TRECVid 2006 Experiments at Dublin City University

Markus Koskela, Peter Wilkins, Tomasz Adamek, Alan F. Smeaton, and Noel E. O'Connor
Centre for Digital Video Processing & Adaptive Information Cluster
Dublin City University, Glasnevin, Dublin 9, Ireland
Alan.Smeaton@computing.dcu.ie

Abstract

In this paper we describe our retrieval system and experiments performed for the automatic search task in TRECVid 2006. We submitted the following six automatic runs:

- `F_A_1_DCU-Base_6`: Baseline run using only ASR/MT text features.
- `F_A_2_DCU-TextVisual_2`: Run using text and visual features.
- `F_A_2_DCU-TextVisMotion_5`: Run using text, visual, and motion features.
- `F_B_2_DCU-Visual-LSCOM_3`: Text and visual features combined with concept detectors.
- `F_B_2_DCU-LSCOM-Filters_4`: Text, visual, and motion features with concept detectors.
- `F_B_2_DCU-LSCOM-2_1`: Text, visual, motion, and concept detectors with negative concepts.

The experiments were designed both to study the addition of motion features and separately constructed models for semantic concepts, to runs using only textual and visual features, as well as to establish a baseline for the manually-assisted search runs performed within the collaborative K-Space project and described in the corresponding TRECVid 2006 notebook paper. The results of the experiments indicate that the performance of automatic search can be improved with suitable concept models. This, however, is very topic-dependent and the questions of when to include such models and which concept models should be included, remain unanswered. Secondly, using motion features did not lead to performance improvement in our experiments. Finally, it was observed that our text features, despite displaying a rather poor performance overall, may still be useful even for generic search topics.

1 Introduction

This year the stand-alone participation of Dublin City University in TRECVid 2006 contains submissions only to the automatic search task. We submitted a total of six fully automatic runs. Unlike our submission last year [6], in which two users collaborated in the search task using a multi-user tabletop input device, the 2006 submission has been developed as a framework for video retrieval research using the traditional single user desktop paradigm. As our independent submission, we designed a set of six experiments for the automatic search task. Furthermore, the retrieval system was also utilized in the collaborative experiments for the manual search task performed within the K-Space project [1], combining the work done in 9 partner organizations and coordinated by DCU. For a description of these other K-Space experiments, see [12].

The rest of the paper is organized as follows.

The retrieval system itself and the features used in these experiments are described in Section 2. The experiments submitted for the fully automatic search task are described in Section 3. Conclusions are then presented in Section 4.

2 Retrieval System

Our automatic retrieval system for experiments this year was a new system constructed to make use of our knowledge gained in query-time automatic weight generation [13]. Highly configurable, it allows a retrieval run to make use of different weight generation schemes, normalization algorithms, feature aggregation points and fusion methodologies, each of which can have an impact on performance. Our system is modular and allows for the insertion of new components, with the retrieval process driven by XML configuration files which fully describe the retrieval method. We now describe the features we used as inputs into our retrieval system.

2.1 Features

Our retrieval system this year made use of text, visual and semantic features. The following subsections describe each of our features and how we utilized each.

2.1.1 Text

Our search system utilized the extracted Automatic Speech Recognition (ASR) transcripts. These transcripts were aligned to shot boundaries. For retrieval we made use of the Zettair search engine [2]. We applied a basic windowing scheme to any returned shots, adding into the result the preceding two shots and the following two shots for each returned result. These additional shots when added were given a down-weighted score of the original shot. We did not make use of the closed caption text provided, but plan to do so in subsequent revisions of our work.

2.1.2 Visual

To facilitate content-based search of query images against shot keyframes our retrieval system made use of four different low-level visual features. Our features are MPEG-7 features and were extracted using the aceToolBox, developed as part of our collaboration in the aceMedia project [3]. We made use of the following features:

- An **Edge Histogram Descriptor (EHD)** is designed to capture the spatial distribution of edges by dividing the image into 4x4 subimages (16 non-overlapping blocks) and edges are then categorised into 5 types (0°, 45°, 90°, 135°and 'nondirectional') in each block. The output is a 5 bin histogram for each block, giving a total of 5x16 = 80 histogram bins.
- A **Local Colour Descriptor (Colour Layout - CLD)** is a compact and resolution-invariant representation of colour in an image. The colour information of an image is partitioned in 64 (8x8) blocks; second, the representative colour of each block is determined by using the average colour in each block.
- **Scalable Colour - SCD)** measures colour distribution over an entire image. It is defined in the hue-saturation-value (HSV) colour space and produces a 256-bin colour histogram, normalised, non-linearly mapped into a four-bit integer value, and then encoded by a Haar transform. This last consists of computing the sum and the difference of adjacent pairs. The sum of adjacent bins leads to a histogram with half the number of bins. Repeating this process four times, we finally obtain a 32-bin histogram. Another form of scalability is achieved by scaling the quantized representation of the coefficients to different numbers of bits. Here the three less significant bits were discarded.
- **Homogenous Texture Descriptor (HDT)** describes directionality, coarseness, and regularity of patterns in images. It is computed by first filtering the image with a bank of orientation and scale sensitive (Gabor) filters, and then computing the mean and standard deviation of the filtered outputs in the frequency domain. In this work we only use the mean values to compute the similarity between the images.

Further details on these visual features can be found in [10]. Our visual features when queried were ranked using Euclidian distance. Earlier work of ours [9] highlights our reasons for employing this metric.

We also made use of a motion estimation feature, provided by Joanneum Research which is described in [12].

2.2 Result Fusion

As mentioned earlier, our system is highly configurable, both in terms of the various algorithms that can be employed in the fusion process, through to the actual fusion stages themselves. Because of this we normalized on our fusion steps to those about to be described, such that we could measure the impact of our usage of semantic features. We only experimented with the use of CombSUM [7] for combining results lists. Our combinations were always normalized first, and for normalization we employed MinMax normalization, formally given by Equation 1.

$$Norm_{score(x)} = \frac{Score_x - Score_{min}}{Score_{max} - Score_{min}} \tag{1}$$

For the fusion of multiple sources of information we often require weights. The weights we employed for this task were dynamically generated at query-time and reflect the degree to which we believe one source of information will provide better performance as opposed to the other sources we have. This process is briefly described in [12], and more fully described in [13].

Future work will involve re-examining these approaches and measuring the impact they have when used with semantic features. We now highlight the order in which our fusion was performed, as illustrated by Figure 1.

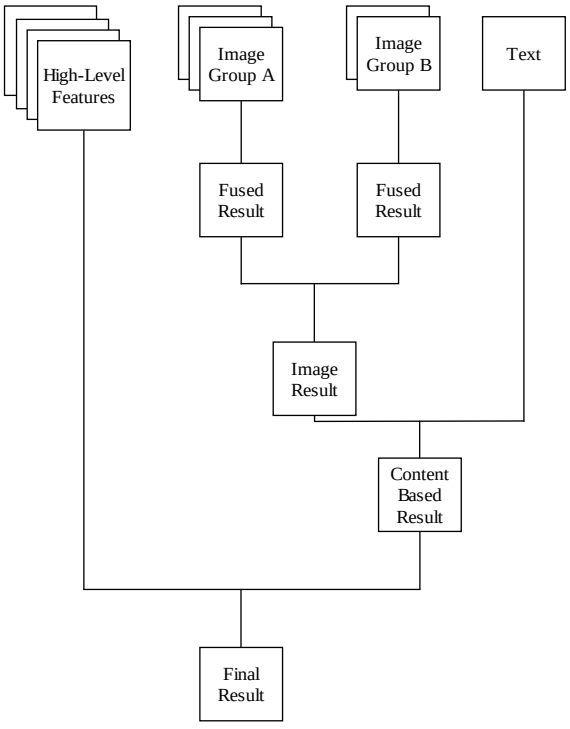

Figure 1: Example framework diagram

We begin with the low-level features. For each example query image we fuse together the results of each low-level search, for instance given an example query image we would fuse together the results of the colour search, edge search and texture search for that image, such that we then had a single visual result list which represented the combined visual search for that image. Second, we then combined

Table 1: An overview of the runs in the automatic search task. Search cues marked with • are used in the corresponding run, cues marked with ◦ are used only with specific queries.

runId	trType	text	visual	motion	concepts	std+ent	MAP
F_A_1_DCU-Base_6	A	•					0.013
F_A_2_DCU-TextVisual_2	A	◦	•				0.026
F_A_2_DCU-TextVisMotion_5	A	◦	•	•			0.024
F_B_2_DCU-Visual-LSCOM_3	B	◦	•		•		0.032
F_B_2_DCU-LSCOM-Filters_4	B	◦	•	•	•		0.031
F_B_2_DCU-LSCOM-2_1	B	◦	•	•	•	•	0.031

the results of each of the example query images searches into one single visual result. Third, we then combined the single visual result with the text search results. At this stage we are left with a single result list which is the combined output of all the previous content-based searches for that topic.

If our run is to employ High-Level semantic features, it is at this stage that they are applied. The semantic features we utilized are described in Section 3.2.

Semantic features were used only as a means to alter the final ranking achieved by the content-based search. This was achieved through a basic filtering approach. Each semantic feature we used provided a score for every shot in the collection as to that shot's likelihood of being a positive example of that semantic feature. Through previous testing we defined a score threshold for each feature, such that shots above this threshold could be said to be definite or 'positive' examples of that semantic feature, whilst shots below the threshold could be seen as 'negative' examples of that semantic feature.

In the use of 'positive' filtering we examined the content-based search result list and if a shot in that result list was found in our 'positive' filter (that is the shot occurred above the threshold we defined for that semantic feature), then we increased the score of that shot by 10%. Conversely, if we were using a semantic feature as a 'negative' example, then if the candidate shot we were examining occurred below the threshold, it's score too was increased by 10%. To clarify, if a shot was found in a positive filter we increased the value of that shot. If a shot was found in a negative filter, we increased the value of that shot because it did not contain the semantic feature we were using. For example if we used a negative 'Studio' filter, then shots which were below the threshold could be seen to not contain a 'Studio' and therefore had their scores boosted, thus possibly raising them up the ranking over shots which did contain a 'Studio'.

3 Automatic Search Experiments

We submitted a total of six runs for the automatic search task. An overview of the runs is given in Table 1. In the experiments we first examined the addition of a motion feature to the baseline runs of using only text and using text and visual features. The run F_A_1_DCU-Base_6 constitutes the required baseline run using only the query text and text features. Second, we incorporated a set of matching pre-existing semantic concept models to the search process. Third, we examined the inclusion of two general concepts (*News Studio* and *Entertainment*) to all search topics as negative concepts to reduce the rankings of shots containing corresponding contents. These experiments are described in more detail below. The results of the experiments are then discussed in Section 3.3.

3.1 Text and visual features

Our baseline run was a pure text-only run against the ASR index. We did not index the closed caption resources available. The query for each topic was the topic description as provided by NIST, we did not employ any query or term expansion. For each shot that was found we applied a basic windowing scheme such that the preceding and following two shots were added into the result set and given a score which was a reduced value of the original shot's score.

For our text and visual run, we combined a selected output of the text result with the output of a visual features. Selected output means that the text result was used only for queries where there was a named entity in the topic description. This choice was based on the work of IBM's TRECVid

submission from 2005 [4]. These were then combined with a visual search, using features as described in Section 2.1.2. Furthermore we then also performed another run that used not only the low-level visual data, but also incorporated motion information.

3.2 Semantic concepts

Next, we performed experiments in augmenting the search with previously constructed semantic concept models. For these experiments, we used the Large Scale Concept Ontology for Multimedia (LSCOM) [5]. LSCOM is an expanded multimedia concept lexicon in development, aimed to contain on the order of 1000 concepts. The current version 1.0 has 856 concepts defined, of which 449 have been used to annotate the TRECVid 2005 development set with a collaborative annotation process.

For the LSCOM concepts, we used concept models based on Self-Organizing Maps (SOMs) described in [11]. The semantic concepts are represented as class models on a set of parallel SOMs trained with multimodal low-level features. More detailed descriptions of this method are given in [8, 11]. In these experiments, we used lists of 2000 best shots for each concept. These were then used as described in Section 2.1.2.

The concept models were included in the search as described. After processing all matching concepts, we sorted the shots in the test collection based on this new qualification value and returned the 1000 best-scoring shots as the result for the query.

The topics in the search task were matched with the semantic concepts using the textual query and synonyms found using WordNet. The query text was first preprocessed and stemmed, and unnecessary words were discarded by using a stoplist and removing the standard beginning of the query phrase. For details on this procedure, see [11]. The matched LSCOM concepts for each search topic are listed in Table 2. The LSCOM annotations were deemed additional data on the common development collection, so the incorporation of these concept models makes the corresponding runs of type B submissions.

For the matching LSCOM concepts, we used separately trained concept models based on Self-Organizing Maps (SOMs). In this approach, the semantic concepts are represented as class models on a set of parallel SOMs trained with multimodal low-level features. More detailed descriptions of this method are given in [8, 11]. In these experiments, we used lists of 2000 highest-scoring shots returned by the detector for each concept as binary classifications on the presence of the concept in question.

The concept models were included in the search as follows. For each concept matching the query text, we mark the shots in the associated list of 2000 best-scoring shots. These shots are awarded a boost of x in their qualification value. Shots associated with concepts marked as negative receive a corresponding negative modifier to their qualification values. After processing all matching concepts, we sort the shots in the test collection based on this new qualification value and return the ranked list of 1000 best-scoring shots as the result for the query.

In the sixth and final experiment, we included the concepts *News Studio* and *Entertainment* as global negative concepts for all queries using the same procedure in order to reduce the number of returned shots containing a studio or general entertainment setting.

3.3 Results

In this section, the results of the experiments are discussed. As an overview, the mean average precision (MAP) values for our runs are shown in Table 1. Based on the MAP values, it can first of all be observed that using the visual features (`F_A_2_DCU-TextVisual_2`) improves the overall results compared to the text-only baseline (`F_A_1_DCU-Base_6`). On the other hand, the addition of the motion feature (`F_A_2_DCU-TextVisMotion_5`) does not seem to improve the results. Second, we can observe an improvement in the results when the LSCOM concepts are incorporated. The MAP values of the three runs utilizing the concept models are very close to each other, indicating that neither the addition of the motion feature (`F_B_2_DCU-LSCOM-Filters_4`) nor the negative *News Studio* and *Entertainment* concepts (`F_B_2_DCU-LSCOM-2_1`) lead to overall improvement of the results compared to the run utilizing only the text and visual features along with the concept models (`F_B_2_DCU-Visual-LSCOM_3`).

Compared to the other submissions for the automatic search task, the MAP scores of our runs are rather modest. The median of all submitted automatic runs (76 in total) was 0.034, whereas the best single submission obtained a MAP of 0.087. More strikingly, the median and maximum of the

Table 2: LSCOM concepts used for each search topic, concepts listed in italics are used as negatives.

Topic	LSCOM Concepts
173 : emergency vehicles	Emergency Vehicles, Ground Vehicles, Police, Vehicle, Explosion Fire, Police Private Security Personnel
174 : tall buildings	Building
175 : leaving or entering vehicle	Ground Vehicles, Vehicle
176 : escorting prisoner	Guard, Police, Soldiers, Police Private Security Personnel, Prisoner
177 : demonstration or protest	Daytime Outdoor, Demonstration Or Protest, Building, People Marching
178 : Dick Cheney	Head Of State, Face, Government Leader, Person
179 : Saddam Hussein	Face, Person
180 : in uniform and in formation	Military Personnel
181 : George W. Bush	George Bush, Head Of State, Walking, Face, Government Leader, Person, Walking Running
182 : soldiers or police	Armored Vehicles, Emergency Vehicles, Ground Vehicles, Police, Soldiers, Vehicle, Weapons, Military Personnel, Police Private Security Personnel
183 : water with boats	Ship, Boat Ship, Waterscape Waterfront
184 : seated at computer	Computers, Sitting, Computer Or Television Screens
185 : reading newspaper	Newspapers
186 : natural scene	Beach, Lakes, Lawn, Oceans, River, Trees, Animal, Mountain, Sky, Vegetation, *Ground Vehicles*, *Vehicle*, *Building*, *Road*
187 : helicopters in flight	Flying Objects, Helicopters
188 : burning with flames	Explosion Fire
189 : seated group in suits and flag	Flags, Group, Sitting, Suits
190 : person and books	Person
191 : adult and child	Adult, Child, Person
192 : kiss on the cheek	Greeting
193 : smokestacks or chimneys	Smoke, Smoke Stack, Tower
194 : Condoleeza Rice	Face, Person
195 : soccer goalposts	Soccer
196 : snow	Snow

text-only baseline runs were 0.036 and 0.048, respectively, indicating that our text features suffered from poor performance.

The conclusions about the relative performances of our runs are not as straightforward, however, when we consider the topicwise results for the runs.

The average precision (AP) results from the six runs for each topic are shown in Figure 2. It can be immediately seen that the relative performances of the runs vary considerably; most of the search topics are extremely challenging for automatic systems and thus, unsurprisingly, score low AP values, whereas the topic 195 has clearly distinct AP scores (maximum of 0.56). It can also be seen that the resulting MAP values are strongly influenced by a small number of topics with higher AP scores.

The text baseline (`F_A_1_DCU-Base_6`), despite having the lowest MAP value, actually scores the highest AP values for 10 of the 24 topics, in some cases (e.g. topics 177, 182, 188, and 196; brief descriptions of the topics are given in Table 2) even with a considerable margin. Furthermore, among these topics there is wide variation; some are specific queries having proper nouns in the textual query (topics 181 and 194) whereas some other topics can be considered rather visual (e.g. topics 188 and 196).

On the topicwise level, the addition of the LSCOM concepts again leads to mixed results. As a general rule, it can be observed that the LSCOM concepts do not seem to work well with specific

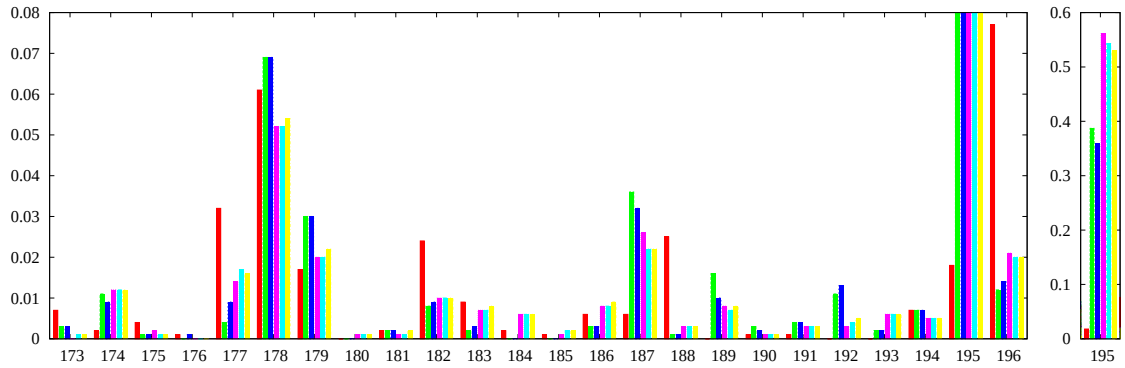

Figure 2: Average precision results for each topic. The six bars for each topic from left to right correspond to the runs listed in Table 1. Topic 195 is shown in correct scale separately on the right.

topics. This is understandable as the concepts are probably too generic for specific queries. For the generic topics it is difficult to find any common explanations for the relative performances with and without the concept models. Adding the concepts degrades results with some of the topics, but the majority of the generic topics still seems to benefit from the additional concepts. Overall, it can be observed that, at least in some cases, it is possible to improve search results by using existing models of semantic concepts even in an automatic setting.

4 Conclusions

In our submission to TRECVid 2006, we adapted a traditional retrieval system. The system was used in the automatic search experiments described in this paper and in the manual runs [12] resulting from the combined effort of the participants in the K-Space project.

Rather surprisingly, the ASR/MT text features, even with a low mean performance compared to the text-only baseline submissions of the other participating groups, gave best results for many topics. This seems to indicate that text should be considered a valuable information source also for general topics, at least with such difficult topics that were included in this year's search task. Based on our experiments, the benefit of using the LSCOM concepts remains inconclusive. This might be caused at least by a number of issues requiring further attention. First, the reliability of the concepts models was not considered in these experiments, although it is relatively easy to estimate based on the training data. Based on these estimates, those concepts that are easier to model should undoubtedly be given more weight. Also, the confidence values returned by the concept detectors could be utilized instead of applying a binary classification. Another question is the suitability of different concepts to a given query. With automatic methods, such as the simple keyword mapping used in these experiments, this is a very difficult problem. Even for human users it is generally difficult to assess whether a given semantic concept model is likely to be helpful in a query without an extensive knowledge of both the retrieval system and target material.

Acknowledgements

This work was partially supported by the Irish Research Council for Science Engineering and Technology, Science Foundation Ireland under grant 03/IN.3/I361 and by the European Commission under contract FP6-027026 (K-Space). We are grateful to the AceMedia project (FP6-001765) which provided us with output from the AceToolbox image analysis tooklit.

References

[1] KSpace Network of Excellence, information at http://www.k-space.eu/.

[2] The Zettair search engine, available from http://www.seg.rmit.edu.au/zettair/.

[3] The AceMedia Project, available at http://www.acemedia.org.

[4] A. Amir, J. Argillander, M. Campbell, A. Haubold, G. Iyengar, S. Ebadollahi, F. Kang, M. R. Naphade, A. Natsev, J. R. Smith, J. Tei, and T. Volkmer. Ibm research trecvid2005 video retrieval system. In *TREC Video Retrieval Evaluation Proceedings*, 2006.

[5] DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia. LSCOM lexicon definitions and annotations version 1.0. Technical Report #217-2006-3, Columbia University, March 2006.

[6] C. Foley, C. Gurrin, G. Jones, H. Lee, S. McGivney, N. O'Connor, S. Sav, A. F. Smeaton, and P. Wilkins. TRECVid 2005 experiments at Dublin City University. In *TREC Video Retrieval Evaluation Online Proceedings*. TRECVID, November 2005.

[7] E. A. Fox and J. A. Shaw. Combination of Multiple Searches. In *Proceedings of the 2nd Text REtrieval Conference*, 1994.

[8] M. Koskela and J. Laaksonen. Semantic concept detection from news videos with self-organizing maps. In *Proceedings of 3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI 2006)*, Athens, Greece, June 2006.

[9] K. McDonald and A. F. Smeaton. A comparison of score, rank and probability-based fusion methods for video shot retrieval. In *Proceedings of CIVR 2005*, 2005.

[10] N. O'Connor, E. Cooke, H. le Borgne, M. Blighe, and T. Adamek. The AceToolbox: Low-Level Audiovisual Feature Extraction for Retrieval and Classification. In *2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, 2005.

[11] M. Sjöberg, H. Muurinen, J. Laaksonen, and M. Koskela. PicSOM experiments in TRECVID 2006. In *TREC Video Retrieval Evaluation Online Proceedings*. TRECVID, November 2006.

[12] P. Wilkins and et al. KSpace at TRECVid 2006. In *TRECVid 2006 – Text REtrieval Conference, TRECVID Workshop, Gaithersburg, Md., 13-14 November 2006*, 2006.

[13] P. Wilkins, P. Ferguson, and A. F. Smeaton. Using score distributions for querytime fusion in multimedia retrieval. In *MIR 2006 - 8th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2006.