A Scheme for Content-Based Image Retrievals for Unrestricted Query Formats

S.R. Subramanya, Punpiti Piamsa-nga, Nikitas Alexandridis, Abdou Youssef
Department of Electrical Engineering and Computer Science,
The George Washington University,
Washington, DC 20052.
{subra,punpiti,alexan,youssef}@seas.gwu.edu

Abstract. *The explosive increase in the amounts of image (and multimedia) data being generated, processed, and used in several computer applications have necessitated the development of image (and multimedia) database systems with newer features such as* content-based indices *and* similarity searches. *This paper presents a scheme for content-based retrievals in image databases for queries-by-example, where the queries can have unrestricted formats. The proposed method is shown to give better retrievals compared to the scheme which scales the query images before retrieval.*

Keywords: Image databases, query-by-content, color histograms, K-tree.

1 Introduction

There has recently been a phenomenal increase in the use of images, along with text and other *multimedia data* such as audio and video, in a variety of computer applications, and this trend is expected to increase remarkably in the future. This has necessitated newer schemes for storage, processing, retrieval, and transmission of such data. Example applications are in digital libraries, radiological image archives, satellite imagery for earth resources, law enforcement, among others. Fast and accurate image retrievals for content-based user queries are crucial for such systems to be useful.

Several schemes have been proposed for image retrievals for content-based queries. [1, 2, 3, 4]. A unified model for multimedia retrieval by content based on K-trees was proposed in [5]. Several schemes including the one proposed in [5] use query-by-example, and assume that the query image is of the same size as the images in the database. In case they are not, then the queries are scaled to the required size. In this paper, we propose an algorithm where this restriction is relaxed, and allowance is made for *unrestricted query formats*, where, (1) the queries could be of smaller sizes than the images in the database (This reflects the common situation where an *image clip* of a smaller size is given as query), (2) the queries are not required to be a square, and (3) the queries are not necessarily aligned with any quadrant of any image.

In this paper, we assume that K-tree is used as the index structure, which has already been built using the database of images, as described in [5]. The processing of *unrestricted* queries and retrieval of relevant images is focussed in this paper. In subsequent discussion, the phrase 'matching (portion of) query with a quadrant' refers to finding the 'distance' between the histograms of the (portion of) query and the data quadrant, and 'best match' refers to the match with the least distance.

The next section gives a description of the model used in our image indexing and retrieval system and formulates problem, Section 3 presents the proposed algorithm. Experimental results are given in Section 4, followed by conclusions.

2 System Model and Problem Formulation

The system for indexing and retrieval of multimedia data that we are developing is based on multi-resolution processing of a unified structure. Currently the system is built to handle images, but further extensions for handling audio and video are under way. For images, the color histograms at various image levels (resolutions) are used as indices. The indices are organized in a K-tree, which facilitates multiresolution processing. The global features appear at the top levels of the tree and local, finer details are stored at the lower levels. This structure enables tuning the searches for the required search speed and the search accuracy. It also supports *progressive transmission* which facilitates quick 'effective' retrievals of large amounts of data over networks.

(The details of the construction and searching in K-tree are given in [5] and will be briefly described in the full paper, if necessary)

The problem is formulated as follows:

Given: a data base of N images of fixed sizes $S \times S$ and a query image Q of size $W \times H$, usually smaller than the image, and the query size not necessarily equal to any quadrant size, and not necessarily aligned with any quadrant.

Objective: retrieve all images from the database which contain a good match of the given query in any portion of the image, with minimal false hits and false misses, as quickly as possible.

Note: (1) The similarity measure is the euclidean distance between the color histograms. (2) False hits are those retrievals which have no visual resemblance to the query in any portion of the image. (3) False misses are those images which are unreported, but indeed have some portion(s) of the image resembling the query.

3 The Proposed Algorithm

The proposed algorithm handles queries of unrestricted format as mentioned in Section 1. We only assume that the queries are rectangular and that they are oriented with their sides parallel to the sides of the images. The algorithm determines the best match for the query in any portion of each of the images in the database and reports a given number of best matches. A brief description of the algorithm is given below followed by the pseudocode.

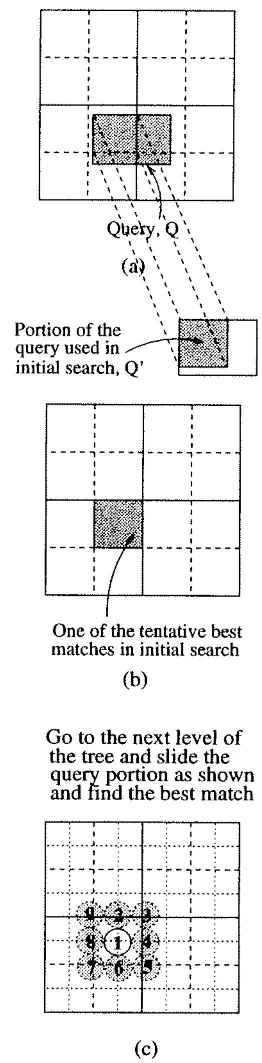

Figure 1: Determination of the best match and alignment of query

First, the level j of the index tree is determined such that the quadrant size S_j at that level is less than or equal to the minimum of the width and height of the query. This is shown in Figure 1(a).

Next, only the initial portion of the query Q' (shown in (b)) is matched with all quadrants at level j and the best few matches (4 in our experiments) are determined. These are referred to as *stage 1* tentative matches and an attempt is made next to find the portion of the data which matches the query portion even better. The query portion is divided into four quadrants, and a descent is made to the next level of the tree. For each of the *stage 1* matches, the query portion is slid around as shown in Figure 1(c). The circles represent the locations in the image where the center of Q' is placed and the match is determined. The best match is determined among the nine positions. This is repeated for all the *stage 1* tentative matches, and the best overall among them is picked as the *stage 2* match. This is taken as the best possible place for the query in the whole image. (Note that this process can be continued until the division of the image reaches the pixel level. This gives better matches but at the cost of increased computation. However, after a certain level, the improvement in the retrievals is not remarkable. In our experiments, we stop at just one lower level–$j+1$ in stage 2 for tentative matches). Having fixed the 'possible' level k and location of the query within the image, we next 'unfold' the query over the image data and determine the distance between the query and data quadrants. First, the distance between the query and data quadrants of size S_k are determined, and then the 'excess portions' of the query are matched by going down the tree levels as necessary. This is illustrated in Figure 2, and described in function FINDDISTANCE. This procedure is repeated for all images and the best β matches are returned.

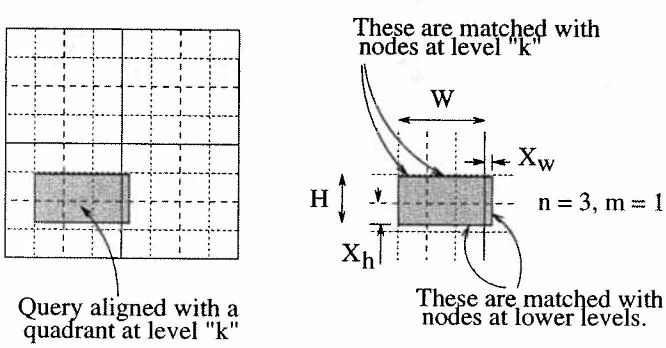

Figure 2: Determination of match distance of query

The pseudocode of the proposed algorithm is given below:

Algorithm 3.1 FINDMATCHES **(in:** K-tree index, Query clip; **out:** Best β matches)

1. **begin**
2. *Find the level j of the K-tree such that the quadrant size at that level $S_j \leq \texttt{MIN}(W, H)$.*
3. *Consider Q', the initial portion of the query, of size S_j as shown in Fig 1(b).*
4. *Find the best b_1 matches of Q' among all quadrants at level j.*
   {These are stage 1 matches}
5. **for** *each of the stage 1 matches* **do**
6. *Consider the four quadrants of Q', descend to level $j+1$ and slide Q' around the nine positions as shown in Fig. 1(c), and find the position of the best match.*
   {Note that the above step can be done iteratively by going down several levels and finding the level k with the best match}
   {This is stage 2 match}
7. **endfor**
8. *Find the best among the best matches obtained in steps 5–7.*
   {Fix the query at the position of the best match of stage 2.}
9. FINDDISTANCE(K − tree, Q, k, pos).
   {Find the distance between the whole query and the portion of image data.}
10. *Repeat steps 2–8 for all images, and return the best β matches.*
11. **end**

Algorithm 3.2 FINDDISTANCE (**in:** K-tree,Q,k,pos; **out:** distance)

1. **begin**
2. $n \leftarrow W/S_k$; $m \leftarrow H/S_k$;
3. $X_w \leftarrow W - n \cdot S_k$; $X_h \leftarrow H - m \cdot S_k$;
4. *Do the matching of the $n \times m$ blocks of the query with the corresponding blocks of the image data, and accumulate the distance.*
   {Next match the 'excess' portions of query(Fig. 2).}
5. *Match the 'n' pieces of the query, each of size $S_k \times X_h$, 'm' pieces of the query, each of size $X_w \times S_k$, and one piece of size $X_w \times X_h$ by suitably subdividing the pieces and descending to appropriate levels and positions of the tree and accumulate the distance.*
6. *Return* distance.
7. **end**

4 Experimental Results

The proposed algorithm has been implemented and used to retrieve images from a database of about 30,000 images, using image clips as queries-by-example. The database is built using images from [6] and [7]. The proposed method is compared with the scheme where the query image clip is normalized to the size of the images in the database (earlier algorithm). Figures 3 and 4 show respectively the retrieval results using the proposed algorithm and the earlier algorithm. In both the figures, the query was the portion of the input whose outline is marked. As shown, the retrievals in the proposed scheme are more relevant (visually) for the given queries.

Note: The visual relevance of the reported matches to the query is better seen in color, since the scheme is based on color histograms.

Figure 3: Retrieval results for query using the proposed algorithm

Figure 4: Retrieval results for query using the earlier algorithm

5 Conclusions

This paper proposed a scheme for content-based retrievals in image databases for queries-by-example, where the queries could have unrestricted formats. The proposed method is observed to be much faster than a brute-force method, and is shown to be more accurate than the scheme which normalized the query size before retrievals. The superiority of the proposed method was established by experiments.

References

[1] Gudivada, V. and Raghavan, V. (Eds.) 'Special Issue on Content-Based Image Retrieval Systems', *IEEE Computer*, Vol.28, No.9, September 1995.

[2] *SPIE: Image Storage and Retrieval Systems*, SPIE, San Jose, CA. Feb. 1992.

[3] Tamura, H. and Yokoya, N. 'Image Database Systems: A Survey', *Pattern Recognition*, Vol. 17, No. 1, 1984, pp29-43.

[4] Chang, S-K. 'Image Information Systems', *Proc. IEEE*, Vol. 73, No. 4, April 1995, pp 754-764.

[5] Piamsa-nga, P, *et. al.* 'A Unified Model for Multimedia Retrieval by Content', *Int'l. Conf. on Computers And Their Applications*, Hawaii, March 1998.

[6] Eastman Kodak homepage. URL: `http://www.kodak.com`.

[7] Smithsonian Institute 'Online Collection of Pictures', URL: `ftp://photo1.si.edu`