Analysis of Using Metric Access Methods for Visual Search of Objects in Video Databases

Henrique Batista da Silva [1]
Zenilton Kleber Gonçalves do Patrocínio Júnior [2]
Silvio Jamil Ferzoli Guimarães [2]

Abstract:

This article presents an approach to object retrieval that searches for and localizes all the occurrences of an object in a video database, given a query image of the object. Our proposal is based on text-retrieval methods in which video key frames are represented by a dense set of viewpoint invariant region descriptors that enable recognition to proceed successfully despite changes in camera viewpoint, lighting, and partial occlusions. Vector quantizing these region descriptors provides a visual analogy of a word – a visual word. Those words are grouped into a visual vocabulary which is used to index all key frames from the video database. Efficient retrieval is then achieved by employing methods from statistical text retrieval, including inverted file systems, and text-document frequency weightings. Though works in the literature have only adopted a simple sequential scan during search, we investigate the use of different metric access methods (MAM): M-tree, Slim-tree, and D-index, in order to accelerate the processing of similarity queries. In addition, a ranking strategy based on the spatial layout of the regions (spatial consistency) is fully described and evaluated. Experimental results have shown that the adoption of MAMs not only has improved the search performance but also has reduced the influence of the vocabulary size over test results, which may improve the scalability of our proposal. Finally, the application of spatial consistency has produced a very significant improvement of the results.

Keywords: Video retrieval, visual vocabulary, metric structures.

1 Introduction

Research involving video retrieval in large databases has had an increasing interested and relevance, mainly, due the increasing popularity of platforms for video sharing and viewing, along with the improvements in the producing technologies (i.e., digital cams, cell phones, and others). According to [24], with the progress in hardware, especially in relation to storage cost, and also with the advances in software, such as video editing tools, nowadays

[1] Departamento de Ciência da Computação, UFMG
{henrique.silva@dcc.ufmg.br}
[2] Instituto de Ciências Exatas e Informática, PUC Minas
{{zenilton, sjamil}@pucminas.br}

there is a great use of this type of media. [24] have also emphasize that the spread of broadband access has greatly contributed to the popularity of videos on the Web, which can be seen by the large amount of videos posted in open tools for multimedia content viewing and sharing available on the Internet.

There is a necessity to provide an efficient solution for video storage. The most common way to search videos is through keywords, since metadata is usually stored for each video. As this approach involves human effort, erroneous and incomplete descriptions can negatively influence the classification/indexing process as well as the retrieval of relevant videos. On the other hands, video retrieval based on visual content searches and localizes all occurrences of a given query image (or short video clip) in a video database. Thus, it is possible to avoid problems with wrong and incomplete descriptions. However, content-based video search is not a simple task, because video data are complex and hard to manipulate, due the large amount of frames and the great number of high dimensional descriptors extracted.

Many research has been made about feature extraction (i.e., image properties, such as color intensity, texture, among others) of video frames, to describe their content and to facilitate video indexing and retrieval [22]. Some approaches in literature, as in [20], consider the search of video clips in a database using a dissimilarity measure between frames, without consider any global information. However, many approaches for content-based retrieval found in the literature have adopted a simple sequential scan during the search over the video database, which makes them unsuitable for large collections. Thus, the adoption of more efficient index structure could accelerate query processing in the video database.

In this work we address the object localization in a video database, in which, given a query image of the object, the aim is to localize all occurrences of the object in the video collection. We also analyze the performance of different metric access methods when applied to the problem of object retrieval in video using visual vocabulary (as artifice to reduce the size of the index). To describe video content, we have used descriptors invariant to changes in camera viewpoint, lighting, and partial occlusions. This process was applied to each key frame of the video to create a visual vocabulary whose words (visual words) are obtained through a clustering process and they are indexed by an inverted file system. The visual vocabulary is important to decrease amount of the visual words in the indexing structure.

We evaluate the search performance by measuring the recall and precision rates of several assessments of objects localization. Were used different objects to evaluate video search. In addition, we have also evaluated the use of different metric access methods (MAM) so as to accelerate query processing. The metrics access methods used were compared with sequential access method. Although those structures have been evaluated before, most of the test results presented in the literature only involve the performance of the search for isolated visual descriptors without any application to a real problem. Results have shown the adoption of MAMs not only has improved the search performance but also has reduced the influence

of the vocabulary size over results, which may improve the scalability of our proposal. We observed the use of visual vocabulary is important for reducing the size index, accelerating the searching for objects in the video, even if it increases the number of false positives in the search results. Finally, we have also described a ranking strategy based on visual words spatial layout and we have evaluated its impact over the results quality.

This article is organized as follows. Section 2 presents main concepts that are useful to this work and to some related works. Section 3 describes our approach for video indexing, including the construction of a visual vocabulary. The approach for the retrieval process is described in Section 4 along with a heuristic strategy for re-ranking baseline search results using information about spatial layout of the frame descriptors. Then, experimental results are presented in Section 5. And, Section 6 presents final remarks and future works.

2 Related Work

Typically, a video consists of a sequence of frames, which are continuously displayed during a period of time [8]. Usually, video segmentation is considered as the first step in video analysis, it can be done through shot boundary detection which aims to identify the boundary between two consecutive shots [8, 21]. So, we use a key frame identification process to reduce the amount of video data without losing too much of its visual content [16, 23].

A major challenge to deal with images and videos is represent their content in order to be efficiently stored and retrieved from databases. In this sense, the simplest way to represent video frames is through low-level image features, such as colors, textures, regions or edges [2]. Thus, the computation of similarity between two frames could be based on the similarity of these descriptors. In order to extract image descriptors, some authors have adopted the concept of local features [18, 28], which is used to detect regions/points of interest (i.e., affine covariant regions). In the literature, there are several methods for detecting and describing regions/points of interest (see [17, 18] for more details).

2.1 Detecting and Describing Regions/Points of Interest

According to [18], affine covariant regions should allow us to recognize objects in images, despite changes in camera viewpoint, illumination and partial occlusion. [18] present several methods to detect affine covariant regions, such as Harris-Affine detector and MSER detector. Harris-Affine recognizes edges or joints in images using information about their curvature, while MSER (*Maximally Stable Extremal Region*) detects blobs – regions in the image that differ in properties such as brightness or color when compared to the surroundings. Once regions/points of interest are detected, they must be described. Many works in the literature have adopted SIFT (*Scale Invariant Feature Transform*), proposed by Lowe [15], which

is invariant to translation, scaling and image rotation, and partially invariant to illumination changes and 3D projection. In [17], several descriptors for local features are compared.

The concept of a bag of words has been used to describe and to search in image database using these local descriptors, an example could be found in [14]. The hole process could be summarize as follows: (i) detection of regions/points of interest; (ii) description of the detected regions/points of interest; (iii) vector quantization of these local descriptors to created a visual vocabulary (similar to a set of terms/words in text retrieval methods); and (iv) computation of a histogram by counting the occurrences of the regions (visual words) in each image (this histogram is called "bag of words"). During the search, query image are described in the same manner, i.e., using a bag of words. Then, are used vector model and the assumptions of document similarity theory to measure the similarity between the query image and the images in the database. The use of sequential scans during the search procedure may not allow the application of this approach to large databases. Therefore, the adoption of a more efficient access method seems to be necessary.

2.2 Indexing and Searching

The databases have evolved to store unstructured data such as images and videos. This kind of content has the disadvantage of being difficult to handle, because the great volume of unstructured multidimensional data which do not present a total order, thus limiting the queries on them. Therefore, it is necessary to use more efficient search models than the ones used on simple structured data. In this context, similarity search arises as a technique to retrieve objects which are similar to the query object [3].

According to [3], similarity could be model using a metric space (i.e., a set of objects and a distance function among them that satisfies the triangle inequality). When images are described using bags of words, they could be seen as points in a multidimensional space and the similarity between two objects is calculated through a distance function between their bags of words. In [25], an approach for video retrieval based on text retrieval methods was presented. In that work, video frames were indexed by an inverted file system (that used a sequential access method in its implementation), while each frame was represented by a bag of words. In the retrieval step, video frames are retrieved using the similarity between the bags of words from the database frames and the bag of words of the query image.

Sequential access method can be used to implement the search in the inverted file (or one can use a more efficient access method such as a multidimensional structure). However, for high dimensional data, metric access methods in general present better performance [29]. Metric access methods store objects using a distance function associated with them. They can be divided into static metric structures, which do not allow insertions and removals of elements after its creation, and dynamic metric structures that are more suitable for scenarios

in which insertions and removals are needed after the index creation [29].

In [3, 29], several metric structures are described. In [4], the authors have proposed the first dynamic metric access method in the literature – M-tree. M-tree is a balanced tree, and it is able to handle dynamic data files and requires no periodic reorganization. In [26, 27], Slim-tree, an extension of the M-tree, was proposed aiming at a performance improvement through the overlap minimization of the space regions (subtrees). In addition to those tree-based metric structures, there are also hash-based metric structures in the literature such as D-Index [5]. The authors' objective in [5] was to minimize the number of disk accesses and the number of distance computations during similarity queries.

In this work, we analyze the performance of those structures (M-tree, Slim-tree, and D-index) when they are applied to the problem of object retrieval in video using visual vocabulary (as an artifice to reduce the size of the index). The selection of those structures are motivated by the following reasons: (i) M-tree was the first dynamic metric structure and it has been extensively used in the literature; (ii) while Slim-tree is an extension of the M-tree which is easily implemented and has shown a performance improvement; (iii) and D-index is a option based on a distinct approach (hash-based) which tries to reduce the number of disk accesses to the lowest possible value. Although those structures have been evaluated before, most of the results presented in the literature only involve the performance of the search for isolated visual descriptors without any application to a real problem [1, 4, 5, 19, 26, 27]. Recently, many authors have conducted assessment that are more related to real problems instead of evaluating the search for isolated descriptors [6, 9, 10, 11, 12]. In this work, we adopt an evaluation procedure related to a real problem – i.e. object localization problem.

3 Video Indexing

In order to successfully retrieve an object, we divide the storage process in the following steps: (i) video summarization; (ii) creation of the visual vocabulary; (iii) and video content description and indexing. Figure 1 presents the proposed framework for video indexing. First, the video is sampled using one key frame per second to reduce its complexity, (more details in section 5.2). The selected key frames are used for both creating the visual vocabulary and indexing the video content. To create the visual vocabulary, a sampling of key frames is made, followed by a process of feature identification and description. In [25], the authors present an approach for video retrieval which is similar to a text retrieval method. In their proposal, a vocabulary consisting of visual words should be created. These visual words are generated by clustering feature descriptors obtained from the video frames. However, they only uses a random sample of the frames generated during video segmentation. According to [25], the idea of using only a sample of video key frames is justified by the high computational cost of vector quantization process.

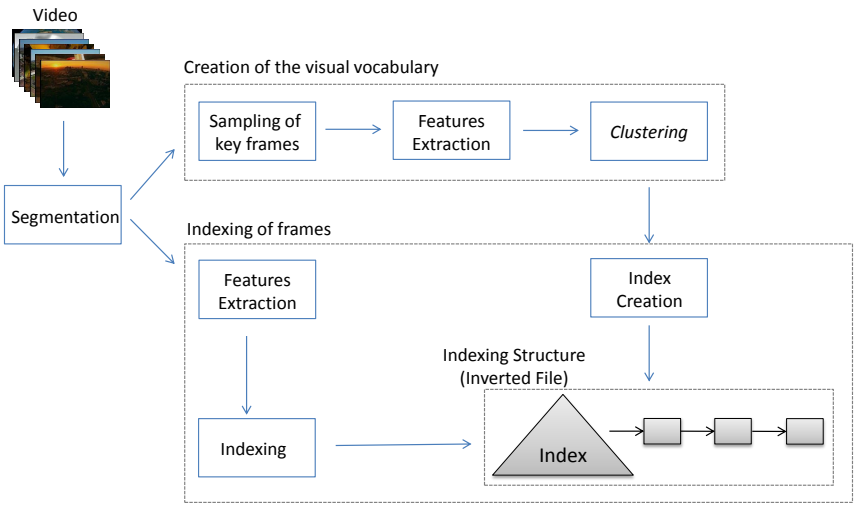


Figura 1. Video indexing framework.

For each selected frame, features extraction is carried out by the detection of affine covariant regions. We have adopted two types of affine covariant region detectors: the MSER detector and the Harris-Affine detector, which identify regions/points of interest in video frames. The choice of such detectors is motivated by the fact that they are complementary: MSER detector identifies stable connected components (blobs), while Harris-Affine detector recognizes image corners. Then, SIFT descriptor was used to describe the regions/points of interest identified by both detectors, producing thousands of 128-dimensional feature vectors for each video frame. Once all descriptors have been generated for each frame of the sample, we applied a clustering algorithm in these descriptors in order to identify visual words. We have used an efficient implementation of K-means algorithm presented in [13]. In which, for a set of data points and a integer k, it determines k centroids. In the iterative process, k-means minimizes the distance of each data point to its centroid. Finally, k-means gives a set of visual words. In [13] was proposed an efficient implementation of Lloyd's k-means clustering algorithm, decreasing the computational cost toward other approaches.

The clustering algorithm was performed separately for each detector, generating two sets of centroids. Each centroid represents a visual word of the vocabulary, and so, two sets of visual words are obtained: one from MSER detector and another from the Harris-Affine detector. Finally, those two sets of visual words are merged into one visual vocabulary. In the vector quantization process, we obtain the cluster along with the frequency of descriptors associated with each cluster. Therefore, the most frequently centroids are eliminated

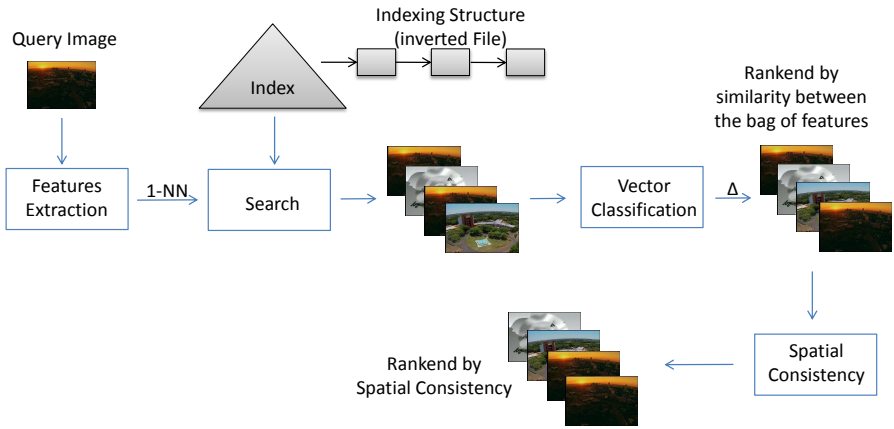


Figura 2. Video retrieval framework.

from the vocabulary, since they may not be discriminative enough. This words are called stopwords [7]. Finally, the visual words were inserted into the index structure used to accelerate the search in the inverted file. The visual vocabulary is important because it reduces the index size of the inverted file and contributes to accelerate the search process.

4 Video Retrieval

The retrieval process is shown in Figure 2. First, the query image of the object was described using the same process applied to the video key frames. Moreover, a bag of words for the query image is also generated. Subsequently, the search is performed and all key frames that have at least one visual word in common with the query image are retrieved along with their bags of words. After that, relevance ranking of the retrieved key frames are calculated using the vector model and the assumptions of document similarity theory, by comparing the deviation of angles between each key frame vector and the query image vector. This could be easily accomplished through the calculation of the cosine of the angle between these vectors (using a dot product), instead of calculating the angle itself.

Finally, a heuristic strategy – named spatial consistency – re-ranks baseline results using information about spatial layout of the regions/points of interest. This helps to reduce the number of false positives and improves retrieval effectiveness.

```
Let F = {F[j] | j ∈ [1, ..., Δ]} be an order list of Δ retrieved frames (after relevance ranking);
for (j ← 1 to Δ) do
    vote[F[j]] ← 0;
    while (There are visual words not marked in the query image) do
        p_c ← choose randomly a not marked visual word in the query image;
        I_marked[p_c] ← true;
        b_c ← search a visual word equals to p_c in the retrieved frame F[j];
        if (b_c is not null) then
            F_marked[j][b_c] ← true;        vote[F[j]] ← 1;
            Let V_{p_c} be the set of k nearest neighbors of p_c in the query;
            Let V_{b_c} be the set of k nearest neighbors of b_c in the frame;
            for (n_{p_c} ∈ V_{p_c}) do
                for (n_{b_c} ∈ V_{b_c}) do
                    if (n_{p_c} = n_{b_c} and not(I_marked[n_{p_c}] or F_marked[j][n_{b_c}])) then
                        vote[F[j]] ← vote[F[j]] + 1;
                        I_marked[n_{p_c}] ← true;
                        F_marked[j][n_{b_c}] ← true;
    Unmark all visual words in the query image;
```

Algorithm 1: Algorithm for spatial consistency.

4.1 Spatial Consistency

Text retrieval methods increase the ranking for documents, in which the searched words appear close together in the retrieved texts. This analogy is especially relevant for querying objects based on an image, in which matched visual words in the retrieved key frames should have a similar spatial arrangement to those of the query image. In [25], an approach was proposed in which key frames are first retrieved using their bags of words and then they are re-ranked based on a measure of their spatial consistency. Spatial consistency can be measured quite loosely by requiring that neighboring matches in the query image lie in a surrounding area in the retrieved key frame. But it can also be measured by requiring that corresponding matches have "almost" the same spatial layout.

Our approach (see Algorithm 1) begins with a random selection of a visual word from the query image, which is then searched in the retrieved key frame. After finding the corresponding visual word, a fixed number of words spatially closer to the selected word are obtained in both the query image and the key frame returned from the database. The number of distinct matches between these two sets of visual words are counted as votes for that key frame. In order to find a fixed number of visual words spatially closer to each other in a frame, we used a KD-tree to index the spatial coordinates of all visual words in the image. Thus, we conducted a search for the k nearest neighbors to obtain the k spatially closest words. The procedure of indexing these words using a KD-tree was done during video indexing phase, therefore the ranking by spatial consistency runs very fast.

Similar to [25], once a visual word is employed to cast a vote, it is marked to prevent further use of it. Then, another word (not yet used) from the query image is randomly selected and the voting process continues until every visual word is marked or there is no words left untested in the query image. The final score for each key frame is obtained by adding the spatial consistency votes to the score previously calculated by relevance ranking. Moreover, the object bounding box could be set to the rectangular area in the key frame that contains all matched visual words. See Algorithm 1 for more details about the spatial consistency.

To accelerate the search process, we calculate the spatial consistency only for a fixed number (Δ) of frames that were previously returned by relevance ranking. Therefore, we consider only the Δ frames more relevant, according to the previous ranking result.

5 Experiments

5.1 Setting of Parameters

In the experiments, one episode of the CBS series The Big Bang Theory was used. The video contains 28682 frames, with 19 minutes and 56 seconds (476 shots and the average size of 59 frames per shot). The page size for M-tree and for Slim-tree was set to 100 entries. The the number of levels of the D-index was set to 9, and an elastic page implementation was adopted (with 100 entries per block). Moreover, the exclusion radius was set to 20 and pivots were randomly selected (2 pivots for the first level and 1 pivot for the others). Euclidean norm was adopted for all three structures. In spatial consistency algorithm, the number of frames analyzed (Δ) was set to 100, while the number of the nearest neighbors considered (k) was set to 15. These parameters were chosen after an analysis of preliminary test results. All experiments were performed on Intel Xeon E5345 2,33GHz, with 8Gb RAM memory, operating system Windows 2008 R1 x64.

5.2 Building Vocabulary

Initially, the video has been sampled in a simple way in which one frame per second was extracted, totaling 1146 frames. From these frames, a random sample of 96 ($\approx$ 8% of total) frames was extracted to create the visual vocabulary (similar proportion was adopted by [25]). Then, 61628 descriptors from the regions detected by the MSER and 86135 descriptors from the points detected by Harris-Affine were generated. In order to build the visual vocabulary, K-means algorithm was applied separately for each type of detector, trying to keep the ratio of 3:5 between Harris-Affine and MSER descriptors, in accordance with [25].

For the experiments, we used four different sizes of vocabularies: 4000, 8000, 16000, and 32000. Each of these vocabularies were indexed using the inverted file, whose index

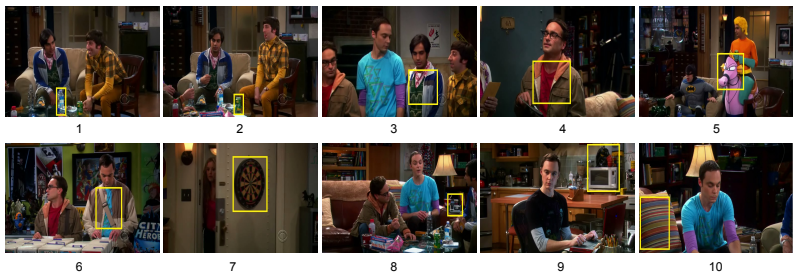


Figura 3. Video frames used for queries. The rectangle in the image indicates the query.

was implemented in four different ways: using sequential access approach and using the structures mentioned in the Section 2.2 (i.e., M-tree, Slim-tree, and D-index).

5.3 Groundtruth Description

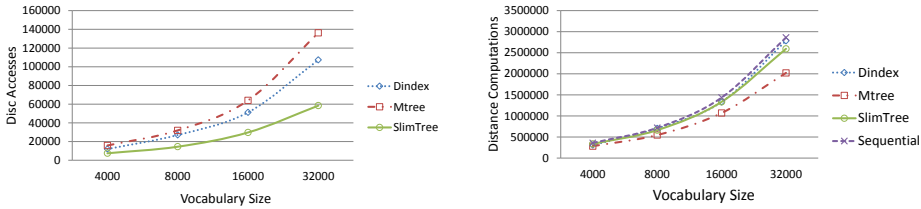
For the queries, we selected 10 different objects from distinct video frames, see Figure 3. We have selected some objects that appear very often throughout the video from different camera angles (and even partially hidden). We have also chosen objects with few occurrences in the video, in order to assess the search results for different scenarios. In order to analyze the test results, a groundtruth was manually created by counting the objects occurrences in the key frames of the video.

5.4 Quantitative Evaluation

In this section, we present experimental results as function of the visual vocabulary size. We evaluate the number of disk accesses performed and the number of distance calculations. In order to measure the impact of spatial consistency, we present the results of two experiments: (i) we ignore the spatial consistency; (ii) the spatial consistency is applied. We used F_1-*Score* to jointly assess precision and recall rates, see Equation 1.

$$F_1\text{-}Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{1}$$

Figure 4(a) shows the average number of disk accesses as a function of the visual vocabulary size without the application of spatial consistency. The values for sequential access method are omitted because they are almost 45 times greater than the values for the others. One should also notice that the behavior of Slim-tree was the least affected by the vocabulary size increase. Slim-tree has always presented the lowest values for the average number of



(a) (b)

Figura 4. (a) Average num. of disk accesses and (b) average num. of distance computations.

disk accesses than the others. Figure 4(b) presents the average number of distance computations as a function of the vocabulary size without the application of spatial consistency. M-tree has presented the best performance, although all the results are very similar.

Table 1 presents the average results for different vocabulary sizes before and after the application of spatial consistency. In general, one can observe a significant improvement in F_1-*Score* after the application of spatial consistency. Moreover the adoption of MAMs not

Tabela 1. Average results before and after the application of spatial consistency.

Access Method	Vocabulary Size	Before Spatial Consistency			After Spatial Consistency		
		Recall	Precision	F_1-*Score*	Recall	Precision	F_1-*Score*
D-index	4000	50.00%	27.60%	35.57%	41.67%	43.40%	42.51%
	8000	52.90%	29.20%	37.63%	44.93%	35.23%	39.49%
	16000	55.43%	30.60%	39.43%	34.78%	43.05%	38.48%
	32000	51.09%	28.20%	36.34%	37.32%	48.82%	42.30%
M-tree	4000	50.00%	27.60%	35.57%	43.48%	32.88%	37.44%
	8000	52.90%	29.20%	37.63%	44.57%	35.55%	39.55%
	16000	55.43%	30.60%	39.43%	34.78%	42.29%	38.17%
	32000	50.72%	28.00%	36.08%	36.96%	50.25%	42.59%
Slim-tree	4000	50.00%	27.60%	35.57%	45.65%	34.62%	39.38%
	8000	52.90%	29.20%	37.63%	44.93%	35.33%	39.55%
	16000	55.43%	30.60%	39.43%	34.78%	42.29%	38.17%
	32000	50.72%	28.00%	36.08%	34.78%	48.73%	40.59%
Sequential	4000	39.49%	21.80%	28.09%	38.04%	24.88%	30.09%
	8000	51.45%	28.40%	36.60%	35.87%	32.04%	33.85%
	16000	48.55%	26.80%	34.54%	36.96%	43.22%	39.84%
	32000	47.83%	26.40%	34.02%	37.68%	54.17%	44.44%

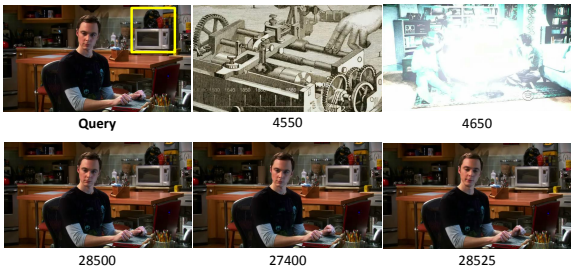


Figura 5. Results for query number 9 without the application of spatial consistency.

only has improved the search performance but also has reduced the influence of the vocabulary size over test results, which may improve the scalability of our proposal. We observed that the use of visual vocabulary is important for reducing the size index (accelerating the search), even if it increases the number of false positives in the search results.

5.5 Qualitative Analysis

In order to assess qualitatively the results, we present in detail the results for query object 9 (see Figure 3) when D-index was used as indexing structure and the vocabulary size is set to 8000. The application of spacial consistency has had a direct impact on the results.

Figure 5 presents the first five (more relevant) key frames returned for the query object 9 without the application of spatial consistency. One could easily see that the first two (and more relevant) key frames returned are false positives, since these two key frames do not contain the selected object (a microwave oven) – they only contain similar visual words which are then responsible for making their bags of words also very similar.

The application of spatial consistency seem to be a key factor in the results improvement. Figure 6 also presents the results of the first five key frames for query object 9 – with spatial consistency. Spatial consistency has had a crucial role to obtain better quality results.

6 Conclusions

This article presents an approach to object retrieval that searches for and localizes all the occurrences of an object in a video database, given a query image of the object. Video key frames are represented by a set of visual words (viewpoint invariant region descriptors) that enable recognition to proceed successfully despite changes in camera viewpoint, lighting,

and partial occlusions. A visual vocabulary formed by those visual words is used to create an index to retrieve all relevant key frames from the video database. We investigate the use of different metric access methods to accelerate the processing of similarity queries. Moreover, we fully describe a ranking strategy – named spatial consistency – that uses visual words spatial layout for re-ranking the results and therefore improving their quality.

Experimental results were used to analyze the behavior of the different metric access methods. We have observed that the adoption of MAMs not only has improved the search performance but also has reduced the influence of the vocabulary size over test results, which may improve the scalability of our proposal. We observed that the use of visual vocabulary is important for reducing the size index, accelerating the searching for objects in the video, even if it increases the number of false positives in the search results.

In general, there was always an increase in the results quality when spatial consistency was used. However, it is important to point out that for sequential access method, this operation consumes nearly half the total search time which is already high. For the other structures, the time spent on the spatial consistency was less significant even in the worst case (30% for M-tree, 20% for Slim-tree, and only 10% for D-index in average for a 4000-word vocabulary). Finally, according to the number of distance computations, M-tree has always presented the best performance compared to the others, while Slim-tree was better when we have analyzed the number of disk accesses performed.

Some future lines of research are: to explore other algorithms for the construction of visual vocabulary; to analyze the impact of efficient ways for creating the index structures, such as the use of bulk-loading algorithms; to investigate the impact of the pivot selection policy over the D-index performance; and to evaluate the adoption of other metric access methods to index a vocabulary of visual words.

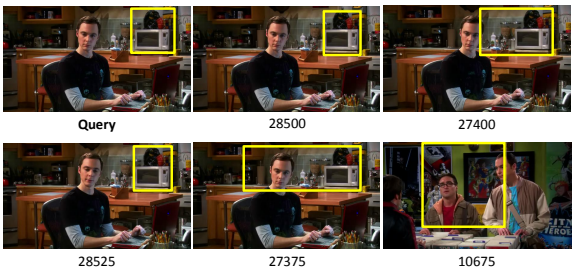


Figura 6. Results for query number 9 with the application of spatial consistency.

7 References

[1] J. Almeida, E. Valle, R. S. Torres, and N. J. Leite. Dahc-tree: An effective index for approximate search in high-dimensional metric spaces. *JIDM*, 1(3):375–390, 2010.

[2] P. Browne and A. F. Smeaton. Video retrieval using dialogue, keyframe similarity and video objects. In *IEEE ICIP*, volume 3, pages 1208–1211, Dublin , Ireland, Sept. 2005.

[3] E. Chavez, G. Navarro, R. Baeza-Yates, and J.L. Marroquin. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, Mar. 2001.

[4] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB*, pages 426–435, San Francisco, USA, 1997.

[5] V. Dohnal, C. Gennaro, P. Savino, and P. Zezula. D-index: Distance searching index for metric data sets. *Multim. Tools App.*, 21(3):9–33, 2003.

[6] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *ACM CIVR*, 2009.

[7] W. B. Frakes and R. A. Baeza-Yates. *Information retrieval: data structures and algorithms*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[8] S. J. F. Guimarães, Z. K. G. Patrocínio, Jr, H. B. Paula, and H. B. Silva. A new dissimilarity measure for cut detection using bipartite graph matching. *Int. Journal of Semantic Computing*, 3(2):155–181, 2009.

[9] T. Homola, V. Dohnal, and P. Zezula. Sub-image searching through intersection of local descriptors. In *SISAP*, number 3, pages 127–128, New York, USA, 2010.

[10] H. Jegou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87(3):316–336, 2010.

[11] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. PAMI*, 33(1):117–128, 2011.

[12] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. PAMI*, 34(9):1704–1716, 2012.

[13] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. PAMI*, 24(7):881–892, 2002.

[14] A.P.B. Lopes, S.E.F. Avila, A.N.A. Peixoto, R.S. Oliveira, and A.A. Araújo. A bag-of-features approach based on hue-sift descriptor for nude detection. In *EUSIPCO*, pages 1552–1556, Glasgow, Scotland, 2009.

[15] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150 – 1157, Washington, USA, Set 1999.

[16] E. Mendi and C. Bayrak. Shot boundary detection and key frame extraction using salient region detection and structural similarity. In *Annual Southeast Regional Conf.*, pages 66:1–66:4, New York, USA, 2010.

[17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. PAMI*, 27(10):1615–1630, 2005.

[18] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1-2):43–72, 2005.

[19] A. Ocsa and E. P. M. Sousa. An adaptive multi-level hashing structure for fast approximate similarity search. *JIDM*, 1(3):359–374, 2010.

[20] Z. K. G. Patrocínio Jr., S. J. F Guimarães, and H. B. Paula. Bipartite graph matching for video clip localization. In *SIBGRAPI*, pages 129–138, Belo Horizonte, Brazil, 2007.

[21] Z. K. G. Patrocínio Jr., S. J. F. Guimaraes, H. B. Silva, and K. J. F. Souza. An unified transition detection based on bipartite graph matching approach. In *CIARP*, pages 184–192, Sao Paulo, Brazil, Nov. 2010.

[22] C. A. F. Pimentel Filho and C. A. S. Santos. A new approach for video indexing and retrieval based on visual features. *JIDM*, 1(2):293–308, 2010.

[23] N. Sebe, M. S. Lew, and A. W. M. Smeulders. Video retrieval and summarization. *CVIU*, 92(2-3):141 – 146, 2003.

[24] J. Shao, H. T. Shen, and X. Zhou. Challenges and techniques for effective and efficient similarity search in large video databases. *VLDB End.*, 1(2):1598–1603, 2008.

[25] J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Trans. PAMI*, 31(4):591 –606, 2009.

[26] C. Traina Jr., A. J. M. Traina, C. Faloutsos, and B. Seeger. Fast indexing and visualization of metric data sets using slim-trees. *IEEE Trans. KDE*, 14(2):244–260, 2002.

[27] C. Traina Jr., A. J. M. Traina, B. Seeger, and C. Faloutsos. Slim-trees: High performance metric trees minimizing overlap between nodes. In *EDBT*, pages 51–65, 2000.

[28] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundation and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.

[29] M. R. Vieira, C. Traina Jr., F. J. T. Chino, and A. J. M. Traina. Dbm-tree: A dynamic metric access method sensitive to local density data. *JIDM*, 1(1):111–128, 2010.