Graphics Programming
Assignment 06

Josh Stallings

October 27, 2021

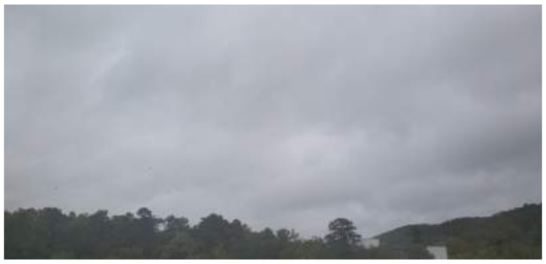

Figure 1: Image 1

1 Introduction

Seam Carving is an algorithm that removes 'seams' from an image in order to resize it in such a way that prevents distortion of the image as much as possible. This is done by creating an energy map, which is an image that highlights the important areas of an image and darkens the unimportant areas. The algorithm works by removed the darkest, so the least important, seams first. This way, the shrunk image remains as close to the original as possible. Seam Carving is useful for when content-aware resizing of an image is necessary. Creation of the energy map can also be used to remove certain objects from an image while keeping the rest intact.

2 Procedure

2.1 Energy Map

An energy map can be created in a variety of ways. Here, the energy map was created by using a Sobel filter to find the edges of the image. After that, the edge map is then eroded and padded with an extra set of pixels on each side of the image. Below is a Python-coded example of the energy map function.

```
def energyFilter(edgeFilter):
    I = edgeFilter
    h = I.shape[0]
    for j in range(h-1):
        row = I[j:j+1]
        row = cv2.erode(row, kernel)
        I[j+1:j+2] += row

    return np.pad(I,([0],[1]),constant_values=I.max()+1)
```

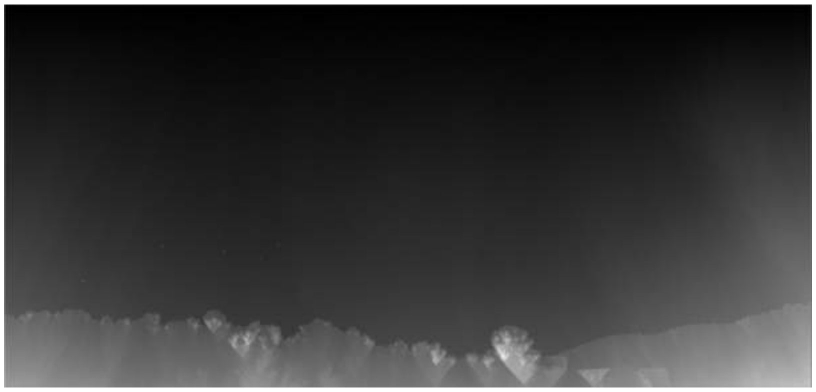

Figure 2: Energy Map

2.2 Finding a Seam

A seam should be carved through the least expensive part of the image; in other words, the smallest value pixels should be removed. The seam is found by starting at the bottom of the image and finding the minimum value pixel, and the next pixel to be deleted is the minimum of the three pixels directly above that one.

```
def findSeam(img, energyMap):
    seam = []
    h = img.shape[0]

    y = h-1
    x = np.argmin(energyMap[y])

    seam.append(x)
    while y:
        y -= 1
        x += np.argmin(energyMap[y, x-1:x+2])-1
        seam.append(x)
    return seam
```

2.3 Deleting a Seam

To delete the seam from the image, the part of the image to the right of the seam has to be shifted over one. This will reduce the image dimension, either height or width, by one. Deleting is similar to finding the seam; starting at the bottom row, each pixel to the right of the pixel to be deleted is shifted over one. That is done with the following code.

```
    h = out.shape[0]
    y = h-1
    while y:
        x = seam[y]
        if y == h-1:
            out[y, x:-1] = out[y, x+1:]
        else:
            out[y, x-1:-1] = out[y, x:]
        y -= 1
    out = out[:,:-1]

    return out
```

3 Assignments

3.1 Finding Seams

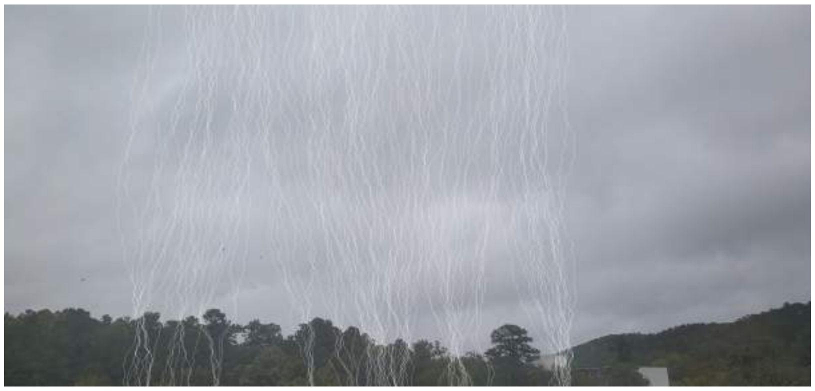

Figure 3: Showing 100 vertical seams

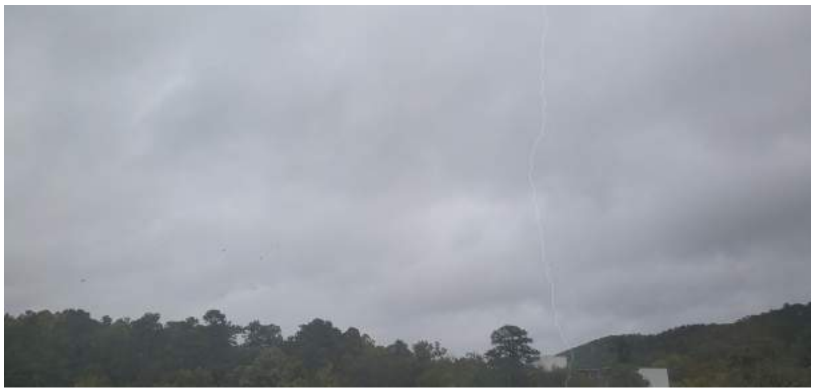

Figure 4: Vertical Seam in Image 1

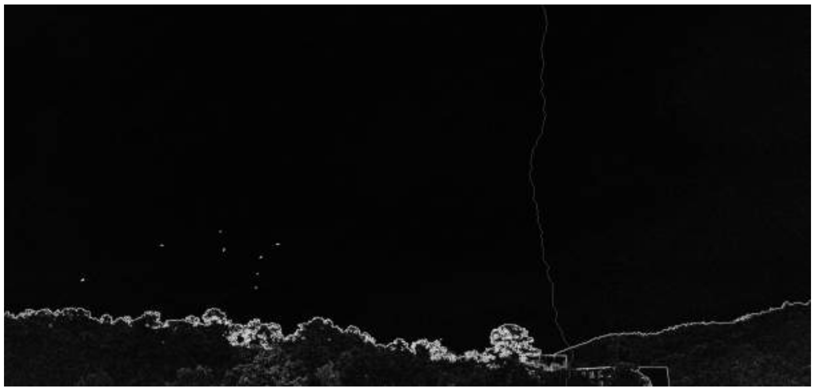

Figure 5: Vertical Seam in Edge Map

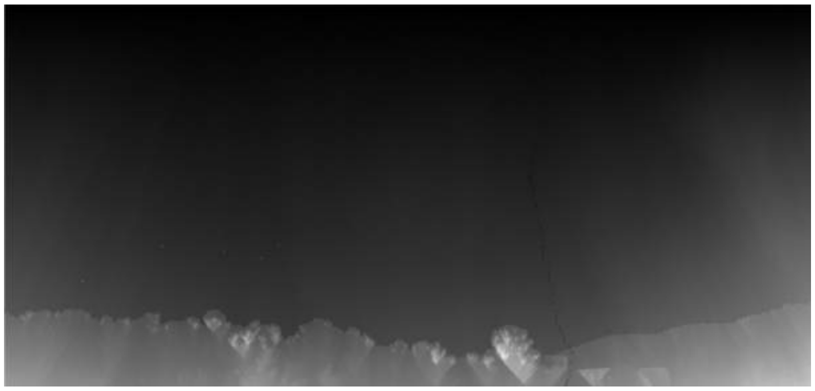

Figure 6: Vertical Seam in Energy Map

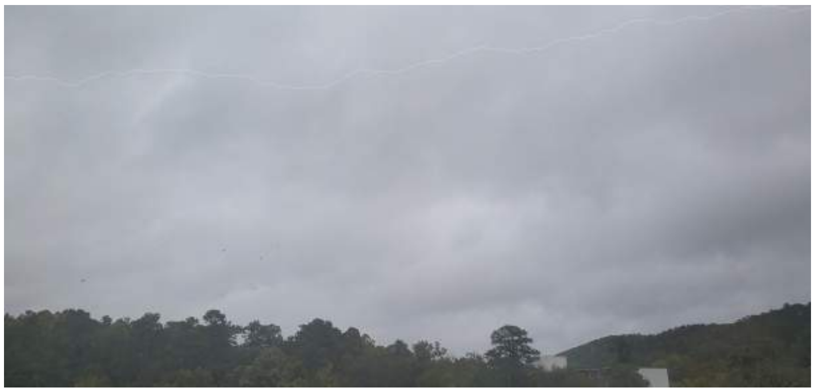

Figure 7: Horizontal Seam in Image 1

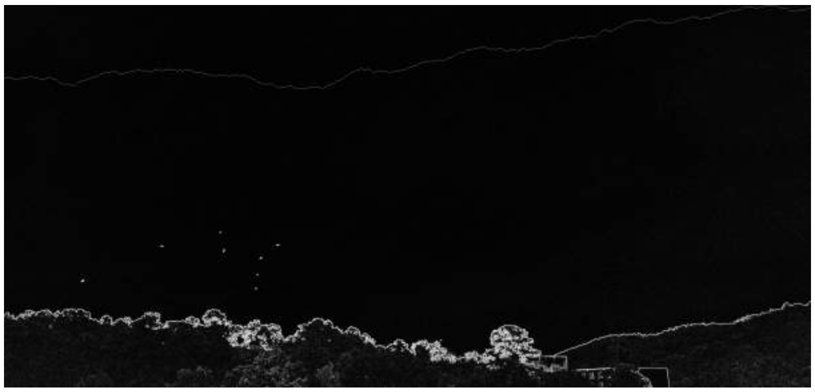

Figure 8: Horizontal Seam in Image 1 edge map

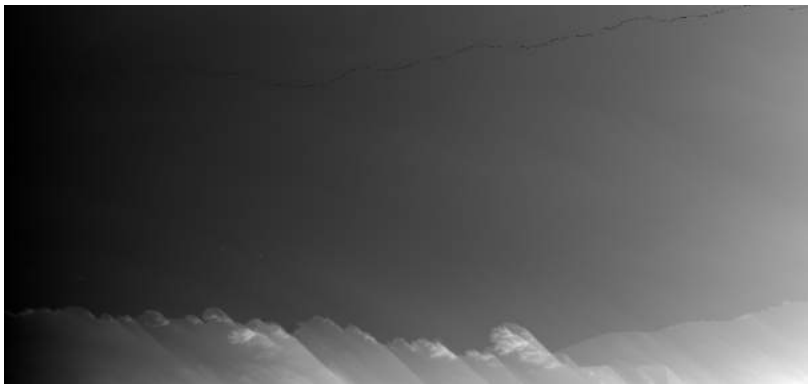

Figure 9: Horizontal Seam in Image 1

3.2 Removing Seams

After the seam is found, it can easily be removed. Many seams can be removed through looping, and certain image dimensions can be achieved by removing enough seams.

Figure 10: One vertical seam removed from Image 1

Figure 11: One horizontal seam removed from Image 1

Since only one seam was removed, these could just be the original images. You have no way of knowing unless you have the files :). To prove it actually works here is the image but with one hundred seams removed.

Some artifacting can be noticed in the vertical seam image. That is a result of the image not being the best for seam-carving, because everything along the bottom is relatively important to the image.

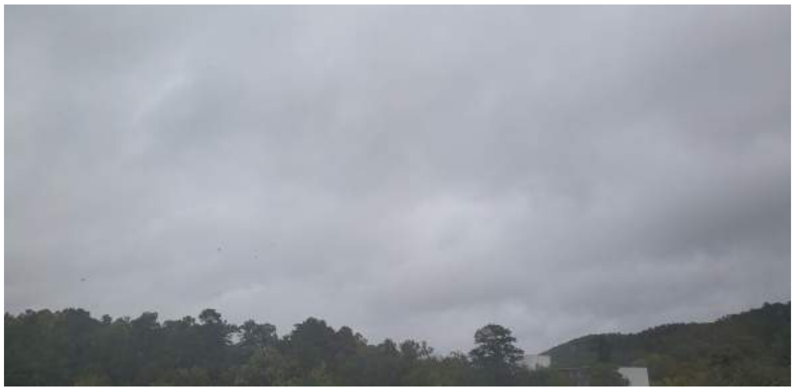

Figure 12: One hundred vertical seams removed.

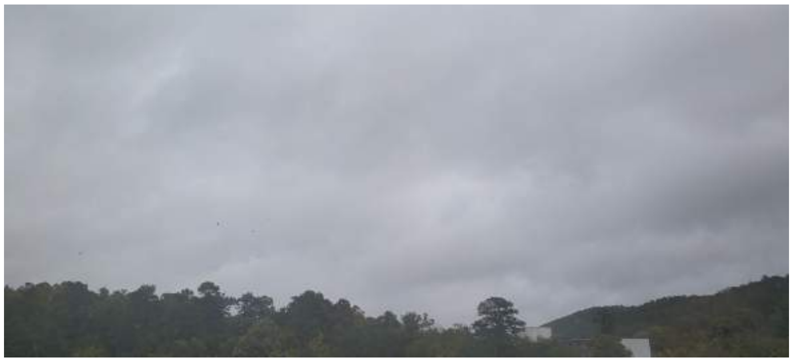

Figure 13: One hundred horizontal seams removed.

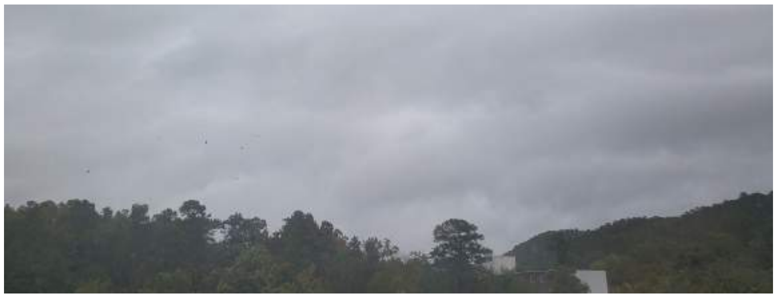

Figure 14: Image 1 resized from 4032 x 1908 to 3200 x 1200

3.3 Resize

Horizontal and vertical seam deletion can be used in combo to attain a specific image size. The results of this may very, as not every image is created equal in the way it is seam carved. Image 1 was originally 4032 x 1908.

4 Conclusion

In conclusion, seam-carving is a smart content-aware method to resizing images. In the future it would be interesting to tackle different methods of producing energy maps, as this would yield different results. It'd also be useful to have some sort of method of knowing whether it would be better to use different types of energy maps in different situations.