Unsupervised Image Retrieval with Similar Lighting Conditions

J. Félix Serrano¹, Carlos Avilés², Humberto Sossa³, Juan Villegas⁴, Gustavo Olague⁵.

¹, ³ Centro de Investigación en Computación – Instituto Politécnico Nacional (CIC- IPN) Av. Juan de Dios Bátiz y Othón de Mendizábal s/n Col. Lindavista C.P.07738 México, D. F.
², ⁴ Universidad Autónoma Metropolitana – Azcapotzalco. Departamento de Electrónica. Av. San Pablo 180, Col Reynosa, México D.F. C.P. 02200
⁵ Centro de Investigación Científica y de Educación Superior de Ensenada CICESE. Ensenada, B.C. México

jfserranotal@gmail.com¹, caviles@correo.azc.uam.mx², hsossa@cic.ipn.mx³, jvillegas@gmail.com⁴, olague@cicese.mx⁵

Abstract

In this work a new method to retrieve images with similar lighting conditions is presented. It is based on automatic clustering and automatic indexing. Our proposal belongs to Content Based Image Retrieval (CBIR) category. The goal is to retrieve from a database, images (by their content) with similar lighting conditions. When we look at images taken from outdoor scenes, much of the information perceived depends on the lighting conditions. The proposal combines fixed and random extracted points for feature extraction. The describing features are the mean, the standard deviation and the homogeneity (from the co-occurrence matrix) of a sub-image extracted from the three color channels: (H, S, I). A K-MEANS algorithm and a 1-NN classifier are used to build an indexed database of 300 images in order to retrieve images with similar lighting conditions applied on sky regions such as: sunny, partially cloudy and completely cloudy. One of the advantages of the proposal is that we do not need to manually label the images for their retrieval. The performance of our framework is demonstrated through several experimental results, including the improved rates for images retrieval with similar lighting conditions. A comparison with another similar work is also presented.

1. Introduction

Most of the objects that can be found in the Internet are images. With this huge distributed and heterogeneous information, people want to search and make use of images over there contained. A great challenge emerges: finding out accurate ways of searching images. Basically, images can be retrieved in two ways, first, text based and second, content-based or query by example based. Text-based retrieval approaches are very well-known and widely used. In this case users are provided with a text area to enter the key words (usually the image file name) on the basis of which image searching is done. It is widely used in Google web based image searching technique. This presents a main drawback: The images in the database are manually annotated using key words. This is known to be a very a time consuming process for almost any database [1], [2]. Also retrieval depends on the human perception based text annotation.

To avoid the above mentioned problems, a second approach, Content-Based Image Retrieval (CBIR) has been proposed by researchers. The term CBIR seems to have originated in the earlier 90’s ([1], [4], [8], [10], and [12]).

CBIR includes research on: Automatic Feature Extraction ([2], [3]), Automatic Feature Extraction with a Semantic Content ([4], [8]) and data representation ([7], [9]). CBIR techniques use features such as texture, color and shape to represent images and retrieves images relevant to the query image from the image database. Among those image features, texture features has been shown very effective and subjective [8].

The rest of the paper is as follows. In section 2 we describe in detail each of the stages of the proposal. In section 3 we show the experimental results obtained.
until now and a comparison with other reported techniques. Finally, in section 5, we conclude and present directions for further research.

2. Methodology

In this section we describe each of the stages (training and testing) of the proposed methodology for the image retrieval with similar lighting conditions applied on sky regions into a database. See Fig. 1.

Training stage: This stage is divided into two main phases as shown in Fig. 1. During the first phase (path A) a set of 300 images with similar lighting conditions (sunny, partially cloudy and completely cloudy) in RGB format (768 x 1024) is first read. Each image is converted to HSI format. From each image, 300 pixels are uniformly selected at random (see Fig. 3(a)). Taking each of the 300 points as the center we open a squared window of size of 10 x 10 pixels around it.

Figure 3(b) shows 20 of the 300 sub-images. To each of the 300 windows the following features are extracted: the mean, the standard deviation [11] and the homogeneity obtained from the co-occurrence matrix [6]. For each window, this is done for the in the channels: hue (H), saturation (S) and brightness (I). The corresponding describing vector for each window of the image has thus nine components: three for H channel, three for S channel and three for I channel.

We take the resulting 90,000 describing vectors (300 for each of the 300 images) and pass them through a K-MEANS algorithm to automatically divide each of these 90,000 features into six clusters. For the 300 images chosen in this paper for training, Table 1 shows how many vectors fall into cluster one, how many vectors fall into cluster two, and so on until cluster six. This numbers give somehow the probability that a given class belongs to the 300 images. Because we do this in an unsupervised manner the six clusters could not have any meaning for as humans, but for the computer, as we will see, this information is very useful to solve the automatic retrieval of images.

We take the resulting 90,000 describing vectors (300 for each of the 300 images) and pass them through a K-MEANS algorithm to automatically divide each of these 90,000 features into six clusters. For the 300 images chosen in this paper for training, Table 1 shows how many vectors fall into cluster one, how many vectors fall into cluster two, and so on until cluster six. This numbers give somehow the probability that a given class belongs to the 300 images. Because we do this in an unsupervised manner the six clusters could not have any meaning for as humans, but for the computer, as we will see, this information is very useful to solve the automatic retrieval of images.

<table>
<thead>
<tr>
<th>Class number</th>
<th>Number of features per class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14647</td>
</tr>
<tr>
<td>2</td>
<td>16106</td>
</tr>
<tr>
<td>3</td>
<td>7104</td>
</tr>
<tr>
<td>4</td>
<td>19155</td>
</tr>
<tr>
<td>5</td>
<td>11848</td>
</tr>
<tr>
<td>6</td>
<td>21140</td>
</tr>
</tbody>
</table>

Table 1. Distribution of the 90,000 features into the 6 chosen clusters.
During the second phase (Fig. 1, Path B), for the same set of 300 images an automatic partition is performed as shown in Fig. 4(a). As shown in this figure each image is divided into $10 \times 10$ regions of $102 \times 76$ pixels per region. For each of these 100 sub-images we take a window of $10 \times 10$ pixels as shown in Fig. 4(b). To each of these 100 windows, we obtain the same describing features: mean, standard deviation and the homogeneity in the three same channels. Each window is described again in the form of vector of nine components. As a result we have 30,000 vectors (100 for each of the 300 images).

To create the indexed database of the 300 images we proceed as follows. We take the 90,000 (300 regions per image, and 300 images) describing vectors obtained in the first training phase (Fig. 1, Path A) and the 30,000 describing vectors obtained in the second phase (Fig. 1, Path B) of training and present them to a $L$-NN classifier.

![Figure 4](image-url)

Figure 4. (a) An image is uniformly divided into 100 sub-images to get 100 describing features. (b) For each sub-image, a window of $10 \times 10$ pixels is selected to compute the corresponding describing vector.

After processing all this data, we obtain an indexed database containing the following information as shown in Table 2.

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>Name of Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>16</td>
<td>23</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>Image 1.jpg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>19</td>
<td>9</td>
<td>9</td>
<td>15</td>
<td>20</td>
<td>Image k.jpg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>23</td>
<td>7</td>
<td>32</td>
<td>19</td>
<td>12</td>
<td>Image 300.jpg</td>
</tr>
</tbody>
</table>

Table 2. Form of the indexed database.

**Retrieval stage:** This stage is divided into the phases shown in Fig. 2. As shown, a query image is presented to the system. To this image the same feature extraction phases used during training are applied. As a result we get 100 describing vectors, these 100 vectors are presented to already trained $L$-NN classifier (reference database is 90,000 random vectors). As a result we get just one vector. This vector contains the probability that each of the six classes (C1, C2, C3, C4, C5 and C6) is contained in the query image. This vector is compared with the 300 vectors saved in the indexed database. To reduce the computing time and to get better retrieval results, we just take into account the two higher components of the six classes. As a comparison distance we use the Euclidean distance.

**Note.** For testing our proposal we have chosen 300 images from a database of Sacre Coeur’s Cathedral (Paris) ($768 \times 1024$). These images were provided by Mauricio Diaz [13].

### 3. Experimental Results

In this section we present the experimental results that validate our proposal.

For this we have 50 images selected from Corel Image Database provided by Julia Vogel [4], and 100 images from the database of [13]. These images are different from those used for training. We presented each of these 150 images to the system and asked it to show the most 10 similar images from the indexed database. Fig. 5 shows a query example. From Fig. 5 we can see for example that the system retrieves correctly 10 images. This gives a 100% of efficiency for this retrieval. To test the efficiency of the proposal we have used the following two measures, $P=$Precision and $R=$Recall:

$$P = \frac{\text{No. of relevant images retrieved}}{\text{Total no. of images retrieved}} \times 100\%$$  \hspace{1cm} (1)  

$$R = \frac{\text{No. of relevant images retrieved}}{\text{Total no. of relevant images in database}} \times 100\%$$  \hspace{1cm} (2)

The first measure represents the number of relevant images retrieved with respect to the total number of images asked to be retrieved for all the image types (or classes). The second measure represents the relevant images retrieved with respect to the total number of images used for training for a given class. Fig. 7 shows that we get 88.14% of efficiency while in [13], the authors get 64.17% of efficiency when using as a query sunny image, (ss) is our chart, (si) is their chart in [13]. Fig. 6 shows that we get 63.75% of efficiency while in [13], authors get a 84.97% of efficiency when using as
a query partially cloudy image, (pcs) is our chart, (pci) is their chart in [13]. Fig. 7 shows that we get 83.24% of efficiency while in [13] a 72.85% of efficiency is obtained when using as a query completely cloudy image, (ccs) is our chart, (cci) is their chart in [13].

Note: Measurements in [13] are only available in an interval of 1 to 4 images for achieve image retrieval, while in our proposal is performed with 12 measurement in an interval of 2 to 100 images for achieve image retrieval.

Figure 6. Performance of our proposal against the method described in [13].

4. Conclusions

In this paper we have described a methodology that allows to automatically retrieving similar lighting condition images from a database. During learning the proposal takes as input a set of images with any lighting conditions applied on sky regions such as: sunny, partially cloudy and completely cloudy. It extracts from them describing features from sets of points randomly and automatically selected. A K-MEANS classifier is used to form six different clusters from the describing features obtained from the randomly and automatically chosen points. A 1-NN classifier is used to build an indexed database from the combination of all the describing vectors. During retrieval the already trained 1-NN classifier is used to retrieve from the indexed database the most similar images given a query image. The experimental results show that our proposal performs better than the reported method in [13]. For this we have used the precision/recall measure. Nowadays we are testing our proposal with much more images and more types of image classes and more cluster regions. Also we are trying to use interest point detectors to select the points from which the describing vectors are going to be computed. We are also going to test with other describing features and other classifiers. One of the advantages of the proposal is that we do not need to manually label the images for their retrieval.

Acknowledgements. J. F. Serrano thanks CONACYT by the scholarship received to complete his doctoral studies. H. Sossa thanks the SIP-IPN under grants 20091421 and 20100468 for the support. Authors thank the European Union, the European Commission and CONACYT for the economical support. This paper has been prepared by economical support of the European Commission under grant FONCICYT 93829. The content of this paper is an exclusive responsibility of the CIC-IPN and it cannot be considered that it reflects the position of the European Union.

References