IMAGE SEARCHING BASED ON PRINCIPAL COMPONENTS ANALYSIS AND INVARINANT MOMENTS

AVILÉS-CRUZ Carlos., GARCÍA AMAYA Antonio., ARÉCHIGA MARTÍNEZ Rene.

Universidad Autónoma Metropolitana-Azcapotzalco, Departamento de Electrónica, Av. San Pablo 180, Col Reynosa, México D. F., 02200 México, Tel: 53189549 Fax: 53946843

Abstract  In this work, an alternative image searching method is proposed. The method is based on principal components analysis and invariant moments (invariant to scale, rotation and translation). It was tested with hundreds of normal images: cell phones, faces, trees, girls, cats, dogs, etc. Principal components analysis is used to characterize an image, and the invariant moments technique is used to extract image features by window size estimation. From an incoming image, a set a features are estimated in order to compare it with a database of images. By using the Euclidean distance, and a Boolean exclusive or (XOR) operation, we obtain a percentage value of likeness. We found that the combination of principal components analysis and invariant moments give excellent results in image identification tasks. An exhaustive study was performed with 500 images. Results include tests over different image sizes and orientation. The robustness of the algorithm is also examined in terms of Gaussian noise.

Key words: Image comparison, Principal Components Analysis, Invariant Moments, and Pattern Recognition.

1 Introduction
At the present time, the number of internet users has grown significantly and with it, the need to search for information in a more efficient and precise way. So it is the case of images such as: faces, circuits, planes, maps, animals, buildings, cities, etc. The necessity to count with a specialized image searcher has increased in applications such as the military, biomedicine, commerce, and education, among others. Up to now, a generic image matching algorithm does not exist for all kind of images.

To have an image searcher is particularly interesting if, for example, we could find a hotel within a map, an interesting site, a museum, etc. (taking as reference the icon that represents each object). The work presented by Wang [16] considers matching an image via "a semantics-sensitive" strategy, finding an indexing by its attribute-properties, solving problem as a searching in database properties; the algorithm was tested on a digital bookstore application. Another approach, by Cordelia [17] does image search based on invariant attributes (in gray level). The algorithm uses invariant moments [1][2][11] to search for an image equal to another one, taking into account that invariant moments are not sensitive to scale, rotation or translation. One of the disadvantages is that it only works for images in gray levels and does the search for some type of structured image.
In the present work, an alternative method for image searching is proposed; the method uses a supervised technique consisting of learning and a testing phases. The comparison scheme is based on correlation techniques[1][2][3] and invariant moments[2].

For the comparison between two images, the proposed method was tested with hundreds of normal images: cell phones, faces, trees, girls, cats, dogs, etc. Principal components analysis is used to characterize an image, and the invariant moments technique is used to extract image features by window size estimation. From an incoming image, a set of features are estimated in order to compare it with a database of images. By using the Euclidean distance, and a Boolean exclusive or operation, we obtain a percentage value of likeness. We found that the combination of principal components analysis and invariant moments give excellent results in image identification tasks. An exhaustive study was performed with 500 images.

The images are in JPG[13] format, which is the common format of images in internet. The structure of the paper is organized as follows. Section 2 describes the image comparison method; Section 3 presents the experimental results; and the conclusions and future work are included in Section 4.

2 Image comparison method

Figure 1 shows the general scheme used to have images ready for comparison. It starts with a color image in jpg format, converts it to a gray image and produces a binary image. Each module is described following the steps shown in Figure 1.

![Fig. 1. General scheme for image comparison](image_url)
• (A) INPUT IMAGE
The "Input image" module just reads from a file. Reading includes: opening, reading and closing the file. Images are in jpg format, therefore, there are 24 bits per pixel (8 bits for red, 8 bits for green, and 8 bits for blue). The image can be of any size.

• (B) IMAGE SIZE NORMALIZATION
Images to be identified or images available in the reference database, can be of any size (NxM). However, in our comparison algorithm all images must be the same size. They are scaled to 160x160 pixels; since the size of the scale could be any, we chose this value arbitrarily. In the results section the effect of other sizes is analyzed. The image size conformation is made via sampling or interpolation of the image.

• (C) GRAY CONVERSION
An image in color (format jpg), is converted to a gray levels image, and quantized to 8 bits, using the following equation:

$$f(x,y) = 0.6f_R(x,y) + 0.3f_G(x,y) + 0.1f_B(x,y)$$

Where $f_R(x,y)$, $f_G(x,y)$, $f_B(x,y)$ are the red, green and blue components respectively.

• (D) NOISE REDUCTION
In this block, the convolution is performed between the image and a Gaussian operator. The Gaussian operator has fixed radius and sigma parameters, and the radius must be greater than sigma. The noise reduction block minimizes the effects produced by the image size normalization module (B).

• (E) IMAGE RANGE NORMALIZATION
Since the Gaussian operator introduces pixel values outside the chosen range, the image range normalization block brings the image within range [0-255] using equation (1).

$$\text{outputpixel} = \frac{255 - \text{inputpixel}}{\text{max}-\text{min}} \times \text{inputpixel} + b$$

(1)

Where max, min are the maximum and minimum values of the image, and $b$ is an offset.

• (F) IMAGE EQUALIZATION
Image equalization is applied in order to obtain, as much as possible, independence of illumination conditions (dark or bright). From the image equalization a uniform histogram is obtained (gray level values appear in the range from 0 to 255).

• (G) THRESHOLDING
The objective of this module is to transform a gray level image into a binary one. The main task of thresholding is to determine if the gray value becomes zero or one. The thresholding value is determined through histogram analysis (defining object range and background range). The equalization stage is fundamental, because if produces a uniform histogram; and the threshold value can be, for example, half of the interval (between 0 and 255 gray levels). Thus gray level values below the threshold will be converted to zero, while those equal to or greater than the threshold value will be converted to one.

• (H) PRINCIPAL COMPONENTS ANALYSIS (PCA)
Principal components analysis is used to characterize each image. The PCA principle can be described as follows:
First we have to estimate the variance/covariance matrix $C_x$, with equation (2).
$$C_{c} = E[(f(x,y) - \bar{X})(f(x,y) - \bar{X})^T]$$  \hspace{1cm} (2)$$

where: \(f(x,y)\) is the window estimation and \(\bar{X}\) is the mean value of \(f(x,y)\).

The new maximal variance and decorrelated values are provided with the Karhunen-Loeve Transform using equation (3)

$$\sum = \sum \cdot \bar{Y} = \sum$$  \hspace{1cm} (3)$$

where: \(\sum = \sum \cdot \bar{Y} = \sum\) is the diagonal eigenvalues matrix and \(A\) is the eigenvector matrix.

(I) FEATURE EXTRACTION (INARIANT MOMENTS)

Invariant Moments are calculated for each image \(f(x,y)\), according to the following definitions.

The definition of \((p+q)\) order invariant moment around the origin is given by:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) \, dx \, dy$$  \hspace{1cm} (4)$$

For an image, equation (4) could be expressed as (5), being \((p+q)\) the order of the central invariant moments:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x-x)^p \, (y-y)^q \, f(x,y) \, dx \, dy$$  \hspace{1cm} (5)$$

where \((x, y)\) are the pixel coordinates, and \(\bar{x}, \bar{y}\) are the average values. The third order central moments are eight non redundant \((p+q)\) combinations \((\mu_{00}, \mu_{20}, \mu_{02}, \mu_{11}, \mu_{30}, \mu_{21}, \mu_{12}, \mu_{31})\). The normalized central moments, \(\eta_{pq}\) are given by:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}}$$ \hspace{1cm} where \(\gamma = \frac{p+q}{2} + 1\)  \hspace{1cm} (6)$$

for \(p + q = 2, 3, \ldots\)

From the second and third moments we get a set of seven invariant moments [2]:

\[
\begin{align*}
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \cdot \eta_{03} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \cdot \eta_{03} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \cdot \eta_{03} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \cdot \eta_{03} \\
\end{align*}
\]

According to [2] this set of moments are invariant to translation, rotation and scaling; thus computing (7) for an image under transformation ensures that its moments will not change significantly (we will see in the results chapter) both from the nature of the digital images and from the images resulting after the transformation process.

(J) IMAGE COMPARISON

a) By Correlation:

Once an image is pre-processed (from block A to block H), the comparison makes use of an exclusive or logic operator (XOR), between images coming from the database and the incoming image. The number of nonzero bits are counted, and the percentage of likeness between the two images is determined by equation (8)

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}}$$ \hspace{1cm} where \(\gamma = \frac{p+q}{2} + 1\)  \hspace{1cm} (6)$$

for \(p + q = 2, 3, \ldots\)

From the second and third moments we get a set of seven invariant moments [2]:

\[
\begin{align*}
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \cdot \eta_{03} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \cdot \eta_{03} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \cdot \eta_{03} \\
\phi &= \eta_{00} + \eta_{11} + \eta_{20} + \eta_{02} + \eta_{30} + \eta_{21} + \eta_{12} \cdot \eta_{31} \cdot \eta_{03} \\
\end{align*}
\]

According to [2] this set of moments are invariant to translation, rotation and scaling; thus computing (7) for an image under transformation ensures that its moments will not change significantly (we will see in the results chapter) both from the nature of the digital images and from the images resulting after the transformation process.

b) By KNN classification:

Where 25600 is determined by the image size (NxM=160x160=25600 in our case).
The K-Nearest Neighbor (KNN) classifier is used in this task. The K value is set to one (for simplicity). We used a modified KNN classifier, to tolerate 5% deviation from mean values (moment values were calculated from reference images). The tolerance percentage value is explained in the results section. The modified KNN classifier works with 7 invariant moments for each image.

Noise Sensitivity
We also evaluated the impact of noise on the images. Random noise could arise from transmission errors. Noise power is given by:

\[ P_N = \frac{\text{Image Power}}{10^{\text{SNR/10}}} \]  \hspace{1cm} (9)

where \( P_N \) is the noise power and SNR is the signal-to-noise-ratio. Once the signal to noise ratio was defined (in order to test over different SNR on image simulating level noise coming from the transmission process), the image power was calculated by equation (10)

\[ \text{Image Power} = \frac{1}{N \times M} \sum_{x=0}^{M} \sum_{y=0}^{N} f(x,y)^2 \]  \hspace{1cm} (10)

Where N and M are the width and height dimensions of the image, respectively.

3 Experimental Results
The searching algorithm was tested with hundreds of images, of different sizes and resolutions, taken from internet. Our database is composed of 500 images: Yugi-oh, cell phones, backpacks, cars, motorcycles, airplanes, watches, cartoons, flowers, houses, etc. Figure 3 shows some examples of images to be identified.

For each image, the image comparison method is applied: from “Input image” to “Principal Components Analysis” blocks (A to H, section 2). The experimental results include robustness analysis to image size as well as noise (Gaussian) sensitivity.
Fig. 3 Some examples of images to be identified: (a) Yugi-oh, (b) cell phone 1, (c) cell phone 2, (d) backpack, (e) car, (f) motorcycle, (g) airplane 1, (h) watch, (i) airplane 2, (j) cartoon, (k) flower, and (l) house
In order to give some examples of how the image comparison works. The first test was developed with an electronic circuit (see Fig. 4). The comparison gives 100% recognition by correlation; and identifies only one image by invariant moments (the same image).

The second example is performed with a car image (see figure 5.(a)). The comparison results produces 5 images: original image at 100% (the same), front view 1 at 89.44%, front view 2 at 87.76%, back view 1 at 91.81%, and finally, back view 2 at 92.15%. (See figure 5). The recognition threshold was fixed at 85%.

(a) Original image to be searched

(b) 100%

(c) 89.441406%

(d) 87.761719
If now we are interested in a cell phone search, we enter into the system the cell phone image we are looking for in the database; and as a result, the system produces different phone cells with their percentages of likeness, see figure 6.
A threshold is fixed to the system in order to produce as results, only the images that have likeness higher than the threshold. In our test, the threshold was fixed at 80 percent. If the percentage is less than 80%, the system produces different images from the one under test (e.g., cars instead of cell phones).

**Robustness to image size**

The image size used for the sampling step was 160x160 pixels. But, what happens if we use different image sizes?, i.e., smaller or greater than 160x160 pixels. For example, taking the car as the test image with sampling scale under or above 160x160 pixels, the system produces the results shown in Table 1.

<table>
<thead>
<tr>
<th>Size (pixels)</th>
<th>% Likeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1200 x 925</td>
<td>100</td>
</tr>
<tr>
<td>960 x 740</td>
<td>99.61</td>
</tr>
<tr>
<td>972 x 750</td>
<td>99.46</td>
</tr>
<tr>
<td>840 x 648</td>
<td>99.27</td>
</tr>
<tr>
<td>720 x 555</td>
<td>99.17</td>
</tr>
<tr>
<td>600 x 463</td>
<td>99.05</td>
</tr>
<tr>
<td>480 x 370</td>
<td>98.80</td>
</tr>
<tr>
<td>360 x 278</td>
<td>98.41</td>
</tr>
<tr>
<td>180 x 139</td>
<td>96.57</td>
</tr>
<tr>
<td>90 x 70</td>
<td>92.90</td>
</tr>
<tr>
<td>45 x 35</td>
<td>88.68</td>
</tr>
</tbody>
</table>

Table 1. Results from the System for the car search as a function of image size.

As a sample of different image size recognition, Figure 7 shows the image car sampled at different resolutions. 7(a) Original image at 1200x925 pixels, 7(b) 360x278 pixels, 7(c) 180x139 pixels, 7(d) 90x70 pixels and 7(e) 45x35 pixels. As we can see the likeness percentage decreases from 100% until 88.68%; however, the likeness percentage is still high (88.68%) for the smaller image (45x35 pixels).
From these results it can be concluded that the greater the image size, the less information is lost, and greater likeness percentage is obtained.

**Noise Sensitivity Results**

The previous results were obtained without noise. Nevertheless, it is important to take into account noise effects, because images are usually contaminated. The noise tests were made taking noise levels of 0.01 to 0.22. Figure 8 and Table 2 show results for different noise levels for the car image (percentage threshold fixed at 85%) for both Principal Components Analysis (PCA), and Invariant Moments.
When we make use of the Internet, very likely we are looking for information we are interested in. In order to do the search, different software engines exist, but they normally work in text, not graphics mode. Thus the idea arose to develop this project: "Seeking of images in Internet". Initially, the idea was to be able to compare any type of image. However, when reviewing the methods used for digital image processing, we had to agree that the problem was not so easy to solve. The images can range from simple to very complex; therefore, at first the project was limited only to comparison of images of electronic symbols in black and white. But the form in which the comparator of images was developed allowed not only the use of images in black and white, but also images in color. The format was not an impediment either, since with the conversion to the "raw" format, images in other formats like GIF, which is used a lot in Internet, can be searched for. In addition to being able to work with the originally proposed images of electronic symbols (bipolar transistors, JFETs, capacitors, amplifiers, and resistors), it became possible to compare images of planes, diagrams and electronic configurations (as it were possible to see from the results) expanding naturally the original project objective.

With the program prototype for the search of images in Internet based on PCA and Invariant Moments, we have obtained satisfactory results with the proposed method. The images found with PCA fulfill the results waited for as far as invariance to rotation is concerned, i.e., images that have

<table>
<thead>
<tr>
<th>Noise level</th>
<th>PCA Found images</th>
<th>Invariant Moments Found images</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>0.01</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>0.02</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>0.03</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>0.04</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>0.05</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>0.06</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>0.07</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>0.08</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>0.09</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>0.10</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>0.12</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0.14</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0.16</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0.18</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.20</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.22</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.24</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.28</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.30</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2, Results from the system for the car search, as a function of noise. Number of images found, with PCA and with Invariant Moments

4 Conclusions and future work

When we make use of the Internet, very likely we are looking for information we are interested in. In order to do the search, different software engines exist, but they normally work in text, not graphics mode. Thus the idea arose to develop this project: "Seeking of images in Internet". Initially, the idea was to be able to compare any type of image. However, when reviewing the methods used for digital image processing, we had to agree that the problem was not so easy to solve. The images can range from simple to very complex; therefore, at first the project was limited only to comparison of images of electronic symbols in black and white. But the form in which the comparator of images was developed allowed not only the use of images in black and white, but also images in color. The format was not an impediment either, since with the conversion to the "raw" format, images in other formats like GIF, which is used a lot in Internet, can be searched for. In addition to being able to work with the originally proposed images of electronic symbols (bipolar transistors, JFETs, capacitors, amplifiers, and resistors), it became possible to compare images of planes, diagrams and electronic configurations (as it were possible to see from the results) expanding naturally the original project objective.

With the program prototype for the search of images in Internet based on PCA and Invariant Moments, we have obtained satisfactory results with the proposed method. The images found with PCA fulfill the results waited for as far as invariance to rotation is concerned, i.e., images that have
similar forms, rotated to any angle, can be found. The image or images found by the calculation of the Invariant moments of HU[2], is identical or very close. It sets the standard to make searches very precise, that is to say, where not only images with similar forms, but also with similar statistical properties can be found.

When noise is added to the image, we found that with up to 0.06 % of noise added to the image, the results are good for the PCA; the calculation of Invariant moments is more sensitive to noise and with up to 0.03 % of noise it is still possible to find the original image. When searching an image of different sizes, the PCA algorithm is able to produce similar images even with a downsizing of 10: 1. If the image is expanded, the algorithm will recognize it without problem, i.e., with an expansion of 1: 10 it will recognize the original image with likeness of 96 %. The required computational effort in the calculation of the PCA and Invariant Moments is considerable, so an improvement would be to include the invariant moments for each image in the database.

The development of this project can continue with future work in which any type of image can be introduced, or with the development of a system that allows to find some element of interest contained in an image. Then, this project does not end here, and it is hoped that it will be retaken by those interested in digital image processing.

References


[7] Brian W. Kernighan, El entorno de programación Unix, Prentice may


