Imperial College at TRECVID

Rui Jesus[*], João Magalhães[+], Alexei Yavlinsky[+], Stefan Rüger[+]

(rjesus@deetc.isel.ipl.pt, j.magalhaes@imperial.ac.uk, a.yavlinsky@imperial.ac.uk, s.rueger@imperial.ac.uk)

[+]Department of Computing
Imperial College London, South Kensington Campus
London SW7 2AZ, UK

[*]Departamento de Engenharia Electrónica
Telecomunicações e Computadores
Instituto Superior de Engenharia de Lisboa
Lisbon, Portugal

Abstract

We describe our experiments for the shot boundary detection, high-level feature extraction and search. In the shot boundary detection task, we employ our proven method based on the calculation of distances between colour histograms of frames over a range of timescales. For the search task, we tested a different system. This year, content based search is complemented with a new relevance feedback method. Results of one interactive run are presented to evaluate the performance of the new system. In the high-level feature detection task we tested two new methods: naïve model and non-parametric density estimation. We evaluated these models with all keywords.

1 Introduction

The retrieval system we use for the search tasks is functionally similar to that used for TRECVID 2004 (Heesch et al., 2004) but with a different relevance feedback method. The search process is based on global and tiled image features. Like last year, we use colour and texture features but we have not modelled the motion aspect in video. One interactive run is carried out to test whether the use of the new relevance feedback method improves performance of the search.

In the high-level feature detection task we tested two new methods: naïve model and nonparametric density estimation. The set of low-level features we used are similar to the ones we use for the search task.

The paper is structured as follows. Section 2-4 describes, respectively, the experiments and results for the shot boundary detection task, the search task and the high-level feature detection task.

2 Shot-Boundary Detection Task

2.1 System

The video shot boundary detection algorithm is broadly based on the colour histogram method. The colour histograms for consecutive frames are compared and a shot change is declared if their difference is greater than a given threshold. This method is extended based on the algorithm of (Pye, Hollinghurst, Mills, and R.Wood, 1998) for detection of gradual transitions that take place over a number of frames, and for rejection of transients such as the effect of a flash-bulb. In order to determine the start and end points for gradual transitions, we employ a method similar to that described by (Zhang, Kankanhalli, and Smoliar, 1993), in which a lower threshold is used to test for the start and end of a gradual transition. Our system is largely unchanged from the last three years, and more details can be found in (Pickering, Heesch, O'Callaghan, Rueger, and Bull, 2002).

2.2 Experiments and Results

We performed ten shot boundary detection runs. The first four runs, Imperial-01 - Imperial-04 were carried out keeping the low threshold, T4, constant, and reducing the high threshold, T16. In runs Imperial-05 - Imperial-08, the low threshold was increased and the same 3 values for the high threshold were used again. The last two runs, Imperial-09 and Imperial-10 used a value for the low threshold falling between the first set of runs and the second set, and a high and low value respectively for the high threshold. The thresholds for these last two runs were determined empirically from previous year's TREC results.

	All		Cuts		Gradual			
	Recall	Prec	Recall	Prec	Recall	Prec	F-Recall	F-Prec
Imperial_01	0,755	0,815	0,830	0,813	0,536	0,822	0,907	0,223
Imperial_02	0,766	0,803	0,831	0,803	0,577	0,800	0,906	0,237
Imperial_03	0,770	0,784	0,828	0,788	0,601	0,768	0,905	0,238
Imperial_04	0,769	0,739	0,825	0,752	0,604	0,692	0,905	0,241
Imperial_05	0,804	0,789	0,856	0,805	0,655	0,731	0,858	0,641
Imperial_06	0,818	0,766	0,856	0,793	0,705	0,683	0,846	0,656
Imperial_07	0,826	0,733	0,855	0,769	0,740	0,632	0,839	0,657
Imperial_08	0,826	0,690	0,852	0,725	0,748	0,595	0,841	0,610
Imperial_09	0,806	0,779	0,847	0,794	0,686	0,727	0,884	0,480
Imperial_10	0,815	0,749	0,846	0,774	0,725	0,675	0,878	0,487
TRECVID Median	0,851	0,796	0,922	0,842	0,701	0,660	0,772	0,717

Table 1 - Shot boundary detection task - results summary.

We show the results for our ten shot-boundary detection runs in Table 1. There was a predictable trade-off between recall and precision as the thresholds were altered. Runs 5, 6 and 9 gave particularly good points at the top right of the precision-recall graphs.

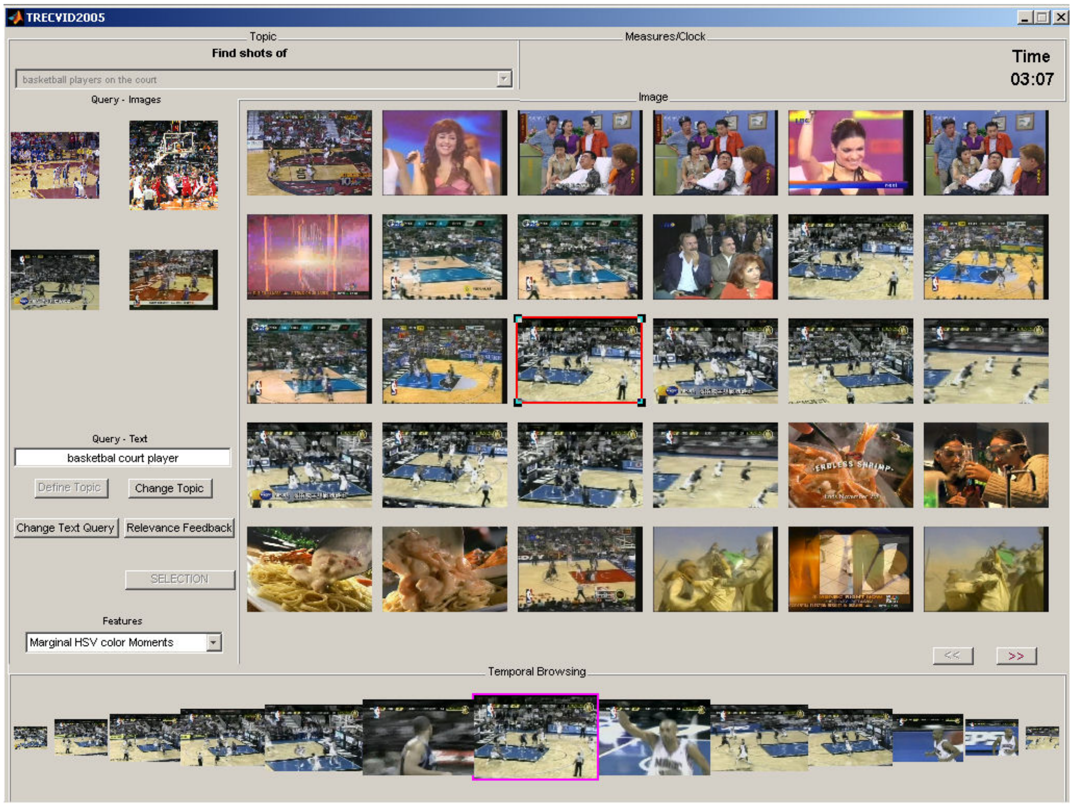
In addition we measured the clock time for each run to get an indication of the efficiency. A run consisted of processing all of the 12 test files by a single system variant. The only differences between each of the runs were the threshold settings. Therefore the total time for each run was the same, approximately 20000 seconds. The timing runs were performed on a Pentium 4, 3.00GHz.

3 Search Task

3.1 Graphical user interface

We have consolidated the paradigms of text-based search, content-based search with relevance feedback and temporal browsing into a unified interface. By providing tight integration of techniques and a rich set of user interactions, we aimed to equip the user with substantial navigational power.

Figure 1 shows the interface layout with one image selected in the search panel. This image is displayed with a red border. The user formulates a query using the left hand panel. Below we outline the stages of the searching process:



Topic – Find shots of: basketball players on the court

Measures/Clock – Time 03:07

Query - Text: basketbal court player

Features: Marginal HSV color Moments

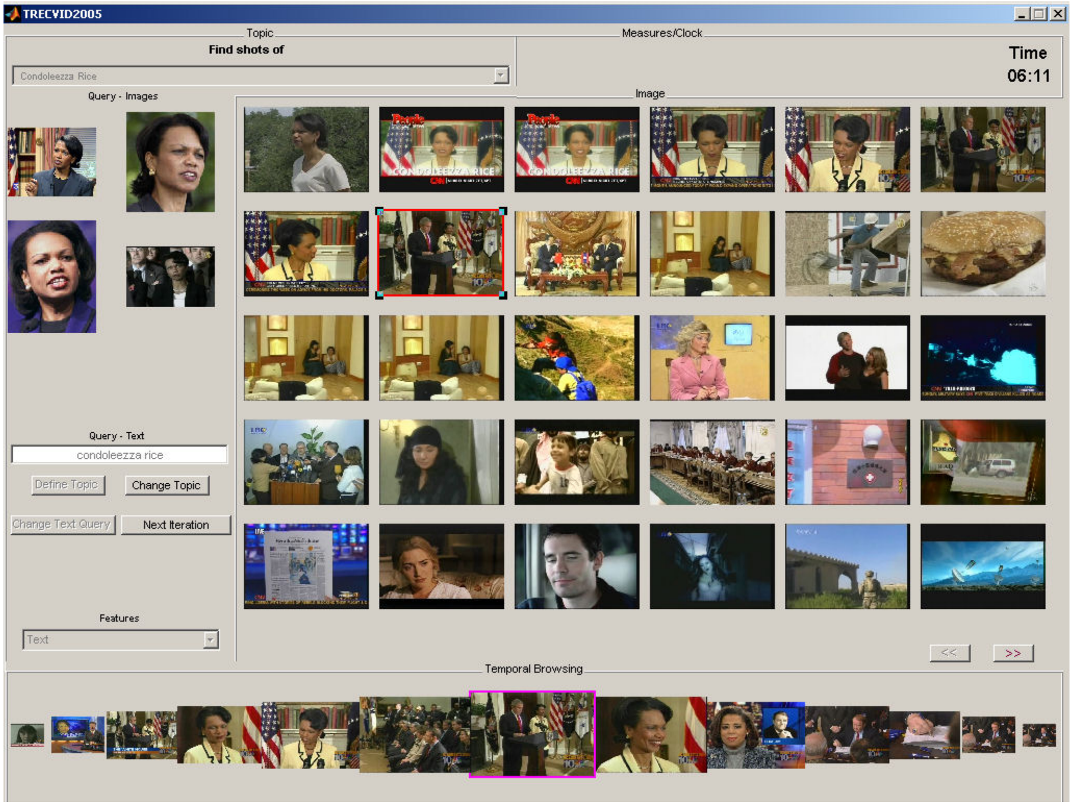
Figure 1 - Initial search result with temporal browsing for the topic "basketball players on the court".

Initial query formulation: query text and images are loaded as per topic and modified as necessary. Initially, by default the system use only text features but the user can modify manually the appropriate pop-up menu and use visual features too.

Browsing of search results: results are returned in a line-by-line page-wise fashion that is common for search engines, with the most relevant image in the top-left corner, and the least relevant in the bottom-right corner. This traditional approach is simple, intuitive and makes maximum use of the available space, helping the user to assess the relevance of many images quickly and use them in the search query if required. In this way, we maximise feedback from the user in the searching process.

Relevance feedback: the search process is divided in two phases. In the first phase, the user is able to modify the query text and the image features, and search again to collect relevant and non relevant images. When the user has a good number of relevant images, he can decide to enter in the second phase of search. The system use the images collected to train the Regularized Least Squares Classifier (RLSC) to perform a binary classification on the database (Jesus, Abrantes, and Marques, 2005). Then, the output of the classifier is used to select a set of images to display (Jesus, Abrantes, and Marques, 2005) and the system asks the user to give his feedback. This information is included in the previous training set and the classifier is retrained. This process is repeated several times.

Temporal Browsing: at the bottom of the window, temporal browsing is provided using a fisheye visualisation. This allows the user to effectively combine temporal browsing with searching. Once a user has located a relevant image, browsing along the temporal neighbours of that image often yields further relevant images.



Topic

Find shots of

Condoleezza Rice

Measures/Clock

Time

06:11

Query - Text

condoleezza rice

Features

Text

Figure 2 - Display of the images obtained in the first iteration of relevance feedback (15 images most relevant and 15 most informative).

Figure 2 shows the interface in the second phase of the search process for the topic "Condoleezza Rice". The displayed images were obtained by the output of the RLSC. The first fifteen images are the most relevant and the other fifteen images are the most informative according to the RLSC.

3.2 Features

For the search task we used 4 low-level texture and colour features as well as the text from the LIMSI transcripts. Like last year, we cut of the bottom of each image as this often contained a news line. To capture local image information, we partition the images into tiles and obtain features from each tile. The final feature vector consists of a concatenation of the feature vectors for individual tiles. For features we used last year's TRECVID, see (Heesch et al., 2004).

Tamura Features - We compute the three Tamura features coarseness, contrast and directionality as proposed in (Tamura, Mori, and Yamawaki, 1978) for each of 9x9 = 81 tiles. For each pixel we compute the values of each of the three features and compute for each tile a 3D histogram. See (Howarth and Rueger, 2004) for additional details and evaluation results.

Gabor Filter - One of the most popular signal processing based approaches for texture feature extraction is the use of Gabor filters. These enable filtering in the frequency and spatial domain. A range of filters at different scales and orientations allows multichannel filtering of an image to extract frequency and orientation information. This can then be used to decompose the image into texture features. Our implementation of the Gabor filter is based on (Manjunath and Ma, 1996). To each image we apply a bank of 4 orientation and 2 scale sensitive filters that map each image point $I(a,b)$ to a point in the frequency domain:

$$W_{nm}(a,b) = \int I(a,b)\, g_{mn}(x-a, y-b)\, dxdy$$

The feature consists of the mean and standard deviation of the modulus of Wnm. Since different filters produce outputs in different ranges, we normalize both mean and standard deviation by dividing by the respective standard deviation obtained for the entire database. We use a tiling of 3x3 based on experiments previously carried out on the Corel collection (Howarth and Rueger, 2004).

RGB color Histogram - Each image is represented by its color histogram with 256 bins.

Marginal HSV colour Moments – For this descriptor, we formed individual histograms for each of the three colour channels and computed the mean and second central moment of each marginal colour histogram.

Bag-of-Words Feature - The text feature was derived by aligning LIMSI ASR transcripts, closed caption data, and OCR data provided by Alexander Hauptmann's group at CMU. Using this textual annotation, we computed a bag-of-words feature consisting for each image of the set of accompanying stemmed words (Porter's algorithm) and their weight. This weight was determined using the standard tf-idf formula and normalised so that the sum of all weights is 1. As this is a sparse vector of considerable size (the number of different words) we store this feature in the form of (weight, word-id) pairs, sorted by word-id.

3.3 Experiments and Results

In the search task we conducted the following experiments:

Interactive runs: One interactive run were carried out. In this run, the users were invited to use content-based search with relevance feedback to retrieve relevant images to the query.

Block design: With a total of four users, 24 topics and one system variants, we chose an experimental design that required each user to execute a total of 6 queries.

The results for the interactive are shown in Table 2. This year we present a different system but the results were very close to the system presented in the last year (MAP = 0.200).

System Variant	MAP
Interactive: Search	0.209
TRECVID Interactive Median	0.226
TRECVID Interactive Max	0.498
TRECVID Interactive Best Run	0.414

Table 2. Search task results for the interactive run averaged over all 24 topics

4 High-level Feature Detection Task

For the high-level feature detection task we evaluated three new methods.

4.1 Naïve Model

As an alternative to learning the models of the concepts in a high-dimensional feature space, we 'mine' the feature space for salient information. Instead of estimating complex concepts with few data we aim at mining the full dataset for its natural patterns (salient clusters) and then learn the causality relation between these natural patterns co-occurrence and the (salient) concepts, see (Magalhães and Rüger, 2005). The algorithm is divided into the following tasks:

1. **Features Pre-Processing:** Process the features, by removing non-relevant features, combining redundant ones, and normalizing them. In this model we use Tamura, Gabor and marginal HSV features. The features are described in the search task section.
2. **Feature Space Density Model:** We use an unsupervised learning algorithm (producing a Gaussian mixture model) to detect salient clusters in the feature space, see (Figueiredo and Jain, 2002).
3. **Learning Cluster-Keyword Models:** The previous step was used to create a set of clusters that will be related to the concepts through generalized linear model.

Each component of the finite-mixture density model captures a pattern of the feature space and therefore a pattern that exists in certain groups of documents. Our model follows a typical generalized additive model comprised by sum of weighted basis functions, see (Hastie, Tibshirani, and Friedman, 2001). The formal expression of this model will be:

$$p(w_j \mid x) = \mathrm{logit}\left(\sum_i \beta_{c_i} \cdot p(c_i \mid x) \right),$$

The variable $\beta_{c_i|w_j}$ is the weight of the component c_i given the keyword *w*. The weights β_{c_i} fall in the interval $]-\infty,1]$ and $[1,+\infty[$, which is why the $\mathrm{logit}(x)$ function is used: it plays the role of returning a value with a probabilistic meaning. Note that $p(c_i \mid x)$ is independent of the training keyword and only the $\beta_{c_i|w_j}$ weights are dependent of training keyword.

A method to compute the linear model weights is based on a proportion between positive and negative examples. The estimation of the β_{c_i} weights is given by the expression:

$$\beta_{c_i} = \begin{cases} \alpha_{c_i} & ,\alpha_{c_i} > 1 \\ -1\!\big/\alpha_{c_i} & ,\alpha_{c_i} < 1 \end{cases}.$$

The estimation of the β_{c_i} weights is computed through the proportion α_{c_i} between the component c_i probabilities of the training keyword w positive examples $a \in \{E_W\}$, and negative examples $b \notin \{E_W\}$. The variable α_{c_i} value is given by the expression:

$$\alpha_{c_i} = \frac{\sum_{a\in\{E_W\}}^{K_P} p(c_i \mid w, x_a) \Big/ K_P}{\sum_{b\notin\{E_W\}}^{K_N} p(c_i \mid w, x_b) \Big/ K_N}.$$

Note that the $\pi_{c_i|w}$ weights fall in the interval $[1,+\infty[$ when the component is more relevant for the positive training samples and fall in the interval $]-\infty,1]$ when the component is more relevant for the negative training samples.

4.2 Nonparametric Density Estimation

We evaluated an approach to automated image annotation described in (Yavlinsky, Schofield, and Rüger, 2005) on this year's TRECVID feature detection task, where each high-level feature is treated as a keyword. We describe this approach briefly below. Denote a keyword as w and a keyframe image feature vector as x. The posterior probability of w given x can then be computed as

$$p(w \mid x) = \frac{f(x \mid w)p(w)}{f(x)}.$$

The density in the feature space conditional upon the assignment of a particular keyword, $f(x|w)$, is estimated using a nonparametric technique known as 'kernel smoothing'. Where x is a vector $(x_1,\ldots, x_d)$ of real-valued image features, we define the kernel estimate of $f(x|w)$ as

$$\hat{f}(x \mid w) = \frac{1}{|T_w|} \sum_{x^{(i)}\in T_w}^{n} k\left(x - x^{(i)}; h\right),$$

where $x^{(1)}\ldots x^{(n)}$ is the sample of images with label w in the training set T_w, where k is a kernel function that we place over each point $x^{(i)}$. We used the following kernel in our experiments:

$$k(t,h) = \frac{1}{h^d}\prod_{l=1}^{d} e^{-\frac{|t_l|}{h_l}},$$

where $t = x - x^{(i)}$. We set each bandwidth parameter h_l by scaling the sample standard deviation of feature l by the same constant λ, computed previously using a cross-validation approach on a different dataset. The prior is defined as

$$p(w) = \frac{1}{|W|},$$

where $|W|$ is the size of the vocabulary. We use a uniform keyword prior because we do not assume that the relative frequency of keywords in the training set is indicative of the true probability with which those keywords would be observed. Finally, we make the approximation $f(x) = \sum_w f(x \mid w) p(w)$ for simplicity. For a given keyword the images are ranked according their posterior probabilities of that keyword.

4.3 Experiments and Results

We evaluated the three models with all high-level features. The used low-level features weren't always the same; the naïve model only used global low-level features of each key-frame; the nonparametric density model used tiled Tamura features and tiled marginal HSV moments (3x3) and the global Gabor filter. Neither models used text, audio or video information.

The naïve model performance is low given the systems that participated in TRECVID. It was expected to have an average performance with this model (Magalhães and Rüger, 2005): the mixture model of the entire dataset constitutes a generalization step, which loses information useful to discriminate between classes. Furthermore, we encountered several problems running this model in TRECVID dataset. Because the dataset grew in size this year, we couldn't run the mixture modeling with a large number of clusters as the algorithm requires, and we couldn't use tiling since it would increase exponentially the number of samples. For these reasons we had to run the model with serious limitations which is now reflected in the low performance illustrated by Figure 3.

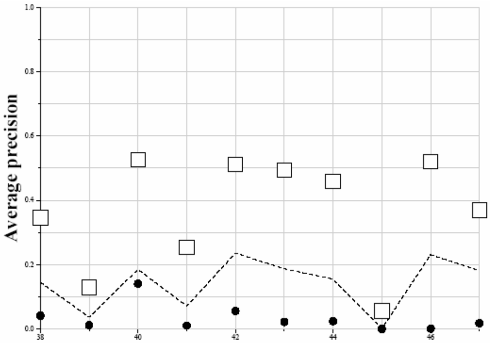


Figure 3 – Naïve model: run score (dot), versus median (---) versus best (box) by feature.

The nonparametric model achieves competitive performance (see Figure 4), which can be attributed to the fact that it retains as much information about the keyword distributions within the training set as possible. This result also reinforces the finding in (Yavlinsky, Schofield, and Rüger, 2005) that global features are suitable for automated image annotation.

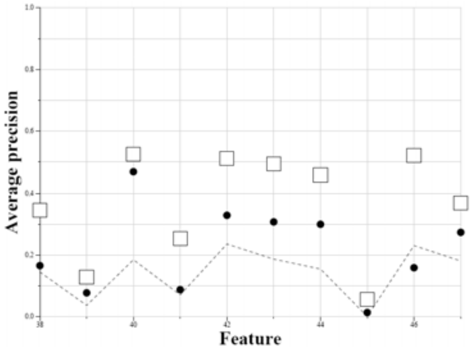


Figure 4 – Nonparametric density estimation: run score (dot), versus median (---) versus best (box) by feature.

Feature		Nonparametric density estimation
38	People walking/running	16,59%
39	Explosion/fire	7,77%
40	Map	46,86%
41	US flag	8,79%
42	Building exterior	32,91%
43	Waterscape/waterfront	30,77%
44	Mountain	30,00%
45	Prisoner	1,38%
46	Sports	15,91%
47	Car	27,39%
Mean Average Precision		**21,84%**

Table 3 – Retrieval evaluation measures of the three models.

5 Conclusion

The new relevance feedback algorithm, Regularized Least Squares Classifier, proved to be as efficient as last year's algorithm. As was expected, text was fundamental to reach the observed MAP.

In the high-level feature detection task, we tested two simple methods. Unfortunately we encountered some practical problems with the naïve model which then caused a poor performance. Using kernel smoothing to model collaboratively annotated training data with only simple visual features achieves competitive performance (mean average precision 0.22 across all topics).

6 References

Figueiredo, M., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(3), 1-16.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference and prediction*: Springer.

Heesch, D., Howarth, P., Magalhães, J., May, A., Pickering, M., Yavlinsky, A., et al. (2004). *Video retrieval using search and browsing.* TREC Video Retrieval Evaluation Workshop, Gaithersburg, MD, USA.

Howarth, P., & Rueger, S. (2004). *Evaluation of texture features for content-based image retrieval.* Int'l Conference on Image and Video Retrieval, Dublin, Ireland.

Jesus, R. M., Abrantes, A. J., & Marques, J. S. (2005). *Relevance feedback in CBIR using the RLS classifier.* EURASIP Conference on Speech and Image Processing, Multimedia communications and Services, Bratislava, Slovakia.

Magalhães, J., & Rüger, S. (2005). *Mining multimedia salient concepts for incremental information extraction.* ACM SIGIR conference on research and development in information retrieval, Salvador, Brasil.

Manjunath, B., & Ma, W. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(8), 837-842.

Pickering, M. J., Heesch, D., O'Callaghan, R., Rueger, S., & Bull, D. (2002). *Video retrieval using global features in keyframes.* TREC Text Retrieval Conference, Gaithersburg, USA.

Pye, D., Hollinghurst, N. J., Mills, T. J., & R.Wood, K. (1998). *Audio-visual segmentation for content-based retrieval.* Int'l Conference on Spoken Language Processing, Sydney, Australia.

Tamura, H., Mori, S., & Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics, 8*(6), 460-472.

Yavlinsky, A., Schofield, E., & Rüger, S. (2005). *Automated image annotation using global features and robust nonparametric density estimation.* Int'l Conference on Image and Video Retrieval, Singapore.

Zhang, H. J., Kankanhalli, A., & Smoliar, S. W. (1993). Automatic partitioning of full-motion video. *ACM Multimedia Systems, 1*(1), 10-28.