

Video-Cuebik: Adapting Image Search to Video Shots

Alexander G. Hauptmann
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
+1 412 268 1448
alex@cs.cmu.edu

Norman D. Papernick
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
+1 412 268 7557
norm@cs.cmu.edu

ABSTRACT

We propose a new analysis for searching images in video libraries that goes beyond simple image search, which compares one still image frame to another. The key idea is to expand the definition of an image to account for the variability in the sequence of video frames that comprise a shot. A first implementation of this method for a QBIC-like image search engine shows a clear improvement over still image search. A combination of the traditional still image search and the new video image search provided the overall best results on the TREC video retrieval evaluation data.

Keywords

Image retrieval, video search, Query By Image Content.

INTRODUCTION

Content-based image retrieval [1] requires an image search engine to find the set of images from a given image collection that is similar to the given query image, where similarity is very subjective. Most research in image retrieval emphasizes features and invariants of single images. Video matching [5] usually involves comparing two sets of extracted features emphasizing motion and color. Little attention has been paid to finding a video sequence based on a single-frame still image.

The QBIC Query by Image Content System and Cuebik

The QBIC system [2] developed at IBM is the standard benchmark for scalable, accurate and efficient image retrieval performance. For the purpose of a simple and clear experiment, our implementation of QBIC (aka Cuebik) simplifies the color space to 256 colors, with only one color (instead of a color histogram) in each of 256 regions based on a 16x16 grid. The partitioning of the color space is derived from the complete RGB histogram of colors for the total collection of images, where the 256 most frequently occurring colors (in RGB space) are chosen as the color subset for that region. The color for a region is determined by the most frequently occurring color in that region which is closest to one of the 256 dominant collection colors.

Let images A and B each be represented as a 256 element region vector, where each region has exactly one color. An image search compares the regions of the new image to every image in the collection. The number of regions that match between a query image and a collection image constitute the similarity to this image in the database.

$$match(a_i, b_i) = \begin{cases} 1 & \text{if } a_i = b_i \\ 0 & \text{if } a_i \neq b_i \end{cases}$$

$$Similarity(A, B) = \sum_{i=0}^{255} match(a_i, b_i)$$

Adapting Cuebik to video search

When examining the image retrieval results from Cuebik, we noticed that a translation of an image by 1 region could result in a completely failed match. This type of translation would occur frequently through a camera pan and zoom or when an object moved slightly from its original position. To make Cuebik more suitable for video images, we decided to allow multiple colors in one region, but only if those colors occurred in that region during the sequence of frames comprising a single camera shot in the video collection. Modeling the range of possible colors in a region over the duration of a shot allows a better approximate match between an image and a video sequence that has roughly the same color composition, despite region mismatches due to object motion or camera pan and tilt. If S is a camera shot comprised of n image frames, let $s_{i,j}$ be the color of region i in frame j, we define shot match and shot similarity as:

$$smatch(a_i, s_i) \equiv \arg \max_{j \in [1..n]} \{ match(a_i, s_{i,j}) \}$$

$$Similarity(A, S) = \sum_{i=0}^{255} smatch(a_i, s_i)$$

Experiments with The TREC VIDEO Collection

The video collection used in our experiment originated from the video retrieval evaluation track in TREC10 [3].

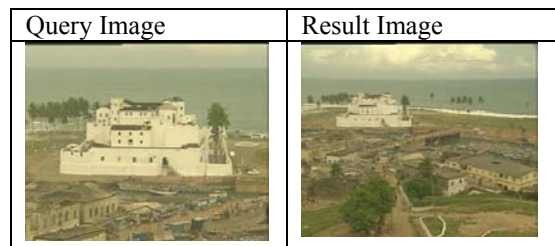


Figure 1. Sample query and result images for Topic33: "Looking for additional shots of this white fort".

It contains about 11 hours of video with approximately 80,000 I-frame images and 8000 shots. There are 32 "known-item" queries, where a complete set of target answers to the queries had been manually marked. Each

video query is consisted of an image or video example and a text description of the desired topic. Systems retrieved up to 100 video shots believed to be similar to the query example. By comparing against the manually marked relevant video shots, we can compute the retrieval accuracy of our video retrieval system.

Evaluation

Because our collection contains only small numbers of relevant items, we adopted the average reciprocal rank (ARR) as our evaluation metric, as in [4]:

$$ARR = \left\{ \sum_{i=1}^k i / r_i \right\} / N_r \quad (2)$$

For a given query, there are a total of N_r relevant items in the collection. If the system retrieves k relevant items, they are ranked as r_1, r_2, \dots, r_k .

ARR rewards relevant items near the top of the retrieval list and de-emphasizes relevant items near the bottom of the list. Since the formula divides by the total number of relevant items for a given query, ‘easier’ queries with more answer items are not favored over ‘difficult’ queries where only a few answer items are relevant.

Recall is measured as the number of relevant items found out of 100 retrieved items over the total number of relevant items. The number of queries answered is the number of queries out of the total of 32, where at least one result was correct in the top 100 results returned.

Results

The results in Table 1 show that compared to the original still-image version of Cuebik, Video-QBIC improves the average reciprocal rank slightly (from .113 to .119), and dramatically improves recall (.222 to .338) as well as finding answers to more different queries (18 instead of just 12). The two different Cuebik versions seemed to find different results, yet the original Cuebik version clearly found some images correctly at a very high rank as indicated by the ARR score relative to recall and the number of queries for which an answer was found. Thus we decided to combine the Cuebik results with the Video-Cuebik results. Since we had no basis for an intelligent combination algorithm, both systems were given equal weight. This combined version showed an improvement in ARR and a slight improvement in recall, but left one additional query unanswered.

For comparison purposes, we also computed an artificial optimal combination score, which gives an upper bound on how well the combination of systems can do, given an oracle, which decides what the best system combination should be on a query-by-query basis.

Discussion

The overall low numbers show that this was a difficult test, about 1/3 of the queries could not be answered by any of the systems participating in the video TREC evaluation. These queries usually require world knowledge outside the

image information or an understanding of image details that far exceeded any current system’s abilities.

Table 1. Contrastive results for 32 queries over an 11-hour video collection consisting of 80,000 i-frame images.

Retrieval Method:	Average Reciprocal Rank (ARR)	Recall	No. Queries Answered
Cuebik	.113	.222%	12
Video-Cuebik	.119	.338%	18
Combined Video-Cuebik + Cuebik	.135	.339%	17
Optimal Combination	.149	.351%	18

Clearly there is benefit to matching video sequences beyond the comparison of individual still images. In this paper we demonstrated an extension to image search for video sequences that tries to capture the color variability in a video shot to allow better matching. When applied to a simplified version of the Cuebik image finding system using an 11 hour video collection and an independent test set of 34 image queries from the TREC video retrieval evaluation, the method clearly improved retrieval performance. Since some query images matched better when a more focused image search was applied, we combined the two versions to retain most of the advantages of each approach. Further work is in progress to apply this principle to more complex image search systems using color histograms and textures.

REFERENCES

1. A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain, “Content-Based Image Retrieval at the End of the Early Years,” IEEE Trans. PAMI, 22(12), pp. 1349-1380, December, 2000.
2. J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, “Efficient Color Histogram Indexing for Quadratic Form Distance,” IEEE Trans. PAMI, 17(7), pp. 729-736, July, 1995.
3. The TREC Video Retrieval Track, <http://www-nlpir.nist.gov/projects/t01v>, 2001.
4. E.M. Voorhees, and D.M. Tice, “The TREC-8 Question Answering Track Report,” The Eighth Text Retrieval Conference (TREC-8), 2000
5. S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong, "A Fully Automatic Content-Based Video Search Engine Supporting Multi-Object Spatio-Temporal Queries," IEEE Trans. on Circuits and Systems for Video Techn., 8(5), pp.602-615, Sep. 1998.
6. A. Mojsilovic, J. Kovacevic, J. Hu, R.J. Safranek, and S.K. Ganapathy, “Matching and Retrieval Based on the Vocabulary and Grammar of Color Patterns,” IEEE Trans. Image Processing, 9(1), pp. 38-54, 2000