Image Retrieval under Image Transformations and Occlusions

Roberto A. Vázquez and Humberto Sossa

Centro de Investigación en Computación – IPN
Av. Juan de Dios Batíz, esquina con Miguel Othón de Mendizábal
Ciudad de México, 07738, México.
Contact: ravem@ipn.mx, hsossa@cic.ipn.mx

(Paper received on July 25, 2007, accepted on September 1, 2007)

Abstract. An important problem in image management is image retrieval. When images in a database are not well organized, their efficient retrieval is a problem. In this paper we describe an image retrieval system able to recover a set of images that satisfy some searching criteria. The proposal works well even if the objects appearing in the images are occluded or suffer affine transformations as rotations and translations. The proposed system is composed of three modules. The first module performs object training and object recognition using associative memories as classification tool. The second module performs image analysis and image organization in a database. The third module allows for image retrieval. Through several experiments we show the efficiency of the system.

1 Introduction

Image retrieval is an important problem nowadays. In the WWW more that 73% of the information is images [8]. Images in this space are, in general, not well or not organized at all. The explosion of image databases and the inefficiency of text-based image retrieval have created an urgent need for effective approaches in image database retrieval. Their search is normally high time-consuming. Many techniques and commercial systems have been proposed in the literature to succeed this task. For several examples, refer to [1-5, 8-11]. Most of these systems use combinations of image-features such as colour or texture to organize the images as a database of images and then to recover them from it. To the final user of the system, normally these image features do not have any meaning or are difficult to interpret and use. To avoid this problem, some systems use images as examples to recover other similar images. For a system to be useful, it is necessary to be designed in accordance to the user's intuition. A person normally knows nothing about describing features, image processing or image analysis. A user of and image retrieval system would like to input directly to the system queries such as:

1. "System, display the set of images with rivers and threes", or
2. "System, show me the images with lions and zebras".

The key problem to most computer vision applications comprising image organization and retrieval is object recognition. Object recognition, in the general case, is still an open problem and it strongly determines the functionality of many systems such as image retrieval systems. In [14] was described an image retrieval system called (SISREC). Given a set of images the system automatically organizes a database. This system was able to recover images from a database in terms of their objects. One of the main restrictions of SISREC was that the objects in the image were not in contact with any other object.

In this paper we describe an improved version of SISREC that is able to recover images now in the presence of object occlusions. The system is composed of three modules that are next described. Is important to mention that modules about to object recognition and image organization are the main contributions of this paper.

2 Description of the system

The system here proposed is composed of three main modules:

1. One of Object Recognition Module (ORM).
2. One of Image Analysis and Structuring Module (IASM), and
3. One of Image Retrieval Module (IRM).

2.1 Object recognition module (ORM)

This module is the most important of the three. Its performance determines the complete efficiency of the other two, and as a result that of the system. This demands using powerful object recognition techniques to get good results. Because of objects can appear occluded we decided to adopt some ideas described in [16] for object recognition under occlusions.

2.1.1. Describing essential parts of an object. For an object to be recognized in an image in the presence of occlusions we first detect its so-called essential parts (EP). An EP part is a part that allows finding out the presence of an object in an image. To detect an EP of an object, we get an image of the object with a background as homogeneous as possible. For an example refer to Figure 1. Then we continue as follows:

1. Manually, with a circular window we first select a region of the image enclosing a candidate EP the object.
2. Inside this window, we apply a standard threshold [12] to get a binary sub-image.
3. We apply to this binary circular image a connected component-labeling algorithm [6] to get all possible connected binary components.
4. We remove small spurious regions with a standard size filter [7].
5. We then calculate well-known first four Hu descriptors, invariant to translations and rotations, to describe locally the selected part. For the details refer to [13].
6. We apply steps 1 to 5 to other parts of the object.
7. We repeat this procedure (steps 1 to 6) to remaining objects.

Special attention has to be put in selecting the set of describing parts. For example, from Figure 1, we might think the head of the bolt and the hole of the washer could be two EPs that will allow differentiation between these two objects. However by comparing the describing features from both parts we have noticed that they could be similar. To select EP or EPs by which we are going to perform object detection under occlusions, we have just taken from the list of pre-selected parts as that EP that allows more discrimination with other objects. For this we have used their corresponding describing features.

2.1.2. The classification tool. An associative memory is a mathematical device specially designed to recall complete patterns from inputs patterns that might be altered with noise. An associative memory M can be viewed as an output-input system as follows: $\mathbf{x} \to \mathbf{M} \to \mathbf{y}$, with $\mathbf{x}$ and $\mathbf{y}$, respectively the input and output patterns vectors. The structure of an associative memory for pattern classification is simply another way to see a neural network. In this work we use an extended associative memory useful to classify real-valued patterns, by assigning them to the class by their index.

(a) (b)

Fig. 1. (a) Head of bolt selected as its essential part. (b) Hole of washer selected as it essential part.

Let $\left(x^{\xi}, i\right)_{\xi=1}^{p}, x^{\xi} \in \Re^{n}, i = 1,\ldots,m$ a set of p fundamental couples (SFC), composed by a pattern and its corresponding class-index. The problem is to build an operator M, using these SFC, that allows classifying the patterns into their classes. This mean, $\mathrm{M} \otimes x^{\xi} = i$ for ξ=1,..., p and that even in the presence of distortions it classifies them adequately, that is $\mathrm{M} \otimes \tilde{x}^{\xi} = i$, where $\tilde{x}^{\xi}$ is an altered version of x^{ξ}. A first approach in this direction was presented in [15]. Operator $\otimes$ is chosen such that when operating vector x^{ξ} with matrix $\mathbf{M}$, produces as result the corresponding index class of pattern x^{ξ}.

Matrix $\mathbf{M}$ is build in terms of a function ϕ as follows:

$$\mathbf{M} = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_m \end{bmatrix} \tag{1}$$

Function ϕ is computing using so-called "sep" operator, which allows us transform a set of relatively close ϕ_i's, into another set of more separated ϕ_i''s. This transformation, as we will next see, allows improving the associative memory's performance. Function ϕ is defined as follows:

$$\phi_{ij} = \gamma_{ij} + \lambda_{ij} \tag{2}$$

where $\gamma_{ij} = \bigvee_{\xi=1}^{p}\left(x_j^{i\xi}\right)$ and $\lambda_{ij} = \bigwedge_{\xi=1}^{p}\left(x_j^{i\xi}\right)$.

Pattern classification is performed as follows. Given a pattern $x^{\xi} \in \Re^{n}$, not necessarily one of the already used to build matrix $\mathbf{M}_{sep}$, class to which x is assigned is given by:

$$i = \mathbf{M} \otimes x^{\xi} = \arg_{l}\left[\bigwedge_{l=1}^{m}\bigvee_{j=1}^{n}\left|m_{lj} - r_{lj}\right|\right] \tag{3}$$

where $r_{lj} = x_j + \gamma_{lj}$.

2.1.3. Building the associative memory for object recognition under occlusions. Once selected the essential feature of each object we would like to recognize, associative memory $\mathbf{M}$ is built as follows:

1. Obtain 20 images of each object in different positions and rotations
2. To each sub-image containing the selected EP, calculate the corresponding Hu's invariants as explained in section 2.1.1.
3. With these values, build corresponding associative memory as explained is Section 2.1.2.

Once trained the classification tool, the output of the ORM is the identity of a given object (from image-features to objects).

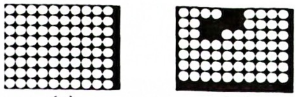
2.2 Image analysis and structuring model (IASM)

This module receives as input a set of *n* images containing one or more instances of the objects already learned by the system even under occlusions. To determine if an instance of an object is in an image we use so-called blocking swapping algorithm (BSA). This algorithm allows extracting information from an image to be used to operate the associative memory already trained. Information is extracted by means of a mask MA of size equal to the original image. This mask is filled with circular windows of ratio of 15 pixels, see Figure 2 (a). To avoid analysing several times the same region of the image, the BSA uses a blocking table. Blocking table is used to decide if a window is used or no, for an example refer to Figure 2 (b). Blocking table allows blocking those regions in the image that have been selected as regions containing the distinctive part of an object. Detection of the essential part of an object is performed in six steps as follows:

For each of the *n* images to be structured:

1. Clear blocking table.

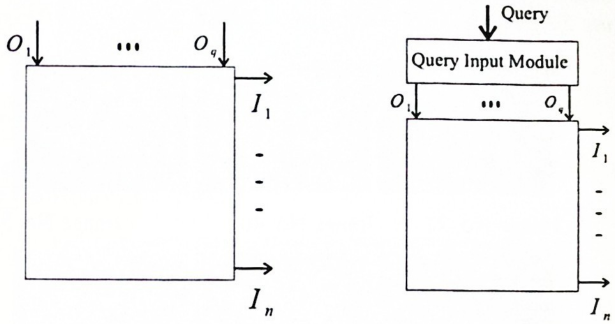
2. Refresh mask **MA**.
3. Apply logical **and** operation between mask **MA** and image to be analysed.
4. Compute first four Hu invariants to each region enclosed by circular window in the new image.
5. Apply associative memory to each obtained describing vector and if vector corresponds to essential part of one of the objects we are looking for, then a vote is given for this object and block associated region in blocking table.
6. If whole image has been analysed then finish stop algorithm, else go back to step 2.



(a) (b)

Fig. 2. (a) Mask used to subtract form image the information to train associative memory. (b) Mask with blocked regions.

The output of this module is a list of pointers from each object to the images that contain this object, organizing automatically the set of images in a table or in a database. In table form each locality of this table contains the evidence that a given object $O_i, i = 1, \cdots, q$ is contained in a given image $I_j, j = 1, \cdots, n$, see figure 3(a).



(a) (b)

Fig. 3. (a)The output of the IASM Module is a table of pointers from objects to images. (b) The IRM module receives queries as inputs and outputs list of images satisfying these queries.

2.3 Image retrieval module

The input to this module is a query in terms of the type of objects an image may contain, see Figure 3(b). Its output is a list of the images satisfying the input query. Examples of queries this module accepts follow:

1. A=1 & C=1, means the system will display all images (if any) containing instances of object A and object C.

2. B=0 & D=1& E=0, means the system will display all images (if any) not containing instances of object B and E, and containing instances of object.

3 Experimental results

Performance of the system was tested with a set of five real objects: a bolt, a washer, an eyebolt, a hook and a dovetail. Images of them are shown in Figure 5. Twenty different images of each of these five objects were used to train the associative memory.

(a) (b) (c) (d) (e)

Fig. 4. The five objects used in the experiments. (a) A bolt. (b) A washer (c) An eyebolt (d) A hook, and (e) A dovetail.

One hundred images containing one or more instances of the objects shown in Figure 4 were used to train the classifiers and to organize these images in terms of theirs objects. Figure 5 shows nine of these images.

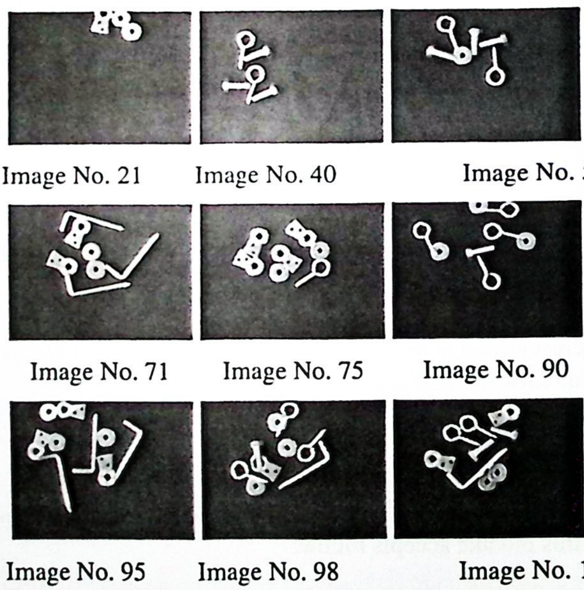


Fig. 5. Nine of the images structured by the system.

With 5 objects and the operators "AND" and "OR", and the restriction that an object is asked to appear or not asked to appear in and image, the number of possible queries that can be input to the system is 1,652: $(A=1 \& B=1)$ is one combination; $(B=1 \& C=0 \mid E=1)$ is another combination.

Once this set of images were organized automatically for the system we test the accuracy of the system to recover images in terms of their objects using only 800 hundred queries (almost the 50% of whole possible queries).

The overall percentage of the system is of 83.0%. This percentage of performance was obtained as follows: Given a query, nt images are output by the system. In only nc of images the desired objects appear and in $nt-nc$ at least one object does not appear (does not satisfy the query). The partial performance percentage for this query is obtained as: nc/nt. By computing this partial percentage for each query and by summing-up the total of partial percentages and by dividing this total by the number of queries we get the total average performance percentage.

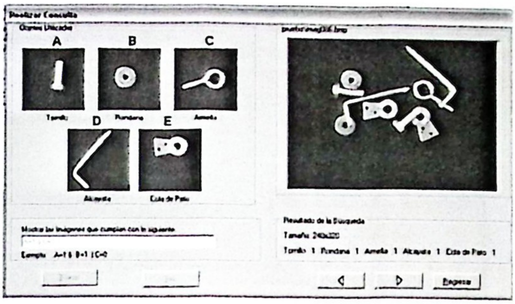


Fig. 6. Image 86 recovered by the system when query A=1 and C=1 is introduced to the system. 43 images were recovered by the system.

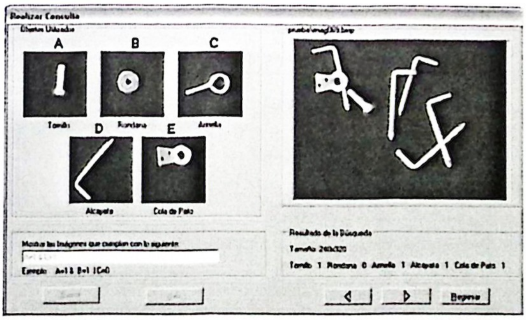


Fig. 7. With the set of 43 image recovered when query A=1 and C=1 is fed to the system, image 69 was recovered. Note there is no an instance of object C, however instances of object A do appear.

Figures 6 and 8 show two of output results. In the first case a set of 43 images were recovered by the system. As you can appreciate in Figure 6, image 86 is output by the system and satisfies the search criterion; only 23 % of recover images do not satisfied the search criterion, one example is given in Figure 7.

In the second case a set of 24 images were recovered by the system. As you can appreciate in Figure 8, image 96 is output by the system and satisfies the search criterion; only 12 % of recover images do not satisfied the search criterion, see for example Figure 9.

The worst results were obtained when the query involves object D. Next we will explain why the low accuracy is obtained when a query involves object D. The total performance of the system depends strongly on the functioning of the ORM. If the system is able to recognize accurately instances of the objects in an image, it can thus be inserted in the right position in the database. Otherwise an erroneous pointer to it will be created producing, of course, at the moment a query is input to the system an erroneous result. The ORM was tested isolated apart from the image retrieval system using the same set of images. The obtained classification results were acceptable. Table 1 summarizes these results.

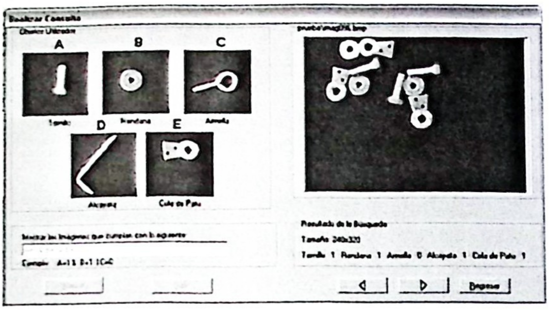


Fig. 8. Image 97 is recovered by the system when query A=1 and B=1 and E=1 is introduced to the system. 24 images were recovered by the system.

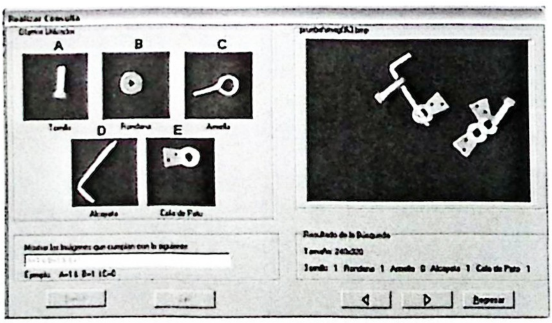


Fig. 9. With the set of 24 image recovered when query A=1 and B=1 and C=1 is injected to the system, image 63 was recover. Note there is not an instance of object B, however instances of object A and E exist.

As you can appreciate in Table 1, the worst classification result was obtained with the hook (Object D). This is because of the essential part selected for the object. This bad classification causes that retrieval system performs a low accuracy when the query involves object D.

4 Conclusion and directions for further research

In this paper we have described a system that is able to first organize a set of input images, and second to allow for recover examples of them given an image query. Three modules integrate the system. The first module performs object training and object recognition. The second module performs image analysis and image organization. The third one allows for image retrieval.

Table 1. Classification results obtained with the object recognition module.

Object	% of classification
Bolt	93%
Washer	92%
Eyebolt	80%
Hook	51%
Dovetail	96%
% total of Classification	**83%**

Through some experiments we have shown the efficiency of the system. The overall performance of the system strongly depends on the correct functioning of the object recognition module. For the set of objects used and the restrictions imposed the system provides good results.

One important feature is that it is close to the end-user that usually ignores everything about image processing and image analysis. It is a first step to design a more general system.

A system with these limitations could be used, for example, to recover photographs of objects circulating through a conveyor belt.

Nowadays, we are working on: 1) a method to obtain automatically the EPS for each object, 2) how to recognize object also in the presence of scale changes, and 3) how to solve the most difficult problem of image retrieval under clutter images. At the end of our research, we hope to count with a system able to structure a set of images (landscapes, portraits, and so on), allowing to recovering them in terms of their objects and their relations.

Acknowledgments. This work was economically supported by SIP-IPN under grant 20071438 and CONACYT under grant 46805.

References

1. A. Del Bimbo, and P. Pal (1997). Visual Image Retrieval by Elastic Matching of User Searches, IEEE Trans on Pattern Analysis and Machine Intelligence, 19(2):121-132.
2. A. Del Bimbo (1999). Visual Information Retrieval, Morgan Kaufmann Publishers, San Francisco, CA.
3. R. Egas et al. (1999). Adapting k-d Trees to Visual Retrieval, Proc. Visual 99, LNCS 1614: 533-540.
4. M. Flickner et al. (1995). Query by image and video content: The QBIC system, IEEE Computer, 28(9):23-32.
5. T. Gevers, and A. Smeulders (1997). PicToSeek: A Content-Based Image Search System for the World Wide Web, Proc. Visual 97, Chicago, pp. 93-100.
6. R. C. Gonzalez and R. E. Woods (2002). Digital Image processing, Second edition, Prentice Hall, Inc 2002.
7. R. Jain et al. (1995). Machine Vision, McGraw-Hill.
8. M. S. Lew (2000). Next-Generation Web Searches for Visual Content, IEEE Computer, 33(11):46-52.
9. M. S. Lew (2001). Principles of Visual Information Retrieval, Advances in Pattern Recognition Series, Springer-Verlag London Limited.
10. J. R. Smith and S. F. Chang (1997). Visually searching the web for content, IEEE Multimedia, 4(3):12-20.
11. L. Taycher, M. Cascia and S. Sclaroff (1997). Image Digestion and Relevance Feedback in the ImageRover WWW Search Engine, Proc. Visual 97, Chicago, pp. 850-91.
12. F. Jiulun and X. Winxin (1997). Minimum error thresholding: A note, Pattern Recognition Letters, 18(8):705-709.
13. M. K. Hu (1962). Visual pattern recognition by moment invariants, IRE Transactions on Information Theory, 8:179-187.
14. H. Sossa et al. (2003). SISREC: A System for Image Retrieval, In proceedings of Forth Mexican International Conference on Computer Sciences (ENC 2003), IEEE Computer Society, 69-73.
15. H. Sossa, R. Barrón, R. A. Vázquez (2004). Real-Valued Pattern Classification based on Extended Associative Memory, In proceedings of Fifth Mexican Conference on Computer Science (ENC2004), IEEE Computer Society 213-219.
16. R. A. Vázquez (2005). Invariant descriptions and associative processing applied to object recognition under occlusions, Master Thesis (In Spanish), CIC-IPN.