# Seam Carving for Content-Aware Image Resizing Recreation

By Josh Adams

CS 6475 Computational Photography

Georgia Tech

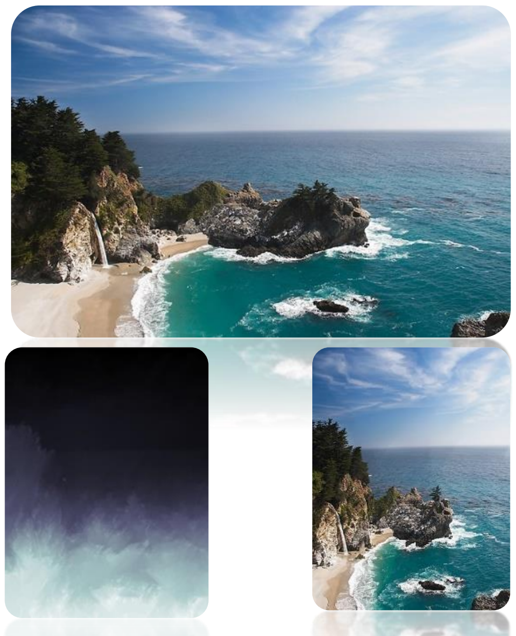
## Abstract

Seam carving is one of the various methods used for content-aware image manipulation. One of the benefits of content-aware manipulation is the ability to keep the information lost relatively low while introducing few artifacts. Seam carving does have limitations, such as an images expansion and reduction is soft limited at least with my implementation to about a 50 percent change. The reason it is a soft limit is that, some images can be modified beyond that limit with very little artifacting, while others may not even be able to reach that threshold. I will be going over the processes I went through to recreate the seam carving algorithm discussed by Shai Avidan and Ariel Shamir.

## Process

*Energy Map:* My implementation of the seam carving algorithm starts by filtering over the image using a sobel filter both in the x and y directions. I then add together the absolute values of the returned x and y arrays. This process ultimately generates my energy matrix which will be used for further processing of the images. The values in the returned array closely corresponds to the overall importance to the image. The higher the value the more important or significate the pixel at that location is.

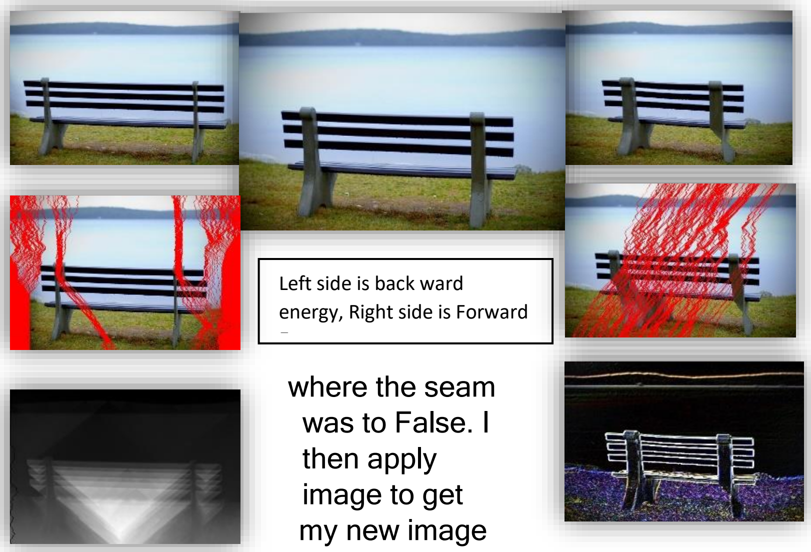
*Seam Path Array:* In the beginning, obtaining the Seam Paths, was the most time-consuming part of my code. I started first by vectorizing a simple function using NumPy and process the arrays this way.


(Top) Figure 5 (2007) – Original, (Bottom-Left) Energy Map of Final Image, (Bottom-Right) Final Image

This was due to the fact I had not implemented this function using dynamic programming. After I changed over to dynamic programming, the performance increase was quite impressive.

*Get Seam:* This function takes the Path array, starting from the bottom of the array we find the index of the smallest value. From that index we work towards the top of the array, by progressively moving upwards to the lowest value in a specified neighborhood.

*Insert/Remove Seam:* Based on whether we are wanting to reduce the size of the image or expand it, this function will behave differently. For remove I create an $NxN$ Boolean matrix with all values set to True at first. I then use an array of values passed into the mask to change all values


Left side is back ward energy, Right side is Forward

where the seam was to False. I then apply image to get my new image with one less dimension because of the removal of the row. For enlarging the image, I have not found an efficient solution, or one that doesn't introduce artifacts into the image. In my implementation I iterate over the rows in the image, I split that row at the point where

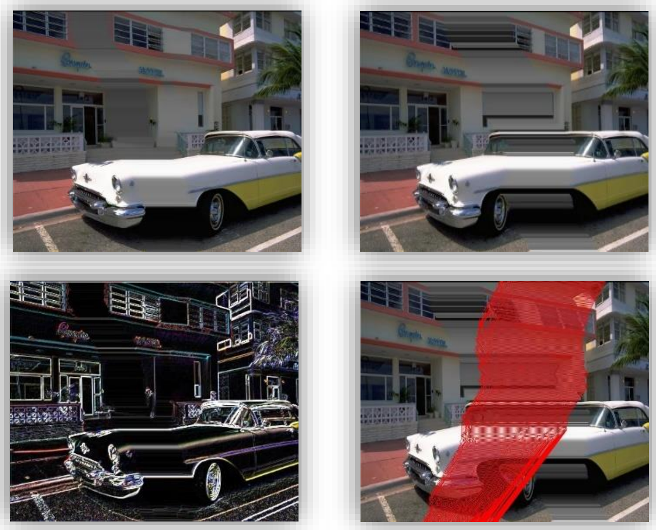


Fig9 (elongated car) – Left side images are of backward energy. Right side images are forward energy

the seam is to be. I fill that seam with an average of the two adjacent pixels. This approach does allow me to enlarge the image, but the results are not appealing.

*Forward Energy:* Is a method to retain more energy within the image, than previous methods, such as backwards energy. With forward energy we look at the effects of removing the seam. The reason we do this is that when we remove a seam the pixels which were adjacent to the seam end up next to each other. This can have more of an impact on the overall energies of the image, than having just left the seam in place.

## Reproductions

(Fig5-2007) The beach scene was the easiest to execute. I believe my result is very close to that of image in the report.

(Fig8-2007) The Dolphin. This expansion did not work very well. I did not see a sound solution for populating the newly added pixels. More research is needed.

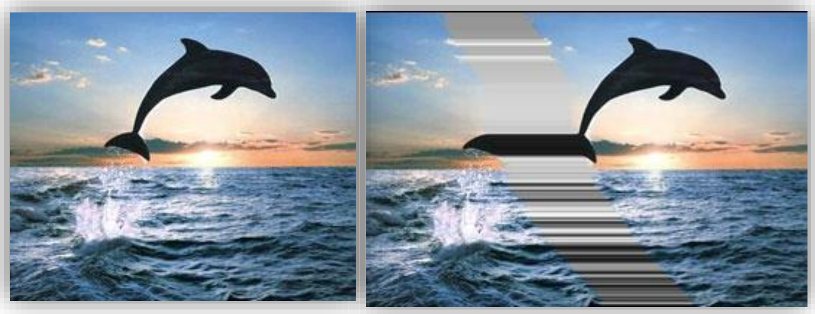
(Fig8-2008) The park bench turned out well. My seams did not go exactly where they did in the paper but that could be a result of how I calculated my energies.

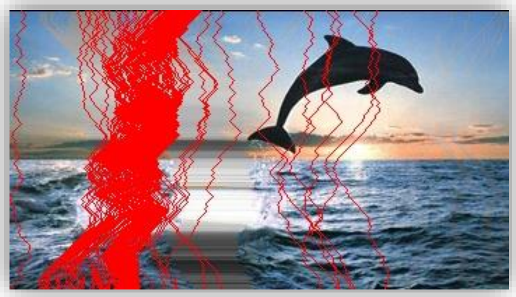
(Fig9-2008) The car elongation. This did not turn out well. I personally really like how the black masked image turned out, but overall the elongation as a failure. Forward Energy(right side) performed worse that backwards energy.

## Difficulties

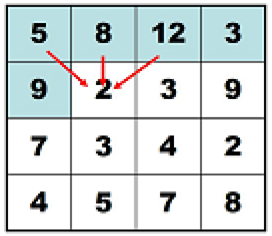
There were three main challenges I faced when attempting to recreate the seam carving and forward energy algorithm. The first was that I did not understand the seam carving mathematical equation.

$$M(i,j) = e(i,j) + \min(M(i-1,j-1), M(i-1,j), M(i-1,j+1))$$





I searched for more information on this and eventually I found “Seam-Carving and Content-Driven Retargeting of Images (and Video) by Michael Rubinstein. While reading over Michaels document, an image of a matrix is what finally helped me understand the equation.



| | | | |
|---|---|---|---|
| 5 | 8 | 12 | 3 |
| 9 | 2 | 3 | 9 |
| 7 | 3 | 4 | 2 |
| 4 | 5 | 7 | 8 |

The second part of this paper was dynamic programming. I personally am not incredibly familiar with dynamic programming. The reason this was difficult was the way dynamic programming was thrown in there. Since I was not the most familiar and the paper stating, ‘The optimal seam can be found using dynamic programming’ or ‘we computer T using dynamic programming’, I was not sure how to apply dynamic programming to this problem. I overcame this by using numpy and its many tools for array manipulation.

The last part I found to be difficult to recreate was image enlarging. My implementation introduces many artifacts into the image, and I have not been able to resolve this issue. I have been able to significantly reduce those artifacts, but many persist. I have not been able to track down the cause of the issues. In the paper it discusses taking the K seams for removal to be used in place for the expansion. I would have liked more information on this process because when I tried this my seams started to focus in one area.

## Video

https://youtu.be/ajF9xHvaW1c

## References

Shai Avidan and Ariel Shamir. *Seam Carving for Content-Aware Image Resizing*

Michael Rubinstein, Ariel Shamir and Shai Avidan. *Improved Seam Carving for Video Retargeting*

Michael Rubinstein with MIT. *Seam-Carving and Content-driven Retargeting of Images (and Video)*

Seam Carving by Xiaofeng Tao (taox)

Numpy

OpenCv