Image Querying

Ilaria Bartolini
DEIS - University of Bologna, Italy
i.bartolini@unibo.it
http://www-db.deis.unibo.it/~ibartolini

SYNONYMS

Image query processing

DEFINITION

Image querying refers to the problem of finding objects that are *relevant* to a user query within image databases (Image DBs). The classical solutions to deal with such problem include the *semantic*-based approach, for which an image is represented through metadata (e.g., keywords), and the *content*-based solution, commonly called *content-based image retrieval* (CBIR), where the image content is represented by means of low-level features (e.g., color and texture). While for the semantic-based approach the image querying problem is transformed into an information retrieval problem, for CBIR more sophisticated query evaluation techniques are required. The usual approach to deal with this is illustrated in Figure 1: By means of a graphical user interface (GUI), the user provides a query image, by sketching it using graphical tools, by uploading an image she/he has, or by selecting an image supplied by the system. Low-level features are extracted for such image (possibly dividing it into regions, see below); such features are then used by the query processor to retrieve the DB images having similar characteristics.

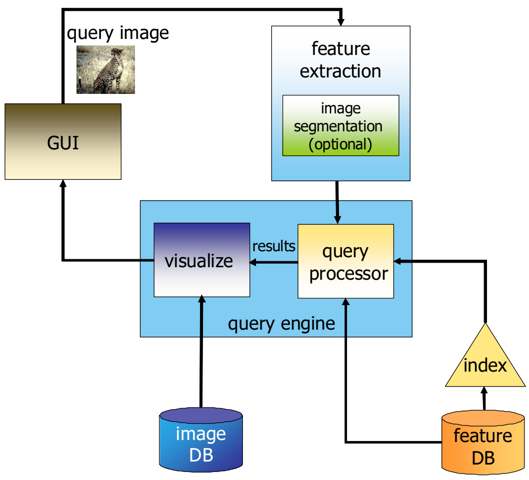

Figure 1: The image querying scenario for CBIR.

How the set of relevant DB images is determined depends on which low-level features are used to characterize image content, on the criterion used to *compare* image features, on how DB objects are *ranked* with respect to the query (based on either a quantitative measure of similarity or qualitative preferences), and, finally, on whether the user is interested in the whole query image or only in a part of it. All these aspects strongly influence the *query evaluation* process.

HISTORICAL BACKGROUND

In spite of the many efforts spent so far, the problem of retrieving relevant objects within image databases (Image DBs) is still a complex task. Following the semantic-based approach, images are described by means of metadata such as keywords, captions, or descriptions and the retrieval is performed over such words using annotations-based information retrieval techniques. In this direction, several solutions have been recently proposed, such as image extensions to public search engines like Google (http://images.google.com/) and Yahoo (http://images.search.yahoo.com/). Such systems consider the contextual information of a crawled image (like the image filename, its title, the surrounding text, etc.) to infer the relevance of the image to the query. In a similar way, some systems, like flickr (http://flickr.com/), assess the relevance of an image to the query by taking into account the characterization of the image provided by the user. However, a such manual image annotation process is expensive and time-consuming. In order to overcome such limitations, there has been a large amount of research done on (semi-)automatic image annotation with the aim to assign meaningful keywords to images by exploiting the information of a pre-annotated set of objects.

With respect to the *content-based image retrieval* (CBIR) solution, the aim is to avoid the use of textual descriptions. This is usually done by using visual similarity to retrieve images, for example, asking for images that are similar to a user-supplied query image, i.e. following the *query by example* (QBE) paradigm fist adopted in the IBM's query by image content (QBIC) system [5]. In particular, each image is characterized using global low-level features, such as color and texture, and the result of a query consists in the set of DB objects that better match the visual characteristics of the target image, according to a predefined similarity criterion, which is in turn based on low-level features [9]. Although CBIR represents a completely automatic solution for the image querying problem, the accuracy of its results is not always completely satisfactory for the user, especially for high-level concept queries, for which low-level features are hardly exploitable due to their low discriminative power. This is largely due to the so-called semantic gap existing between the concept of similarity as percept by the human brain and the one implemented by the system. The effectiveness of this approach still calls for improvement: The use of relevance feedback techniques could be of help, but it is still not enough to reach acceptable levels of accuracy.

More recently, the *region-based image retrieval* (RBIR) approach has been proposed, which has lead to promising results. With respect to the case in which images are represented by means of global descriptors, RBIR is able to characterize the image content in a more precise way by segmenting each image into a set of homogeneous regions from which a set of low-level features are extracted. As a consequence, most of the modern image database systems adopts the RBIR paradigm in order to improve the retrieval accuracy [8, 1, 2, 4, 6, 10]. Almost all such systems treats each region as a separate query and somehow aggregate the so-obtained partial results in order to derive the final answer. This property introduces a number of new interesting query processing problems with respect to the case in which the segmentation is not considered. Among these, which constraints must be satisfied by the aggregation rule in order to provide the query result and which criterion has to be followed to order the DB images with respect to the query. Finally, with RBIR, new query types, such as partial queries, are supported.

SCIENTIFIC FUNDAMENTALS

The general approach followed by RBIR systems is to divide an image I into a set of homogeneous regions, i.e., set of pixels that share similar visual characteristics, and to represent each of them by means of a set of low-level features, such as color and texture. Thus, any image I is seen as a complex object. Regions comparison is obtained by defining a *region similarity function*, s_R, able to produce a scoring value which quantifies their visual similarity. Given a query image I^q, the set of relevant DB images to I^q is computed starting from the similarities between the query regions and the regions of DB images. This requires first to somehow *match* regions of the query to regions of DB images, by using the proper aggregation of region similarities, and then to *rank* DB images so as to produce the query result (see Figure 2). Formally, the image querying problem can be concisely formulated as follows:

Problem 1. *Given a query image I^q composed of regions, an image database $\mathcal{IDB}$, where each image $I \in \mathcal{IDB}$ is composed of regions, and a region similarity function, $s_R(R_i, R_j)$, that for each pair of regions, (R_i, R_j), returns*

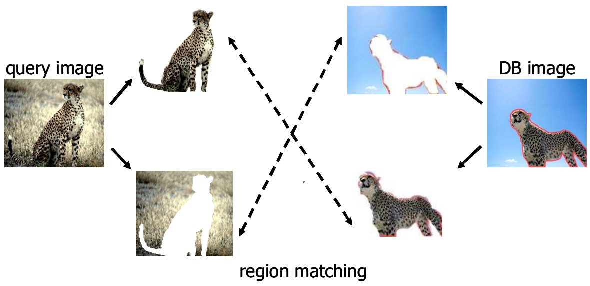

Figure 2: The similarity between the query image and a DB image is assessed by taking into account the similarity between matched regions.

their similarity score, determine the set of relevant *images in* $\mathcal{IDB}$ *wrt* I^q*.*
Instantiating the general problem can be done in different ways, since different coordinates are involved in the definition of what "relevant" actually means. Among these coordinates, the rules according to which a region of the query can be coupled to regions of a DB image (conventionally called *matching type*) and the aggregation modality applied to the region similarity scores in order to assess the query result (i.e., the *ranking model*).

Matching Type. The matching type defines which set of constraints applies when the component regions of the query image $I^q = \{R_1^q, \ldots R_n^q\}$ have to be matched to the component regions of a DB image $I = \{R_1, \ldots R_m\}$. Two relevant cases for matching types are the *one–to–one* $(1-1)$ and the *many–to–many* $(n-m)$ matching types. In the $1-1$ case, each region of image I^q is associated to at most one region of I, and vice versa. In particular, each matching has to be *complete*, i.e., if $n > m$ (respectively, $n < m$) then only $n - m$ (resp., $m - n$) regions of I^q (resp., I) have to remain unmatched (refer to Figure 3 for an example).
With the $n - m$ matching type, each region of I^q can be associated to many regions of I, and vice versa. This, however, could lead to undesired (pathological) results: For example, a single region of the query could be matched to *all* regions of a DB image. This has been termed "the two tigers problem" in [11] since it arises when a single region (a tiger) of the query image is very similar to multiple regions of a DB image (e.g., containing two tigers). A special case of $n - m$ matching that avoids this problem is the Earth Mover Distance (EMD) matching [7], where variable-sized pieces of regions are allowed to be matched (the size of each region defines the maximum amount for its matching). This contrasts with the $1 - 1$ matching, where elements of fixed size (i.e., regions) are matched individually.

Ranking Model. Two generic models of ranking are possible: *k-Nearest Neighbors* (k-NN) and *Best Matches Only* (BMO) [3]. The k-NN ranking model (also known as Top-k selection) requires to define the image similarity of a DB image I with respect to a query image I^q, $s_I(I^q, I)$, by means of a *numerical scoring function* (sf), such as the average, which aggregates the region similarity scores into a global similarity value. In particular, among all valid matchings that satisfy the constraints of the specific matching type, the rationale is to select the one that maximizes the aggregated score. This can be modelled as an optimization problem whose solution depends on the particular choice of the scoring function. For the most commonly used functions efficient algorithms exist: For example, when using the average function with the $1 - 1$ matching type, the problem takes the form of the well-known *assignment problem*, while with the $n-m$ (EMD) matching type (see above), this corresponds to the *transportation problem*. For both such problems the optimal solution can be efficiently found without performing an exhaustive search; however, in the general case, the optimal matching can not be easily found. Figure 3 shows an example of matching for a query image I^q with 3 regions and a DB image I with 4 regions under the assumption of $1 - 1$ matching type and the average scoring function. Similarities between regions of I^q and regions of I are arranged in a matrix. Circled cells are those involved in the matching. Note that, since $n < m$, in valid matchings $4 - 3 = 1$ region of I remains unmatched.
Finally, given the query image I^q and two DB images I_1 and I_2, image I_1 will be considered more similar than I_2 to I^q iff $s_I(I^q, I_1) > s_I(I^q, I_2)$ holds. In such a way it is possible to linearly order DB images and return to the user only the k highest scored ones.

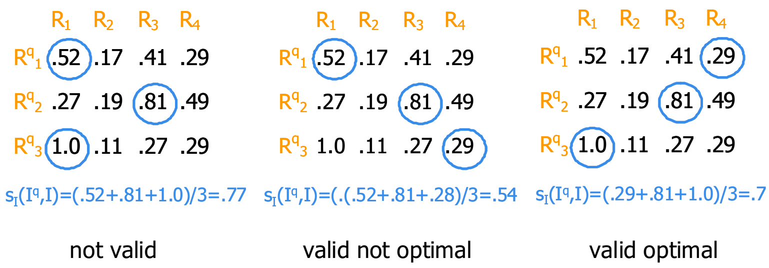

Figure 3: Example of similarity assessment between images I^q and I when adopting the 1-1 matching type and the average scoring function: not valid matching (left), valid not optimal matching (center), and valid and optimal matching (right). If an alternative matching type (e.g., the general $n-m$ matching) is considered, the left matching could become valid (and optimal).

The main limitation of the k-NN ranking model is that the choice of a particular scoring function clearly influences the final result, i.e., different scoring functions will likely yield different results. This can lead to missing relevant images, because the choice of the scoring function is a difficult task for the user. Moreover, the use of scoring functions limits the expressive power of queries that can be submitted to the system, since all of them will always define a simple linear order on objects: This might prevent their applicability to modern multimedia systems asking for more flexibility in querying [3]. In the BMO model, the result of the query depends on a specific *preference relation* $\succ_p$, where $\succ_p$ is only required to define a strict partial order over images. Image $I_1 \in \mathcal{IDB}$ is in the query result if and only if no other image $I_2 \in \mathcal{IDB}$ is *better* than (or *dominates*) I_1 according to $\succ_p$. Clearly, preference relation $\succ_p$ is based on regions similarity scores (see Figure 3).
Thus, even if region scores are numerical (by definition), the BMO ranking model does not need to aggregate them using a scoring function. Actually, the result of a BMO query with image I^q is the set of undominated images in $\mathcal{IDB}$, i.e., all and only those images for which no better image (with respect to I^q and to $\succ_p$) can be found in the database.

When considering together the matching type and ranking model coordinates, different scenarios are derived. In the following, algorithms for k-NN and BMO queries are provided by considering the simplest way to solve the image querying problem, i.e., when using a sequential scan of the DB. Note that the efficiency of such solutions is clearly quite limited. It is possible to derive efficient algorithms [2, 11, 3] by exploiting index structures, such as multi-dimensional or metric indices (built either on regions of the DB images or on the DB images themselves).
The steps described in Algorithm 1 show the logic of the sequential algorithm for k-NN queries, named k-NNSeq, to determine the k nearest neighbors of the image query I^q: Given the image query I^q, the scoring function sf, and the cardinality of the result k, the algorithm correctly returns the k images that are most similar to I^q according to sf. This algorithm covers both cases of $1-1$ and $n-m$ matchings.

Algorithm 1 k-NNSeq

Require: I^q: query image, k: cardinality of result, $\mathcal{IDB}$: image DB, sf: scoring function
Ensure: set of relevant k images

```
for all images I ∈ IDB do
    for all regions R_j ∈ I do
        for all regions R_i^q ∈ I^q do
            compute s_R(R_i^q, R_j)
    compute matchings between regions R_i^q ∈ I^q and regions R_j ∈ I
    select the matching that maximizes s_I(I^q, I) by means of sf
return the k images having the highest overall similarity scores s_I
```

Algorithm 2, named BMOSeq, describes the main steps for the sequential evaluation of BMO queries, with the assumption of $1-1$ matching: Given the image query I^q and the preference relation $\succ_p$, the algorithm correctly returns set of undominated images with respect to the query I^q.

Algorithm 2 BMOSeq

```
Require: I^q: query image, IDB: image DB, ≻_p: preference relation
Ensure: set of undominated images
1:  U ← ∅
2:  for all images I_1 ∈ IDB do
3:      for all regions R_j ∈ I_1 do
4:          for all regions R_i^q ∈ I^q do
5:              compute s_R(R_i^q, R_j)
6:      U ← U ∪ {I_1}
7:      for all images I_2 ∈ U do
8:          if I_1 ≻_p I_2 then
9:              U ← U \ I_2
10:         else if I_2 ≻_p I_1 then
11:             U ← U \ I_1
12:             break (for at line 7)
13: return images in U
```

It is also important to consider the type of images the user is interested in. The above description deals with *full image* search, i.e., when the user is interested in all regions of the query, but other possibilities exist that introduce minor modifications in the query evaluation process.

For example, *part-of* queries request DB images whose regions are all matched to some query region (the presence of unmatched query regions is not penalized). Two other query types are introduced when the user is given the possibility to select, possibly exploiting a suitable graphical interface, only a subset of query regions: In *partial match* queries the user is looking for DB images containing selected regions of the query (the presence of other regions in the DB image should not be penalized); on the other hand, with a *contains* query DB images are requested to contain selected query regions only (other existing regions reduce the image similarity, differently from the case of part-of queries).

KEY APPLICATIONS

Image querying is an important tool for many modern multimedia applications, such as *digital libraries*, *e-commerce* (where electronic catalogues have to be browsed and/or searched), *edu-tainment* (for example, to search in clipart repositories, or to search and organize personal photo albums in mobile phones or PDAs).

Another interesting application area is the one related to (semi-)automatic *image annotation techniques*, which can be based on assigning to a unlabelled image I the keywords associated to the DB images most similar to I.

Finally, image querying techniques have been also profitably used in *image classification*, for example, to search for similar logo images, for copyright infringement issues, and for the detection of pornography images.

CROSS REFERENCES

IMAGE DATABASE
LOW-LEVEL IMAGE CONTENT ANALYSIS (COLOR, TEXTURE, SHAPE)
IMAGE SIMILARITY
IMAGE RETRIEVAL AND RELEVANCE FEEDBACK
IMAGE SEGMENTATION
ANNOTATION-BASED INFORMATION RETRIEVAL
TOP-K SELECTION QUERIES ON MULTIMEDIA DATABASES
MULTIDIMENSIONAL INDEXING

RECOMMENDED READING

[1] S. Ardizzoni, I. Bartolini, and M. Patella. Windsurf: Region-based image retrieval using wavelets. In *Proceedings of the 1st International Workshop on Similarity Search (IWOSS'99)*, pages 167–173, Florence, Italy, Sept. 1999.

[2] I. Bartolini, P. Ciaccia, and M. Patella. A sound algorithm for region-based image retrieval using an index. In *Proceedings of the 4th International Workshop on Query Processing and Multimedia Issues in Distributed Systems (QPMIDS'00)*, pages 930–934, London/Greenwich, UK, Sept. 2000.

[3] I. Bartolini, P. Ciaccia, V. Oria, and T. Özsu. Flexible integration of multimedia sub-queries with qualitative preferences. *Multimedia Tools and Applications*, 33(3):275–300, June 2007.

[4] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Proceedings of the 3rd International Conference on Visual Information Systems (VISUAL'99)*, pages 509–516, Amsterdam, The Netherlands, June 1999.

[5] M. Flickner and H. S. Sawhney and J. Ashley and Q. Huang and B. Dom and M. Gorkani and J. Hafner and D. Lee and D. Petkovic and D. Steele and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, Sept. 1995.

[6] A. Natsev, R. Rastogi, and K. Shim. WALRUS: A similarity retrieval algorithm for image databases. In *Proceedings 1999 ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, pages 396–405, Philadelphia, PA, June 1999.

[7] Y. Rubner and C. Tomasi. *Perceptual Metrics for Image Database Navigation.* Kluwer Academic Publishers, Boston, MA, Dec. 2000.

[8] J. R. Smith and S.-F. Chang. VisualSEEk: A fully automated content-based image query system. In *Proceedings of the 4th ACM International Conference on Multimedia (MM'96)*, pages 87–98, Boston, MA, Nov. 1996.

[9] 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 Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, Dec. 2000.

[10] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, Sept. 2001.

[11] R. Weber and M. Mlivoncic. Efficient region-based image retrieval. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management (CIKM'03)*, pages 69–76, New Orleans, LO, Nov. 2003.