Exploiting Redundancy in Binary Features for Image Retrieval in Large-Scale Video Collections

Noa Garcia
garciadn@aston.ac.uk
George Vogiatzis
g.vogiatzis@aston.ac.uk

Computer Science Group
Engineering and Applied Science
Aston University, Birmingham

Inspired by multimedia searching applications in smartphones, this work proposes a system for large-scale scene retrieval in movies, i.e. searching frames within video collections given a sample image. The system asks the user to take a picture of the film during video playback and it returns the corresponding frame of the video sequence.

Despite other methods have been previously proposed in the field of scene retrieval [2, 4, 6, 7], the novelty of this work relies on its scalability. Whereas previous works were able to process one [7], two [2, 4] or up to seven [6] movies, experiments conducted here show that our method can process over 40 movies and 7 million frames. This is achieved by combining two techniques. Firstly, frame redundancy is exploited by tracking and summarizing local binary features [3] into *key features*. Secondly, key features are indexed in a kd-tree structure for fast search of frames.

The processing is performed in two phases: the training and the query phase. During the training phase, frames from a video collection are processed and indexed, whereas in the query phase, a sample image is used to find a similar frame in the indexed collection.

In the training phase, BRIEF local binary features [3] are extracted from every frame and tracked along time by matching descriptors in consecutive frames. Suppose two consecutive frames f_n and f_m have a set of features $\{x_N\}$ and $\{y_M\}$, respectively. Two features x_i and y_j located in positions p_{x_i} and p_{y_j} in the pixel space, respectively, are allocated in the same track if they are unique nearest neighbours:

$$\mathrm{d}(x_i, y_j) = \min\big(\mathrm{d}(x_k, y_j) | k \in N\big) < th_1 \tag{1}$$

$$\mathrm{d}(x_i, y_j) = \min(\mathrm{d}(x_i, y_k) | k \in M) < th_1 \tag{2}$$

where $\mathrm{d}(x_i, y_j)$ is the Hamming distance between x_i and y_j, as well as if their spatial distance is less than a threshold:

$$\Big(\sum |p_{x_i} - p_{x_j}|\Big)^{1/2} < th_0 \tag{3}$$

The set of binary features allocated in a same track, $\{z_K\}$, is aggregated into a single binary vector known as key feature by computing majorities [5]. That is the b-th bit of the key feature k is computed as:

$$k[b] = \begin{cases} 1 & \text{if } \frac{1}{K}\sum_{k=1}^{K} z_k[b] \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

To avoid adding noisy features to the system, only stable tracks longer than a fixed number of frames, L_T, are considered. Consecutive frames that share visual similarities are grouped into shots. The boundaries of different shots are detected when f_n and f_m have no common tracks.

Binary key features are indexed in a kd-tree [1] for fast search. We modify the classical kd-tree approach so that it can index and search binary features. In this case, each decision node of the kd-tree has an associated dimension, dim, such that all query vectors, v, with $v[dim] = 1$ belong to the left child. Otherwise, vectors belong to the right child. The value dim is chosen such that entropy is maximum. Leaf nodes have as many as S_L indices pointing to the features that ended up in that node. A first-in first-out (FIFO) queue, Q, keeps record of the already visited nodes to backtrack and explore them later.

In the query phase, binary features are extracted from the input image and assigned to its nearest set of key features by searching down the kd-tree. Each key feature votes for the shot it belongs to. Finally, the set of frames contained in the most voted shot are rapidly compared against the input image by applying Hamming distance between their vectors.

For evaluation, a collection of 40 movies and a query set of 25142 images captured by a webcam, along with their corresponding ground truth frames, are used. The values used are $th_1 = 20$, $th_0 = 100$, $L_T = 7$, $S_L = 100$ and $B = 50$. Accuracy is computed as $\mathrm{Acc} = \frac{\text{No. Visual Matches}}{\text{Total No. Queries}}$, where the number of visual matches is measured by computing the similarity between the ground truth frame and the retrieved frame.

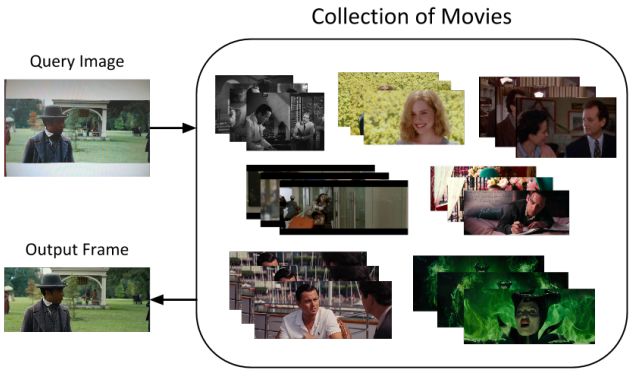

Figure 1: Scene retrieval diagram. Top-left: query image. Right: video collection. Bottom-left: output frame.

	BF	KT	KF	Ours
Memory	2.53GB	2.53GB	762MB	**61MB**
Accuracy	**0.98**	0.96	0.93	0.94

Table 1: Comparison between different systems. BF: Brute Force. KT: Kd-Tree [1]. KF: Key Frame [8].

Two experiments are conducted. In the first one, a movie consisting of 196572 frames is used to compare our proposal against three other systems: a brute force (BF), the system described in [1] and the key frame extractor from [8]. Table 1 shows that although the performance is similar in all the four systems, the amount of processed data is drastically reduced when applying our method, cutting down the memory requirements by 42.5 times with respect to BF and [1] and 12.5 times with respect to [8].

In the second experiment, the scalability of our approach is explored by increasing the database up to 40 movies. The amount of data is reduced from 3040 million features to only 58 million key features. The total accuracy over the 40 videos is 0.87, reaching values of 0.98 and 0.97 in some films.

In conclusion, a system for searching frames within large-scale video collections given a static image is presented. As a result of this work, videos would be more interactive and possibly augmented with extra content that can enhance viewer experience, such as director's commentary, educational information or purchasable items present in the scene.

[1] M. Aly, M. Munich, and P. Perona. Distributed kd-trees for retrieval from very large image collections. *Proc BMVC*, pages 1–11, 2011.

[2] A. Anjulan and N. Canagarajah. Object based video retrieval with local region tracking. *Signal Processing: Image Communication*, 22 (7):607–621, 2007.

[3] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF : Binary Robust Independent Elementary Features. *Proc ECCV*, pages 778–792, 2010.

[4] O. Chum, J. Philbin, M. Isard, and A. Zisserman. Scalable near identical image and shot detection. In *Proc CIVR*, pages 549–556, 2007.

[5] C. Grana, D. Borghesani, M. Manfredi, and R. Cucchiara. A fast approach for integrating orb descriptors in the bag of words model. In *IS&T/SPIE Electronic Imaging*, pages 866709–866709, 2013.

[6] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. *IEEE CVPR*, 2:2161–2168, 2006.

[7] J Sivic and A Zisserman. Video Google: a text retrieval approach to object matching in videos. *Proc IEEE ICCV*, pages 2–9, 2003.

[8] Z. Sun, K. Jia, and H. Chen. Video key frame extraction based on spatial-temporal color distribution. In *IIHMSP*, pages 196–199, 2008.