Video Retrieval using Global Features in Keyframes

Marcus J Pickering[1], Daniel Heesch[1], Robert O'Callaghan[2], Stefan Rüger[1] and David Bull[2]

[1] Department of Computing, Imperial College London,
180 Queen's Gate, LONDON SW7 2BZ, UK
{m.pickering,dh500,srueger}@doc.ic.ac.uk
[2] Image Communications Group, Centre for Communications Research, University of Bristol,
Woodland Road, BRISTOL BS8 1UB, UK
{r.j.ocallaghan,dave.bull}@bristol.ac.uk

Abstract. We describe our experiments for the shot-boundary detection and search tasks for the TREC-11 video track. Our shot-boundary detection scheme is based on a multi-timescale detection algorithm in which colour histogram differences are examined over a range of frames. Our search efforts are based on a system which brings together a number of global features encompassing colour, texture and text features derived from speech recognition transcripts into a unique relevance feedback system.

1 Introduction

Early attempts at content-based video retrieval were based on keywords attached to shots, and this proved very effective for video types such as broadcast news [2, 10]. However, we now live in a multimedia world and, as demonstrated by the search topics for this year's video track, there is potential for querying video in many different ways.

In this paper, we present our system of retrieval of video shots based on global features derived from keyframes. The keyframes are the output of a shot boundary detection process, which we describe in Section 2. Our search system with relevance feedback is described in Section 3.

2 Shot boundary detection task

2.1 System

The video shot boundary detection algorithm is broadly based on the colour histogram method, where the colour histograms for consecutive frames are compared and, if their difference is greater than a given threshold, a shot change is declared. This method is extended, based on the algorithm of Pye et al [12] 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.

Each frame is divided into 9 blocks, and for each block a histogram is determined for each of the RGB components. The Manhattan distance between corresponding component histograms for corresponding blocks in two frames is calculated, and the largest of the three is taken as the distance for that block. The distance between two frames is then taken as the median of the 9 block distances. This helps eliminate response to local motion.

A difference measure is defined as follows:

$$d_n\left(t\right) = \frac{1}{n}\sum_{i=0}^{n-1} D\left(t+i, t-n+i\right),$$

where $D(i, j)$ represents the median block distance between frames i and j.

If, at frame f, the value for $d_{16}(f)$ is greater than an empirically determined threshold T_{16}, the frame is examined for the presence of a shot change.

A cut is declared at frame f if the following conditions hold:
- $d_8(f) > T_8$ (where $T_8 = 0.4T_{16}$)
- $d_n(f) > d_n(f + \delta)$ for all $n \in \{2, 4, 8\}$ and all $\delta \in \{-2, -1, 1, 2\}$ (cuts show characteristic coincident peaks for all d_n).
- $d_n > 1.3d_{n/2}$

If no cut was declared, a gradual transition is declared if the following conditions hold:
- $d_{16}(f) > d_{16}(f \pm \delta)$ for all $\delta \leq 16$
- Peak value of d_8 in range $f \pm 16$ occurs within $f \pm 5$

In order to determine the start and end points for gradual transitions, we employ a method similar to that described by Zhang [19], in which a lower threshold, T_4, is used to test for the start and end of a gradual transition. At each frame, the d_4 difference is compared to the threshold. If $d_4 < T_4$ then the frame is marked as a potential start of a transition. If, on examination of successive frames, d_4 falls below T_4 again before a shot change is detected, this potential start is scrapped and the search continues. Following the detection of a shot change, the end point of the transition is declared as the point at which d_4 first falls below the threshold again, following the shot change. The d_4 timescale is used because it is fine enough to pinpoint accurately the moment at which the change begins, but also introduces a tolerance to any momentary drop in the difference which may occur in the process of the change.

It has been suggested that automatic threshold setting can improve performance [13], but we found no empirical evidence to support this with our algorithm. We were, however, able to improve on our TREC-10 system [11] by using empirical data to determine the cut and gradual definition rules.

2.2 Experiments

We performed six shot boundary detection runs. The first three runs, KM-01 – KM-03 were carried out keeping the low threshold, T_4, constant, and reducing the high threshold, T_{16}. In runs KM-04 – KM-06, the low threshold was increased and the same 3 values for the high threshold were used again.

2.3 Results

	All		Cuts		Gradual			
	Recall	Prec	Recall	Prec	Recall	Prec	F-Recall	F-Prec
KM-01	0.826	0.843	0.883	0.895	0.682	0.707	0.673	0.608
KM-02	0.845	0.798	0.889	0.863	0.733	0.648	0.658	0.618
KM-03	0.859	0.720	0.893	0.803	0.773	0.553	0.650	0.612
KM-04	0.825	0.813	0.888	0.880	0.665	0.645	0.471	0.603
KM-05	0.833	0.755	0.891	0.832	0.685	0.578	0.477	0.444
KM-06	0.836	0.688	0.885	0.755	0.711	0.536	0.477	0.356
TREC Avg	0.760	0.790	0.852	0.835	0.527	0.603	0.551	0.713

Table 1. Shot boundary detection task – results summary

We show the results for our six shot-boundary detection runs in Table 1. All six runs gave good results for overall precision and recall, comparing favourably with the average of all systems (shown as "TREC Avg" in Table 1). System KM-01 appeared to give the best balance between precision and recall overall, suggesting that further experiments with a higher T_{16} threshold may be worthwhile.

The frame-recall and frame-precision results (F-Recall and F-Prec respectively in Table 1) give an indication of the accuracy of the system for detection of gradual transitions. Our relative performance here was not as good, and this perhaps reflects the fact that little time was devoted to tuning the algorithm for setting the start and end times of gradual transitions. Results could almost certainly be improved here by adjusting the parameters of this algorithm.

3 Search task

3.1 Overview

For each shot of each video, we take a representative *keyframe*, defined as the middle frame of the shot. The shot boundaries were prescribed by NIST for the task. For each keyframe, a number of feature vectors are pre-computed. The descriptors are then combined in an integrated retrieval model such that the overall distance between a query set $\hat{Q}$ and an image T is given by a convex combination of the distance values computed for each descriptor.

$$D(\hat{Q}, T) = \sum_{d} w_d \mathrm{Dis}_d(\hat{Q}, T) \tag{1}$$

where $\mathrm{Dis}_d(\hat{Q}, T)$ denotes the distance for descriptor d between query set $\hat{Q}$ and T, $w_d \in [0, 1]$ and $\sum_d w_d = 1$. The descriptor-specific distance values are computed using the k-nn method, described in Section 3.3.

We combined 6 features, described in the next section. Following retrieval, the user has the option to apply relevance feedback, through a system described in section 3.4.

3.2 Features

HSV Colour Histograms. Retrieval from image databases using only colour was one of the earliest content-based retrieval methods [4, 9, 14]. There is an abundance of colour spaces [5, 16, 18], virtually all of which are 3-dimensional owing to the human perception of light using three different cones as receptors in the retina. Colour histograms are quantised distributions in the 3-dimensional colour space of all pixels of one image. The corresponding feature vector is a list of the proportions of pixels which fall into the respective 3-dimensional colour bins; its length depends on the granularity of the colour bins. Here, we do *not* use 1-dimensional component-wise histograms since (as with all marginalisations) information about the underlying colours would be lost.

HSV [16] seems to be intuitive to humans. The hue coordinate H encodes the underlying pure colour tone of a colour circle. The saturation S reflects the pureness of the colour (the less pure the colour the more grey is mixed into it, S is zero for greys). V and L are both measures, albeit differently defined, for the apparent brightness or luminosity. When expressing the difference of two colours humans tend to use HSL or HSV coordinates ("more in direction of magenta", "purer than", "darker than") rather than RGB components.

HSV and HSL are both cylindrical colour spaces with H being the angular, S the radial and V or L the height component. This brings about the mathematical disadvantage that hue is discontinuous wrt RGB coordinates and that hue is singular at the achromatic axis $r = g = b$ or $s = 0$. As a consequence we merge, for each brightness subdivision separately, all pie-shaped 3-d HSV bins which contain or border $s = 0$. The merged cylindrical bins around the achromatic axis describe the grey values which appear in a colour image and take care of the hue singularity at $s = 0$. Saturation is essentially singular at the black point in the HSV model and at both black and white points in the HSL model. Hence, a small RGB ball around black should be mapped into the bin corresponding to $hsv = hsl = (0, 0, 0)$, or $hsl = (0, 0, 1)$ respectively for white, to avoid jumps in the saturation from 0 to its maximum of 1 when varying the singular RGB point infinitesimally. There are several possibilities for a natural subdivision of the hue, saturation and brightness axes; they can be i) subdivided linearly, ii) so that the geometric volumes are constant in the cylinder and iii) so that the volumes of the nonlinear transformed RGB colour space are nearly constant. The latter refers to the property that few RGB pixels map onto a small dark V band but many more to a bright V interval of the same size; this is sometimes called the HSV cone in the literature. We use the HSV model with a linear subdivision.

Convolution filters. For this feature we use Tieu and Viola's method [15], which depends on the definition of highly selective features that are determined by the structure of the image, as well as capturing information about colour, texture and edges. By defining a vast set of features, each feature is such that it will only have a high value for a small proportion of images, and by discovering a number of features which distinguish the example set in question we are able to perform an effective search.

The feature generation process is based on a set of 25 primitive filters, which are applied to each of the three colour channels to generate 75 feature maps. Each of these feature maps is rectified and downsampled before being fed again to each of the 25 filters to give 1875 feature maps. The process is repeated a third time, and then each feature map is summed to give 46,875 feature values. The idea behind the three stage process is that each level 'discovers' arrangements of features in the previous level. The feature generation process is computationally quite costly, but only needs to be done once and then the feature values can be stored with the image in the database.

Text. Our text feature is derived from the speech recognition transcripts supplied by Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI). A full text index was built using the Managing Gigabytes search engine [1] and queries formed from the XML data supplied with each query. Managing Gigabytes supplies a numerical relevance value which is used when weighing features.

HMMD Colour Histogram. The recently introduced HMMD (Hue, Min, Max, Diff) colour space, which is used in the MPEG-7 standard, is derived from the HSV and RGB spaces. The hue component is the same as in the HSV space, and max and min denote the maximum and minimum among the R, G, and B values, respectively. The diff component is defined as the difference between max and min. Three components are sufficient to uniquely locate a point in the colour space and thus the space is effectively three-dimensional. Following the MPEG-7 standard, we quantize the HMMD non-uniformly into 184 bins with the three dimensions being Hue, Sum and Diff (sum being defined as $(max + min)/2$) and use a global histogram. See Manjunath and Ohm [6] for details about quantization.

Colour Structure Descriptor. We use a second descriptor defined in HMMD space that lends itself better to capturing local image structure. A 8×8 structuring window is used to slide over the image. Each of the 184 bins of the HMMD histogram contains the number of window positions for which there is at least one pixel falling into the bin under consideration. This descriptor is capable of discriminating between images that have the same global colour distribution but different local colour structures. Although the number of samples in the 8×8 structuring window is kept constant (64), the spatial extent of the window differs depending on the size of the image. Thus, for larger images appropriate sub-sampling is employed to keep the total number of samples per image roughly constant. The bin values are normalized by dividing by the number of locations of the structuring window and fall in the range $[0, 1]$ (see Manjunath and Ohm [6] for details).

Illumination Invariant Colour Descriptors. Recognition and retrieval via colour are heavily influenced by variations in the scene illumination conditions. This places undesirable limitations on the use of raw colour features in content-based applications. In an attempt to attain some robustness to variation in lighting conditions, we use the set of illumination-invariant descriptors defined by O'Callaghan and Bull [8]. These are histogram, rather than pixel, based features and are calculated using invariant moments of the distribution in RGB space. In the current implementation, they are applied on a global basis to each key-frame. As such, they provide a small number of features (specifically 21), which describe the colour distribution of the scene, invariant to changes in the colour and intensity of the illuminant. Spatial variation of the illumination over the scene is neglected and a diagonal model of illumination change is assumed.

The utility of these colour descriptors was previously demonstrated by O'Callaghan and Bull [8] on a constrained dataset [3], of images of colourful man-made objects, under varying illumination. One of the objectives of our search task submission was to evaluate the performance of such features in a "real" retrieval application in comparison with conventional methods.

3.3 Retrieval using k-nearest neighbours

Retrieval is performed using the k-nearest neighbour approach, which is based on the intuitive notion that if we have seen and already identified something, then anything we later see that is the same, or similar, (based on some defined characteristics) is probably the same kind of thing. So, we provide positively and negatively classified examples, and then classify all test images according to their proximity to the examples.

We use a variant of the distance-weighted k-nearest neighbour approach [7]. Positive examples are supplied by the user, and a number of negative examples are randomly selected from the database. The distances, for descriptor d, from the test image T_i to each of the k nearest positive or negative examples (where 'nearest' is defined by the Euclidean distance in feature space) are determined, and a distance measure calculated as follows:

$$\mathrm{Dis}_d(\hat{Q}, T_i) = \frac{\sum_{n \in N} (\mathrm{dist}(T_i, n) + \varepsilon)^{-1}}{\sum_{q \in \hat{Q}} (\mathrm{dist}(T_i, q) + \varepsilon)^{-1} + \varepsilon}$$

where $\hat{Q}$ and N are the sets of positive and negative examples respectively amongst the k nearest neighbours, such that $|\hat{Q}| + |N| = k$. ε is a small positive number to avoid division by zero. Images are ranked according to $\mathrm{Dis}_d(\hat{Q}, T_i)$.

3.4 Relevance feedback

Retrieved images $T_1, T_2, \ldots$ are displayed as thumbnails such that their respective distance from the centre of the screen is proportional to the dissimilarity $D_s(\hat{Q}, T_i)$ (given by Equation 1) of thumbnail T_i to the query set $\hat{Q}$. Using this semantics of thumbnail location on the screen, the user can provide relevance feedback by moving thumbnails closer to the centre (meaning they are more relevant than the system predicted) or further away (indicating less relevance). The user effectively supplies the system with a real vector of distances $D_u(\hat{Q}, T_i)$, which, in general, differ from the distances $D_s(\hat{Q}, T_i)$ which the system computes using the set of weights w_d. The sum of squared errors

$$\begin{aligned} \mathrm{SSE}(w) &= \sum_{i=1}^{N} \left[D_s(\hat{Q}, T_i) - D_u(\hat{Q}, T_i) \right]^2 \\ &= \sum_{i=1}^{N} \left[\sum_{d} w_d \mathrm{Dis}_d(\hat{Q}, T_i) - D_u(\hat{Q}, T_i) \right]^2 \end{aligned} \tag{2}$$

gives rise to an optimisation problem for the weights w_d such that (2) is minimised under the constraint of convexity. Using one Lagrangian multiplier we arrive at an analytical solution w' for the weight set which changes the distance function. We get a different ranking of images in the database and, with (1), a new layout for the new set of top-retrieved images on the screen.

3.5 Experiments

We carried out four runs to investigate the effects of various combinations of features and of relevance feedback:

1. All features + using relevance feedback.
2. Illumination invariant, Text and Convolution features only.
3. All features. (Baseline for run 1).
4. CSD, Text and Convolution features only. (Baseline for run 2).

3.6 Results

Topic	I_B_KM-1_1	M_B_KM-2_2	M_B_KM-3_3	M_B_KM-4_4
75	0.172	0.146	0.142	0.146
76	0.487	0.540	0.545	0.442
78	0.000	0.188	0.000	0.172
80	0.081	0.009	0.146	0.071
81	0.138	0.000	0.000	0.000
83	0.133	0.028	0.000	0.024
84	0.260	0.250	0.050	0.258
92	0.121	0.021	0.011	0.033

Table 2. Search task – results for topics for which at least one variant of our system achieved average precision greater than 0.100

In Table 3.6 we show the results for topics for which our system achieved average precision greater than 0.100. The topics for which we achieved our best results are, at first glance, surprising – topics 75 and 76 were both queries requiring specific personalities, Eddie Rickenbacker and James H Chandler. Our system was not designed to detect faces. However, both queries contained film of quite distinctive colouring, and the Chandler query contained query shots from within the test set.

Using our four runs we hoped to show that relevance feedback improved performance and that the use of illumination invariant features improved performance, but results were not completely conclusive for either hypothesis.

As we carried out our interactive (relevance feedback) run (I_B_KM-1_1) the retrieved shots were certainly visually much better with each round of relevance feedback, though this is not spectacularly clear from the numerical results. There was some improvement in the average precision for most topics, an observation which is reinforced by the 95% confidence interval for the difference between the performance means of the results for this run and its baseline (M_B_KM-3_3) which, while not proving statistical significance, does suggest an improvement in performance using relevance feedback. The perceived performance improvement may simply be due to the fact that relevance feedback re-ordered the rankings – and with better top-ranked results the user's overall impression is one of greater satisfaction. Some topics (for example 81 – football players, 83 – Golden Gate Bridge, 88 – US maps) benefitted significantly from the application of relevance feedback.The interactive runs in TREC model a search scenario where someone, such as a librarian, searches on behalf of someone else who ultimately judges the returned results. This is different from our model of relevance feedback where the searcher is the one who judges and uses the results. It is also important to note that a user may often be more content with one or two good results, highly ranked, than with retrieving every relevant item in the database.

Calculation of the 95% confidence interval for the difference between the means of the results of the illumination invariant run (M_B_KM-2_2) and its baseline (M_B_KM-4_4) showed that the introduction of the illumination invariant feature brought about no overall improvement in results. However, performance was improved in a number of specific topics, for example, topics 90 – snow covered mountains, and 91 – parrot.

With hindsight, the experiments could have been better designed; in some cases the limited number of features used in the second run (run M_B_KM-2_2) performed better than the combination of all features (run M_B_KM-3_3), suggesting that a philosophy of "more features is better" does not necessarily hold. Some further experiments could be carried out to discover which combinations of features work best – whether there are some features that are consistently good and some that are consistently unhelpful, and whether some features facilitate good results in the presence or absence of other particular features.

4 Conclusions

Our shot boundary detection scheme was shown to work very effectively, with particularly good performance on cuts. There is some potential for improvement of its accuracy on gradual transitions, though detection was good and well above average.

In the search task, we have shown that the use of global features in keyframes, with the addition of relevance feedback, can make an effective contribution to retrieval. Although the recall levels were not spectacular, subjectively the top-ranked results were good on many of the topics.

Further experiments are required to determine which are the best combinations of features, and which features contribute significantly (positively or negatively) to the retrieved results.

Acknowledgements: This work was partially supported by the EPSRC, UK.

References

1. Managing Gigabytes search engine. http://www.cs.mu.oz.au/mg/.
2. M. G. Brown, J. T. Foote, G. J. F. Jones, K. Spärck-Jones, and S. J. Young. Automatic content-based retrieval of broadcast news. In *Proceedings of the Third ACM Multimedia Conference*, Apr. 1995.
3. Computational Vision Lab - Simon Fraser University. Data for computer vision and computational colour science. Available: http://www.cs.sfu.ca/~colour/data/.
4. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.
5. R. Jackson, L. MacDonald, and K. Freeman. *Computer Generated Colour*. Wiley, 1994.
6. B. S. Manjunath and J.-S. Ohm. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11:703–715, 2001.
7. T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.
8. R. J. O'Callaghan and D. R. Bull. Improved illumination-invariant descriptors for robust colour object recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 2002.
9. A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases, 1994.
10. M. J. Pickering. Video archiving and retrieval. http://km.doc.ic.ac.uk/video-se/, 2000.
11. M. J. Pickering and S. M. Rüger. Multi-timescale video shot-change detection. In Voorhees and Harman [17].
12. D. Pye, N. J. Hollinghurst, T. J. Mills, and K. R. Wood. Audio-visual segmentation for content-based retrieval. In *5th International Conference on Spoken Language Processing, Sydney, Australia*, Dec. 1998.
13. J. R. Smith, S. Srinivasan, A. Amir, S. Basu, G. Iyengar, C.-Y. Lin, M. Naphade, D. Ponceleon, and B. Tseng. Integrating features, models, and semantics for TREC video retrieval. In Voorhees and Harman [17].
14. M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
15. K. Tieu and P. Viola. Boosting image retrieval. In *5th International Conference on Spoken Language Processing*, Dec. 2000.
16. D. Travis. *Effective Color Display*. Academic Press, San Diego, CA, 1991.
17. E. M. Voorhees and D. Harman, editors. *Proceedings of the Tenth Text REtrieval Conference (TREC-10)*, 2001.
18. G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas*. Wiley, 2nd edition, 1982.
19. H. J. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *ACM Multimedia Systems*, 1:10–28, 1993.