# Image Recognition Using Polar Histograms

*Pablo Suau, Ramón Rizo, Mar Pujol*

Department of Computer Science and Artificial Intelligence
University of Alicante, Alicante (Spain)
{pablo,rizo,mar}@dccia.ua.es

## Abstract

This work proposes a new technique for object recognition based on histogram comparison by means of Chi-Square distance and polar coordinates, applied to the particular problem of artificial landmark recognition for mobile robot navigation. This scheme makes our approach less sensitive to some shape alterations, rotation and displacement, and also simpler than other recognition methods. The experiments prove that this technique achieves an high number of right guesses when the landmark database is large enough.

## 1. Introduction

Over the years several techniques have been used in image recognition: Fourier Descriptors, Invariant Momentum, and so on.

A very useful tool for image analysis is the histogram. Usually a histogram is created counting the number of times a particular intensity value appears in the image (this has been generalized several times to introduce colour information [4]). We could apply tresholding to the histogram to achieve a simple image segmentation ([5]). Image quality can often be improved by manipulation of the histogram ([6]). Even these histograms could reveal image features in the image that were previously unobservable.

Another useful tool for image analysis and image recognition are polar coordinates ([2]). They allow to match figures with rotation and displacement variance, even with some little changes in shape.

We have joined these two techniques in an structure called polar histogram that, applied to our particular problem (artificial landmark localization and recognition for mobile robot navigation), has given us a great number of right guesses when the landmark database is large enough to satisfy a mobile robot navigation based on artificial landmarks with a very low computational cost. This paper focus on the polar histogram technique, making a brief introduction to the context of the problem where it has been applied.

This brief introduction to the context where this technique has been applied is made in section 2. Section 3 explains how to build this structure called polar histogram, and section 4 explains how to compare polar histograms to achieve an object recognition using an object database. Section 5 shows some experimental results, and finally section 6 includes some conclusions and information about future work.

## 2. Context of the problem

Although in this paper we are focusing more on the polar histogram recognition process, we must also explain in which situation we are applying this technique. We will summarize in this section the complete context where the polar histograms have been used with success.

Our objective was to localize the nearest landmark inside a digital image containing one or more landmarks, obtained by a camera placed on a mobile robot, and to extract the symbol inside it to recognize it, after a comparison with a set of symbols stored in a database. Some examples of landmark are shown in Figure 1: a landmark is square shaped, with a blue border and a black symbol inside this border. As we can see, the symbols inside the landmarks have been taken from real roadsigns

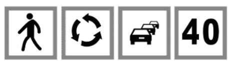


Figure 1: *Landmark examples.*

Our approach for landmark localization and recognition was based on [1], with some modifications. Figure 2 shows the complete process from the moment the image is obtained from the camera on the robot to the moment the symbol inside the landmark is recognized. This process can be summarized with these steps:

- Colour segmentation: a colour quantization is applied to the image, reducing it to eight basic colours ([1]). A binary image is created containing the pixels of the original image with the landmark's border color (one of these eight basic colours).

- Landmark localization: from the binary image corresponding to the landmark borders, we try to localize the nearest one, by means of horizontal

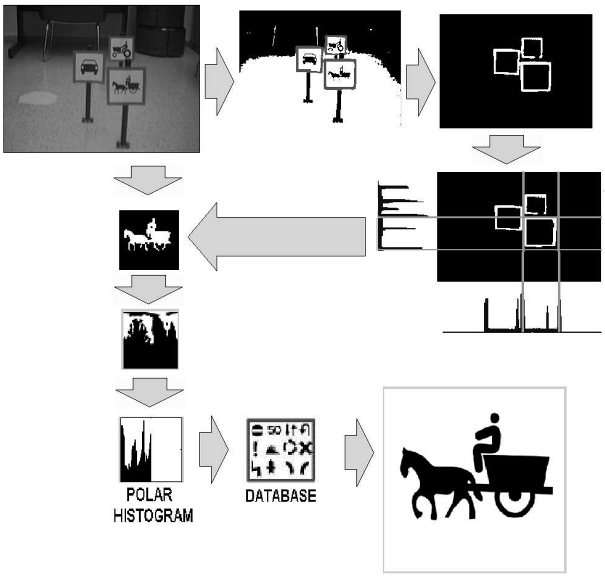


Figure 2: *The complete localization and recognition process.*

and vertical projections. We don't use stereo vision (we have only one camera on top of the robot), so we don't have depth information. Therefore we consider that the nearest landmark is the biggest one, the landmark with the greatest number of pixels. The localization is done following following [1], but instead of using projections as the total sum of blue pixels in each row and column, we use the maximum sum of consecutive blue pixels, avoiding some localization problems.

- Landmark's symbol extraction: once the nearest landmark has been detected, we apply the k-means algorithm only to the part of the original image where this nearest landmark is placed. As a consequence, we create a binary image with the same size than the images stored in the database, containing only the symbol.

- Recognition: a histogram is created from the polar coordinates of the extracted symbol, it is compared with the histograms created from symbols at the database, and the recognized landmark is shown on screen.

The recognition approach we present is done during this last step of the process and will be explained in detail in the next sections. But although we are focusing on recognition, we must mention the k-means algorithm for the landmark symbol segmentation. Once the image region inside the nearest landmark is extracted to a new image, we can observe that the symbol's V value (from HSV colour model) is quite different to the rest of the region's V values. So we can use a simple k-means algorithm to split the points of the symbol image into two clusters, if we select the maximum and the minimum V vales as initial cluster centers. This way we can have a good clustering even if light conditions change.

## 3. Building Polar Histograms

In order to recognize the symbol inside the nearest landmark, we must find some way to compare it with the symbols stored in the database. We need a robust classification method, so we can differentiate the symbols without being affected by little changes in shape, orientation and displacement (the scale variations have been solved before, because the symbol extracted from the image is scaled and stored in a image with the same size than the database ones). Some works have proven that using polar coordinates allows an efficient and low computational cost two dimensional irregular shape comparison, invariant to displacement and rotation (on the plane of the image, no 3D rotations) [2].

After the previous localization steps we have the symbol extracted from the nearest landmark in a binary image with the same size than the images included in the landmark database. This image is represented by means of cartesian coordinates, and it must be transformed into an image with polar coordinates, using the gravitational center of the symbol as the pole and a polar axis which origin is that pole (an example is shown in Figure 3).

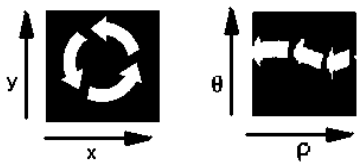


Figure 3: *Example of symbol represented with cartesian coordinates (left) and with polar coordinates (right) using the center of the symbol as pole.*

If the image is stored in an array, we must translate from a cartesian image where the $x$ coordinate increases in each column and the $y$ coordinate increases in each row to a polar image where the distance $\rho$ increases with each column and the angle $\theta$ with each row. Using the equations (1) and (2) we can know which cartesian pair ($x$,$y$) corresponds to each polar pair ($\rho$,$\theta$). This translation can be done in two ways: calculating the polar coordinates for each cartesian pair in the original image, or calculating the cartesian coordinates corresponding to each polar pair in the destination image. The first method is more inefficient, because we can calculate the value of some of the positions in the destination image more than once, and also some gaps can appear. So it seems better to calculate for each position of the destination image the corresponding cartesian coordinates and to assign the pixel value at these coordinates in the original image.

$$x = \rho \cdot cos(\theta) \quad (1)$$

$$y = \rho \cdot sin(\theta) \quad (2)$$

Finally, from the polar image, we can obtain an histogram that represents the original symbol. In the polar image, the distance $\rho$ increases with each column; so, all the pixels in the same column are at the same distance from the symbol's gravitational center in the original image. If we add all the pixels with value 1 in each column in the polar image, we generate an histogram that indicate us for all the distances from the gravitational center of the symbol, how many pixels have value 1 (an example is shown in Figure 4). This histogram is rotation invariant (because we use polar coordinates and the camera is always straight) and displacement invariant (because we use the gravitational center of the figure as polar center). We have also an scale invariant representation after having scaled the symbol to a fixed size in previous steps. We call this structure *polar histogram*.

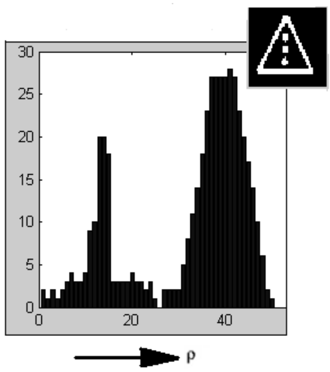


Figure 4: *An example of polar histogram.*

## 4. Comparing Polar Histograms

If we have followed the same process for all the symbols stored in the database, and if we have obtained the polar histogram for each of them, the landmark recognition is as simple as to check which of the histograms of he database symbols is more similar to the histogram of the landmark in the image caught by the robot camera. In order to test this similarity several histogram comparison methods, like Kolmogorov-Smirnov([3]) test or Chi-Square distance, could be used. After some experimentation, Chi-Square distance was used (Kolmogorov-Smirnov was not suitable to our problem).

The Chi-Square distance, applied to two histograms, can give us a weighted average of the difference between all the positions of them, so it could tell us which of the histograms in the database is more similar to the image symbol's histogram. We can calculate this distance $\chi^2$ between two histograms *i* and *j* using (3) and (4).

$$\chi^2_{ij} = \sum_k^n \frac{(H_i(k) - \hat{H}(k))^2}{\hat{H}(k)} \quad (3)$$

$$\hat{H}(k) = \frac{H_i(k) + H_j(k)}{2} \quad (4)$$

Although the Chi-Square distribution is not symmetric, the Chi-Square distance has this property, so it can fit our purposes.

## 5. Experimental results

Finally we show some experimental results. The images caught by the camera on the robot had a size of 320x240 pixels, and the images stored in the landmark database had a size of 96x96 pixels (so, the nearest landmark's symbol extracted from the image would be scaled to a size of 96x96 too).

The symbols stored in the database where obtained from SEÑALECTICA (http://iris.cnice.mecd.es/bancoimagenes/senales/), a vectorial image repository of real roadsigns. A group of test sets with several images caught by the camera on the robot (from 89 to 380 images) were created to estimate the localization and recognition error rates. The first one had 10 landmarks stored in the database, the second one 20 landmarks, the third one 30 landmarks, and so on. The last one had 100 landmarks (Figure 5). All the landmarks from the databases appeared in at least 3 images in the corresponding test set. In these images appeared from 1 to 3 landmarks, at different distances.

Figure 5: *Some of the landmarks used in the experiments.*

The localization error rate in the previous steps was between 1-3%. To calculate the error recognition rates we ignore the images where the localization is not correct. First we could see the effect of changing the size of the polar images from where we calculate the polar histograms. Figure 6 shows the recognition error rate when we have 100 landmarks in the database for different polar image resolutions.

With a low number of polar histogram elements there is not enough information in order to achieve an adequate

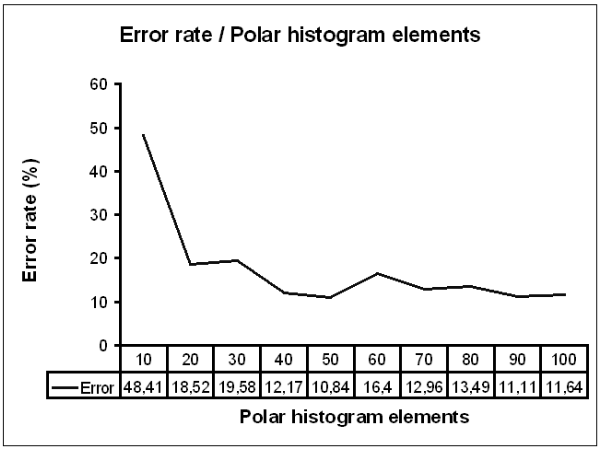


Figure 6: *Influence of the polar histogram number of elements on error rate.*

recognition. From the moment we use 40 elements, the error rate converges, so we will use histograms with 50 elements, because they gave us the lower error rate and with an higher number of elements the computational cost is also higher with no additional benefits.

Now we can calculate the error rate of our approach for the test cases described before. Figure 8 shows the recognition error rate for different number of landmark symbols in the database. The error rate increases as long as we increase the number of landmarks in the database (with the highest growing error difference being between the test set with 30 landmarks and the test set with 40 landmarks). There is an exception when there are 70 landmarks, probably caused by the use of symbols more adapted to recognition.

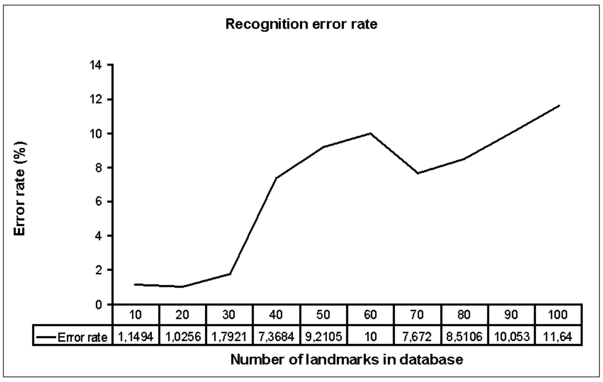


Figure 7: *Recognition error rate for different number of landmarks in database.*

As we can see, the mixture of polar histograms for image characterization and Chi-Square distance for image recognition results in a low recognition error rate. This error rate is low enough to allow a correct robot navigation guided by artificial landmarks.

## 6. Conclusions

A fast method to recognize symbols inside artificial landmarks to help in mobile robot navigation with a low error rate has been presented. This method is based on the comparison of polar histograms, an structure created from the polar coordinates of the symbols we want to recognize, taking advantage of the combination of histograms and polar techniques. An high number of right guesses is achieved when the number of symbols in the database is high enough.

Our approach could be improved including other kinds of classification criteria, trying not to increase the computational complexity of our algorithm. In our case it was not necessary, but it could be interesting to introduce colour information to the polar histogram structure. Our final goal is to use this approach into a real robot platform and study how it works.

## 7. Acknowledgements

This work has been financed by Spanish Comisión Interministerial de Ciencia y Tecnología (CICYT) Project Number TIC2001-0245-c02-02

## 8. References

[1] Hsien, J.C., Chen, S.Y., “Road Sign Detection and Recognition Using Markov Model”, 14th Workshop on Object-Oriented Technology and Applications (OOTA 2003), Taiwan, 529–536, 2003.

[2] Bernier, T., Landry, J.A., “A New Method for Representing and Matching Shapes of Natural Objects”, Pattern Recognition, 36(8), 1711–1723, 2003.

[3] González, T., Shani, S., Franta, W.R., “An Efficient Algorithm for the Kolmogorov-Smirnov and Lilliefors Tests”, ACM Transactions on Mathematical Software (TOMS), 3(1), 60–64, 1977.

[4] Chang, P., Krumm, J., “Object Recognition with Color Coocurrence Histograms”, IEEE International Conference on Computer Vision and Pattern Recognition, 1999

[5] Albiol, A., Torres, L., Delp, E.J., “An Unsupervised Color Image Segmentation Algorithm for Face Detection Applications”, International Conference on Image Processing (ICIP01), Thessaloniki (Greece), 681-684, 2001.

[6] Battiato, S., Bosco, A., Castorina, A., Messina, G., “Automatic Global Image Enhancement by Skin Dependent Exposure Correction”, IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP 2003), Trieste (Italy), 2003