Practical 3 Part 2 Hit-or-Miss transform

February 16, 2021

Jaya Parmar, MATLAB filename hitnmiss.m

1 Introduction

Our aim is to perform Hit or Miss transform on the two images components.jpg' and 'componentsStrEl.jpg'

2 Method

We start by reading in the images and converting them to grayscale and thereafter binarizing them,

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

$$i1 = imread('components.jpg') \tag{2}$$

$$im1 = rgb2gray(i1) \tag{3}$$

$$im2 = rgb2gray(i2); \tag{4}$$

$$bw1 = im2bw(im1) \tag{5}$$

$$bw2 = im2bw(im2) \tag{6}$$

The binarized images are shown in Figure 1. figure(1), subplot(2,1,1),imshow(bw1),title('Original images'); subplot(2,1,2),imshow(bw2),impixelinfo;

The original binary image is shown in Figure 1.

$$figure(1), subplot(2, 1, 1), imshow(bw1), title('Original\ images') \tag{7}$$

$$subplot(2, 1, 2), imshow(bw2), impixelinfo \tag{8}$$

We make the first structuring element now. Both the images have rectangle images. So we can make the foreground structuring element like a small square.

The hit and miss transform needs the structuring element to be a combination of ones and zeros rather than just ones, hence we make SE1 below

$$SE1 = [00000; 01110; 01110; 01110; 00000]; \tag{9}$$

The background structuring element SE2 as a complement of SE1. Basically the pixels from (1,2) to (5,4) in the 5X5 matrix is the complement of SE1. The first and fifth column are the 'dont care' elements.

$$SE2 = [01110; 00000; 00000; 00000; 01110]; \tag{10}$$

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

$$bg1 = (\ bw1) \tag{11}$$

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

$$e1f = imerode(bw1, SE1) \tag{12}$$

$$e1b = imerode(bg1, SE2) \tag{13}$$

The final resulting image is

$$f1 = e1f - e1b \tag{14}$$

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

Repeating the process for second image

$$e2f = imerode(bw2, SE1) \tag{15}$$

$$e2b = imerode(bg2, SE2) \tag{16}$$

$$f2 = e2f - e2b \tag{17}$$

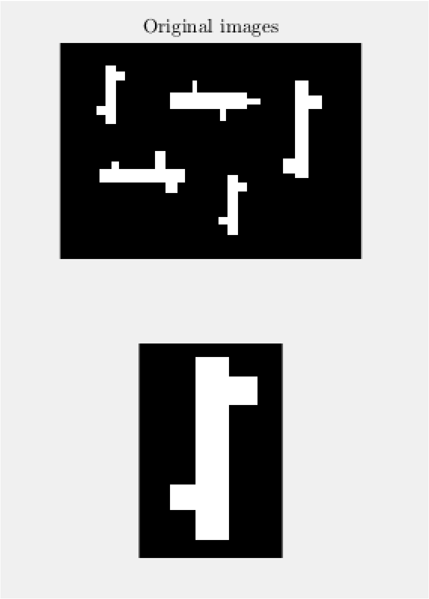
gives the results as shown in Figure 2

$$figure(2), subplot(2, 1, 1), imshow(f1), title('Transformedimages') \tag{18}$$

$$subplot(2, 1, 2), imshow(f2) \tag{19}$$

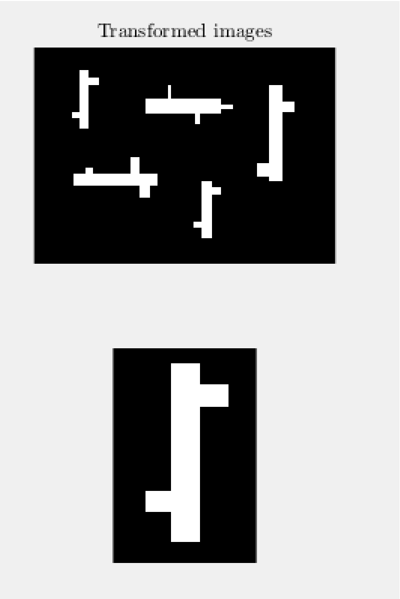
3 Results

Figure 1



Original images

Figure 2



Transformed images

4 Discussion

The Hit-or-Miss Transform gave very accurate results of image segmentation. It is because it takes into account both foreground and background while evaulating the target shape.

5 Appendix

```
set(0, 'defaulttextinterpreter','Latex');
i1 = imread('components.jpg');
i2 = imread('componentsStrEl.jpg');
im1 = rgb2gray(i1);
im2 = rgb2gray(i2);
bw1=im2bw(im1);
bw2=im2bw(im2);

figure(1),
subplot(2,1,1),imshow(bw1),title('Original images');
subplot(2,1,2),imshow(bw2),impixelinfo;

SE1 = [0 0 0 0 0; %Structuring element foreground
       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; %Structuring element background
       0 0 0 0 0;
       0 0 0 0 0;
       0 0 0 0 0;
       0 1 1 1 0];

bg1 = (~bw1);               %background bw1
e1f =imerode(bw1,SE1);      %erode foreground
e1b = imerode(bg1,SE2);     %erode background
f1 = e1f-e1b;               %add results

bg2 = (~bw2);
e2f = imerode(bw2,SE1);
e2b = imerode(bg2,SE2);
f2 = e2f-e2b;

figure(2),
subplot(2,1,1),imshow(f1),title('Transformed images');
subplot(2,1,2),imshow(f2);
```