Region Detection Report

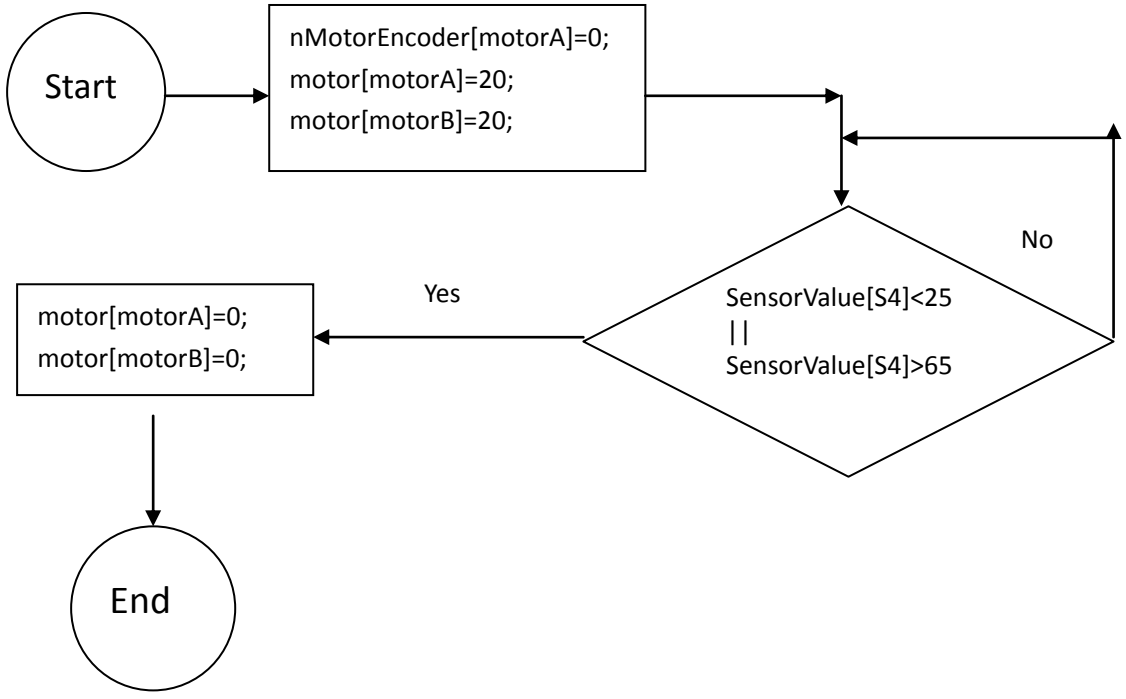
1. Introduction

Project region detection is a project using MATLAB to analysis data as well as plot graphs. We need to plot four graphs for: original data, smoothed data, derivative data, and peak of derivative data. By observing the data's change in smooth data graph, we can know color's change. After we find peak value, we minus every peak value with their previous one, and this method can let we know every color's width.

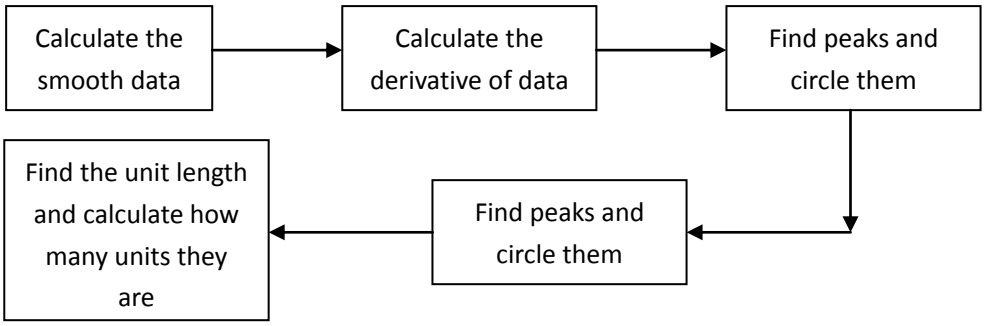
In this project, the most challenge thing is finding peak value part. Some peaks are very close to other peak, but only the highest peak is real, we need to find way to get value of real peak. We can create an array to store all peaks. To begin with, we scan all value, if a point bigger than its previous point and smaller than last one, then we say it is a peak. We also need to set an interval; if we find two or more peaks, we compare these peaks, and store the biggest one into an array.

2. Data capture software implemented in RobotC.

```
nMotorEncoder[motorA]=0; // first we need to initialize the encoder
motor[motorA]=20;  // set the speed
motor[motorB]=20;
if (SensorValue[S4]<25  ||  SensorValue[S4]>65) //if the color is not red , black and yellow
{
    motor[motorA]=0; // stop the robot
    motor[motorB]=0;
}
```



3. Data analysis software implemented in MATLAB.



If we want to find the peaks, firstly, we need to change the data smoothly. So we take every 10 adjacent values and calculate the average of them. We declare a new array to store these values. By the way, there are 9 values should disappear in this array, but we do not need to add the last 9 values, because the x-axis of the plot is index of the smoothed data, we can also draw the plot without adding 9 values.

```
subplot(4,1,2);
purecolor=color;
for i=1:length(purecolor)-9
    purecolor(i)=sum(color(i:1:i+9)) / 10;
end
plot(purecolor);
xlabel('Index');
ylabel('color');
title('Smoothed data');
```

Then, we want to calculate the derivative, so we take every 3 values of smoothed data to calculate the derivative values. For example, there are 3 values: x1, x2, and x3. The value is (x3-x1)/2. And we need to get the absolute value of them, because some of them are minus. We store these values in a new array.

```
subplot(4,1,3);
diffcolor=( purecolor(3:1:end)-purecolor(1:1:end-2) ) ./ 2;
diffcolor=abs(diffcolor);
plot(diffcolor);
xlabel('Index');
ylabel('Derivative of colors');
```

Next we must use the derivative of colors. So we have set a value (0.3), if the derivative of colors is greater than this value, we can say this position maybe a peak, and we need to check this peak until we have found the next peak. Assume we can find the next peak, if the index of next peak minus the index of current peak smaller

or equal to 40 and the next peak is higher than the current peak, we can say the current peak is not the real peak, and we cover the false peak by the next peak, otherwise the current peak is the real peak. At the same time, we store the index of peaks and y-value of peaks in 2 arrays, and circle them at the next step.

```
%Find peaks
subplot(4,1,4);
plot(diffcolor);
peakcolorx(1)=-100; %We should initialize the first variable
peakcolory(1)=-100; %assume we have a peak, becuase we need to use peakcolorx(total==1) at the begin
total=1;
for i=1:length(diffcolor)
    if diffcolor(i)>0.3  &&  diffcolor(i)>=diffcolor(i-1)  &&  diffcolor(i)>=diffcolor(i+1)
        if i-peakcolorx(total)<=40  &&  diffcolor(i)>peakcolory(total)%Two peaks are close and the first is false.
            peakcolorx(total)=i;
            peakcolory(total)=diffcolor(i);
        elseif i-peakcolorx(total)>40 %There is no two peaks are close and we need to add one
            total=total+1;
            peakcolorx(total)=i;
            peakcolory(total)=diffcolor(i);
        end
    end
end
```

For circling peaks, we only need to plot the index of peaks array and y-value of peaks array on the plot with red circle.

```
for i=1:total-1 % we should delete the first peak, becuase it is just temp and make code simply.
    peakcolorx(i)=peakcolorx(i+1);
    peakcolory(i)=peakcolory(i+1);
end
peakcolorx(total)=0;
peakcolory(total)=0;
total=total-1;

finalx=peakcolorx(1:total);
finaly=peakcolory(1:total);


plot(diffcolor);
hold on;
plot(finalx,finaly,'ro');
hold off;
xlabel('Index');
ylabel('Peaks');
title('Find peaks');
```

Finally, we need to display the results. We need to find the shortest color interval, and we set the length of this interval to be unit length. And we calculate the others intervals how many units they are. And we set some intervals to determine which color they are. And then print them.

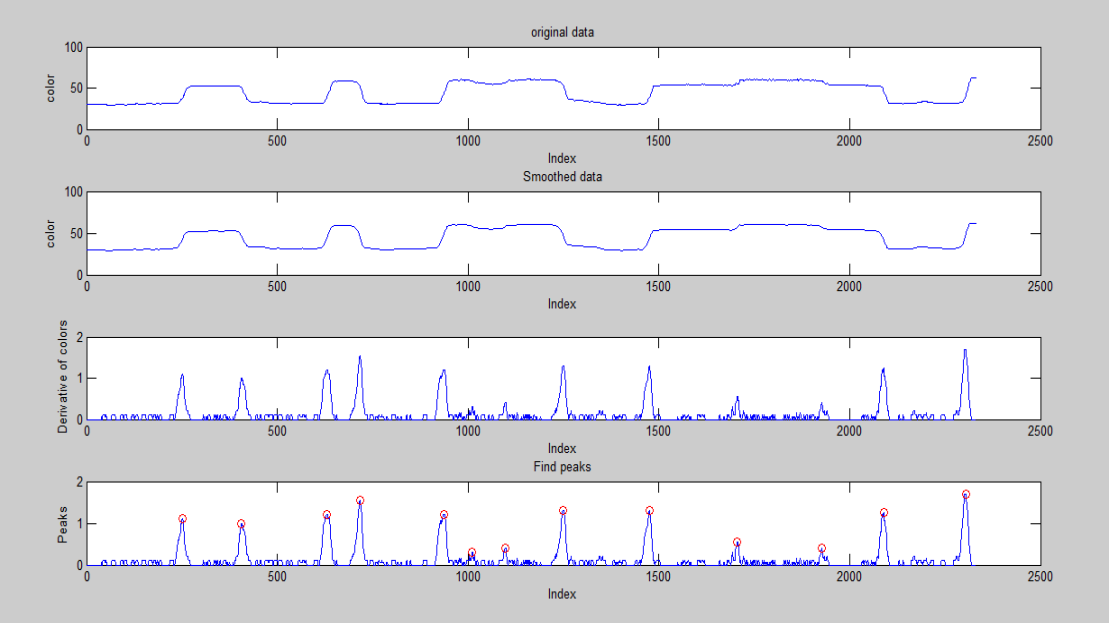
```
total=total-1;
unit=finalx(1);    %shortest distance of one color
for i=2:total      %find which is the shortest dis.
    if finalx(i)-finalx(i-1)<unit
        unit=finalx(i)-finalx(i-1);
    end
end

disp('Table of results:');

current_color=0;
last_position=0;
for i=1:total
    len=( finalx(i)-last_position ) / unit;   %calculating num of units.
    %round(len)
    if i>1
        current_color=color(finalx(i)-30);
    end
    if current_color<40
        display(sprintf('region %d is black , width is %d',i,round(len)) );  %let value = the int it close.
    elseif current_color>40  &&  current_color<57
        display(sprintf('region %d is red  , width is %d',i,round(len)) );
    else
        display(sprintf('region %d is yellow  , width is %d',i,round(len)) );
    end
    last_position=finalx(i);
end
```

4. Summary

Here are our four plots and table of results.



region 1 is black, width is 3
region 2 is red, width is 2
region 3 is black, width is 3
region 4 is yellow, width is 1
region 5 is black, width is 3

region 6 is yellow, width is 1
region 7 is red , width is 1
region 8 is yellow, width is 2
region 9 is black, width is 3
region 10 is red, width is 3
region 11 is yellow, width is 3
region 12 is red, width is 2

From this project, we learn how to deal with graphs and data; to come up with different way to satisfy different requests. The more important is we find way to solve problem.