Ashton Gray

ECE 579IP – Project 4

Watershed:

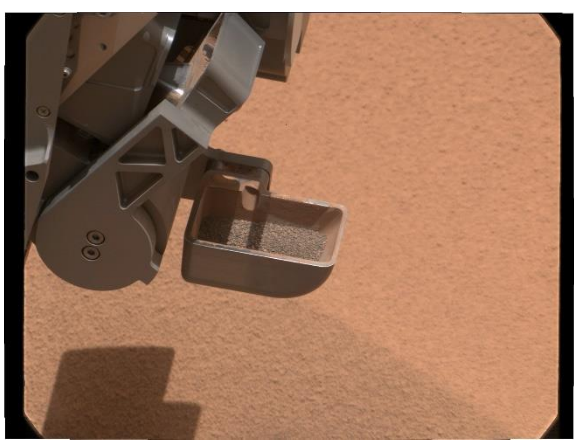
Background:

Watershed is another algorithm that can perform image segmentation. It does this by treating a grayscale image like a topographic map, where brightness levels are the elevation. The uniform low light levels are treated as basins. At each basin, the gray level is increased, starting with the lowest one, until all basins are at the same height and the water does not cross a pre-determined watershed line. The watershed line is where if you placed "water", it would fall either way. If when increasing the gray level, a watershed line is crossed, the value of the line is increased and a "dam" is created. To create the dam, you check before the watershed line is crossed, and take the union of the basins before they cross. Then the intersection of that and the next step, where the basins are combined. A dilation is then used that stays within the intersection. A dilation makes boundaries of objects larger. Then a second dilation that are within the intersection and would create spilling. These are the pixels that are increased in brightness level to create the dam. We will be performing the watershed on the gradient of the image. However, one issue with this, is that over-segmentation will be produced due to irregularities in the gradient. To fix this, we will be using markers that define where the basins are.

Skimage

To perform the watershed algorithm, we will be using the Sci-Kit image package. First, the gradient will be created by using: "rank.gradient()". This takes in the grayscale image, and a structuring element. The structuring element used will be a disk, where the radius is a parameter of the problem. Next, we will be creating a second gradient, with a larger disk radius of the first gradient for another parameter. This will then be labelled using "label()" to create markers. Both the first gradient, and the markers are used as inputs to watershed().

Image 1



Original Image 1

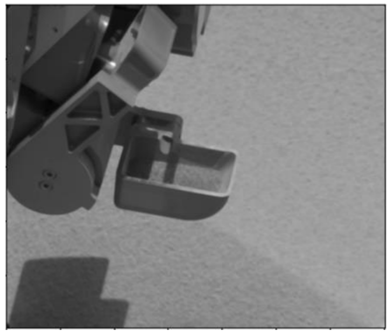
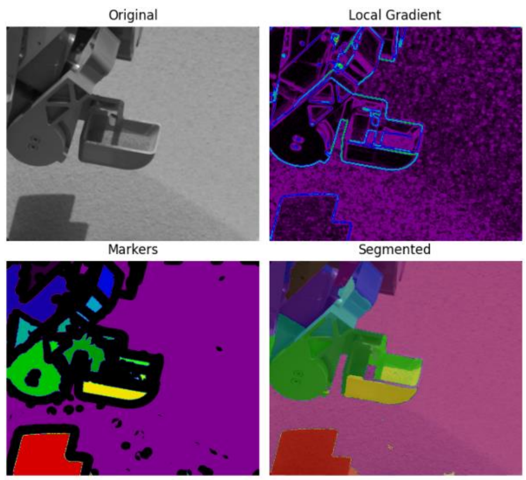


Image 1 Grayscale

Looking at the grayscale image, there are clear edges between the rover and soil. Therefore, using grayscale for the gradient should yield good results.



Values Used:

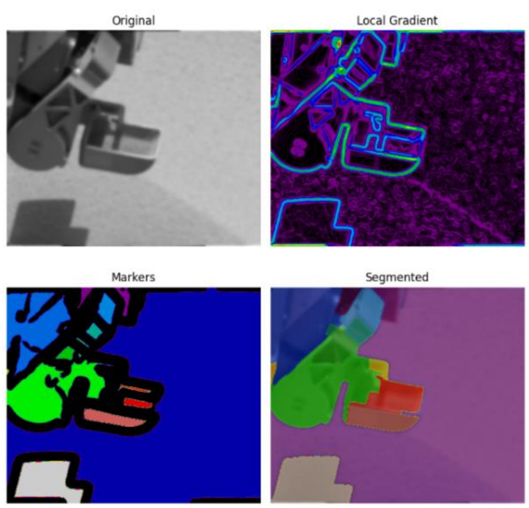
Local Gradient Disk Radius: 5

Markers Disk Radius: 25

Markers: < 30

Image 1 Grayscale, Local Gradient, Markers, Watershed Image

As you can see with the local gradient, it was true that the edges of the rover and soil are clearly defined. However, in the markers image, we can see that parts of the soil had enough variance to create it's own marker, and therefore when the watershed was performed has it's own segmentation. To fix this, we can do more blurring in the pre-processing using a gaussian blur of sigma = 7.



Values Used:

Local Gradient Disk Radius: 5

Markers Disk Radius: 25

Markers: < 30

Blur = 7

Image 1 Blurred Grayscale, Local Gradient, Markers, Watershed Image

As you can see, the blurring lead to much better results. Now, the soil is all one segment. However, you can notice that part of the scoop is still missing. This is a limitation to watershed, where if the edges are not able to be defined as clearly, it will not be able to segment between two objects. In this case, I was not able to get a clear edge in the scoop without also getting clear edges in the soil. You can also notice that the rover is still in multiple segments. This is because there are many different, uniform areas of the rover, and therefore will create multiple edges.

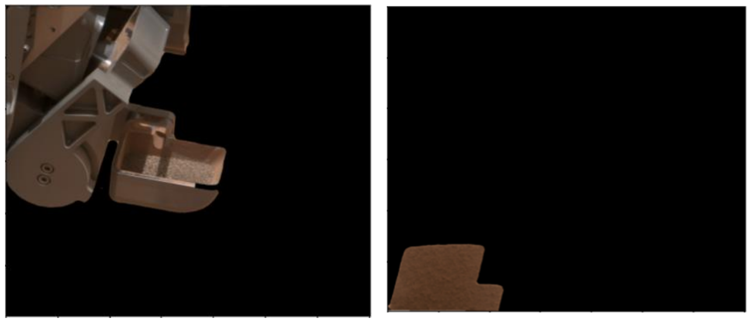


Image 1 Watershed Segmentation

Image 2:

Original Image 2

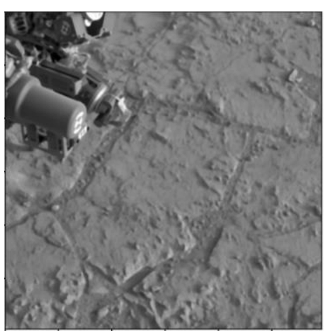


Image 2 Grayscale

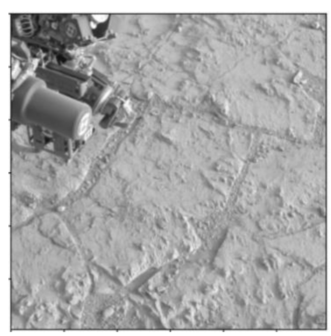
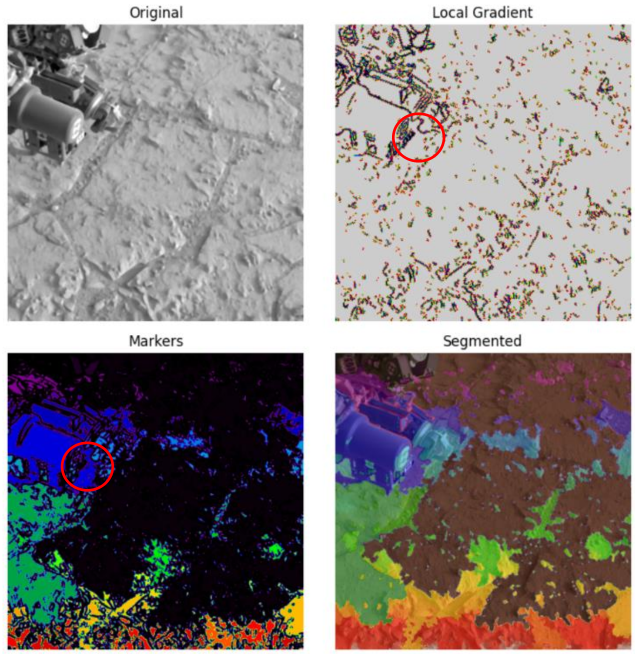


Image 2 Red Channel

We know that the edges of the object are extremely important for the gradient. Therefore, when looking at the grayscale image (which is the average of RGB channels), we can see that the edges between rover and soil are not well defined. Instead, when looking at just the red channel of the image, we can see that the edges between rover and soil are much more defined, since there is not much of a red hue in the rover. This is what we will use for the gradient.



Values Used:

Local Gradient Disk Radius: 3

Local Gradient: < 40

Markers Disk Radius: 4

Markers: < 20

Image 2 Red Channel, Local Gradient, Markers, Watershed Image

This was the best rest results achieved using watershed on image 2. Looking at the local gradient, we can see the outline of the rover, however, in many spots, the lines are unable to connect. This leads to issues in the markers such as within the red circle. Here, you can see the marker bleeding from within the rover to the soil due to the gradient line not being fully closed. This is a limitation to the algorithm, because it relies heavily on distinct edges in the gradient. For this image, there were many edges in the soil which led to gradient lines, and therefore segmentations, that could not be fixed by blurring due to losing the edges of soil and rover.

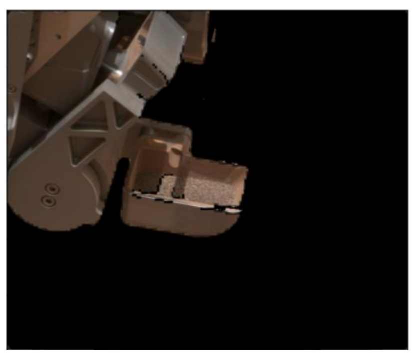
<u>Watershed + GrabCut</u>

To fix the over-segmentation of rover and background for each image shown using watershed, we can use a method called grabcut. Grabcut extracts foreground from background with information from the user. This can be done in two different ways, first is by using a box around the object to explicitly tell the algorithm where the object is, or by using a mask to show wherever it is white, the object is within it for foreground, and if it is black, it is known background. This masking method is what will be used on top of watershed segmentation. To do this, the union of all labels from the watershed that make up the rover will be labelled white (foreground), and the rest of the labels that make up the soil will be labelled as black (background).

Image 1:

Watershed Segmented Image 1

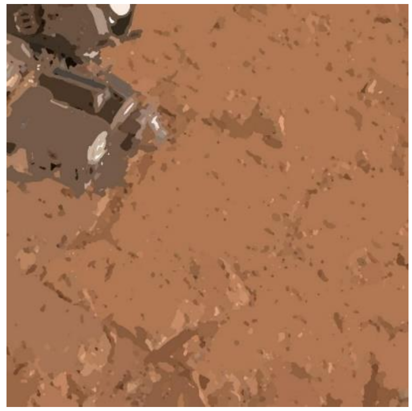
Manually Labelled Mask From Watershed Segmentation



Watershed + Grabcut Results

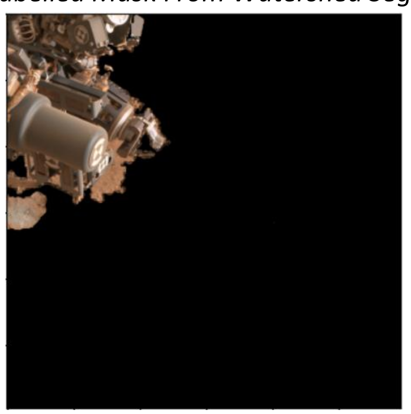
As you can see, this actually led to worse results than from the watershed itself. This is because the mask predicts where background could still be, and therefore parts of the rover were defined as background, such as the top of the scoop. This is due to the watershed results already segmenting very close to what the actual image should be.

Image 2:



Watershed Segmented Image 2

Manually Labelled Mask From Watershed Segmentation



Watershed + Grabcut Results

In this instance, the addition of grabcut created much better results. This is because it was able to remove parts of the soil away from edges of the rover, combine all the segments of soil together, and combine all the parts of the object together. However, it is still not perfect. It still thought that parts of the soil were part of the object. This is due to the limitations of the segmentation, that lead to a less than ideal mask of the object, including a lot of the soil in the foreground.

GrabCut

Mentioned above, the grabcut method also works by using a rectangle that surrounds the object, to give better information to the algorithm of what is foreground vs background. We will test this method by itself without any previous segmentation.

Image 1

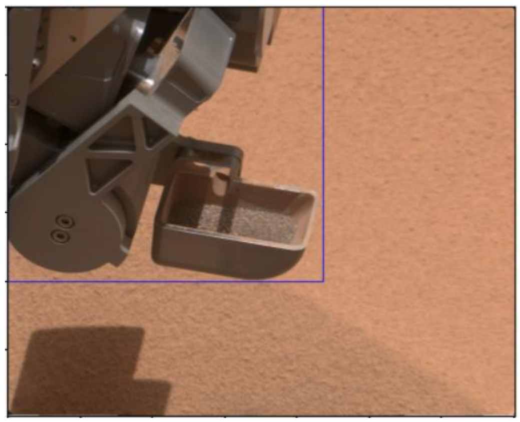
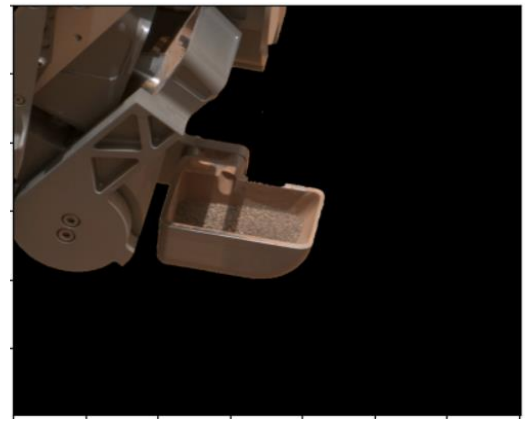


Image 1 With Bounding Rectangle



Grabcut Segmentation

As you can see, the results of grabcut are extremely good. The inputted user information is a strong advantage for the algorithm in deciding what is foreground vs background. There still was a slight issue with the top of the scoop, where it was deemed as background, this could be because of the color similarities between that part of the scoop and surrounding soil

Image 2

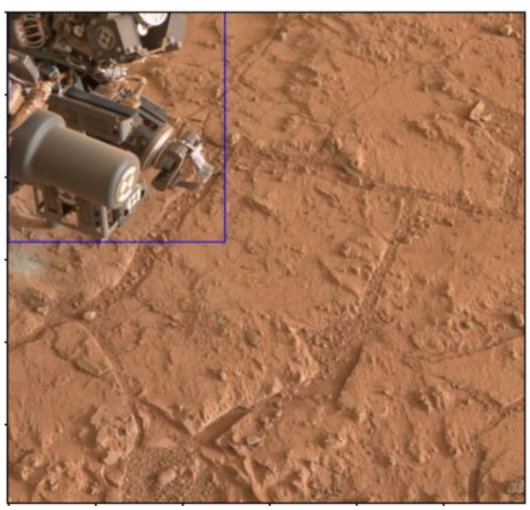
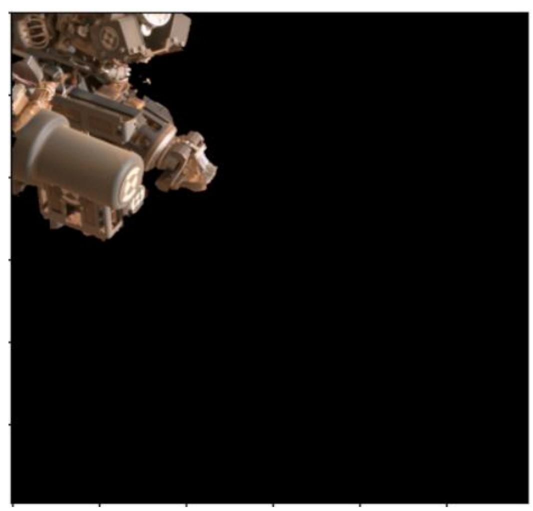


Image 2 With Bounding Rectangle



Grabcut Segmentation

As you can see, the results of grabcut are extremely good. The inputted user information is a strong advantage for the algorithm in deciding what is foreground vs background. Again, there is a slight loss of part of the rover, which is due to the similarity in color of rover and background.

Conclusion:

In conclusion, the watershed algorithm proved to be very powerful for image segmentation, and showed good results. However, had a limited success due to the need of having well defined, connecting edges of the object in the gradient image. We also saw, that when an over-segmentation is created from watershed, manually creating a mask and using the grabcut on both images proved to make the algorithm even stronger. Lastly, grabcut showed that with any user input, high-accuracy segmentation can be created with ease.

Best Unsupervised Segmentation:

Original Images

Image 1

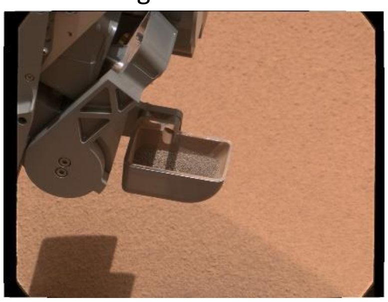


Image 2

Image 3

Image 4

Image 1:

The best segmentation performed for image 1, was using just Felzenszwalb, with scale = 600, Sigma = 3.5, and Min Size = 10000. Even though the edges were not perfect, there was no loss of rover to the soil, which many other algorithms had happen. Such as seen in this report with watershed.

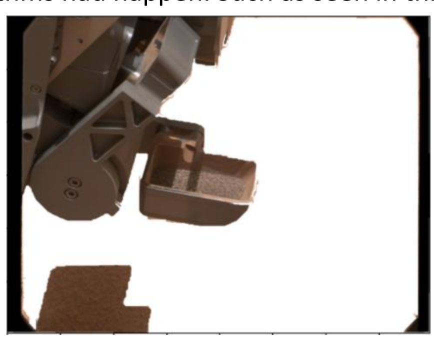


Image 1 Best Segmentation

Image 2:

The best segmentation performed for image 2, was also using Felzenszwalb, with scale = 80, Sigma = 20, and Min Size = 17000. This image was especially hard to segment due to the many edges of soil, along with the many edges in rover. Therefore, most edge-based algorithms had a very hard time with this image. However, this segmentation kept the most detail of the rover without losing parts of the rover to soil. The only downside was including the soil with the segment.

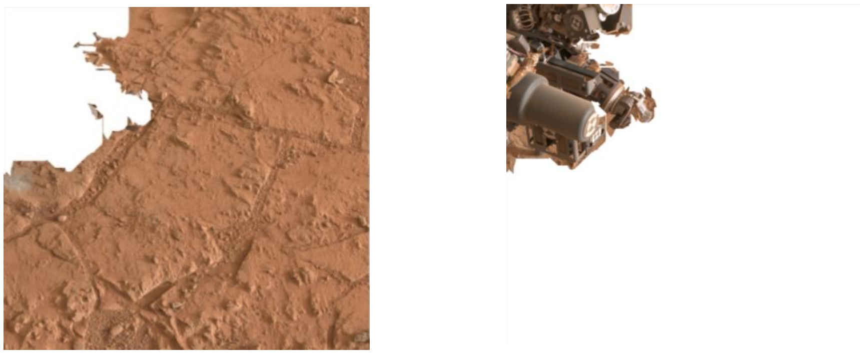


Image 2 Best Segmentation

Image 3:

The best segmentation performed for image 3, was using Felzenszwalb, with scale = 230, Sigma = 15, and Min Size = 30000. This image was hard to segment due to the dark rocks in the image looking like their own objects to the segmentation algorithms. However, this was not an issue for felzenszwalb. Though, in almost every segmentation, the silver aspects of the rover were lost.

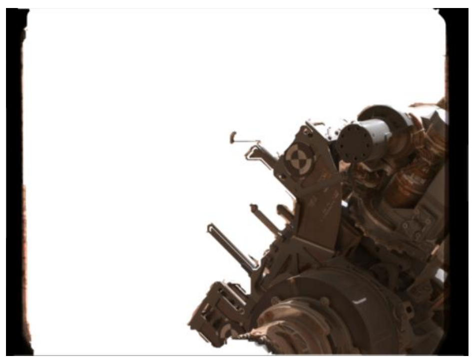


Image 3 Best Segmentation

Image 4:

Lastly, the best segmentation for image 4 was found when hierarchical merging was used on an average color RAG, with felzenswalb over-segmentation. The values used were Scale = 2.5, Sigma = 2, and Min Size = 15000 for felzenszwalb, and threshold = 0.5 for hierarchical merging. This lost very little detail of the rover to the soil, however some soil was still included as the rover.

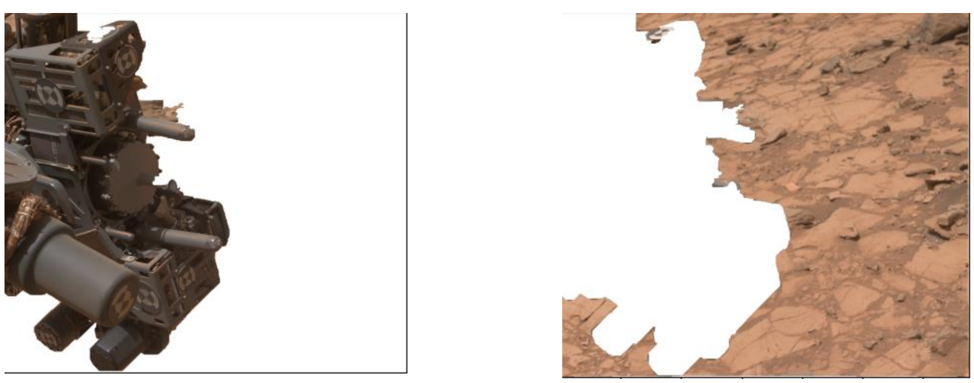


Image 4 Best Segmentation