|---|---|
| **SYS** | **MAP** | **Notes** |
| B_AL-06Beta_4 | 0.07255 | baseline |
| B_AL-06Beta_3 | 0.061 | $\chi^2$ cliques, 101 concepts, CML+I |
| B_AL-06Beta_1 | 0.05915 | $\chi^2$ cliques, 39 concepts, CML+I |
| B_AL-06Beta_2 | 0.0591 | $\chi^2$ cliques, 39 concepts, CMLT+I |
| B_AL-06Beta_6 | 0.0587 | LRT cliques, 101 concepts, CMLT+I |
| B_AL-06Beta_5 | 0.05735 | LRT cliques, 39 concepts, CML+I |
| MEDIAN | 0.0755 | |

Table 4: Table summarizing the high-level feature detection results in terms of mean-averaged precision (MAP). The bottom column shows median TRECVID results. The notes column indicates the similarity measure by which cliques were formed, the number of concepts from which cliques were formed, and the interaction model applied.

directly.

## 2.3 Information fusion via random fields

The remaining runs were based on the discriminative random field (DRF) model of [9]. This model combines discriminative classifiers with pairwise interaction representing contextual information. We adapted this approach to the semantic labeling context. The corresponding probabilistic model for the binary concept vector $\mathbf{y}$ given the observed features $\mathbf{x}$ is :

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

As before, we use logistic classifiers for the association terms $A_i(y_i, \mathbf{x})$, in (2). Thus we need to identify which concepts are related, i.e. which concepts are connected by an edge in our graph. For three of the runs (B_AL-06Beta_1, B_AL-06Beta_2, B_AL-06Beta_3), 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 clique. These cliques define the neighborhoods $\mathcal{N}_i$ for each label $y_i$. This approach allows us to jointly infer all labels in $\mathbf{y}$ simultaneously in hopes of exploiting inter-label dependencies. For the remaining two runs we constructed a similar graph using a likelihood ratio test rather than the chi-squared test (B_AL-06Beta_5,B_AL-06Beta_6). The other difference in the graphs we built were the number of concepts used. For three of the graphs, we only considered concepts among the 39 selected for the evaluation (B_AL-06Beta_1, B_AL-06Beta_2, B_AL-06Beta_5). For two of the graphs, we greedily used a subset of the 101 features considered in the MediaMill set [8] to construct the graph, including only features most significantly related to those 39 in the evaluation (B_AL-06Beta_3, B_AL-06Beta_6).

In contrast to [9], we wanted to include case by case interaction terms. Comparing this equation to (2), we use a linear form for the interaction term:

$$I(y_i, y_j, \mathbf{x}) = \sum_{k \in \mathcal{K}} \lambda_{ij}^{(k)} f_{ij}^{(k)}(y_i, y_j, \mathbf{x}) = \mathbf{\Lambda_{ij}}^T \mathbf{F_{ij}}(y_i, y_j, \mathbf{x}) \quad . \tag{3}$$

Below, we present two models for the interaction terms.

### 2.3.1 The CML+I model

The first model, denoted CML+I, captures inter-concept co-occurrence. These features are defined for each pair of concepts $y_i, y_j$ that are connected in our graph (i.e. not for all pairs). Thus, we have the indexed family of interaction potential functions:

$$\begin{aligned}
f_{ij}^{(0)}(y_i, y_j, \mathbf{x}) &= \begin{cases} 1 & y_i = y_j = 0 \\ 0 & \text{otherwise} \end{cases} \\
f_{ij}^{(1)}(y_i, y_j, \mathbf{x}) &= \begin{cases} 1 & y_i = 1, y_j = 0 \\ 0 & \text{otherwise} \end{cases} \\
f_{ij}^{(2)}(y_i, y_j, \mathbf{x}) &= \begin{cases} 1 & y_i = 0, y_j = 1 \\ 0 & \text{otherwise} \end{cases} \\
f_{ij}^{(3)}(y_i, y_j, \mathbf{x}) &= \begin{cases} 1 & y_i = y_j = 1 \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{4}$$

By modeling the four possible combinations separately, we hope to capture all types of inter-concept co-occurrence within the model. For this model the index set for the interaction potentials is simply $\mathcal{K} = \{0, 1, 2, 3\}$. This interaction model was proposed in [10] for text categorization.

### 2.3.2 The CMLT+I model

We define a second model to capture concept-feature co-occurrence, combining ideas from [10] and [11]. We first quantize the low level visual features $\mathbf{x}$ across the training set using k-means. Denote this discrete representation for the low-level features as $\mathcal{Q}(\mathbf{x}) \in \{0, \cdots, Q\}$. For this model, denoted

CMLT+I, we define interaction potentials:

$$
\begin{aligned}
f_{ij}^{(0,q)}(y_i, y_j, \mathbf{x}) &= \begin{cases} 1 & y_i = y_j = 0, \mathcal{Q}(\mathbf{x}) = q \\ 0 & \text{otherwise} \end{cases} \\
f_{ij}^{(1,q)}(y_i, y_j, \mathbf{x}) &= \begin{cases} 1 & y_i = 1, y_j = 0, \mathcal{Q}(\mathbf{x}) = q \\ 0 & \text{otherwise} \end{cases} \\
f_{ij}^{(2,q)}(y_i, y_j, \mathbf{x}) &= \begin{cases} 1 & y_i = 0, y_j = 1, \mathcal{Q}(\mathbf{x}) = q \\ 0 & \text{otherwise} \end{cases} \\
f_{ij}^{(3,q)}(y_i, y_j, \mathbf{x}) &= \begin{cases} 1 & y_i = y_j = 1, \mathcal{Q}(\mathbf{x}) = q \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{5}
$$

The index set for the interaction potentials is $\mathcal{K} = \{(i, q) : 0 \leq i \leq 3, 0 \leq q \leq Q\}$.

### 2.4 Summary

Results are summarized in Table 4. The performance of these systems was relatively disappointing, and we believe this is because we did not approach model induction with sufficient care. We can see that the use of the chi-squares inter-concept similarity measure outperformed the likelihood ratio test for graph induction. The use of the full 101 available concepts for graph induction also seems to provide a performance improvement. There are clearly unresolved bugs in our system that prevent us from drawing any strong conclusions at this point. We are continuing our fault analysis of the submitted systems and hope to provide more complete results in subsequent papers.

## 3 Interactive Search

### 3.1 Summary of submitted runs

The interactive search system we used in 2006 is evolutionaly different from our 2005 system. We submitted 6 runs for the interactive search task. Our 3 searchers had greatly varying experience with the system and completed 4, 7, and 13 topics in order from least to most experienced respectively. 6 different methods were used to augment the list of user-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. FXPAL1LN: LSA text query with bracketing
2. FXPAL2LNC: Combined LSA text query and concept-based ranking with bracketing
3. FXPAL3UN: Lucene text query with bracketing
4. FXPAL4UNC: Combined Lucene text query and concept-based ranking with bracketing
5. FXPAL5LNP: Combined LSA text query and concept-based ranking using only positive concept examples with bracketing
6. FXPAL6UNP: Combined Lucene text query and concept-based ranking using only positive concept examples with bracketing

All runs were fully interactive, type B (due to the user of the MediaMill high level features[8]), condition 1. The system names are loosely acronymed with the following decoding:

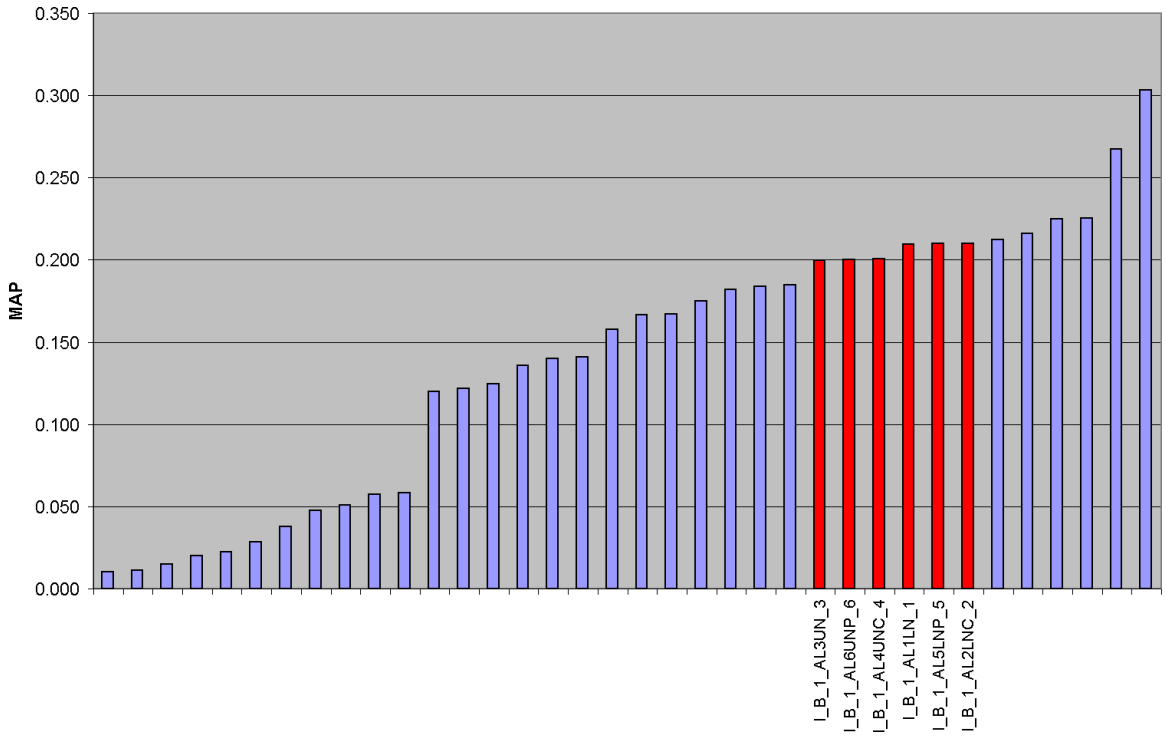


Figure 3: MAP performance of all interactive search submissions with FXPAL submissions shown in red.

---

**L** Latent Semantic Analysis (LSA) based text query/similarity scoring used
**U** Keyword (Lucene) text query/similarity scoring used
**N** Neighboring shots are included in the results
**C** Concept vectors (high level features) are used in measuring shot similarity
**P** Concept similarity is measured using only positive examples

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

| System | MAP score | Notes |
|---|---|---|
| FXPAL 2 LNC | .2101 | Post-processing same as 2005 system LNC |
| FXPAL 5 LNP | .2100 | |
| FXPAL 1 LN | .2095 | Post-processing same as 2004 text-only system LSA1 and 2005 system LN |
| FXPAL 4 UNC | .2005 | |
| FXPAL 6 UNP | .2000 | |
| FXPAL 3 UN | .1995 | |

Table 5: MAP scores for the 6 systems in performance order. MAP scores shown to 4 digits to expose the differences.

---

In 2006 we submitted the interactive run with the 7th highest MAP score behind submissions from 3 other groups. This is similar to our relative performance in 2004 and 2005. Figure 3 shows the

MAP performance of the FXPAL runs against the entire set of interactive search submissions. This year we had 1 searcher session for each topic so all variation in our runs is from system differences. The MAP performance between different systems is very small. The range within the two clusters of 3 submissions less than .001 and the total range from best to worst MAP is only 0.0106. Table 3.1 lists the runs in performance order. From this ordering It can be observed that the concept-enhanced systems outperform the text-only systems and the systems using both positive and negative examples outperform those using only positive examples. We have yet to perform statistical significance analysis on these results but similarly small differences in 2005 proved to have a high level of statistical significance.

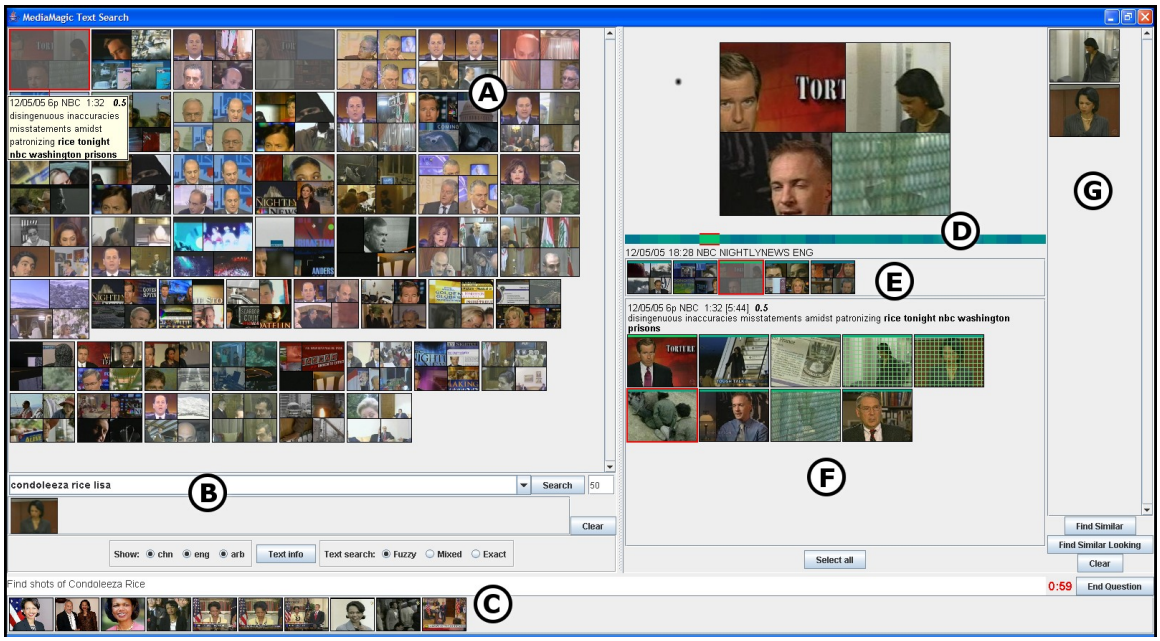


Figure 4: Interactive system interface. (A) Story keyframe summaries in the search results (B) Search text and image entry (C) TRECVID topic display (D) Media player and keyframe zoom (E) Story timeline (F) Shot keyframes (G) Relevant shot list

## 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 and 2005 systems [2, 6].

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 lists. We use a 2 level video segmentation with an automatically generated story-level segmentation supplementing the reference shot segmentation[12]. Text-based latent semantic analysis (LSA) of the transcripts is used to build the story-level segmentation. User text searches are performed with exact


Figure 5: Transcript feedback dialog window. Shows transcript text for the selected shot or story with query terms and related terms highlighted. Note that 'Condoleeza' does not appear in the dictionary because it is never correctly transcribed.

---


(a) (b)

Figure 6: A story summary quad (a) and the complete set of 9 shots contained in the story. The search target was Condoleeza Rice (the text query was "condoleeza lisa rice")

---

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. 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 [12]. 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 [13] 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 [14] (for keyword search) and our latent semantic indexing system (for fuzzy text search) Color correlograms [15] 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 [14] back end and ranks each story based on the tfidf values of the specified 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 [16], 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 [15]. Any shot thumbnail in the interface can be dragged into the query bar (Figure 4 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 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

We take advantage of the 101 concepts provided by the University of Amsterdam[8] to provide a foundation for concept-based similarity measurements. Each shot has an associated 101 element vector describing the likelihood of each of the high-level features. For the concept distance between two shots we use the mean absolute distance (normalized L1) between their concept vectors.

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

New in 2006 the user can alternatively choose to perform a "find similar looking" operation on a set of selected shots. In this operation the selected shots are used to perform an image-based search. It is equivalent to putting all the selected shots in the image-query area and clearing the text search box, and thus isn't extending the capabilities of the system but rather provides a significant shortcut.

## 3.6 Interface Elements

The interactive search system is pictured in Figure 4. 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 where the searcher can also choose which languages include in the results. 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 6 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 4 A and E). A red overlay on a shot icon indicates that it has been explicitly excluded from the relevant shot set (see Figure 4 F). A green overlay on a shot icon indicates that it has been included in the results set (see Figure 4 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 4 D.

### 3.6.3 Transcript Dialogs

An optionally displayed dialog is pictured in Figure 5. This provides information about the underlying transcript and text query operation. The dialog shows the transcript from the selected shot or story along with terms related to the query (determined from the latent semantic space) and query terms that are not contained in the dictionary. Query term and related terms are highlighted in the transcript pane, and transcript terms are highlighted in the related terms pane.

## 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 employs 6 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 7.

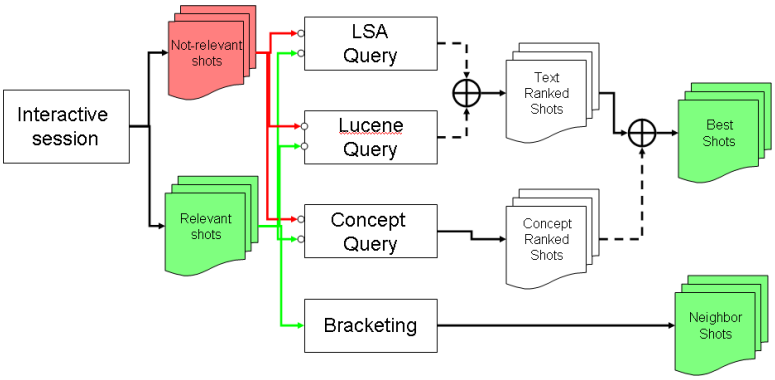


Figure 7: Process flow for the various methods of performing queries to augment the shots identified by the user during the interactive session.

---

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 engine.
- The transcript text from the shots marked relevant is used to form a text query which is passed to the Lucene text query engine.
- 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. We have not yet evaluated the contribution of neighboring shots in 2006, but in previous years the use of neighboring shots has provided a significant boost in MAP scores.

### 3.7.1 LSA-based Text 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 Lucene-based Text 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 Lucene text query. This query is applied to the unjudged shots and the highest scoring ones retained for the result list.

### 3.7.3 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 distance is computed, yielding a positive and negative similarity measure for each candidate shot.

### 3.7.4 Combining Measures

The concept similarity measure(s) and the text similarity measures were averaged with equal weighting to form a final ordering from which to select likely shots. In all cases in 2006 bracketing is used and 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 and before the highest ranked unjudged shots.

# 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] M. Cooper, J. Adcock, H. Zhou, and R. Chen FXPAL Experiments for TRECVID 2005. *Proceedings of TRECVID 2005*, 2005.

[7] R. Yan, M.-Y. Chen and A. Hauptmann. Mining Relationship between Video Concepts using Probabilistic Graphical Model. *Proc. IEEE International Conference On Multimedia and Expo (ICME)*, 2006.

[8] Cees G.M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W.M. Smeulders. The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia. *In Proceedings of ACM Multimedia*, Santa Barbara, USA, October 2006.

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

[10] N. Ghamrawi, and A. McCallum. Collective multi-label classification. *Proceedings of the 14th ACM international Conference on information and Knowledge Management CIKM '05*. ACM Press, New York, NY, p. 195-200, 2005.

[11] J. Jeon, V. Lavrenko, R. and Manmatha. Automatic image annotation and retrieval using cross-media relevance models. *Proceedings of the 26th Annual international ACM SIGIR Conference on Research and Development in informaion Retrieval*. ACM Press, New York, NY, 119-126, 2003.

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

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

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

[15] 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.

[16] 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