Assignment Cover Sheet

Name	Peter Anderson
Enrolment ID	10180629
Module Code	CMP3641M
Module Title	Computer Vision and Robotics
Module Co-ordinator	Tom Duckett

Submission Regulations

The School of Computer Science requires that all submissions adhere to the guidelines as articulated in the *Presentation of assessed written work* document which can be found on the Assessments of every module site on Blackboard.

By checking the confirmation box below you are acknowledging that you have read these guidelines and adhere to the conditions within it.

Plagiarism

Part B of the University Regulations is concerned with Academic Offences, including collusion, use of misleading material, plagiarism and cheating. Please ensure you read these regulations.

You are required to keep an electronic copy of any text or code-based coursework and may be asked to resubmit this at any time.

Your work may be copied and stored for the purposes of assessment, which may include double and second marking, review and/or moderation by external examiners approved by the University, and plagiarism detection.

Extensions

Extensions may be granted for extenuating circumstances. Only the Extensions Officer can grant extensions. Applications will only be accepted if they are on the *Extensions Requests Form* which can be found on the Useful documents area of your Technology Level Site on Blackboard.

EXTENSIONS WILL NOT BE GRANTED AFTER THE DATE OF THE HAND IN

If you have an extension for this submission print your Authorisation Code here:

Type a "Y" character to confirm that you have read the University regulations and that you understand them and that the work that you have submitted is your own. Y

Date of submission **06/12/2012**

Computer Vision and Robotics

Assignment 1

1 Task 1

1.1 The Problem

There are a number of images supplied; these images contain a number of potatoes with a dark background. This assignment requires certain characteristics of the potatoes to be output. To solve the problem computer vision techniques will be utilised alongside MATLAB to segment the image and extract the information about the potatoes.

Ten images have been provided for this problem. One algorithm must be written that will perform all actions required to extract all the information about each potato in each image.

1.2 How Was This Done

A variety of different computer vision techniques have been used to solve this problem. This section will explain what techniques were used and why they were used.

The images provide are not perfect, the background of the image is not a single solid colour. However, the entire background is significantly darker then the potatoes. Initially the image is converted into a grey scale image; this makes it easier to filter out any unwanted noise from the image, such as the background. Figure 1 shows a RGB image converted to greyscale.

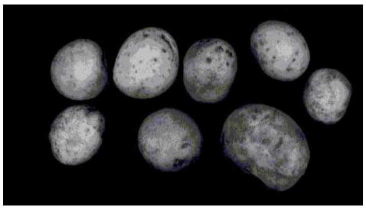


Figure 1: GreyScaleImage

Next a Mean filter is used to reduce the amount of noise in the supplied images. A Mean filter is applied to the grey scale image; the resulting image is subtracted from the grey scale image to create a ‘filtered’ image. This image contains all noise that needs to be removed from the grey scale image. The filtered image is now subtracted from the grey scale image to leave a noise free image.

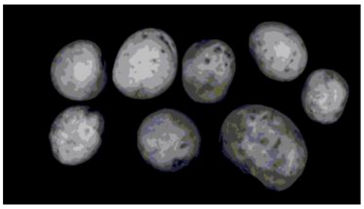


Figure 2: Mean Filter

Figure 3: Filtered Image

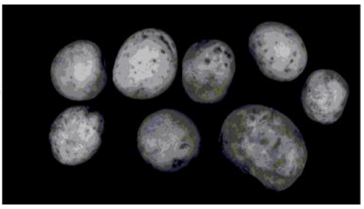


Figure 4: Noise Removed

Next we need to convert the image into binary so that we can perform labelling on all connected components. First we need to use a threshold value to ensure that all the background is converted to black, leaving all the potato to be converted to white. The threshold values used here range from 0 to 20. Figure 5 shows the image converted to binary.

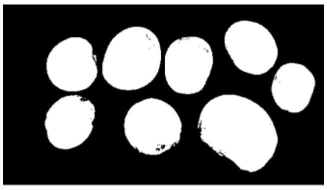


Figure 5: Binary image

Once the image has been converted to binary gaps appear in the potatoes, these holes can be filled using the ‘imfill’ function provided in MATLAB. This function looks for black areas that are encompassed by white areas and converts them to white. Figure 6 shows the binary image filled.

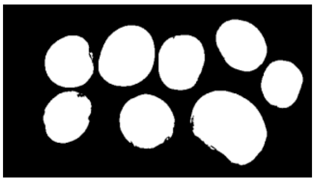


Figure 6: Filled image

After filling the image there are still segments that should be connected to the potatoes, but are not. These gaps can be breached by using morphological operations such as erode, dilate and close. Figure 7 shows the binary image after these morphological operations have been applied to the image.

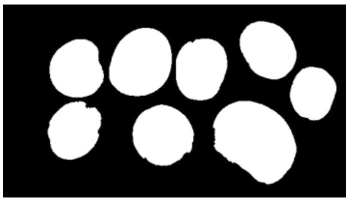


Figure 7: Binary after morphological operations

Now there is a template available that shows where all the potatoes are located and where their edges are. It is possible to count the number of potatoes in the image by using a function called bwlabel. This function searching the image for connected components, which just means contiguous blocks of the same colour. Using this function with the label2rgb, dilate and subtracting and adding images, it was possible to give all the potatoes a blue outline. Figure 8 shows the outline image being added to the original image of the potatoes.

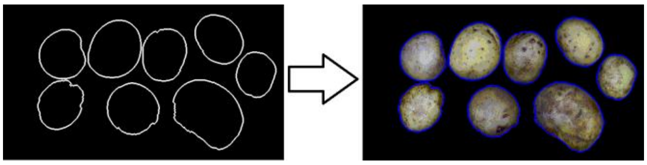
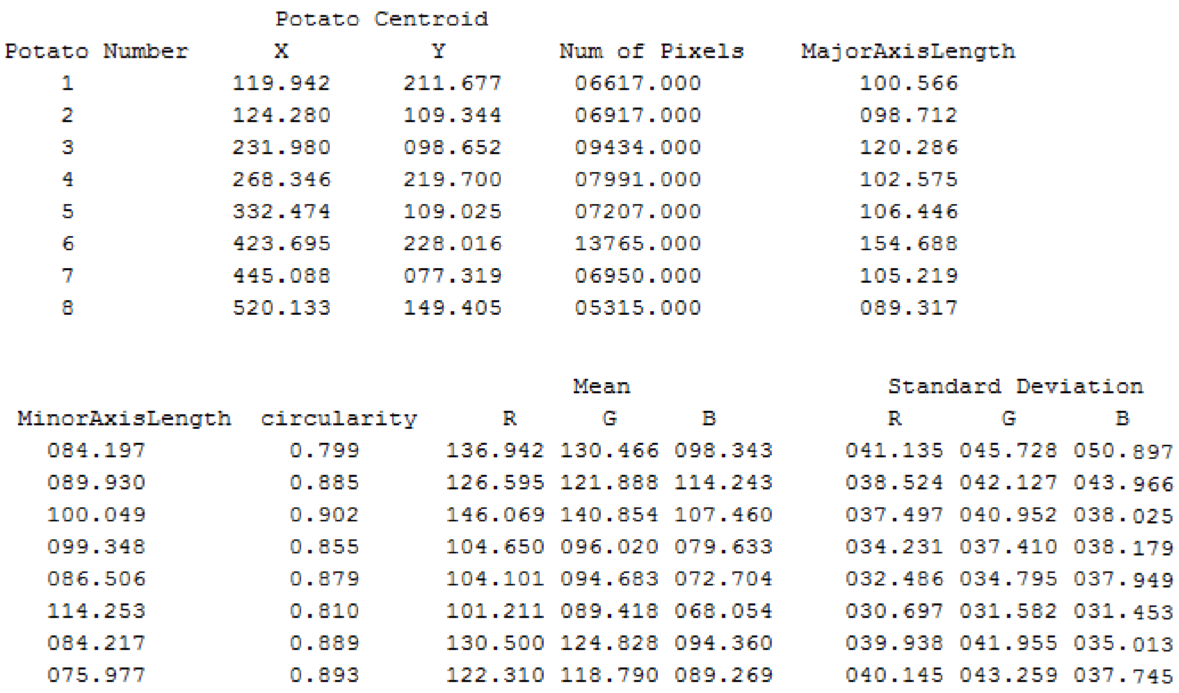


Figure 8

Using this information from the bwlabel is was possible to output information about each potato. This is showing in figure 9.



```
                       Potato Centroid
Potato Number         X            Y          Num of Pixels    MajorAxisLength
     1             119.942      211.677      06617.000            100.566
     2             124.280      109.344      06917.000            098.712
     3             231.980      098.652      09434.000            120.286
     4             268.346      219.700      07991.000            102.575
     5             332.474      109.025      07207.000            106.446
     6             423.695      228.016      13765.000            154.688
     7             445.088      077.319      06950.000            105.219
     8             520.133      149.405      05315.000            089.317

                                                 Mean                        Standard Deviation
 MinorAxisLength  circularity          R        G        B             R        G        B
   084.197            0.799        136.942  130.466  098.343       041.135  045.728  050.897
   089.930            0.885        126.595  121.888  114.243       038.524  042.127  043.966
   100.049            0.902        146.069  140.854  107.460       037.497  040.952  038.025
   099.348            0.855        104.650  096.020  079.633       034.231  037.410  038.179
   086.506            0.879        104.101  094.683  072.704       032.486  034.795  037.949
   114.253            0.810        101.211  089.418  068.054       030.697  031.582  031.453
   084.217            0.889        130.500  124.828  094.360       039.938  041.955  035.013
   075.977            0.893        122.310  118.790  089.269       040.145  043.259  037.745
```

Figure 9

1.3 Limitations

This method of detecting potatoes is not without its flaws. This method does not detect the entire potato; in some cases it has missed the edges of the potato or event small chunks. The problem with this is that you do not get entirely accurate information from the image. This algorithm also assumes that the background is a very dark colour, and that the potatoes are not similar in shade to the background.

2 Task 2

2.1 The Problem

This problem involves segmenting potatoes from a non-static background. There are a number of images supplied, some with just rollers on a conveyor belt and other images with potatoes on the rollers.

This problem requires a model of the background to be created in order to segment the potatoes out of the image. One algorithm much work for all images with no alterations.

2.2 How was this done?

This solution made use of a colour model for the background. This was an ‘m by n by p’ matrix with each channel being stored in each dimension (e.g. m = Red channel). This was built but going through each pixel of the background images, and setting each colour that was found to true in the colour model.

Once this model was built it required a few tweaks to make sure that it picked up as much of the background image as possible. Because this is a non-static background it means that there could be colours similar to the ones in the background images in the test images. This problem was partial solved by applying morphological operations to the colour models, to increase the number of colours flagged to be the background, that are close to already flagged colours. Once the colour model was finished it was used to remove all flagged colours from the test images. Figure 10 shows a sample image after colour removal.

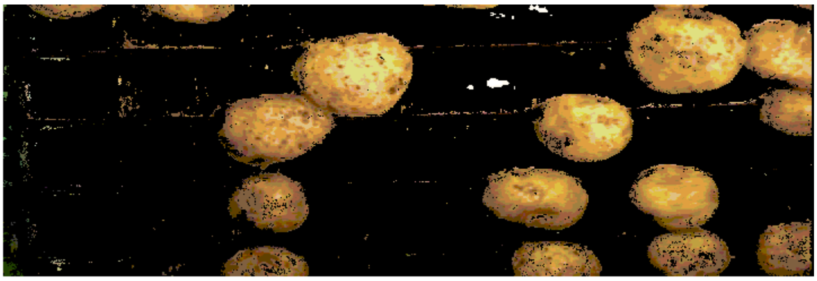


Figure 10: Background removed using colour model

As figure 10 shows now all the background is removed and some of the potatoes are removed as well. To resolve this problem a similar method that was described in **2.1** was used to removed noise from the image. Also all connected components that consisted of less than a thousand pixels were removed from the image. The reason for this is because there is a minimal chances that there are any potatoes that are smaller than a thousand pixels. Figure 11 shows the image after all noise has been removed.

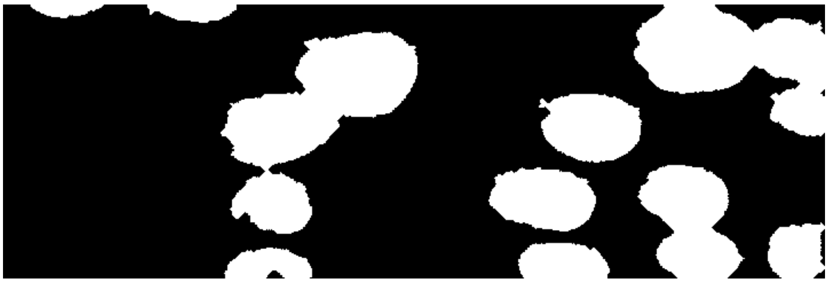


Figure 11: Noise removed

Once all noise has been removed a template image is left (figure 11). This template is used to remove the background from the original image whiles keeping as much of the original information of the potatoes as possible. Figure 12 shows the final image.

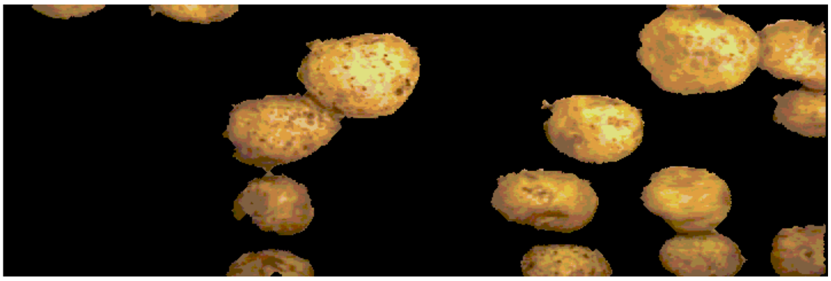


Figure 12: Final image

2.3 Limitations

This algorithm does not work perfectly, some parts of the potatoes are removed, and not all of the background is removed in all the images. The reason why some of the background is not removed is because the colour of the background is so similar to the potato the algorithm cannot distinguish between the two. In other cases the whole potato is removed; this only accurses when a small part of the potato show up on the edge of the image. This happens because the connected component is less than one thousand pixels in size.