Measuring diameter of a circle within an image

Based on the **IEEE** paper “*A New Method of Circle’s Center and Radius Detection in Image Processing*” by **Zhang Mingzhu and Cao Huanrong**

COURSE PROFESSOR: NUTAN BANSODE

Made by:
TETA06 Aman Kumar
TETA10 Krushna Garkal
TETA29 Pranav Abute

YouTube Link:

https://www.youtube.com/watch?v=lfct4xnqIT0&t=161s

CONTENT

- INTRODUCTION
- CIRCLE DETECTION
- HOUGH TRANFORM
- FINDING THE CENTER AND RADIUS (Theoretically)
- OBJECTIVE
- FLOW CHART
- IMPLEMENTATION IN MATLAB
- RESULTS
- APPLICATIONS

INTRODUCTION

- Measuring objects within an image or frame can be an important capability for many applications where computer vision is required instead of making physical measurements. This application note will cover a basic step-by-step algorithm fo ct and measuring its diameter.

CIRCLE DETECTION

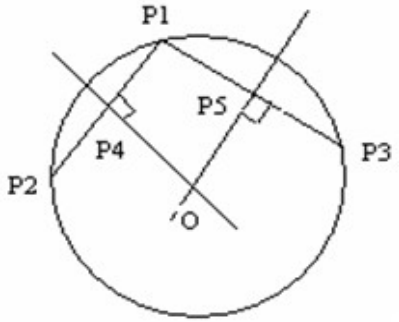
- In order to detect any circular object we need to detect the edges of the object .
- Edges are the points in the image where there is a sharp change in the color.
- So for detecting circles we need to first make the background of the image different maybe lighter or darker than the image.
- Once that is done we can easily distinguish the sharp color change in the image.

HOUGH TRANSFORM

- Hough transform is a feature extraction technique used in image analysis, computer vision and digital image processing.
- The purpose of the technique is to find imperfect instances of objects within certain class of shapes.
- We will be taking the idea from this algorithm to find the centers and radii of the detected circle.

FINDING THE CENTER AND RADIUS (Theoretically)

Once a circle is detected following calculation is done to find center and radius.



Assuming the detected circle

Here we make chord P1P2 and P1P

then bisects both the chord and the point

where they meet is the center of the circle O. We have the coordinates of P1 as (x1,y1) P2 as(x2,y2) and P3 as (x3,y3) similarly we will consider the coordinate of P4 as(x4,y4) and P5 as (x5,y5)

$$P4.\begin{cases} x_4 = x_1 + \dfrac{x_2 - x_1}{2} \\ y_4 = y_1 + \dfrac{y_2 - y_1}{2} \end{cases} \quad (1)$$

$$P5.\begin{cases} x_5 = x_1 + \dfrac{x_3 - x_1}{2} \\ y_5 = y_1 + \dfrac{y_3 - y_1}{2} \end{cases} \quad (2)$$

The line of the chord $P1P2$ is: $y = k_1 x + b_1$.

Here into: $k_1 = \dfrac{y_2 - y_1}{x_2 - x_1}$, $b_1 = y_1 - k_1 x_1$.

The line of the chord $P1P3$ is: $y = k_2 x + b_2$.

Here into: $k_2 = \dfrac{y_3 - y_1}{x_3 - x_1}$, $b_2 = y_1 - k_2 x_1$.

The perpendicular bisectors of the two chords are:

$$\begin{cases} y_{11} = k_{11} x + b_{11} \\ y_{22} = k_{22} x + b_{22} \end{cases} \quad (3)$$

Here into: $k_{11} = \dfrac{x_1 - x_2}{y_2 - y_1}$, $b_{11} = y_4 - k_{11} x_4$

$k_{22} = \dfrac{x_1 - x_3}{y_3 - y_1}$, , $b_{22} = y_5 - k_{22} x_5$.

So according to the above formulas we can calculate the center's coordinate of (x_0, y_0).

Hereinto: $x_0 = \dfrac{b_{22} - b_{11}}{k_{11} - k_{22}}$, $y_0 = k_{11} x + b_{11}$.

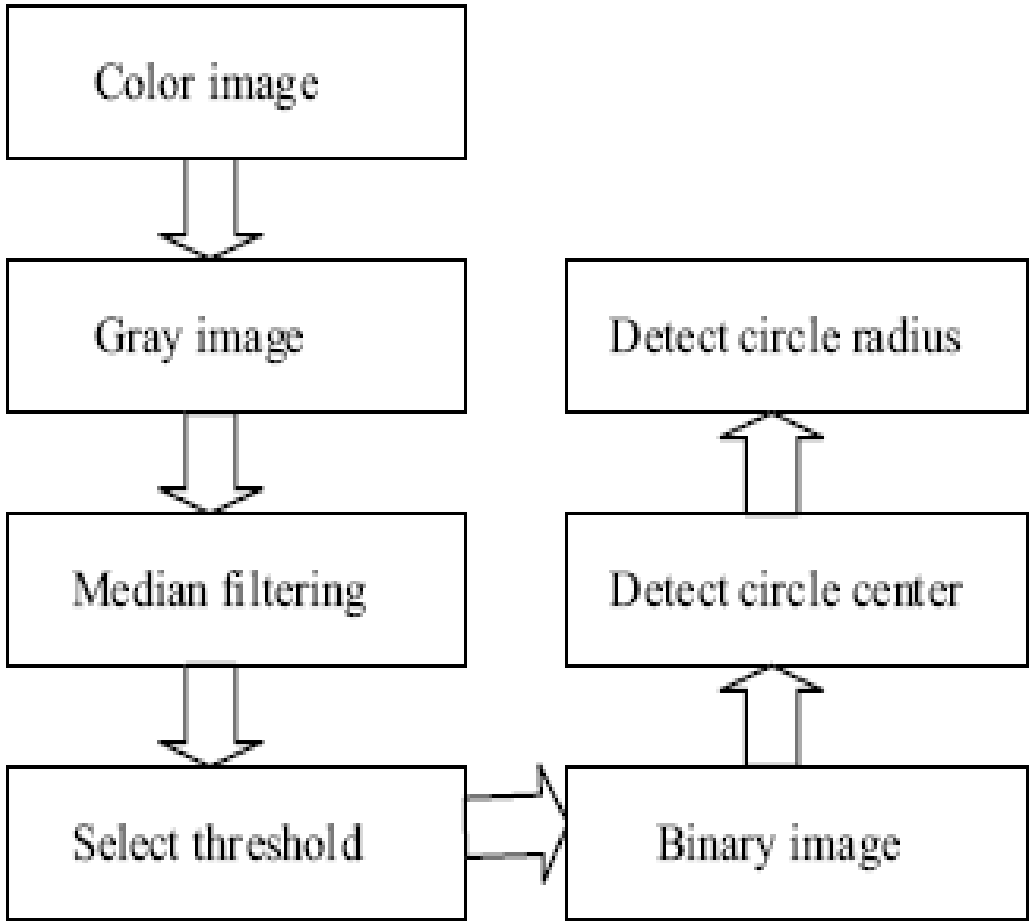
And the radius is:

$$r = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \quad (4)$$

OBJECTIVE

- To detect the circles within the image.
- To find the centers and radii of that detected circles.

FLOW CHART


Color image
Gray image
Median filtering
Select threshold
Binary image
Detect circle center
Detect circle radius

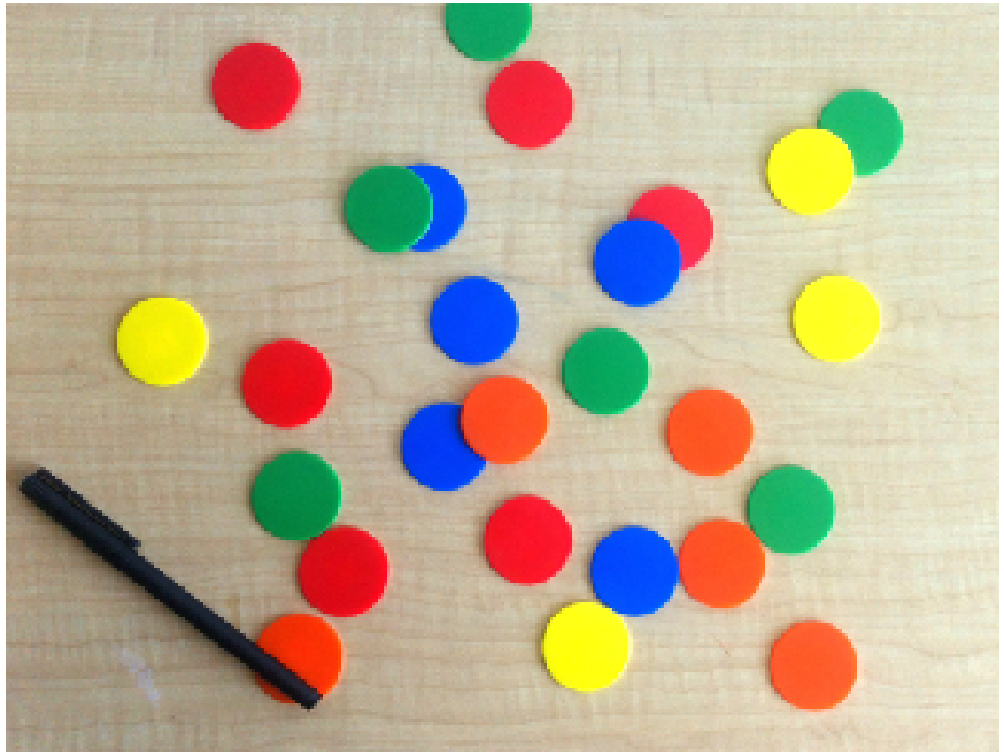
IMPLEMENTATION IN MATLAB

CODE :

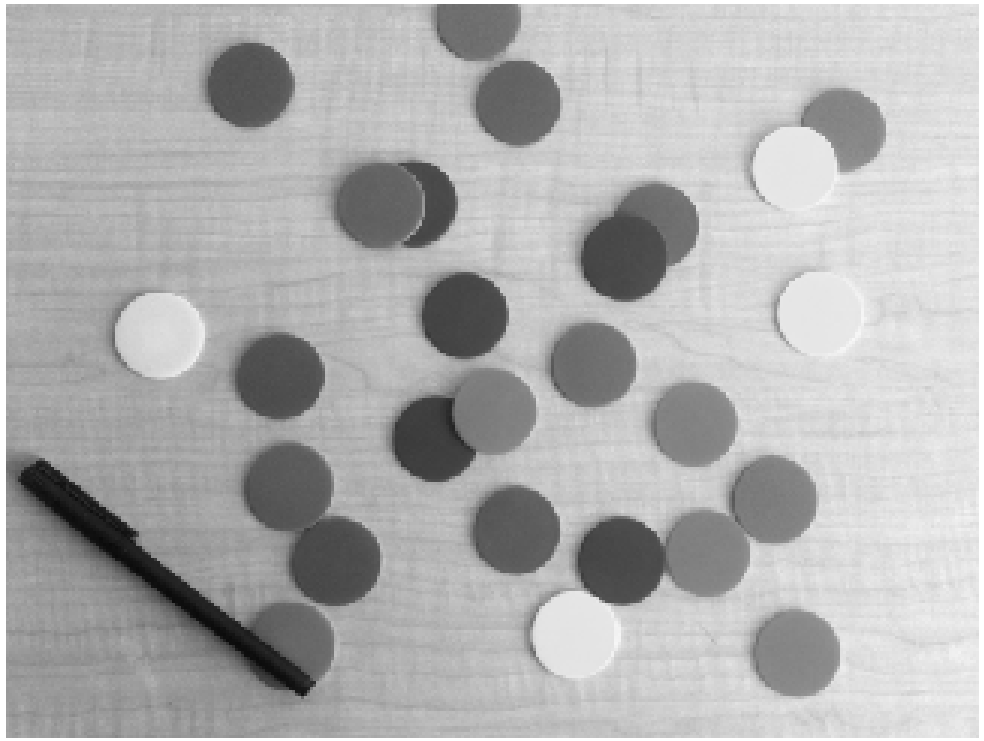
```
% Detecting and Measuring Circle's Parameters
 %Implemented by - Aman Kumar (TETA06)
% Krushna Garkal (TETA10)
% Pranav Abute (TETA29 )

    %% Load Image
rgb = imread('coloredchips.png'); % Saving the image into a variable
figure(1)
imshow(rgb) % Displaying the image
    %% Determine Radius Range for Searching Circles
d = imdistline; % to get an approximate estimate of the radii of various objects.
delete(d) % Remove the imdistline tool.
    %% Initial Attempt to Find Circles
gray_image = rgb2gray(rgb); % To see the grayscale version of this image.
figure(2)
imshow(gray_image) % Displaying the image
```

```
%The background is quite bright and most of the chips are darker than the background. But, by
default, imfindcircles finds circular objects that are brighter than the background.
%So, set the parameter 'ObjectPolarity' to 'dark' in imfindcircles to search for dark circles.

[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark')

%% Increase Detection Sensitivity
[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark', ...
'Sensitivity',0.92) %Increasing the sensitivity to 0.92

%% Draw the Circles on the Image
h = viscircles(centers,radii);
[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark', ...
'Sensitivity',0.92);

delete(h) % Delete previously drawn circles
figure(3)
imshow(rgb)
h = viscircles(centers,radii);
```

RESULTS



ORIGINAL IMAGE



DISTINGUISHING FROM BACKGROUND

OUTPUT

centres =
232.2309 227.4810
414.9079 164.4902
216.3884 72.8850
261.3064 326.0969
398.3695 221.1736
330.0556 195.4413
314.2498 260.8037
357.6694 309.7785
413.0249 326.1936
303.2436 135.8660
452.3346 253.1764
358.8867 82.3400
494.8046 293.9452
431.8169 147.9789
530.7653 96.9731
506.3809 375.5448

radii =
23.3627
22.9843
22.9531
23.7828
22.9803
22.8865
22.7151
23.0176
23.0064
22.9279
23.0301
22.6955
22.9003
22.4965
22.3829
23.7109

ALL DETECTED CIECLES

APPLICATIONS

- In medical research : Where ever the size of circles are very small and detection of the circle and it’s parameters are not possible to be found by human eyes. This can be very useful.
- In mechanical industries: circular shape tools can be detected and its parameter can checked accurately as all the tools should be accurate and in proper size.
- In space agencies : This can be very effective in calculating the size of the planets or stars which are not possible without computer vision

THANK YOU
ANY QUESTIONS?