Practical 3 Part 2 Algorithm for segmenting IR image

February 17, 2021

Jaya Parmar, MATLAB filename FLIRimage.m and IRimage.m

1 Introduction

Our aim is to try two segmentation approaches, one for the IR image and other for the RGB version

2 Method

We will first show the segmentation for RGB image and later move to IR image.

2.1 Segmentation of RGB image

The Matlab program is named FLIRimage.m
Before starting the pre-processing, we check the resolution of IR and RGB version of image by iminfo. The IR image is 'FLIR0110.jpg' and the RGB version is 'FLIR0110rgb.jpg'. Both havethe same spatial resolution of 72X72.

We start by reading in 'FLIR0110rgb.jpg' and converting it to grayscale and thereafter binarizing it.

$$i1 = imread('FLIR0110rgb.jpg'); \tag{1}$$

$$i11 = rgb2gray(i1) \tag{2}$$

Before binarizing, we perform a histogram equalisation to enhance the contrast.

$$hq1 = histeq(i11) \tag{3}$$

$$bw1 = im2bw(hq1) \tag{4}$$

The binary image is shown in Figure 1.

$$figure(1), imshow(bw1), title('Original\ binary\ image'), impixelinfo \tag{5}$$

Now we use the hit and miss algorithm as it gave good segmentation results on component images in previous exercise. The choice made to avoid under and over segmentation.

We make the structuring element for the foreground.

$$SE1 = [00000; 01110; 01110; 01110; 00000]; \tag{6}$$

The background structuring element SE2 as a complement of SE1.

$$SE2 = [01110; 00000; 00000; 00000; 01110]; \tag{7}$$

Since bw1 is the foreground, background is everything not foreground i.e.

$$bg1 = (\ bw1) \tag{8}$$

Now we erode the foreground and background with their structuring elements as below

$$e1f = imerode(bw1, SE1) \tag{9}$$

$$e1b = imerode(bg1, SE2) \tag{10}$$

The final resulting image is

$$f1 = e1f - e1b \tag{11}$$

The minus sign before e1b will change the white to black and black to white for the background image.

Please see the Figure 2 in results section.

$$figure(2), imshow(f1), title('Segmented\, RGB\, image'); \tag{12}$$

2.2 Segmentation of IR image

We show two segmentation techniques for the IR image. First, we segment using the image histogram and second we segment using multithresh.
There are no well-defined edges in the objects hence we will not use the morphological segmentation and therefore the image is not binarized.
The segmentation techniques are based on the thermal temperature colors in different sections of the image.

We read in the image and convert it to gray scale.

$$i = double(imread('FLIR0110.jpg'))/255 \tag{13}$$

$$i2 = rgb2gray(i) \tag{14}$$

Now we plot the IR image, its histogram and the grayscale image as below, see Figure 3 in results section

$$figure(1), subplot(3, 1, 1), imshow(i), title('IRimage') \tag{15}$$

$$subplot(3, 1, 2), imhist(i), title('IRhistogram') \tag{16}$$

$$subplot(3, 1, 3), imshow(i2), title('Grayimage'), impixelinfo \tag{17}$$

1. **Segmentation using histogram values**
   In this technique, we read the image histogram from Figure 3 and find the peaks. These are the objects to be segmented. Using these histogram values, we find the coordinates of the pixels in the grayscale image and replace them with the RGB pixel values from the original IR image. The segmented image is stored ina new matrix so lets first initialize the new matrix and its channels.

$$fr = zeros(size(i2)) \tag{18}$$

$$fg = zeros(size(i2)) \tag{19}$$

$$fb = zeros(size(i2)) \tag{20}$$

$$new = zeros(k, l, 3) \tag{21}$$

   We find 5 peaks in the histogram as below

   - The CPU in the histogram range 0.2 to 0.3
   - The instrument between CPU and monitor in the histogram range 0 to 0.1
   - The red horizontal edges of the monitor in the histogram range 0.45 to 0.55
   - The yellow middle of the monitor in the histogram range 0.7 to 0.8
   - The pinkish white on the top of temperature bar in the histogram range 0.85 to 1.0

Now with the help of impixelinfo, we find the coordinates in the grayscale image corresponding to above histogram ranges and replace them with the RGB values of IR image.
Code is shown below.

$$f1 = find(i2 < (0.3) < \&i2 > (0.2)) \tag{22}$$

$$fr(f1) = 0.08 \tag{23}$$

$$fg(f1) = 0.29 \tag{24}$$

$$fb(f1) = 0.62 \tag{25}$$

$$f2 = find(i2 > (0)\&i2 < (0.1)) \tag{26}$$

$$fr(f2) = 0.06 \tag{27}$$

$$fg(f2) = 0.08 \tag{28}$$

$$fb(f2) = 0.34 \tag{29}$$

$$f3 = find(i2 > (0.45)\&i2 < (0.55)) \tag{30}$$

$$fr(f1) = 0.08' \tag{31}$$

$$fg(f1) = 0.29 \tag{32}$$

$$fb(f1) = 0.62 \tag{33}$$

$$f1 = find(i2 < (0.3)\&i2 > (0.2)) \tag{34}$$

$$fr(f3) = 0.92 \tag{35}$$

$$fg(f3) = 0.13 \tag{36}$$

$$fb(f3) = 0.31 \tag{37}$$

$$f4 = find(i2 > (0.7)\&i2 < (0.8)) \tag{38}$$

$$fr(f4) = 0.82 \tag{39}$$

$$fg(f4) = 0.78 \tag{40}$$

$$fb(f4) = 0.07 \tag{41}$$

$$f5 = find(i2 > (0.85)\&i2 < (1.0)) \tag{42}$$

$$fr(f5) = 0.96 \tag{43}$$

$$fg(f5) = 0.90 \tag{44}$$

$$fb(f5) = 0.81 \tag{45}$$

Let us fill these values in our new matrix.

$$new(:,:,1) = fr \tag{46}$$

$$new(:,:,2) = fg \tag{47}$$

$$new(:,:,3) = fb \tag{48}$$

and see the results in Figure 4.

$$figure(2), subplot(2,1,1), subplot(2,1,1), imshow(i,[]) \tag{49}$$

$$subplot(2,1,2), imshow(new,[]), title('Segmentedusinghistogramvalues') \tag{50}$$

2. **Segmentation using multithresh**
   In this technique, we use Matlab's inbuilt thresholding function. Since we identified 5 objects, we use multithresh as graythresh would contrast the image in black and white. This would not work well since we are segmenting using thermal image colors.

   Let us start by using the temperature values given in the image as min and max values

$$min_temp = 22.4/100 \tag{51}$$

$$max_temp = 40.9/100 \tag{52}$$

   We are interested in a range of temperature values between min and max.

$$range = [min_tempmax_temp] \tag{53}$$

   We will now filter the image using its own self as the guide or filter. Also, since we are given the temperature values, we will use them to accomodate the range of temperature. There is an inbuild Matlab function imguidedfilter which can do this work. We choose the parameter "DegreeOfSmoothing' as the square of difference between min and max temperature. This is to capture the entire temperature range of the image.

$$smoothValue = diff(range).^2 \tag{54}$$

$$J = imguidedfilter(i2,'DegreeOfSmoothing', smoothValue) \tag{55}$$

   Now threshold the image for five threshold values identified in the first method

$$thresh = multithresh(J,5) \tag{56}$$

The five 'thresh' values are 0.2057 0.3073 0.4489 0.6122 0.7719. Note the difference in these values compared to the manually thresholded values from the histogram in method 1.

The final step is to segment the image. We use these threshold values to quantize the image. imquant function does this for us.

$$L = imquantize(J, thresh) \tag{57}$$

Let us now see the results in Figure 5.

$$figure(3), subplot(2,1,1), imshow(i, []), ('OriginalIRimage') \tag{58}$$

$$subplot(2,1,2), imshow(label2rgb(L)), title('Segmentedusingmultithresh') \tag{59}$$

3 Results

Figure 1

Figure 2

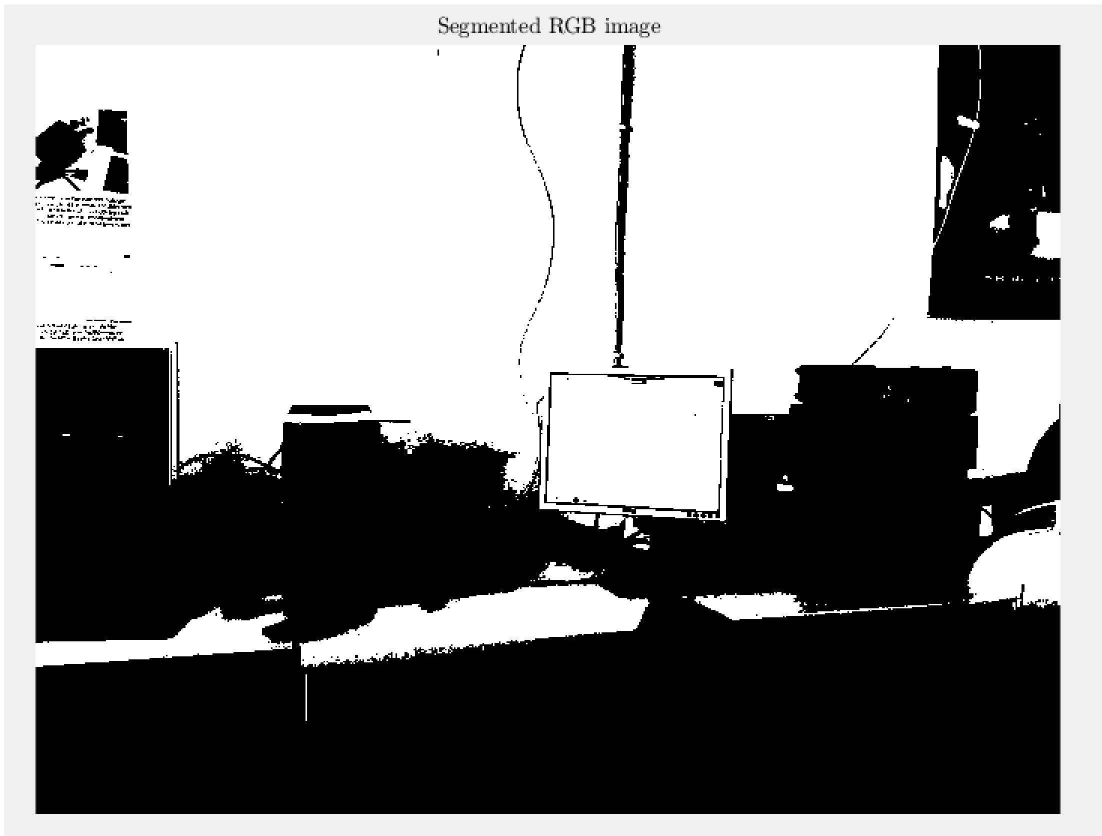


Figure 3

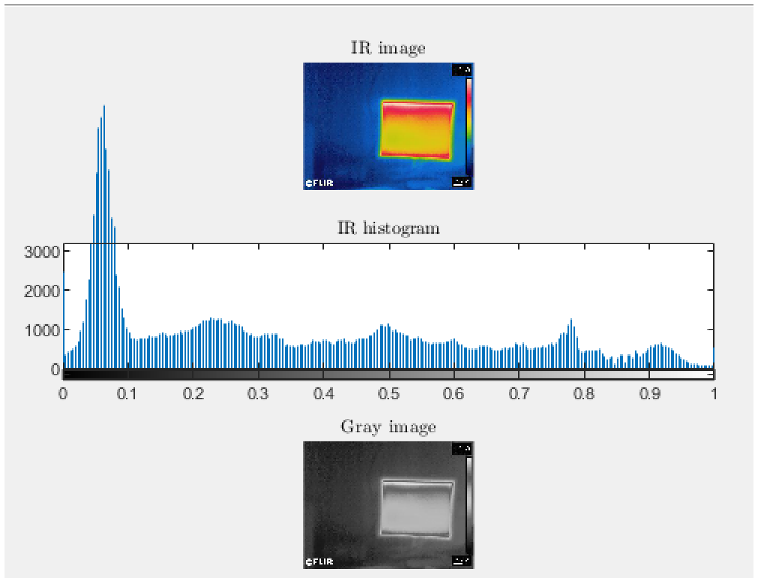

IR image
IR histogram
3000
2000
1000
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Gray image

Figure 4

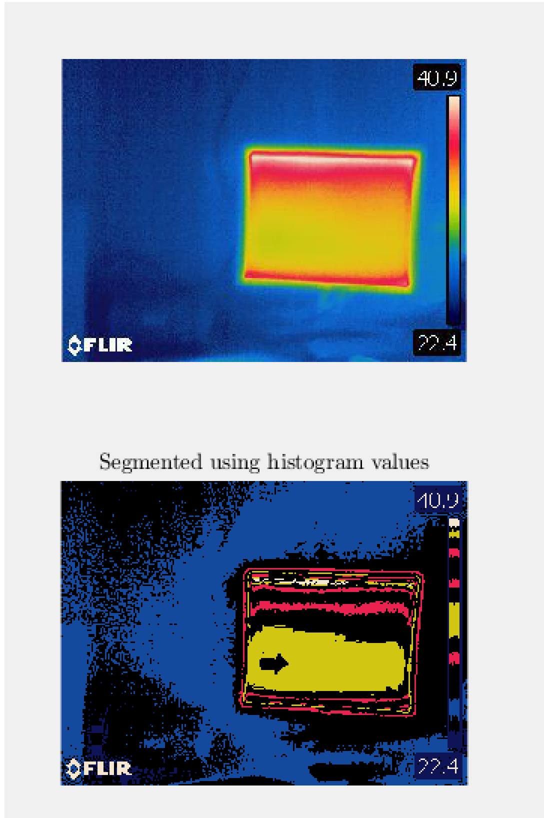
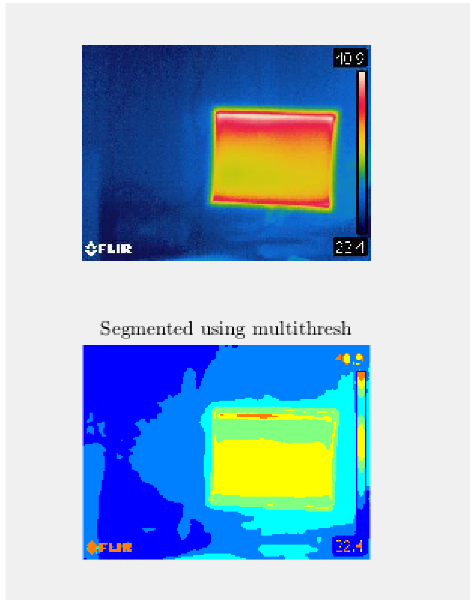


Figure 5



4 Discussion

We have segmented the IR image using a manual method and segmentation functions inbuilt in Matlab. Let us compare the two results (Figure 4 and Figure 5) by seeing the segmentation results.

- The black or left out area by the manual method is the dark blue and the cyan looking background by the multithresh method. For example look the cyan behind the monitor and in the temperature bar.
- The CPU region is light blue in both the images.
- The center of the monitor is yellow although the area captured is smaller in the manual method.
- We could use thresholding values from multithresh to improvize on the manually selected histogram values so that the segmented images from two methods look more alike.

Question 17. Can you notice any drawbacks with the segmentation in the IR comparing to the RGB? Discuss it.

IR or thermal images do not show all objects segmented clearly like the RGB image to the human eye. Instead the segments are based on the objects emissivity i.e. ability to emit energy which is measured in temperature scale. Hence if there are two seperate objects with equal emissivity lying adjacent to each other, they will appear as one big object of same temperature or color intensity. Also, our segmentation techniques are based on this color intensity hence the two objects will be segmented as one.

5 Appendix

1. RGB image segmentation

```
%same spatial resolution 72X72
%fi = imfinfo('FLIR0110rgb.jpg');
%gi = imfinfo('FLIR0110.jpg');
set(0, 'defaulttextinterpreter','Latex');
i1 = imread('FLIR0110rgb.jpg');
i11= rgb2gray(i1);
hq1 = histeq(i11);
bw1=im2bw(hq1);

figure(1),
imshow(bw1),title('Original binary image'),
   impixelinfo;

%Use hit and miss algorithm
SE1 = [0 0 0 0 0;
       0 1 1 1 0;
       0 1 1 1 0;
       0 1 1 1 0;
       0 0 0 0 0];

SE2 = [0 1 1 1 0;
       0 0 0 0 0;
       0 0 0 0 0;
       0 0 0 0 0;
       0 1 1 1 0];

bg1 = (~bw1);
e1f =imerode(bw1,SE1);
e1b = imerode(bg1,SE2);
f1 = e1f-e1b;

figure(2),
imshow(f1),title('Segmented RGB image');
```

2. IR image segmentation

```
set(0, 'defaulttextinterpreter','Latex');
i = double(imread('FLIR0110.jpg'))/255;
i2 = rgb2gray(i);
```

```
[k,l] = size(i2);

figure(1),
subplot(3,1,1),imshow(i),title('IR image');
subplot(3,1,2),imhist(i),title('IR histogram');
subplot(3,1,3),imshow(i2),title('Gray image'),
   impixelinfo;

%% Segmentation using histogram values
fr = zeros(size(i2));
fg = zeros(size(i2));
fb = zeros(size(i2));
new = zeros(k,l,3);

f1 = find(i2<(0.3) & i2>(0.2)); %CPU
fr(f1) = 0.08;  % Replace with RGB pixel values of
    original IR image 'i'
fg(f1) = 0.29;
fb(f1) = 0.62;

f2 = find(i2>(0) & i2<(0.1)); %Instrument between
   CPU & monitor
fr(f2) = 0.06;
fg(f2) = 0.08;
fb(f2) = 0.34;

f3 = find(i2>(0.45) & i2<(0.55));  %red edges of
   the monitor
fr(f3) = 0.92;
fg(f3) = 0.13;
fb(f3) = 0.31;

f4 = find(i2>(0.7) & i2<(0.8)); %yellow middle of
   the monitor
fr(f4) = 0.82;
fg(f4) = 0.78;
fb(f4) = 0.07;

f5 = find(i2>(0.85) & i2<(1.0)); %white in the
   temperature bar
fr(f5) = 0.96;
fg(f5) = 0.90;
fb(f5) = 0.81;

new(:,:,1) = fr;
new(:,:,2) = fg;
```

```
new(:,:,3) = fb;

figure(2),
subplot(2,1,1),imshow(i,[]);
subplot(2,1,2),imshow(new,[]),title('Segmented
   using histogram values');

%% Segmentation using multithresh
min_temp = 22.4/100;
max_temp = 40.9/100;
range = [min_temp max_temp];
smoothValue = diff(range).^2;
J = imguidedfilter(i2,'DegreeOfSmoothing',
   smoothValue);  %Filter the image guided by min
   & max temp range

thresh = multithresh(J,5); % We identified 5
   segments as above
L = imquantize(J,thresh);

figure(3)
subplot(2,1,1),imshow(i,[]),('Original IR image');
subplot(2,1,2),imshow(label2rgb(L)),title('
   Segmented using multithresh');
```