VIDEO SEARCH USING A VISUAL DICTIONARY

Emilie Dumont and Bernard Mérialdo

Institut Eurécom
Département Communications Multimédia
2229, route des Crêtes -B.P. 193
06904 Sophia-Antipolis cedex - France
{dumont, merialdo}@eurecom.fr

ABSTRACT

This paper presents an original approach to video shot retrieval, which is an adaptation of the common text-based paradigm. The idea of our approach is to describe images using a small number of visual elements chosen in a Visual Dictionary. The user may select some elements from the Visual Dictionary to compose a query and search for specific video shots. In our approach, we automatically compute those visual elements to compose a Global Visual Dictionary, and we select the most representative to build a Query Visual Dictionary. We propose a method to evaluate automatically the efficiency of this Query Visual Dictionary for retrieving shots, and we present experimental results on the TrecVideo BBC Rushes task.

1. INTRODUCTION

With the advance of the multimedia technology, large collection of videos in various formats are becoming available to the public: collections to web pages, or even video databases. The search, retrieval, or indexing of this data based on content is a challenge which is the focus of many research projects. The success of text-based retrieval motivates us to investigate analogous techniques which can support the querying and browsing of video data. However, images differ significantly from text both syntactically and semantically in their mode of representing and expressing information. Thus, the generalization of information retrieval in the image and video domain is non-trivial.

In text document retrieval, the most commonly used document representation is a vector where each entry represents the importance of a particular word in a document. The work introduced in this article uses a similar approach for searching video shots. So, in the video domain, a shot is represented by one or more keyframes, and a keyframe is represented by a vector of visual elements where each entry represents the importance of a particular visual element in the shot.

For text document retrieval, a user types words to compose a query and the system returns a ranked list of documents. The higher the document is in the list, the more it is relevant to the query. For querying video shots, we propose that the user selects visual elements, from which the retrieval system returns a ranked list of video shots. Again, the higher the shot is in the list, the more it will be considered relevant to the query.

In our approach, we use a two step process to build our visual dictionary. First, we create a large Global Visual Dictionary (GVD), by clustering blocks from all training keyframes, then we construct a Query Visual Dictionary (QVD) composed of the most discriminant classes. This is to insure that the size of the Query Visual Dictionary remains limited, so that the user may efficiently select the visual elements to build the visual query. Using the QVD, an image can be encoded as a vector of number of occurrences of the visual elements. Based on this image representation, information retrieval techniques developed in the text domain can thus be generalized for video shot retrieval.

The rest of the paper is organized as follows : first, we recall some of the most relevant works on visual dictionaries. In the next section, we explain our method to construct and optimize the visual dictionary. Then, we propose a method for visual dictionary evaluation. And finally, we show the power of the visual dictionary through experiments on the TrecVid BBC Rushes data.

2. RELATED WORKS

The first content-based image retrieval (CBIR) systems [1] [3] [8] proposed a set of image indexing methods based on low-level features (colour, texture, shape) fully automatic, but these methods could not capture the semantic information in images.

Picard was the first to develop the general concept of a visual thesaurus by transforming the main idea of text dictionary to a visual dictionary [5]. One year later, she proposed examples of a visual dictionary based on texture, in particular the FourEyes system [6]. But no experiment was carried out in order to show the quality of these systems.

A first method consists in building a visual dictionary from the feature vectors of segmented image regions. In [11], the authors use a self-organizing map to select visual elements, in [4] SVMs are trained on image regions of a small number of images belonging to seven semantic categories and in [2], regions are clustered by similar visual features with a competitive agglomeration clustering. And then, images are represented as vectors based on this dictionary. The semantic content of those visual elements depends primarily on the quality of the segmentation.

Elliptical affine regions are represented by scale invariant feature transform. The regions detected in each frame of the video are tracked, and the estimation of the descriptor for a scene region is computed by averaging the descriptors throughout the track. And finally, to create the visual vocabulary, they use K-means clustering in [7]. This method cannot be used on a set of images and requires a tracking method.

To create a visual dictionary, authors of [12] segment images in blocks and use a combination between a Generalized Lloyd Algorithm and Pairwise Neighbor Algorithm on a training set. Results presented use blocks of small sizes (lower than 4x4 pixels), that does not make it possible to make visual queries.

3. VISUAL DICTIONARY

To apply the text document retrieval paradigm to video retrieval, the first step is to create a Query Visual Dictionary QVD. Using the QVD, an image can be encoded as a vector of visual element components and a user can compose queries. Our approach for the construction of the QVD is a two step process:

- **Creation of a Global Visual Dictionary (GVD):** In textual document retrieval, documents are based on dictionaries of several hundred thousand words. Users know most of these words, and are able to type them to compose a query without having to access the complete list. But in visual document retrieval, there is not universal visual dictionary. Thus, our first step is to analyze the keyframes from the training videos and to automatically build a large set of visual elements to compose the GVD.
- **Creation of a Query Visual Dictionary (QVD) :** Users cannot ”type” those visual elements, so they have to select them in a list. In order to keep the selection process reasonable, the list should have a relatively small size. This is why we select the most discriminant visual elements to create the QVD.

3.1. Visual Elements

Our approach is based on the idea of using a small and fixed number of visual elements. Those visual elements should be automatically computable, so that an image can be automatically described in terms of those visual elements. They should also have some interpretable representation for the user, so that the user can understand the relationship between the representation in visual elements and the content of the image, and also that he can select some of those elements to compose a query during a search activity.

A visual element is an image area. While a large number of visual elements may be considered, for example the ”indoor/outdoor” attribute could be such a visual element, we focus in the present work on the construction of a visual dictionary of image blocks, either through a color representation or a texture representation, see figure 1.

Fig. 1. Examples of visual elements : left, based on colour and right, based on texture

For example, if we work with a regular grid of $4 * 4$ blocks : an image generates 16 blocks and the block size is $W/4 * H/4$ where H is the height of an image and W is the width.

3.2. Global Visual Dictionary

Each image is divided into blocks, and for each block we construct two vectors : a colour feature vector (HSV histogram) and a texture feature vector (Gabor filters). We cluster independently the colour and the texture vectors

using the K-Means algorithm, with a predefined number of clusters (Nc and Nt) and using the Euclidean distance.

We build the Global Visual Dictionary by selecting the feature vector which is closest to the centroid for each cluster, so this GVD is composed by two dictionaries : Dc containing only colour elements and Dt containing only texture elements. Each image block is then associated to one color and one texture visual element.

3.3. Query Visual Dictionary

Finally, we construct our QVD by selecting the most discriminative vectors. We define the discriminative power of a vector v as :

$$dis(v) = \log(\frac{1}{1 + tf(v)})$$

where $tf(v)$ is the total number of occurences of the visual element v in all images.

The QVD is composed of the N feature vectors with the highest discriminative power (note that we can either process color and texture independently, or mix them during the selection process). Figure 2 shows an illustration of the process of the construction of the QVD.

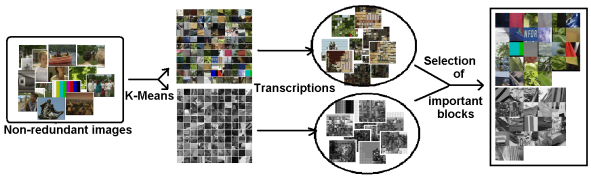

Fig. 2. Creation of the visual dictionary

4. SHOT REPRESENTATION

For each video in the video database, a shot boundary detection is done to obtain a set of shots. And for each shot, a keyframe is extracted, then each keyframe is encoded by the QVD.

4.1. Shot boundary detection

To detect transitions, we use the window approach presented by [10]. A moving window consists of two equal-sized half windows, surrounding a current frame. It is shifted through the video frame-by-frame. Each frame is represented by a HSV histogram, computed on a grid of 16 equal-sized regions. The four central regions are often affected by rapid object movement, so they are not used in the histogram construction.

For cut detection, we use a ranking-based method. Frame similarity is the sum of the inter-region similarities. Each frame, in the moving window, is ranked by decreasing similarity to the current frame. The number of pre-frames that are ranked in the top half of the rankings is monitored. When a cut is passed, the number of top ranked pre-frames rises to a maximum and falls to a minimum within a few frames.

For gradual detection, pre-frames and post-frames are combined into two distinct sets of frames. The average distance of each set to the current frame is computed. The ratio between the pre-frame set distance and the post-frame set distance is monitored. The end of most gradual transitions is indicated by a peak in the PrePostRatio curve.

For each shot, the central image is extracted as the keyframe of the shot.

4.2. Image encoding

To encode an image using a QVD, first the image is decomposed into blocks, for each block, a feature vector is extracted and then, each block is replaced by the nearest visual element in a QVD. Figure 3 illustrates an image encoding. It can also be considered as a list of visual elements, in format to a text document, defined by a list of keywords.

Fig. 3. An image is described in terms of visual elements: on the left the original image and on the right, the image encoding

Each visual element is defined by $\mathbf{v_f} \in \mathcal{R}^{d_f}$, $f \in F$ where F is the set of the selected features, d_f is the dimension of the feature vector. For each image block, and each feature, we find the closest visual element in QVD by the Euclidian distance. The image feature representation $\mathbf{I}$ is based on the frequency of the visual elements within the image, so $\mathbf{I} \in \mathcal{R}^N$, $\mathbf{I} = \{w_1, \ldots, w_N\}$, where N is the

QVD size, and w_v is the number of occurences of the visual element v in the image.

5. ARTIFICIAL SEARCH FOR VIDEO SHOTS

The search for video shots is performed as follows : first a user composes a query by selecting visual elements in the QVD, then the system will compare the query vector to the keyframe descriptions using a similarity measure, and will return a ranked list of shots. As human experimentation is always difficult and expensive, we propose a new approach: the Artificial Search (AS), described in the next section, that allows an entirely automatic evaluation.

5.1. Interactive Search

There are several ways in which the QVD may be used to search for information inside the content of video files. For example, it is possible to conduct an interactive search in the following manner: initially, all video files are available, and represented as a line of micro-icons. Then, the user may select one of the visual elements in the QVD. This identifies a set of video files which contain this visual element, so that the visual result is displayed as a line of bigger micro-icons. In the QVD, the visual elements which do not appear anymore in the list of selected video files are grayed, and the user may select another relevant visual element, to filter the selected list further. We have not built such a system yet, but we can show simulations of this interface. Figure 4 shows a simulation of this progressive refinement process.

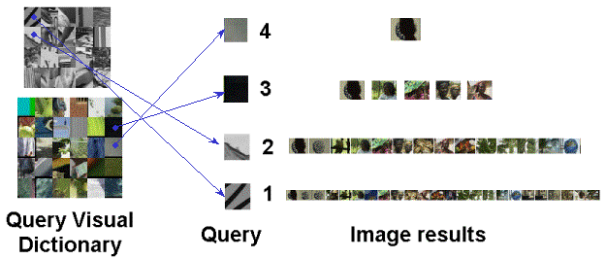

Fig. 4. Simulated interface to search among BBC Rushes

5.2. Selection of a keyframe set

In order to build the set of all training keyframes, we process all video files in the database, we perform shot detection and for each shot, we select a keyframe. We remove redundancy by a hierarchical classification. To create the training set, keyframes are classified by a hierarchical agglomerative clustering algorithm. Each keyframe is represented by a HSV histogram and 12 Gabor filters. The distance between two images is computed as the Euclidean distance, and the distance between two clusters is the average distance across all possible pairs of images of each cluster. When the clustering is finished, we select for each cluster the image which is closest to the centroid of the cluster. Those selected images will compose the set of non-redundant images. Figure 5 shows an illustration of this process.

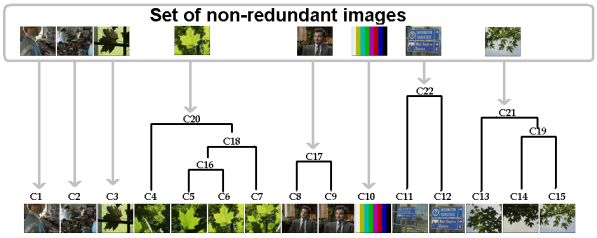

Fig. 5. Illustration of the hierarchical agglomerative clustering algorithm

5.3. Automatic Evaluation

We propose to evaluate the quality of the QVD through an Artificial Search (AS) procedure. The idea of the experiment works as follows: assume that we want to identify a keyframe from the training set, that is, we can look at the image, and we want to recover the video file that it was extracted from, and its frame number within this file. From the image, we can identify the most adequate visual elements from the QVD and get a ranked list of relevant images. The rank of the original image in this list is a measure of the efficiency of the visual dictionary in describing the image content. We can average this rank through a large number of images to provide a global measure for the performance of the visual dictionary. Figure 6 shows an illustration of this process.

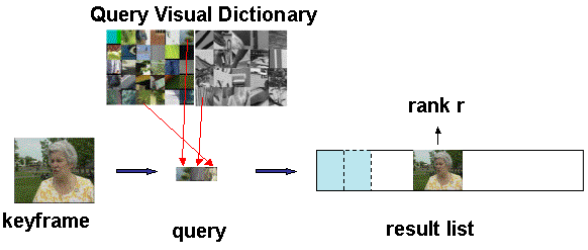

Fig. 6. Artificial Search Experiment

This process can be easily simulated if we can provide a reasonable mechanism to automatically compose the visual query by selecting visual elements based on the original keyframe. Indeed, we automate this selection with the

following algorithm: for each visual element of QVD, we calculate the quality $q_i(v)$ of this element to belong to a potential query for the considered keyframe by :

$$q_i(v) = \begin{cases} \log(\frac{1+N}{1+tf(v)}) & \text{if } v \in i \\ 0 & \text{otherwise} \end{cases}$$

where i is the keyframe, v the visual element, N the total number of blocks in all images and $tf(v)$ the number of occurences of the visual element v in all images.

For each image i, we can define $rank(i)$ as the rank of the original image i in the result list. Then, the performance of the QVD is the average rank over all keyframes :

$$AverageRank = \frac{1}{N}\sum_{i=1}^{N} rank(i)$$

where N is the number of keyframes. This evaluation can be conducted completely automatically.

6. RESULTS AND DISCUSSION

6.1. Dataset

Experiments are conducted on the video data which is used in the "BBC Rushes" task of TrecVid'06 [9]. It represents a total of over 40 hours of video. The video files contain unedited footage recorded for the preparation of video programs. There is no edition, and the data is highly redundant, as typically only 5% of it would be kept in the final program. As explained previously, we process all those video files, we perform shot boundary detection and we extract a set of non-redundant keyframes. For those video files, we found a set of 1759 non-redundant images.

6.2. Experimental protocol

We have considered several parameters in our experimentations:

- different block sizes: images are split into a regular grid with either $12*10$, $10*8$, $8*5$, $5*4$ or $4*2$ blocks;
- different sizes for the Global Visual Dictionary GVD (this is the number of clusters in the K-Means, ranging from 25 to 1500);
- different sizes for the Query Visual Dictionary QVD: 25, 50, 75, 100 and 200.

From the original image, we construct the query by selecting the $Nclics$ most important visual elements which appear in the image.

6.3. Experimental results

6.3.1. Global Visual Dictionary size

The Global Visual Dictionary is obtained by clustering the block vectors with the K-means algorithm. There is one clustering for the color vectors and another for the texture vectors, and we consider an equal number of clusters in each case: N_c = N_t. The size of the GVD is the sum N_c + N_t. The cluster representatives are the visual elements. Then we select the most discriminant visual elements to construct the Query Visual Dictionary QVD, and we perform an artificial query with $Nclics = 2$ (i.e a query is composed of the best two visual elements). In this experiment, images are split according to a $8*5$ regular grid. The figure below shows the average rank of the original image with respect to the size of the GVD, while each curve refers to a given size for the QVD.

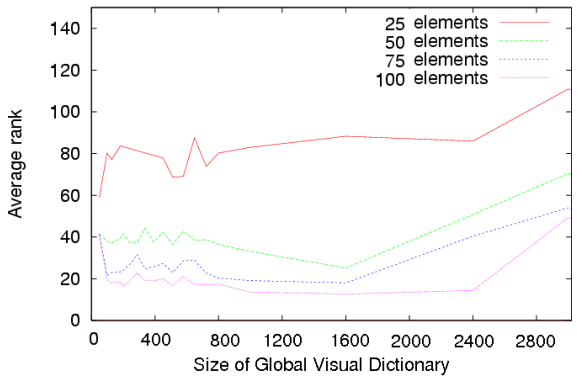

Fig. 7. Impact of the size of the GVD

This figure shows that the size of the Global Visual Dictionary is not such a critical factor for the quality of the Query Visual Dictionary. Indeed, the performance curves show a very stable behaviour, except when the size of the GVD becomes very high, because then the visual elements become very specific.

6.3.2. Query Visual Dictionary size

The QVD size is the number of elements that an image can be encoded and it is too, the number of visual elements that is proposed, to a user, to make queries. Figure 8 shows the average rank of the original image for various sizes of QVD in the case where $Nclics = 2$, images are split into $8*5$ blocks. The X axis indicates the QVD size, while the different colour curve indicate the number of clusters using in K-Means.

The evaluation curve for the QVD size shows that

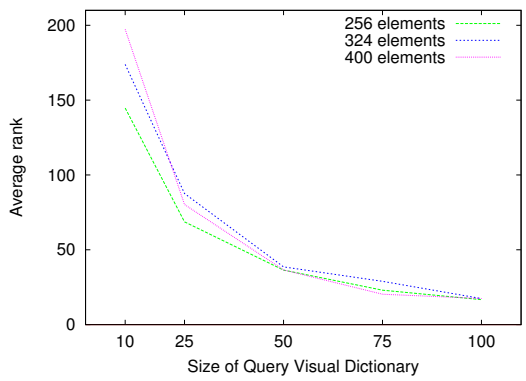

Fig. 8. Impact of the size of the QVD

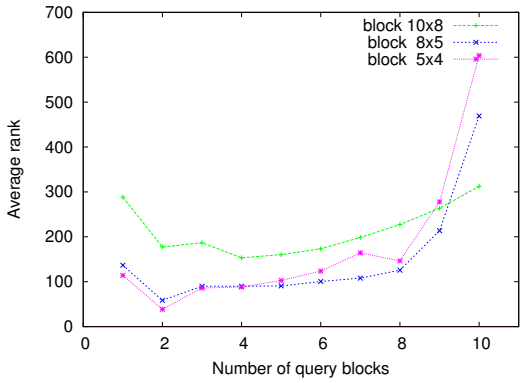

Fig. 9. Evaluation of the QVD for different block sizes

the average rank decreases rapidly with the number of visual elements in QVD. This is expected, as those visual elements are used both to encode images and to build queries. The higher the number of visual elements, the lower the number of images associated to each element, and the lower the average rank of the original images. But visual elements are displayed to the user so that (s)he can select the ones that make the query, so the effective number of visual elements should be kept to a reasonable size, because of display size and selection time constraints. Also, if too many visual elements are kept, some elements may be visually very similar, which will make them difficult for the user to distinguish and use properly.

6.3.3. Block size

The block size is obviously a very important factor in the whole process. Smaller blocks allow a more precise description with fewer elements, while bigger blocks may contain more information, but require a larger number. Figure 9 shows the average rank of the original image for various block sizes and various values of $Nclics$. The size of the GVD is 648 and the size of the QVD is 50.

Those curves that for each block size, there is an optimal value for $Nclics$. When blocks are large, this optimal value is small, because those blocks are very discriminant. On the contrary, when blocks are small, a larger number is required to best identify the desired image. Note that the query is considered as a strict logical AND, so that when $Nclics$ exceeds the number of valid blocks in the original image, the query returns all images with equal rank, which causes the average rank to increase. Figure 10 shows examples of blocks with different sizes.

Fig. 10. Examples of block sizes : the first block on the left is an example of a block obtained by an image divided in 10 by 8 blocks, so that the size of this block is 35 pixels wide and 36 pixels high.

Visual elements are intended to represent semantic concepts when building queries. For example, if a user searches a video shot containing water, woods and sky, (s)he could select three elements, one per concept, each identified by a given visual element. See an illustration of this process in figure 11. With the current image size available in the BBC Rushes, blocks arranged according to $8 * 5$ and $4 * 2$ grids seem to be quite adequate for this task.

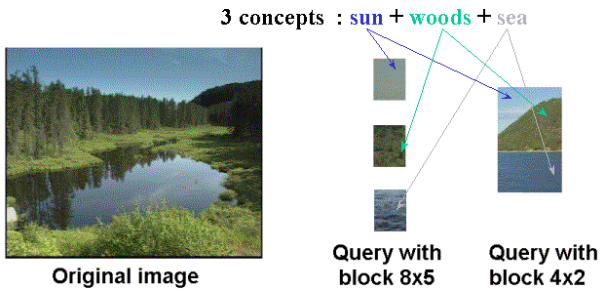

Fig. 11. Sample of a query

We have not considered the case of a dictionary composed of blocks of varying sizes. This remains an open re-

search issue.

6.3.4. Visual Features

Up to now, we have considered a Query Visual Dictionary composed of the most discriminant visual elements chosen among color and texture feature vectors. We can compare this combination with what would happen if we had considered a single feature, either color or texture. Based on the best-case block size $8 * 5$, a GVD size of 648 and a QVD size of 50, figure 12 shows the comparison between the combined QVD and the QVDs respectively based on color or texture only. This shows a substantial improvement for the combination. Note that the combined dictionary contains 35% of texture vectors and 65% of color vectors.

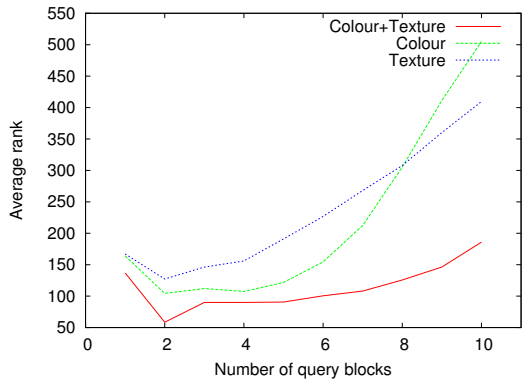

Fig. 12. Evaluation of the QVD for different features

7. CONCLUSION

This paper has demonstrated an original approach to video retrieval by using a visual dictionary. We have detailed the various steps involved in the construction of the dictionary, and proposed a methodology to automatically evaluate its performance. Finally, we have presented a set of experiments to compare various block sizes and dictionary sizes. Although many issues remain to be explored, such as the relevance of visual elements, we expect that this type of approach will provide a useful component in future frameworks for video navigation and search.

Acknowledgement

The research leading to this paper was supported by the Institut Eurecom and by the European Commission under contract FP6-027026, Knowledge Space of semantic in-ference for automatic annotation and retrieval of multimedia content - K-Space.
BBC 2006 Rushes video is copyrighted. The BBC 2006 Rushes video used in this work is provided for research purposes by the BBC through the TREC Information Retrieval Research Collection.

8. REFERENCES

[1] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*. Springer, 1999.

[2] J. Fauqueur and N. Boujemaa. New image retrieval paradigm: logical composition of region categories, 2003.

[3] Yihong Gong, H.C. Chua, and X.Y. Guo. Image indexing and retrieval based on color histograms. *International Journal of Multimedia Tools and Applications*, 2, 1996.

[4] Joo-Hwee Lim. Categorizing visual contents by matching visual "keywords". In *VISUAL '99: Proceedings of the Third International Conference on Visual Information and Information Systems*, pages 367–374, London, UK, 1999. Springer-Verlag.

[5] R. Picard. Toward a visual thesaurus, 1995.

[6] Rosalind W. Picard. A society of models for video and image libraries. *IBM Systems Journal*, 35(3/4):292–312, 1996.

[7] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. *iccv*, 02:1470, 2003.

[8] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000.

[9] TRECVID. Digital video retrieval at NIST. http://www-nlpir.nist.gov/projects/trecvid/.

[10] Timo Volkmer, S.M.M. Tahaghoghi, and Hugh E. Williams. RMIT University at TREC 2004. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of the TRECVID 2004 Workshop*, Gaithersburg, Maryland, USA, 2004.

[11] Ruofei Zhang and Zhongfei (Mark) Zhang. Hidden semantic concept discovery in region based image retrieval. *cvpr*, 02:996–1001, 2004.

[12] Lei Zhu, Aidong Zhang, Aibing Rao, and Rohini K. Srihari. Keyblock: an approach for content-based image retrieval. In *ACM Multimedia*, pages 157–166, 2000.