Image Processing Project Blog

Entry 5

Sam Evans

# Morphology

“Morphology*is a broad set of image processing operations that process images based on shapes.” [1]*

In order to improve the results of cleaning the images from blog entry 3 I would need to employ morphological Matlab operations.



Figure 1: Blog Entry 3 Results

A complete list of the morphological operations is available on the mathworks website [2]. There are a lot of operations which could be useful however after much testing I chose to use just a couple.

* bwareaopen(ROI, 100)
* bwmorph(ROI,’fill’)

bwareaopen () removes areas from the region of Interest (ROI) that are a part of group of less than 100 pixels in this case.

bwmorph () applies a particular function to the ROI depending on the second argument. In this case, ‘fill’, will fill isolated interior pixels. [3]. I would also use the argument ‘bridge’ which bridges unconnected pixels.

## Code

%%%%%%%%%%%%%%%%%%%%%%%%%%% Initilisation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Clear the command window, workspace and figures.

clc**;**

clear all**;**

close all**;**

% Reads in the image for testing and converts it to double which is

% neccessary for matrix operations. The image is then converted into a

% greyscale image.

I **=** imread**(**'Vehicles1.png'**);**

% I = imread('Vehicles2.png');

% I = imread('Vehicles3.png');

% I = imread('Vehicles4.png');

% I = imread('Vehicles5.png');

% I = imread('Vehicles6.png');

% I = imread('Vehicles7.png');

I **=** im2double**(**I**);**

G **=** rgb2gray**(**I**);**

R **=** I**(:,:,**1**);**

G **=** I**(:,:,**2**);**

B **=** I**(:,:,**3**);**

%%%%%%%%%%%%%%%%%%%%%%%%%%% Contrast Streching %%%%%%%%%%%%%%%%%%%%%%%%%%

% Contrast streching widens the range of intensity values of an image. The

% number of times a certain intensity value occurs in G, is set as COUNTS,

% and the intensity value itself is set in X. The minimum and maximum

% intesity values are found, excluding intesity values that appear less

% than 100 times in the image. This is done to avoid small intesity value

% peaks from affecting the streching. The contrast is then streched using a

% simple linear transform.

**[**COUNTS**,** X**]** **=** imhist**(**G**);**

**for** i **=** 1**:**length**(**X**)**

**if** COUNTS**(**i**)** **>** 300

Gmin **=** X**(**i**);**

**break**

**end**

**end**

**for** i **=** length**(**X**):-**1**:**1

**if** COUNTS**(**i**)** **>** 300

Gmax **=** X**(**i**);**

**break**

**end**

**end**

Gstr **=** **(**G **-** Gmin**)** **/** **(**Gmax **-** Gmin**);**

%%%%%%%%%%%%%%%%%%%%%%%% Adaptive Thresholding %%%%%%%%%%%%%%%%%%%%%%%%%%

% Form of segmentation that sets pixels to either a foreground or

% background value based on an intesity value threshold. Adaptive differs

% from global by setting unique thresholds for different sections of an

% image.

%

% A square(cookie) of 7x7 pixels is thresholded for every pixel. the height

% and width of the filtered image are used so as not to go over the edge of

% the image. If the pixel is above the mean intensity value of the cookie -

% C (in this case 7/225) then it is set to 1(White). If not, it is set to 0

% (Black).

**[**H**,** W**]** **=** size**(**Gstr**);**

Gthr **=** Gstr**;**

n **=** 2**;**

half **=** floor**(**n **/** 2**);**

**for** i **=** half **+** 1**:**H **-** half

**for** j **=** half **+** 1**:**W **-** half

Cookie **=** Gstr**(**i **-** half**:**i **+** half**,** j **-** half**:**j **+** half**);**

Gthr**(**i**,** j**)** **=** Gstr**(**i**,** j**)** **>** mean**(**Cookie**(:))** **-** 12 **/** 255**;**

**end**

**end**

ROI **=** **(**1**-**Gthr**);**

SE **=** strel**(**'square'**,**1**);**

% ROI = bwmorph(edgex,'clean');

% ROI = imerode(edgex, SE);

% ROI = imdilate(edgex, SE);

% ROI = imopen(edgex, SE);

% ROI = imclose(edgex, SE);

% ROI = bwmorph(edgex,'branchpoints');

% ROI = bwmorph(edgex,'bridge');

ROI **=** bwmorph**(**ROI**,**'fill'**);**

ROI **=** bwmorph**(**ROI**,**'bridge'**);**

ROI **=** bwareaopen**(**ROI**,** 100**);**

Output**(:,:,**1**)** **=** ROI **.\*** R**;**

Output**(:,:,**2**)** **=** ROI **.\*** G**;**

Output**(:,:,**3**)** **=** ROI **.\*** B**;**

figure**,**imshow**(**Output**)**

# Results

The results were much improved as the backgrounds were increasingly removed. Unfortunately, there were still some minor background elements still persisting. However, the frame of the car is still surviving the morphological operations.

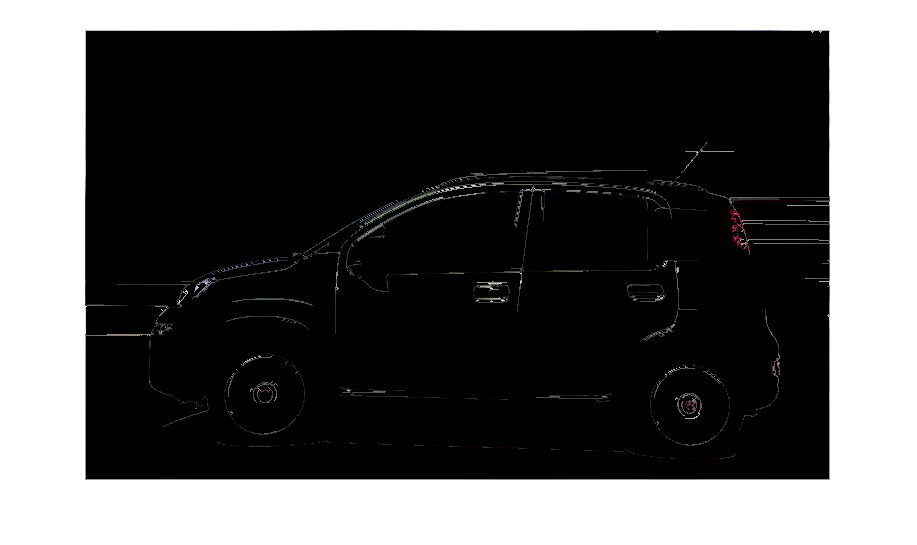


Figure 2: Vehicle 1 New Result



Figure 3: Vehicle 2 New Result

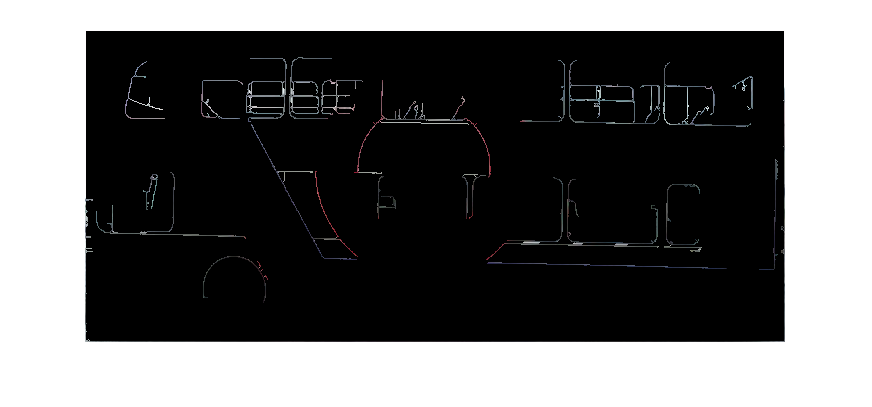


Figure 4: Vehicle 5 New Result

# Conclusions

The majority of the background image has been removed so with a little tweaking hopefully the rest will be removed. Otherwise I may need to crop the vehicle out of the image in order to demonstrate how they will be identified.

# References

[1] "Morphological Dilation and Erosion - MATLAB & Simulink - MathWorks United Kingdom", Uk.mathworks.com, 2017. [Online]. Available: https://uk.mathworks.com/help/images/morphological-dilation-and-erosion.html. [Accessed: 22- Apr- 2017].

[2] "Morphological Operations - MATLAB & Simulink - MathWorks United Kingdom", Uk.mathworks.com, 2017. [Online]. Available: https://uk.mathworks.com/help/images/morphological-filtering.html?requestedDomain=www.mathworks.com. [Accessed: 22- Apr- 2017].

[3] "Morphological operations on binary images - MATLAB bwmorph - MathWorks United Kingdom", Uk.mathworks.com, 2017. [Online]. Available: https://uk.mathworks.com/help/images/ref/bwmorph.html. [Accessed: 22- Apr- 2017].