ABSTRACT

A method for two-dimensional shape recognition is described in this paper as is an extension to 3D. First the object is scanned using laser with the help of a hardware system designed inside. Then, the object’s center of mass is calculated, with this calculation the method is able to obtain a unique signature or model of the object. This signature is compared with the existent signatures, using a correlation coefficient, a knowledge base and an inference process, in order to decide the object’s shape. Actually the method has been implemented as a Windows application that simulates it. It is also able to distinguish between objects with an almost exact signature, for example a star compared with a pentagon, furthermore, the object’s position does not matter, neither do the discontinuities, nor the rotation grade that it has. This method provides a simple way to object recognition using faster and easier algorithms, furthermore it does not require expensive equipment, like video cameras, it is very flexible and robust, since it is not necessary to have exactly the same objects in the knowledge base. And finally, it can be trained for storing more objects in its knowledge base.

KEY WORDS: Applications, Computer vision, Intelligent information systems, Expert systems.

1. INTRODUCTION

Shape recognition is usually related with computational vision systems using cameras or other devices for obtaining images, and the use of algorithms for digital image processing (filters, escalation), segmentation and recognition [1, 7-13].

The proposed method is built from the vision system (hardware) to the recognition software.

The hardware system consists of a plain rectangular area (fig. 1,b), where the object (fig. 1,c) will be placed, and using laser (fig. 1,a) and its respective sensors (fig. 1,e) the system should be able to obtain all the information needed for dispatching to the computer, where the recognition and the robot arm’s control (fig. 1,d) software is stored. Using the robot’s arm the object is placed inside the area, without really being careful of the location or position where the object is placed. Once inside, the laser sweeps (scans) from one side of the area, in other words, it would be obtaining its “shadow”. This shadow is stored, then the robot’s arm rotates the object a certain quantity of degrees, depending of the required resolution, so the system obtains another shadow, and so on, until the object rotates completely and gets its pattern/signature (fig. 2). This signature is analyzed to determine what kind of object it would be, a simple method for this could be the maximum and minimum counting.

The main problem is that this method generates the same signature for objects like: a pentagon and a star, fig. 2, a) and fig 2, b), accordingly. However, it is not possible to distinguish objects that have closed angles, assuming the
maximum and minimum method, it would identify five maximum/minimum for both objects. That is why it was changed so as to make the laser sweeping from the top to obtain a picture of the area with the shadow of the object projected on it; in which way the signature is calculated from inside the object, this procedure is detailed in the next part of this theme.

Now, while making an analysis of the methods and algorithms used for object recognition it was found that, the majority of the actual methods obtain the images using cameras, and the use of these imply:

- Good resolution.
- Focus.
- Camera-Computer interface.
- Adequate illumination.
- Camera fixing.

The cost of this is too high. Once the image is obtained it takes a lot of computing time to extract the objects using processing algorithms and image segmentation [1, 7-13]; moreover, the object identification is made using conventional software (like data bases or implicitly inside the source code) [2,3], and in some cases the software includes artificial intelligence techniques like neural networks [4], that imply a high computational processing and training time. The closest to the method proposed in this paper is illustrated by Guillermo Sampallo [2], this, in addition to the image processing is necessary to guide the object to a main axis using its inertia momentums, because if they are not used, the result is mistakenly obtaining its signature (fig. 2), aside from additional calculations like cubic interpolations, makes the method more robust, and increases the computational cost.

There are few methods for object recognition that do not use cameras, as is described by J.R. Llata [5], who evaluates and compares the application of different kinds of expert systems to recognize prismatic objects and faces, those systems use ultrasonic sensors to get a representation of the objects, and probabilistic distribution functions as a parameter to recognize them. However, this method is used to identify 3D-objects using ultrasonic equipment (very expensive) and statistical information (a lot of processing).

The purpose of this work is for the development of a cheaper method (computational and financial), which involves its own hardware for object acquisition, and easier and faster recognition algorithms.

This paper’s contribution, is the construction of a hardware system that does not require a video camera to obtain the object image, and for the object recognition algorithm (in this method), the object’s position, orientation, size and shape does not matter.

### 2. RECOGNITION METHOD

The method consists of four steps:

I. Obtaining the points.
II. Center of mass calculation.
III. Sampling and signature construction.
IV. Object recognition.

Now, while making an analysis of the methods and algorithms used for object recognition it was found that, the majority of the actual methods obtain the images using cameras, and the use of these imply:

- Good resolution.
- Focus.
- Camera-Computer interface.
- Adequate illumination.
- Camera fixing.

The cost of this is too high. Once the image is obtained it takes a lot of computing time to extract the objects using processing algorithms and image segmentation [1, 7-13]; moreover, the object identification is made using conventional software (like data bases or implicitly inside the source code) [2,3], and in some cases the software includes artificial intelligence techniques like neural networks [4], that imply a high computational processing and training time. The closest to the method proposed in this paper is illustrated by Guillermo Sampallo [2], this, in addition to the image processing is necessary to guide the object to a main axis using its inertia momentums, because if they are not used, the result is mistakenly obtaining its signature (fig. 2), aside from additional calculations like cubic interpolations, makes the method more robust, and increases the computational cost.

There are few methods for object recognition that do not use cameras, as is described by J.R. Llata [5], who evaluates and compares the application of different kinds of expert systems to recognize prismatic objects and faces, those systems use ultrasonic sensors to get a representation of the objects, and probabilistic distribution functions as a parameter to recognize them. However, this method is used to identify 3D-objects using ultrasonic equipment (very expensive) and statistical information (a lot of processing).

The purpose of this work is for the development of a cheaper method (computational and financial), which involves its own hardware for object acquisition, and easier and faster recognition algorithms.

This paper’s contribution, is the construction of a hardware system that does not require a video camera to obtain the object image, and for the object recognition algorithm (in this method), the object’s position, orientation, size and shape does not matter.

### 2. RECOGNITION METHOD

The method consists of four steps:

I. Obtaining the points.
II. Center of mass calculation.
III. Sampling and signature construction.
IV. Object recognition.

Now, while making an analysis of the methods and algorithms used for object recognition it was found that, the majority of the actual methods obtain the images using cameras, and the use of these imply:

- Good resolution.
- Focus.
- Camera-Computer interface.
- Adequate illumination.
- Camera fixing.

The cost of this is too high. Once the image is obtained it takes a lot of computing time to extract the objects using processing algorithms and image segmentation [1, 7-13]; moreover, the object identification is made using conventional software (like data bases or implicitly inside the source code) [2,3], and in some cases the software includes artificial intelligence techniques like neural networks [4], that imply a high computational processing and training time. The closest to the method proposed in this paper is illustrated by Guillermo Sampallo [2], this, in addition to the image processing is necessary to guide the object to a main axis using its inertia momentums, because if they are not used, the result is mistakenly obtaining its signature (fig. 2), aside from additional calculations like cubic interpolations, makes the method more robust, and increases the computational cost.

There are few methods for object recognition that do not use cameras, as is described by J.R. Llata [5], who evaluates and compares the application of different kinds of expert systems to recognize prismatic objects and faces, those systems use ultrasonic sensors to get a representation of the objects, and probabilistic distribution functions as a parameter to recognize them. However, this method is used to identify 3D-objects using ultrasonic equipment (very expensive) and statistical information (a lot of processing).

The purpose of this work is for the development of a cheaper method (computational and financial), which involves its own hardware for object acquisition, and easier and faster recognition algorithms.

This paper’s contribution, is the construction of a hardware system that does not require a video camera to obtain the object image, and for the object recognition algorithm (in this method), the object’s position, orientation, size and shape does not matter.

### 2. RECOGNITION METHOD

The method consists of four steps:

I. Obtaining the points.
II. Center of mass calculation.
III. Sampling and signature construction.
IV. Object recognition.

Now, while making an analysis of the methods and algorithms used for object recognition it was found that, the majority of the actual methods obtain the images using cameras, and the use of these imply:

- Good resolution.
- Focus.
- Camera-Computer interface.
- Adequate illumination.
- Camera fixing.

The cost of this is too high. Once the image is obtained it takes a lot of computing time to extract the objects using processing algorithms and image segmentation [1, 7-13]; moreover, the object identification is made using conventional software (like data bases or implicitly inside the source code) [2,3], and in some cases the software includes artificial intelligence techniques like neural networks [4], that imply a high computational processing and training time. The closest to the method proposed in this paper is illustrated by Guillermo Sampallo [2], this, in addition to the image processing is necessary to guide the object to a main axis using its inertia momentums, because if they are not used, the result is mistakenly obtaining its signature (fig. 2), aside from additional calculations like cubic interpolations, makes the method more robust, and increases the computational cost.

There are few methods for object recognition that do not use cameras, as is described by J.R. Llata [5], who evaluates and compares the application of different kinds of expert systems to recognize prismatic objects and faces, those systems use ultrasonic sensors to get a representation of the objects, and probabilistic distribution functions as a parameter to recognize them. However, this method is used to identify 3D-objects using ultrasonic equipment (very expensive) and statistical information (a lot of processing).

The purpose of this work is for the development of a cheaper method (computational and financial), which involves its own hardware for object acquisition, and easier and faster recognition algorithms.

This paper’s contribution, is the construction of a hardware system that does not require a video camera to obtain the object image, and for the object recognition algorithm (in this method), the object’s position, orientation, size and shape does not matter.

### 2. RECOGNITION METHOD

The method consists of four steps:

I. Obtaining the points.
II. Center of mass calculation.
III. Sampling and signature construction.
IV. Object recognition.
The sensing precision depends on the amount of sensors conforming the line (fig. 4,d), or a sensor can be placed moving in an axis perpendicular to the laser, using a stepper motor and the precision is determined by the amount of steps of movement and the laser's angle.

The information received by the sensors is transformed into a bitmap (BMP), like the one shown (fig. 5). The method for object recovery is very simple, the BMP is manipulated obtaining the pixels that are different to the background color, which represent the object and then are assigned coordinates according to a cartesian axis (fig. 5). This step's result is a list of coordinates (x,y) with the points of the object.

II. Center of mass calculation.

From the points obtained in the previous step it is possible to calculate the object's center of mass, using the formulas:

\[
M_y = \Sigma m_i x_i \quad M_x = \Sigma m_i y_i \\
X = M_y / m_{total} \quad Y = M_x / m_{total}
\]

Where \( m_i \) is the mass of each point, in this case is equal to the unit (1), because a point is the smallest unit in our space; \((x_i, y_i)\) correspond to the coordinates \((x, y)\) of each point, \( m_{total} \) is the sum of all the points' masses. The pair \((X,Y)\) is the center of mass. Finally, is necessary to translate this point \((X,Y)\) to the origin of the cartesian map by using the translation matrix \([6]\):

\[
\begin{pmatrix}
1 & 0 & -T_x \\
0 & 1 & -T_y \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
X \\
Y \\
1
\end{pmatrix}
\]

as a result \((X=0, Y=0)\). The same operation is applied to all the points to obtain somewhat as shown in figure 6.

III. Sampling and signature construction

With the center of mass as a reference, the method searches the same y coordinate into the points' list of the object, as a result of this operation two points are obtained, the first one chosen having the major x coordinate, thus is obtained the point from the start to the center's right (zero degrees), as is shown in figure 7.

Once the starting point is calculated, a cycle is made to go over the 360 degrees around the center of the mass. As we have the cartesian coordinates of each point that conforms the object's perimeter, its polar coordinates are obtained and the cartesian map is discarded; the new coordinates are calculated using the following formulas:

\[ P(x,y) \rightarrow P(r,\theta). \]
\[ r^2 = x^2 + y^2 \quad \text{sen} \ \theta = y / r. \]

As a result of this step, a vector is built (the signature) with all the radius (r) normalized within a range of [0 1] and the angles of all the points arranged from minor to major according to its angle, as shown in figure 8.

![Figure 8](image)

**IV. Object recognition**

Now having the object’s signature, it is necessary to compare it with all the objects that are in the knowledge base in order to determine whether the object has been registered or is totally new for the system. For this purpose the correlation coefficient (\( \rho \)) is used, statistical calculation, a knowledge base and an inference process. Each procedure is detailed below:

**IV.i) Correlation Coefficient (\( \rho \)).** This coefficient determines the equal grade of two samples; in this case, the samples are the radius vector of the knowable object (r1) and the object’s vector to recognize (r2). This equality is quantified in a range [0 1], being 1 when the objects are identical; and if it tends to 0 it is because the likeness is low. In order to make the calculus the vectors must have the same size (N). Next, the procedure to follow for the calculation of \( \rho \) is shown:

\[
S_1 = \sum_{i=1}^{N} r_{1i} \quad S_2 = \sum_{i=1}^{N} r_{2i} \\
S_{12} = \sum_{i=1}^{N} r_{1i} \cdot r_{2i} \\
S_{c1} = \sum_{i=1}^{N} (r_{1i})^2 \quad S_{c2} = \sum_{i=1}^{N} (r_{2i})^2 \\
\sigma_1 = S_{c1} - (S_1^2 / N) \quad \sigma_2 = S_{c2} - (S_2^2 / N) \\
\sigma_{12} = S_{12} - (S_1 \cdot S_2 / N) \\
\rho = \frac{\sigma_{12}}{\sqrt{\sigma_1 \cdot \sigma_2}}
\]

In case the vector’s size is not equal, because the size can be different from the object to recognize to the size of the knowable object, it is solved by taking the vector with the minor N and a new vector is built for the vector with the larger size using the radiuses whose angle is the same in both vectors, so they have the same size.

**IV.ii) Knowledge Base.**

The Knowledge Base uses files containing the knowable objects. There are two kind of files:

a) For each knowable object, the object’s name and the file’s name is taken from the user, and it is a text file (TXT). These files store the information obtained in the Sampling and signature construction step:

- The quantity object’s points, this quantity is obtained from the polar coordinates.
- The object’s polar coordinates, that is formed by the pair (angle, radius).

b) A general file called objects.txt, that contain a list of objects in the knowledge base. Before creating the file for each knowable object, the object’s name is placed in this file

**IV.iii) Inference Process.**

The inference algorithm is based on the correlation coefficient of each object, for all the knowable objects (into the knowledge base) it is necessary to calculate the correlation coefficient (IV.i), and the value of \( \rho \) is obtained, and those coefficients are sorted in descendent order. The quantity of knowable objects can be as large as desired. Following are the rules of the inference algorithm:

If an object with a correlation coefficient over zero exists, then:

the object’s name is taken as the most similar one.

If two or more objects with a correlation coefficient over zero exist, then:

the difference is obtained between the highest correlation coefficient with the correlation coefficient of each object, the objects with less difference or the same as 0.2, are taken as the most similar ones.

If the previous rules do not apply, then:

The user is notified that knowable objects do not exist in the knowledge base, and it is asked to add it in the knowledge base with the name given by the user.
As a result of the inference process, the knowable object’s name is obtained and its similar percentage ($\rho$), or it is added into the knowledge base.

### 2.1 ANALYSIS

The running time of this algorithm (method) depends of the quantity of points in the step I (Obtaining the points), assume N, also depends of the knowledge base size (M). In the worst case when N and M have the same size, the running time is $O(n^2)$. In other hand, best case, if the absolute difference between N and M is big then the running time is either $O(c*M)$ or $O(N*c)$, that is $O(n)$ [14].

### 3 GENERALIZATION OF THE METHOD

This method is generalized to recognize tridimensional objects, to do this it is necessary to modify the four steps described above. To obtain the points, the hardware system must be modified, in this form it is possible to obtain the object’s tridimensional representation; for example, a similar technique can be used like the catscan (used in medicine to acquire a representation of the human brain). Also, the center of mass calculation and the sampling process of the tridimensional object are modified to get the signature, that is totally carried out, but its calculus is expensive. The other steps are maintained without modification, but considering tridimensional objects.

### 4 SIMULATION AND PROTOTYPE

To this moment, a prototype has been built, which consist of a Windows® application programmed using Microsoft® Visual C ++. Now each option is explained in the menu bar:

**File.** This menu option contains two choices; the first one is to open a bitmap file (BMP), once opened, the object contents is displayed; and the second one is to exit the application. The objects that this application accepts must be in a bitmap format with 256 color or gray scale. The object’s border must be black (RGB 0,0,0), the background may be white or any other color except black. The recommended size is 120 x 120 pixels, the accepted sizes must have a width divided by 8, for example 8, 16, 32, 40, and so on.

**Signature.** This choice gets the pattern that represents the current object’s signature, making a comparison between the angle and the radius using its polar coordinates, and is displayed on the screen.

**Training.** Opens a dialog box asking the name to be added into the knowledge base. If the given name already exists, the system requests another name.

**Recognition.** This choice identifies the current object, comparing it with objects into the knowledge base. It also, shows a message with the name of the most similar object, and the value of the correlation coefficient ($\rho$, described in the 2, IV.i section). If there are more similar objects their names are shown and $\rho$ too. If there are not any similar objects the system notifies this fact.

**Help.** Shows a dialog box with the system’s information.

### 5 RESULTS

The analysis of results for the described method is obtained using the prototype and generating four known objects that conform the knowledge base. (Table I).

#### Table I

<table>
<thead>
<tr>
<th>Object</th>
<th>Image / Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star</td>
<td><img src="image" alt="Star Image" /></td>
</tr>
<tr>
<td>Pentagon</td>
<td><img src="image" alt="Pentagon Image" /></td>
</tr>
<tr>
<td>Square</td>
<td><img src="image" alt="Square Image" /></td>
</tr>
<tr>
<td>Circle</td>
<td><img src="image" alt="Circle Image" /></td>
</tr>
</tbody>
</table>
To evaluate the robustness and the flexibility of the method, some cases of test are analyzed and are used as a parameter of comparison among other recognition methods, those cases are objects with: different size, discontinuities, different rotation angle, random position, similar closed angles (fig. 2) and asymmetric similarity.

Following are descriptions of the results for each case comparing an object with the knowable objects from Table I:

a) Objects with different size.

Object to recognize:

Result: “Star” with equality grade $\rho = 0.6946$ and “Pentagon” with $\rho = 0.5577$.

b) Objects with discontinuities.

Object to recognize:

Result: “Pentagon” with equality grade $\rho = 0.8877$.

c) Objects with different rotation angle.

Object to recognize:

Result: “Square” with equality grade $\rho = 0.2767$, and “Circle” with $\rho = 0.2159$.

d) Objects with similar closed angles.

In this case the object to recognize can be the star or the pentagon, from table 1; and the result in both cases an equality grade $\rho = 1$ is obtained.

e) Asymmetric similar objects.

Object to recognize:

Result: “Pentagon” with equality grade $\rho = 0.7303$.

6. CONCLUSION

This method has a simple way to recognize objects using easy and fast calculations comparing it with other methods. Furthermore, it is able to recognize more objects than any other classical methods, increasing its knowledge base. It is also flexible and robust, since it is not necessary to have the object to recognize completely equal to the object stored in the knowledge base. The way to get the object’s image does not require using processes like: filtering, umbralization, etc. [1, 7-13]. Another important advantage is that the method is not sensitive to the position of the object, in other words, the object does not require to be in an exact position, nor rotation grade it has.

If the object to be recognized is too small, comparing it with the knowable object in the knowledge base, the values of $\rho$ can be negatives, so the method does not work correctly.

At this point the method has been tested with few objects in its knowledge base (around 10), for this reason it has not been explored on a great amount of objects. So it is open to build a system that reads automatically from a list of objects (object’s shape and name) to feed the knowledge base (for example a quantity of 100), training them, recognizing them and generating test cases from the list of objects so it determines what percentage of the objects it recognizes.

On the other hand, this method is unable to recognize objects that contain other(s) object(s) inside themselves, for example:
Being this problem open to be solved. It is also open to recognize tridimensional objects that contain other(s) object(s) inside themselves. For example, to recognize a sphere inside another sphere.

7. ACKNOWLEDGEMENTS

To the Computation and Informatics Area of the School of Engineering for the support offered in the development of this method. To Mirna García DUI’s professor for her help in the grammar review.

8. REFERENCES.


