International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 09
May 2016

Designing of GUI for classifying Globular Entities

AVinod Kumar & BDr. Rajeev Ratan
[A]Research Scholar, MVN University, Palwal, Haryana, India
[B]Associate Professor, MVN University, Palwal, Haryana, India
EMAIL: vinodsorout92@gmail.com

Abstract
In this paper, to evaluate the performance of space-time trellis code we calculate pairwise error probability (PEP) expressions. We then use these PEP expressions to calculate union bounds on the performance of space-time trellis codes. A MATLAB based approach is proposed and implemented to evaluate and compare performance of 4-psk with different transmitting and receiving antennas. The performance of STTC codes is evaluated by giving the performance as a function of SNR against frame error rate (FER).

Key Words: Space-time trellis coding; pairwise error probability; diversity; multiple transmit antenna; frame-error rate; outage capacity.

Introduction
Boundary detection constitutes a crucial step in many computer vision tasks. A boundary map of an image can provide valuable information for further image analysis and interpretation tasks such as segmentation, object description. etc. Fig. 1 shows an image and the associated boundary map as marked by human observers. It can be noted that the map essentially retains gross but important details in the image. It is hence sparse yet rich in information from the point of scene understanding. Extracting a similar boundary map is of interest in computer vision.

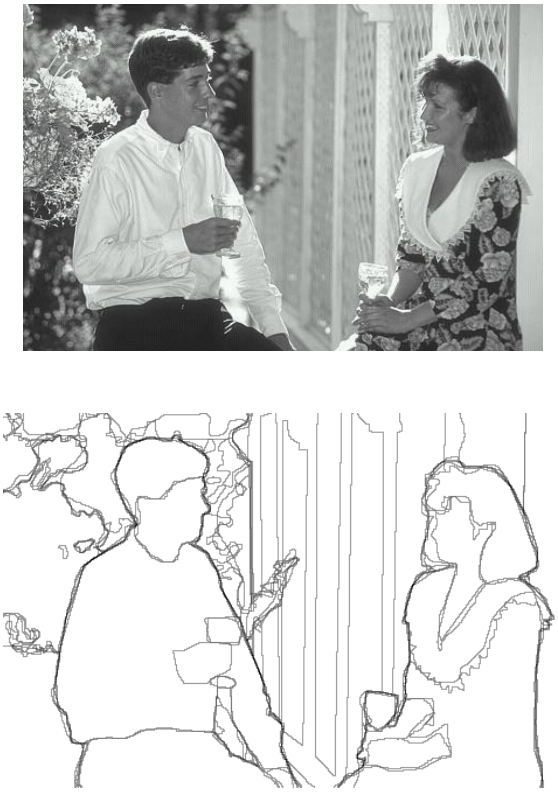


Figure 1: (a) Example image (b) Human-marked segment boundaries.

Figure 1 shows boundaries marked by 4-8 observers. The pixels are darker where more observers marked a boundary [1].

Boundary detection constitutes a crucial step in many computer vision tasks. A boundary map of an image can provide valuable information for further image analysis and interpretation tasks such as segmentation, object description etc. Figure 1 shows an image and the associated boundary map as marked by human observers. It can be noted that the map essentially retains gross

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 09
May 2016

but important details in the image. It is hence sparse yet rich in information from the point of scene understanding. Extracting a similar boundary map is of interest in computer vision. The problem of boundary detection is different from the classical problem of edge detection. A boundary is a contour in the image plane that represents a change in pixel's ownership from one object or surface to another [2]. In contrast, an edge is defined as a significant change in image features such as brightness or color. Edge detection is thus a low-level technique that is commonly applied toward the goal of boundary detection.

Methodology

Real world images are processed in our visual system to produce boundaries. These images are characterized by color, texture 1 and non-texture (only regular luminance/color based) regions. Thus, boundaries can arise due to the adjacency of any of these regions in natural images. Some of these that can occur in grey scale images (which is the focus of this paper) are shown in figure 2: luminance-luminance or LL boundary, texture-luminance or TL boundary, and texture-texture or TT boundary. Our goal is to classify objects based on their roundness using boundaries (a boundary tracing routine).

The Computational Model may be shown as shown in figure 1:

In MATLAB, the function 'regionprops' is used to measure the image properties. Here are some basic properties computed without using the function.

To find the Roundness:

- Roundness of an object can be determined using the formula:

Roundness=(4*Area*pi)/(Perimeter.^2)

If the Roundness is greater than 0.90 then, the object is circular in shape. Result=

(4*8*3.14)/10.2426.^2=0.9582

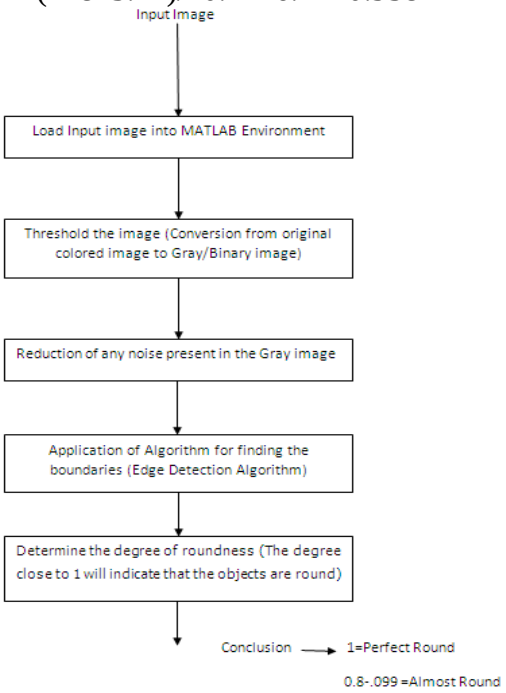


Figure 2: Computational Model for the proposed scheme

GUIDE:

GUIDE, the MATLAB Graphical User Interface Development Environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools greatly simplify the process of laying out and programming GUIs as shown in figure 3.

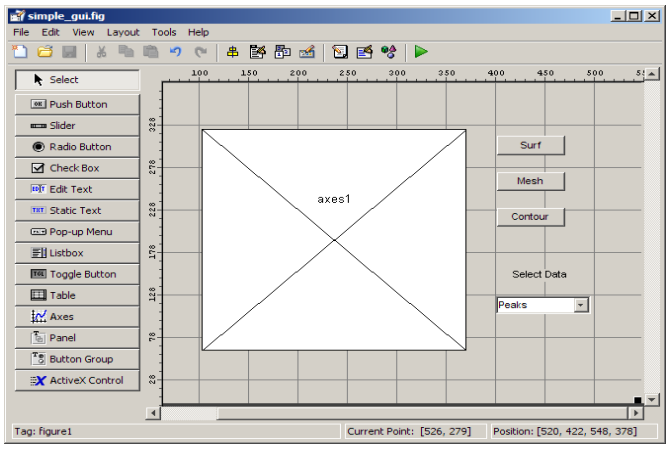


Figure 3: Ready GUI showing names in component Palette

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 09
May 2016

Implementation:

The following steps will be carried out to complete the work:

- **Step 1: Read Image**

Load the Input image to MATLAB Environment

- **Step 2: Threshold the Image**
  Convert the image to black and white in order to prepare for boundary tracing using bwboundaries.
- **Step 3: Remove the Noise**
  Using morphology functions, remove pixels which do not belong to the objects of interest.
- **Step 4: Find the Boundaries**
  Concentrate only on the exterior boundaries. Option 'noholes' will accelerate the processing by preventing bwboundaries from searching for inner contours.
- **Step 5: Determine which Objects are Round**
  Estimate each object's area and perimeter. Use these results to form a simple metric indicating the roundness of an object:
- **Step 6: Determine the degree of roundness**
  (The degree close to 1 will indicate that the objects are round)

All this will be done using MATLAB Environment. M- Files will be developed for different stages and will be executed in MATLAB and the results will be scrutinized.

Results:

The following steps will be carried out to complete the work:

Step 1: Read Image

The loaded Input image to MATLAB Environment may be shown as:

Step 2: Threshold the Image

Convert the image to black and white in order to prepare for boundary tracing using bwboundaries.

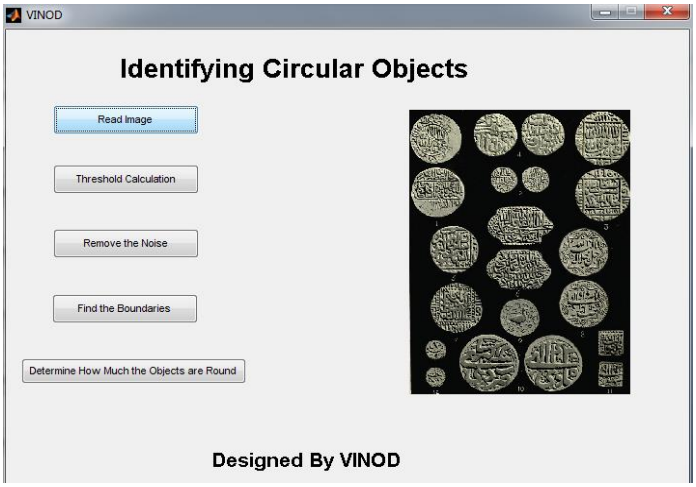


Figure 4: Loading Input Image in MATLAB

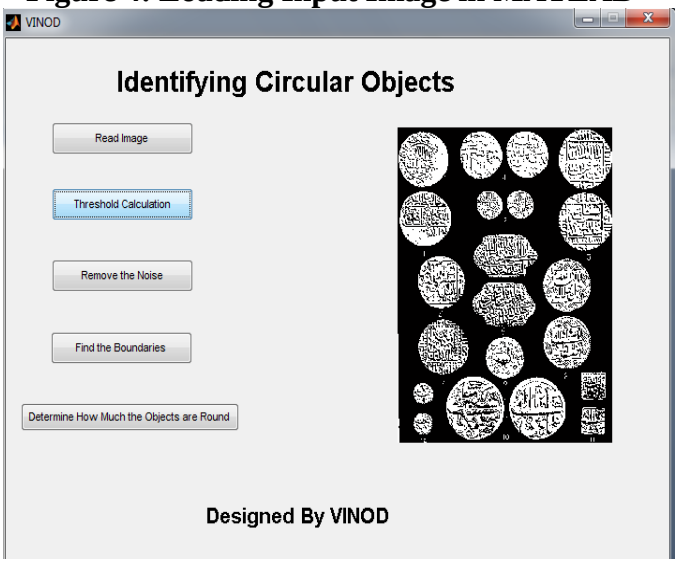


Figure 5: Threshold Calculation

Step 3: Remove the Noise

Using morphology functions, remove pixels which do not belong to the objects of interest.

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 09
May 2016

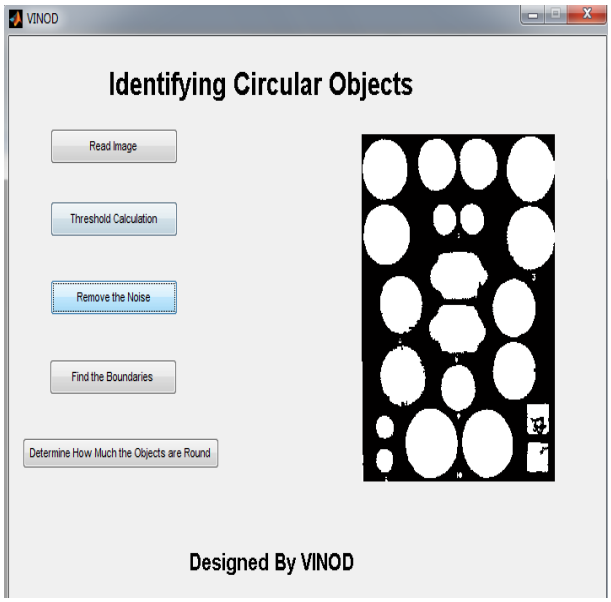


Figure 6: Noise Reduction

Step 4: Find the Boundaries

Concentrate only on the exterior boundaries. Option 'noholes' will accelerate the processing by preventing bwboundaries from searching for inner contours.

Step 5: Determine which Objects are Round

Estimate each object's area and perimeter. Use these results to form a simple metric indicating the roundness of an object:

Estimate each object's area and perimeter. Use these results to form a simple metric indicating the roundness of an object:

metric = 4*pi*area/perimeter^2.

This metric is equal to one only for a circle and it is less than one for any other shape. The discrimination process can be controlled by setting an appropriate threshold. In this example use a threshold of 0.94 so that only the pills will be classified as round.

Use regionprops to obtain estimates of the area for all of the objects. Notice that the label matrix returned bybwboundaries can be reused by regionprops.

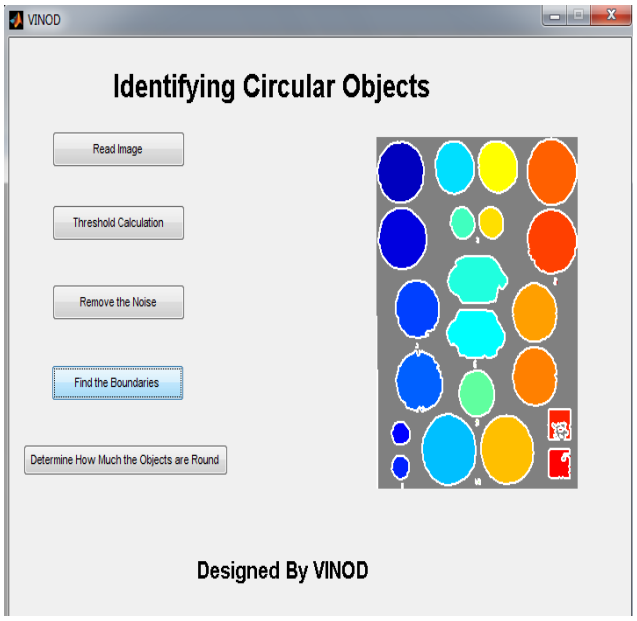


Figure 7: Boundary Detection

Step 6: Determine the degree of roundness

(The degree close to 1 will indicate that the objects are round)

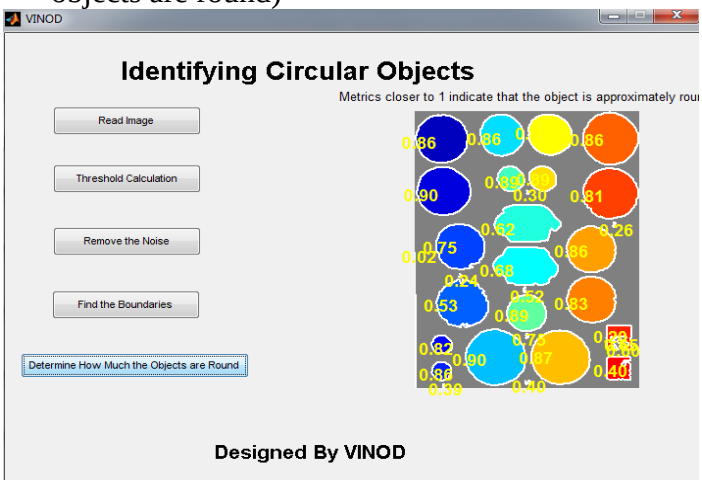
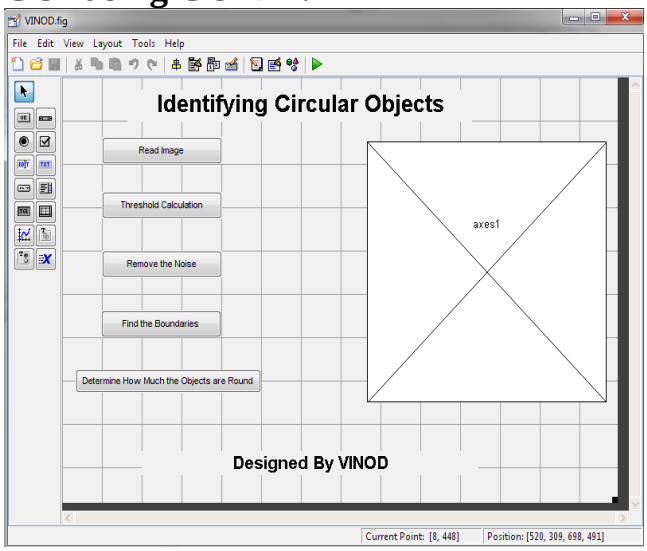


Figure 8: Determination of roundness of objects

GUI using GUIDE:



Conclusion:

Thus it is clear that the degree close to 1 will indicate that the objects are round. Thus semi circular and half circular objects can be found out by computer in any input image. All this has been done using MATLAB Environment. M- Files will be developed for different stages and will be executed in MATLAB and the results have been scrutinized.

References:

[1] Wei-Ying Ma and B. S. Manjunath, "EdgeFlow: A Technique for Boundary Detectionand Image Segmentation", IEEE Transactions on Image Processing, vol. 9, no. 8, August 2000

[2] Cosmin Grigorescu, Nicolai Petkov*, Michel A. Westenberg, "Contour and boundary detection improved by surround suppression of texture edges" Elsevier, Image and Vision Computing 22 (2004) 609–622

[3] Borislav Marinov, Veselina Gospodinova, "Adaptive Procedure For Boundary Detection In Satellite Images 3rd International Conference On Cartography And Gis 15-20 June, 2010, Nessebar, Bulgaria.

[4] Gonzalez, "Fundamentals of Digital Image Processing", Chapter 5, The segmentation Problem

[5] Rainer Lienhart, "Comparison of Automatic Shot Boundary Detection Algorithms", Microcomputer Research Labs, Intel Corporation, Santa Clara, CA