Robotic Vision:
Rotated Clock

José Ramón Romero Chávez (A01700318)
A01700318@itesm.mx
Tecnológico de Monterrey, Campus Querétaro

Background

To have a basic understanding of Sherlock UI and its implemented algorithms

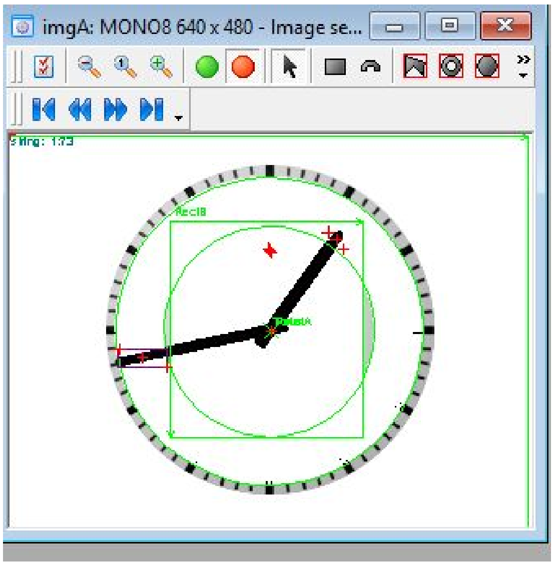
Objective:

Identify the hour shown by the clock in the image by using sherlock elements

Implementation

- Once we have opened the image set, we have to pre-processing it.
    - Set a reference point in the center of the image that will be also the center of the clock
    - Using the center as reference, create a Torus element using Sherlock UI
    - In the area generated by the torus, use the "Connectivity - Binary " Algorithm to set fing black blobs and its corresponding centroids
    - Generating a region of interest we can find the black blob for the reference (point)
- Getting the time and Angles
    - By using the difference in X and Y between the referenced center of the clock and the centroids and trigonometric functions (atan2) it is possible to define the angle generated and round it to its corresponding match to hours and minutes
    - With the pointwe can get the angle delta and reference for the other points
    - Finally it is necessary to validate for angles between 0-30 in order to express the proper time
- Generate a string by concatenating all the information calculated and finally show it on the screen

Results:



```
var X1, Y1, X2, Y2, XRef, YRef, oc1, ac1, oc2, ac2, ang1, ang2;

Vars.Point1 = Vars.Points[0];
Vars.Point2 = Vars.Points[1];
X1 = Vars.Point1[0];
Y1 = Vars.Point1[1];
X2 = Vars.Point2[0];
Y2 = Vars.Point2[1];
XRef = Vars.PointRef[0];
YRef = Vars.PointRef[1];

if (X2 == 0 && Y2 == 0) {
        X2 = X1;
        Y2 = Y1;
}

oc1 = X1 - XRef;
ac1 = YRef - Y1;
oc2 = X2 - XRef;
ac2 = YRef - Y2;

var ax = Vars.axis.x;
var ay = Vars.axis.y;

var dX = ax - XRef;
var dy = ay - YRef;

var dist = Math.sqrt((dy * dy) + (dX * dX));

var sina = dy / dist;

var radians = Math.asin(sina);

var dg = radians * (180 / Math.PI);


if (dX >= 0 && dy >= 0) {//quadrant 1

        dg = 90 + dg;


}
else {


        if (dX >= 0 && dy < 0) { //quadrant 4
        dg = 90 + dg;

        }

        else {

        if (dX < 0 && dy < 0) { //quadrant 3

                //degrees=180-degrees;
                dg = (90 * 3) + dg * -1;




        }
        else {

                if (dX < 0 && dy > 0) { //quadrant 2

                dg = (90 * 3) - dg;


                }
        }
        }
}



Vars.angulo = (dg);
ang1 = Math.atan(oc1 / ac1) * (180 / Math.PI);
ang2 = Math.atan(oc2 / ac2) * (180 / Math.PI);

ac1 = ac1 + dg;
ac2 = ac2 + dg;
if (ac1 < 0) {
        ang1 = ang1 + 180;
} else if (oc1 < 0) {
        ang1 = ang1 + 360;
}
if (ac2 < 0) {
        ang2 = ang2 + 180;
} else if (oc2 < 0) {
        ang2 = ang2 + 360;
}

//


var mh = (Math.ceil(Math.floor((((dg * 12) / 360)))));
var mmin = (Math.ceil(Math.floor((((dg * 60) / 360)))));


var anh = Math.floor(Math.ceil(ang2) / 30);
var anm = (Math.floor(Math.ceil(ang1) / 6));
Vars.angulo = mh;


Vars.Hours = anh;
Vars.Minutes = anm;

if (Vars.Hours == 0) {
        Vars.Hours = 12;
}
```