A Comparison of Score, Rank and Probability-based Fusion Methods for Video Shot Retrieval

Kieran Mc Donald and Alan F. Smeaton

Centre for Digital Video Processing, Dublin City University, Dublin, Ireland

Abstract. It is now accepted that the most effective video shot retrieval is based on indexing and retrieving clips using multiple, parallel modalities such as text-matching, image-matching and feature matching and then combining or fusing these parallel retrieval streams in some way. In this paper we investigate a range of fusion methods for combining based on multiple visual features (colour, edge and texture), for combining based on multiple visual examples in the query and for combining multiple modalities (text and visual). Using three TRECVid collections and the TRECVid search task, we specifically compare fusion methods based on normalised score and rank that use either the average, weighted average or maximum of retrieval results from a discrete Jelinek-Mercer smoothed language model. We also compare these results with a simple probability-based combination of the language model results that assumes all features and visual examples are fully independent.

1 Introduction

The purpose of video retrieval is to locate video from a collection that meets a user's information needs. In this paper we address the general video retrieval task, as supported by the TRECVid search task, which expresses search topics in terms of a text description coupled with multiple image and video examples. Because video retrieval is situated in a diverse feature environment, it potentially requires the combination somehow of many different features. These include text (automatic speech recognised text, closed caption text, video optical character recognition text), audio features (e.g. monologues, music, gun firing), visual features (colour, texture, shape), motion features (cameras and objects), high-level concepts ('visual keywords' such as outdoors, indoors, landscape, faces) and other specific audio-visual models such as for identifying specific people, animals or objects. Early fusion methods, which combine features before performing matching, are not practical for such a large number of features due to the high dimensionality of any combined representation. Late fusion methods, which are the topic of this paper, perform matching on individual features and fuse these matching scores. Late fusion can potentially support adaptive fusion methods when relevance information is available and also allows the use of tuned retrieval models (or completely different retrieval models) for each feature.

At their most basic, late fusion methods combine the scored and ranked retrieval results from different systems/models/features in order to improve upon the best individual retrieval result. Traditional fusion techniques in information retrieval can be broadly divided into rank and score-based [6]. Rank-based methods such as Borda count combine separate search results based on summing the rank position of documents from different result lists. An extension to this combination method is weighted Borda count, which gives preferential weight to specific search result lists. Traditional score-based combination methods include CombSUM, which sums the multiple retrieval scores, and CombMNZ which sums the scores from truncated results lists (such as top 1000) and multiplies the average by the number of retrieval models that returned it [3]. Weights are predominantly included though a linear interpolation of scores. When combining heterogenous retrieval models/features normalisation of retrieval scores is necessary and generally involves linear normalising the results from 0 to 1. Quite different approaches include distribution modelling [4] and logistic regression [5], which attempt to learn a relationship between scores/ranks and relevance.

Fusion is very important in the video search task. Smith *et al.* [7], reports on many score-based fusion methods used in an interactive video retrieval experiment but does not cross compare their performances. In [8] Westerveld *et al.* combine their visual language model results with the text language model results using the joint probability of generating both features assuming independence between modalities and combine the results of multiple visual examples using round-robin (minimum rank). Yan *et al.* [9] use a boosted co-training approach that trains the weights for combining concept and low-level feature results with text-based results on a per-query basis. In [10] the search topics were automatically classified into one of four classes (named people, named objects, general objects, scenes) and they used query-class dependent weights for fusing results in a hierarchical mixture of experts framework. Yavlinsky *et al.* [11] compared support vector machines with the standard fusion methods CombMIN, CombMAX, CombSUM and Borda count for the task of combining text and visual feature results on TRECVid 2003 but found that no fusion method improved on the results of text.

In this paper we investigate standard fusion methods based on scores, ranks and probability for single visual example search (fusing multiple visual features), for multiple visual example search and for multiple modality search. We evaluate fusion methods for visual retrieval models based on the results of Jelinek-Mercer smoothed language model for three visual features (regional colour, edge and texture) on three video retrieval collections (TRECVid 2002, 2003 and 2004). We successfully used the same features and retrieval model (discrete Jelinek-Mercer language model) in our TRECVid 2004 automatic search submission and the discrete language model has previously been studied in [1] but achieved poorer results due probably to their discrete feature representation, which was high-dimensional and lacked x, y location information. The contribution of this paper is in empirically establishing effective fusion methods for supporting different types of video search such as single feature, multiple feature, multiple example

or multimodal search that achieve state-of-the-art performance in the TRECVid video search task.

The rest of this paper is organised as follows: In section 2 we describe the fusion methods that we will evaluate in this paper, while in section 3 we describe our experiment setup. In section 4 we present and discuss our fusion results and finally in section 5 we summarise our conclusions.

2 Fusion for Multi-Modal, Multi-Example Video Retrieval

We investigate the fusion of retrieval model results in order to combine (A) the multiple visual features, (B) the multiple visual examples and (C) the multiple modalities text and visual. The combination (A) supports the retrieval of video shots using a single visual example and in our experiments involves the automatic fusion of colour, edge and texture retrieval models. The combination (B) supports visual-based retrieval of video shots using a query with multiple visual examples (images and/or videos) and involves the automatic fusion of results from possibly quite disparate image or video examples. The combination (C) supports the retrieval of video shots using a query which has both text and multiple visual examples and for which the combination would involve very different and possibly conflicting result sets. We also investigate the fusion of results for multiple visual examples using a single visual feature, which provides support for users who wish to use a single visual feature in their search.

The multi-example multi-feature search can be performed in two different sequences. Firstly, visual features can be combined for each visual example and then the visual examples' scores are fused, or secondly, the visual features can first be separately combined for each visual example and then the scores for each visual feature can be combined. Due to score normalisation and result list truncation these different sequences do not yield exactly the same results.

We combine results using fusion methods originally investigated for fusing the results of multiple text search engines [2, 3]. These fusion methods are computationally inexpensive and have been shown to be quite effective on truncated result lists such as top 1000 results for text retrieval. Truncating the result lists is beneficial as it reduces the amount of information transferred between nodes within a video retrieval server that is distributed across multiple machines. We compare fusion methods based on normalised score and rank that use either the average, weighted average or maximum of individual results as the combination function. We also compare these results with a probabilistic combination that assumes all features and examples are fully independent and which does not truncate the result lists. We will use the following notation to refer to each fusion strategy:

- *CombJointPr* - multiply the probabilities of individual retrieval models (or add the log-likelihoods).
- *CombSumScore* - add the normalised scores of the top N results (ie. traditional CombSUM).

- *CombSumRank* - add the normalised ranks of the top N results (ie. traditional Borda count)
- *CombMaxPr* - order by the maximum of the probabilities.
- *CombMaxScore* - order by the maximum of the scores - same as CombMaxPr when inputs are probabilities.
- *CombMaxRank* - order by the maximum of the normalised rank score (ie. round-robin/order by increasing rank removing duplicates).
- *CombSumWtScore* - weighted average of the normalised scores of the top N results.
- *CombSumWtRank* - weighted average of the normalised ranks of the top N results (weighted Borda count).

For all score and rank based fusion methods we truncate the input result lists to their top N results. As in [3] we define normalised rank as

$$norm\_rank_{shot} = \frac{N + 1 - rank_{shot}}{N} \tag{1}$$

where N is the number of shots in the truncated result list and normalised score is defined as

$$norm\_score_{shot} = \frac{score_{shot} - score_{min}}{score_{max} - score_{min}} \tag{2}$$

where $score_{min}$ is the score of the lowest ranked shot in the truncated result list. When combining features we truncate the feature result lists to their top 1000 results (N=1000), but when combining results from multiple visual examples we truncate the visual examples' result lists to $N = M/num\_visual\_examples$, where M is a value between 1000 and 3000 and is empirically chosen for each fusion method by tuning on separate topics and video collection.

We use log-query-likelihoods as our score for each shot's text and visual language model's retrieval results since the generative probabilities for our visual features are extremely small and cannot be directly represented using double precision floating point numbers. As a result we are limited in how we can directly combine these probabilities but one simple combined generative model is to assume that all the features and visual examples are independent, which is straightforward to calculate by adding the log-probabilities. For some fusion tasks, especially combining visual and textual results, it would be beneficial to combine the generative probabilities using a finite mixture model (linear interpolation) but as yet we have not evaluated this approach, which we believe would be more beneficial than using joint probability for combining text and visual results as it allows for the influence of the visual model to be reduced - the joint probability of text and visual features allows the visual features probabilities to overwhelm the combination since it is the result of the product of probabilities for each pixel in the query image whereas the text probability is the result of the product of only a few probabilities of the search terms.

We prefer the normalised score fusion methods over the normalised rank fusion methods as we believe that the distribution of scores holds valuable information that is lost when normalising based on rank. Ideally the result sets

being fused have somewhat similar relevant documents and dissimilar irrelevant documents. If this is the case then combining results using the average function should be preferable to the max function because averaging should reduce the noise from a single query image/feature's results whereas using the max function assumes that a document which matches well a single query image/feature is preferable. We can think of averaging as indicating the document should somewhat match all features/examples (AND logic), whereas max implies that a relevant document need only match a single feature/example (OR logic).

3 Experiment Setup

We perform automatic retrieval experiments, where by "automatic" we mean that retrieval does not involve iterative refinement from end-users, on the TRECVid 2002, 2003 and 2004 collections and search topics. The TRECVid 2002 collection consists of advertising, educational, industrial and amateur videos from the early 20th century to mid seventies, while the TRECVid 2003 and 2004 collections contain TV news programmes, broadcast in the late 1990's on the ABC, CNN and C-SPAN channels. TRECVid search topics are motivated from the needs of professional video searchers who request video shots that contain specific or generic people, things, actions and/or locations (e.g. shots of people moving a stretcher, a handheld weapon firing, Boris Yeltsin, flood waters). Search topics request video shots and are formulated as a multimedia query that contains a text description of the information need plus multiple image and video examples.

We represent the video shot content using four features ASR text, HSV colour, Canny edges and DCT-based texture. The visual features are all calculated using a 5x5 grid based representation thus providing a limited but still potentially beneficial amount of positional information. The HSV colour is quantised into a 16x4x4 multidimensional histogram (16 hue by 4 saturation by 4 brightness levels). The Canny edge direction feature is quantised into 64 directions with the first direction centred on the horizonal axis and non-edge pixels are counted in an extra bin for each image region. The DCT feature quantises the first 5 DCT coefficients of the brightness band of the YCbCr colour space into 3 bins each with the quantisation boundaries for each DCT coefficient calculated across the whole keyframe collection so that the marginal distribution of a specific DCT coefficient uniformly populates its quantisation bins. The DCT transform is calculated for non-overlapping 8x8 pixel blocks in the image. The visual features representations were chosen for our official TRECVid 2004 automatic discrete language model experiments and their selection was based on their performance on the TRECVid 2003 collection. This implies that our visual results for TRECVid 2003 collection are somewhat biased though still useful for comparing fusion models.

For the ASR text feature, we use the hierarchical Jelinek-Mercer smoothed language model [8] that smoothes a shot with the text from adjacent shots, from the enclosing video and from the collection. For the visual features we use a discrete language modelling approach. In the language modelling approach shots

are ranked by the probability of their language model generating the query, an approach known as query-likelihood. Jelinek-Mercer smoothing uses a collection model (distribution of the events in the whole collection of the visual feature) to adjust the empirical distribution of the features so as to better handle low-frequency (particularly zero frequency) events and to reduce the importance of frequent events. Its retrieval status value is

$$RSV_{q,d} = \log \Pr_{JM}(q|d) = \sum_t f_{q,t} \times \log \Big((1-\lambda) \Pr_{ML}(t|d) + \lambda \Pr_{ML}(t|\mathcal{C}) \Big) \quad (3)$$

where t is a symbol from the visual feature's discrete language (histogram bin index), $f_{q,t}$ is the frequency of the symbol in the query (visual example), $\Pr_{ML}(t|d)$ is its empirical probability in a document (video shot), and $\Pr_{ML}(t|\mathcal{C})$ is its empirical probability within the whole collection.

For each experiment we tune the retrieval models that have free parameters on an independent search collection so as not to bias our experiment. The tuning process is automatic and identifies a single parameter setting over all tuning topics that optimises mean average precision (MAP). For experiments with the TRECVid 2002 and 2004 search topics the parameters are tuned on TRECVid 2003 search topics and collection, while the parameters for retrieval and fusion models on TRECVid 2003 search topics are tuned using TRECVid 2002. When reporting results for each fusion task in terms of mean average precision (MAP) and precision at cutoff 10 and 100 we indicate whether the difference between these result and our best fusion result is statistically significant according to the Wilcoxon sign-rank test at 95% significance level. We furthermore aggregate all results from the three collections and test whether the overall best result is significantly better than the other fusion methods.

4 Results

The results for all our experiments are shown in Table 1 which we primarily discuss in terms of MAP unless otherwise stated.

Multiple Features, Single Example Fusion The first fusion experiment, Vis*CET, is for single visual query-by-example and combines colour, edge and texture features. All fusion methods except CombJointPr fail to improve on the colour-only results for TRECVid 2002, while all fusion methods except CombSumWtRank improve on the colour only results for TRECVid 2003 and 2004. The results for TRECVid 2003 and 2004 indicate that CombSumRank and CombSumScore achieve similar results and are statistically significantly better than CombSumWtRank and CombSumWtScore. The aggregated result for the three collections indicates that CombSumScore is best and is significantly better than the same two weighted fusion methods. Overall, we find it a little surprising that the weighted variants do not perform as well as a simple average considering that colour performs better that the other two single features. This indicates the difficulty in tuning weights for combining visual features.

Table 1. Fusion results in terms of mean average precision (MAP) and precision at cutoff 10 (P10) and 100 (P100) for TRECVid 2002, 2003 and 2004 search tasks. The aggregated (Agg.) column shows the MAP for all topics from the three collections. Bolded results are the highest for each fusion task, while underlined results are statistically significantly poorer than these (Wilcoxon sign-rank test, 95% significance level).

Features	*Fusion*	*TRECVid 2002*			*TRECVid 2003*			*TRECVid 2004*			*Agg.*
		MAP	*P10*	*P100*	*MAP*	*P10*	*P100*	*MAP*	*P10*	*P100*	*MAP*
Colour		**.0153**	**.041**	**.018**	**.0238**	.080	.037	**.0088**	**.039**	**.015**	**.0159**
Edge		.0092	.036	.017	.0105	.036	.024	.0078	.022	.012	.0092
Texture		.0073	.023	.015	.0226	**.082**	**.040**	.0061	.022	.010	.0127
Vis	**JointPr**	**.0156**	.042	.018	.0244	.080	.038	.0093	.039	.015	.0164
**CET*	**WtRank**	.0110	.045	.022	.0230	**.084**	.037	.0061	.022	.010	.0136
	WtScore	.0143	**.052**	.020	.0252	.083	.039	.0113	.040	.016	.0173
	SumRank	.0116	.044	.022	.0247	.076	**.049**	**.0132**	**.041**	**.018**	.0172
	SumScore	.0126	.049	**.023**	**.0262**	.081	.047	.0130	.040	.017	**.0180**
VisExs	**JointPr**	.0069	.024	.016	**.0536**	**.100**	**.063**	.0024	.017	.011	.0215
**Colour*	**SumRank**	.0146	.056	.022	.0364	.084	.058	.0142	.043	.034	.0219
-only	**SumScore**	.0152	.044	**.023**	.0400	.072	.058	**.0174**	**.052**	**.036**	**.0244**
	MaxPr	**.0231**	.056	.020	.0221	.048	.035	.0017	.022	.009	.0160
	MaxRank	.0230	**.060**	.020	.0162	.048	.031	.0016	.017	.008	.0139
VisExs	**JointPr**	.0042	.016	.012	.0061	.004	.022	.0031	.017	.009	.0045
**Edge*	**SumRank**	.0081	.036	.019	.0132	.040	**.026**	.0234	.074	**.035**	.0147
-only	**SumScore**	**.0142**	**.072**	**.020**	**.0133**	**.048**	.022	**.0255**	**.078**	.027	**.0174**
	MaxPr	.0111	.028	.008	.0126	.044	.023	.0033	.009	.003	.0092
	MaxRank	.0108	.028	.007	.0038	.024	.016	.0032	.009	.003	.0060
VisExs	**JointPr**	.0123	.024	.016	.0363	**.120**	.054	.0016	.013	.007	.0172
**Texture*	**SumRank**	.0074	.016	.019	.0331	.088	.057	.0054	.013	**.012**	.0156
-only	**SumScore**	**.0142**	**.032**	**.020**	**.0417**	.116	**.061**	**.0057**	**.030**	.009	**.0209**
	MaxPr	.0120	.028	.018	.0196	.068	.030	.0005	.004	.003	.0110
	MaxRank	.0116	.028	.017	.0086	.060	.020	.0004	.004	.003	.0070
VisExs	**JointPr**	.0071	.032	.016	**.0564**	.100	**.067**	.0036	.030	.012	.0229
**Vis*	**MaxPr**	**.0216**	**.092**	.034	.0145	.040	.026	.0016	.017	.007	.0129
	SumRank	.0114	.032	.028	.0382	.068	.065	.0244	.043	.024	.0247
	SumScore	.0172	.048	.032	.0394	.060	**.067**	.0272	.074	.027	.0280
	MaxRank	.0204	.088	.033	.0502	**.120**	.064	.0234	.087	.023	.0316
	MaxScore	.0205	.088	.033	.0500	.120	.065	.0231	.087	.023	.0314
**CET*	**SumRank**	.0193	.068	**.035**	.0433	.088	**.067**	**.0413**	**.139**	**.037**	**.0344**
	SumScore	.0174	.064	.030	.0450	.084	.061	.0356	.100	.028	.0326
	WtRank	.0161	.068	.023	.0493	.100	.061	.0128	.048	.014	.0264
	WtScore	.0213	.064	.029	.0503	.092	.066	.0245	.074	.021	.0322
Text-Only		.1605	.264	.117	.1405	.252	.113	.0686	.209	.091	.1247
TextVis	**JointPr**	.0071	.032	.016	.0564	.100	.067	.0036	.030	.012	.0229
	SumScore	.1326	.212	.096	.1211	.244	.118	**.0862**	.230	.097	.1140
	SumRank	.1134	.172	.096	.1255	.228	.116	.0595	.109	.088	.1005
	WtRank	.1589	.232	.118	.1530	.288	.114	.0700	**.257**	.093	.1289
	WtScore	**.1715**	**.268**	**.121**	**.1633**	**.292**	**.126**	.0830	.243	**.102**	**.1408**
% Impr. on Text		6.9	1.5	3.4	16.2	15.9	11.5	21.0	16.3	12.1	12.9

Single Feature, Multiple Example Fusion: We performed single feature, multi-example fusion experiments for colour (VisExs*Colour-only), edges (VisExs*Edge-only) and texture (VisExs*Texture-only). The overall best performing fusion method is CombSumScore, which is clearly the best fusion method for combining the visual examples for the edge and texture features on the separate collections, while for the colour feature, it is the best method for TRECVid 2003 and the second best for TRECVid 2004. On the three collections and three features it is never statistically significantly bettered by another fusion method in terms of the three performance measures. Surprisingly CombMaxPr (CombMaxScore) and CombMaxRank (round-robin) perform quite poorly and are overall significantly poorer for the aggregated collection results. We believe this implies that the TRECVid topics visual examples are more cohesive than we previously thought. For the most part CombSumRank again performs slightly worse that CombSumScore indicating the slight benefit of using the scores. The CombJointPr method performs best on TRECVid 2003 but its performance is quite erratic and nearly always lower than CombSumScore on other features and collections in terms of the three performance measures. The only difference between these two methods that effects ranking is that CombSumScore normalises and truncates the scores before averaging. In investigating this we found that truncation of results slightly hurts performance and that the normalisation of scores accounts for the improvement in results of CombSumScore over CombJointPr.

Multiple Features, Multiple Example Fusion: In our VisExs*Vis multi-feature multi-example visual experiments we combine the visual features using CombSumScore for each example and then combine the multiple visual examples, while in our VisExs*CET multi-feature multi-example experiments we combine visual examples separately for the three features using CombSumScore and then combine the results of the multiple visual features. In the case of the CombJointPr both these orderings produce exactly the same results, however the other fusion methods are not symmetric to the order of fusion. The VisExs*CET CombSumRank (Borda count) fusion performs consistently better than the other fusion methods. Again both it and the respective CombSumScore perform similarly but this time CombSumScore has the slightly lower results and again neither fusion method is significantly bettered by another fusion method. We believe the previous fusion task (in particular the truncation of results) may have reduced the usefulness of the scores for this fusion task. Even though CombJointPr performs best in terms of MAP and precision at cutoff 100 for TRECVid 2003, the aggregated collection results indicates that it is significantly poorer than the CombSumRank method. The performance of this fusion method (and others) on TRECVid 2003 is largely due to two topics and this accounts for how the mean performance on this collection without taking into account the statistical tests can mislead. The CombMaxScore and CombMaxRank methods perform well on TRECVid 2002 and 2003 but perform relatively poorly on TRECVid 2004. CombSumWtScore also performs well on TRECVid 2002 and TRECVid 2003 but significantly worse than CombSumRank on TRECVid 2004. TRECVid

2004 fusion methods are tuned on the very similar TRECVid 2003 collection and the general underperforming of fusion methods with weights indicate how delicate this process is and the possible need for a large set of tuning topics or the classification of topics into sub-groups.

Multimodal Fusion The multimodal fusion results (TextVis) which combine the ASR text retrieval results with the retrieval results of multiple visual examples (specifically CombSumRank, VisExs*CET) indicates that CombSumWtScore is the best multimodal fusion strategy for this task and shows positive improvement in terms of MAP and precision at cutoff 10 and 100 for all three collections. These results are representative of the current state-of-the-art for automatic video retrieval experiments and improve but not statistically significantly on our previous submitted TRECVid 2004 automatic video retrieval results (MAP 0.078), which achieved the highest MAP of the submitted automatic TRECVid video retrieval runs. This improvement is solely due to better fusion strategies since we did not change any of the features or retrieval models. The CombJointPr fusion performs very poorly, actually achieving the same performance of visual-only searching, due to the fact that the visual probabilities for a large sample of pixels overwhelms the generative text probabilities for a small sample of text in the joint probability. This effect was expected but the magnitude in overwhelming the good performance of text was not. The difference between optimal weights is again highlighted by the result of CombSumScore which achieves the highest MAP of 0.0862 for TRECVid 2004 though not significantly better than CombSumWtScore.

5 Conclusions

We combined results for the text and visual features using variations of data-fusion methods originally developed for combining the results of multiple text search engines. We found consistent results indicating that CombSumScore is best for combining a single visual feature over multiple visual examples and that CombSumWtScore is best for combining text and visual results for TRECVid type searches. Our experiment results also indicated that CombSumScore (and CombSumRank) are best for combining multiple features for a single query image. Our results for multi-example multi-feature visual search, while less clear cut, indicate that features should first be fused separately for the visual examples and then these features' scores should be fused using CombSumRank or CombSumScore. In our experiments all the retrieval models and fusion models have been trained and tested on separate collections and therefore our experiments should represent a fair comparison of fusion strategies. The limitations of the current study is that it is possible our findings could be tied to the particular retrieval model (discrete Jelinek-Mercer langauge model) and the particular set of visual features. Our future work entails improving the visual features and evaluating fusion methods for alternative retrieval models (e.g. L1) and features. Our current results highlight problems with tuning weights for combining visual features which is likely exacerbated when trying to fuse more visual features.

6 Acknowledgments

This work was partially supported by Science Foundation Ireland under grant No. 03/IN.3/I361. The authors wish to acknowledge the support of the Informatics Directorate of Enterprise Ireland.

References

1. A. P. de Vries and T. Westerveld. A comparison of continuous vs. discrete image models for probabilistic image and video retrieval. In *Proceedings of IEEE International Conference on Image Processing (ICIP-2004)*, 2004.
2. E. Fox and J. Shaw. Combination of multiple searches. In *Proceedings of the 2nd Text REtrieval Conference TREC-2*, pages 243–252. NIST Special Publications 500-215, 1994.
3. J. H. Lee. Analyses of multiple evidence combination. In *Proc. of the 20th Intl. Conf. on Research and Development in Information Retrieval (SIGIR'97)*, pages 267–276, 1997.
4. R. Manmatha, F. Feng, and T. Rath. Using models of score distributions in information retrieval. In *Proceedings of the LM Workshop 2001*, pages 91–96, 2001.
5. J. Savoy, A. Le Calve, and D. Vrajitoru. Report on the TREC-5 experiment: Data fusion and collection fusion. In *Proceedings of TREC-5*, pages 489–502, 1997.
6. A. F. Smeaton. Independence of contributing retrieval strategies in data fusion for effective information retrieval. In *Proceedings of the 20th BCS-IRSG Colloquium*, Grenoble, France, April 1998. Springer-Verlag Workshops in Computing.
7. J. Smith, A. Jaimes, C.-Y. Lin, M. Naphade, A. Natsev, and B. Tseng. Interactive search fusion methods for video database retrieval. In *IEEE International Conference on Image Processing (ICIP)*, pages 741–744, 2003.
8. T. Westerveld, T. Ianeva, L. Boldareva, A. P. de Vries, and D. Hiemstra. Combining information sources for video retrieval: The Lowlands team at TRECVID 2003. In *Proceedings of TRECVid 2003*, 2004.
9. R. Yan and A. G. Hauptmann. Co-retrieval: A boosted reranking approach for video retrieval. In *Proceedings of the Third International Conference on Image and Video Retrieval (CIVR 2004)*, pages 60–69. Springer-Verlag, 2004.
10. R. Yan, J. Yang, and A. G. Hauptmann. Learning query-class dependent weights in automatic video retrieval. In *Proceedings of ACM Multimedia 2004*, pages 548–555, New York, NY, Oct. 2004.
11. A. Yavlinsky, M. J. Pickering, D. Heesch, and S. Rüger. A comparative study of evidence combination strategies. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2004.