Seam Carving for Content-Aware Image Resizing Research Replication

Dylan Fiedler
Georgia Tech University
CS6475 - Research Replication

Abstract

This goal of this paper is to replicate the results of the previously published **Seam Carving for Content-Aware Image Resizing** by **Shai Avidan** and **Ariel Shamir**.

1 Introduction

A supplementary video of this project can be found at

https://drive.google.com/file/d/1ynH6MJJo_gcsSbySdE_U4JD9nsmXXR2F/view?usp=sharing

1.1 Background

In their original work, **Shai Avidan** and **Ariel Shamir** displayed how to properly adjust the size of images using a technique referred to as *seam-carving*. In this paper I attempt to recreate the results of their work using similar methods and formulas while sharing my learnings and mishaps.

1.2 The Operator

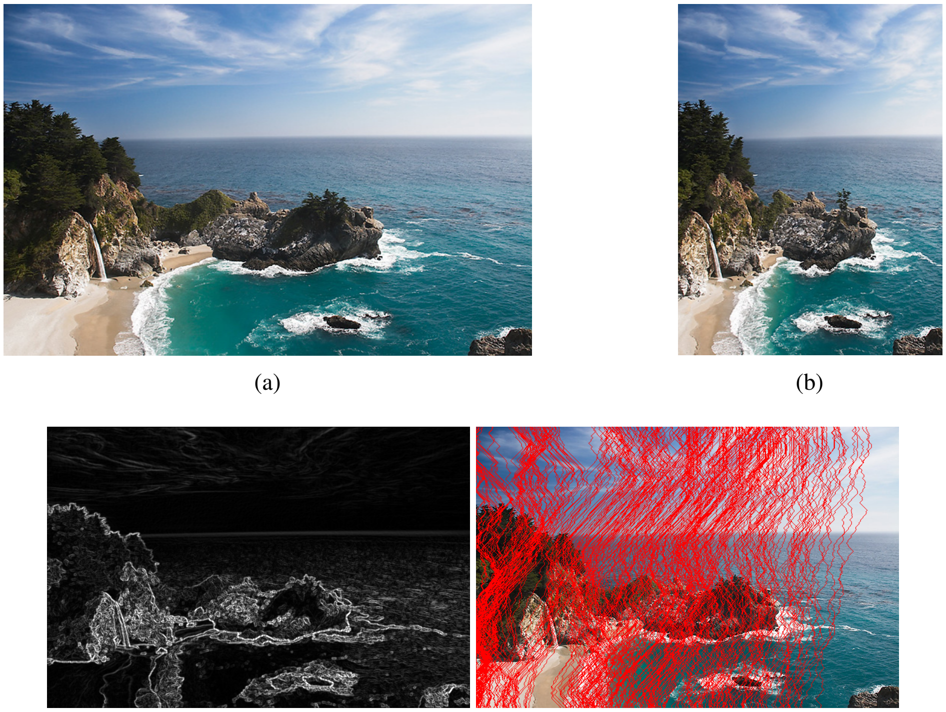
In order to resize images in a context-aware manner, we first must determine what pixels should be removed from the original image. To do so, we must first generate an energy map of the image and than use that function to remove unnoticatble, low-energy, pixels from the image. In their original research **Seam Carving for Content-Aware Image Resizing** by **Shai Avidan** and **Ariel Shamir** used the following simple energy formula

$$e_1(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|$$

Conversely, in this paper we used a Sobel gradient energy function with a 3x3 kernel to produce the energy maps that you see in Figures 1 and 2.

2 Image Reduction

Given any energy function we can now reduce the width of an image. We could take varying approaches such as removing all the lowest energy pixels from the image or by removing the lowest energy columns in the image. However, with those approaches we would eventually destroy important (high-energy) pixels from the image or create artifacts that fail to maintain the main content of the image.



(a) (b)

(c) (d)

Figure 1: Comparing the original image (a) with the compressed image (b). Figure (c) shows the entropy energy map of the image while (d) shows the lowest energy seams that were removed from the original image.

Instead, and similar to Avidan and Shamir, we determined the *vertical seams* with the lowest cumulative energy in the image and then removed those seams. If our image **I** is *m x n* than each seam will be composed of *n* pixels. In order to find the lowest energy seams we must leverage dynamic programming. For each row in the image, we determine the cumulative energy of for all possible paths for each point (i, j) using the formula

$$\begin{aligned} M(i,j) \quad &= \quad e(i,j)+ \\ &\quad\quad \min(M(i-1,j-1),M(i-1,j),M(i-1,j+1)) \end{aligned}$$

What this formula does is take the energy of the current pixel, then determine the energy for the three neighboring pixels that are below the current point (i, j). It then adds the location of the pixel with the lowest energy (amongst those three) and adds it to the current path, eventually traversing down the entire image and creating a *v*ertical seam.

Once we have the energy for every seam in the image, we then remove the required number of seams that have the lowest total energy. By doing so, you maintain the rectangular shape of an image, while removing only seams that have the lowest total energy, making their removal almost unnoticeable. In Figure 1(c) you can see the energy map that was produced, and in Figured 1(d) you can see the lowest energy seams that were identified and removed, resulting in the final image 1(b).

It is important to note that if you were to reduce an image by a large amount you would eventually create artifacts in the image as eventually higher energy seams would have to be removed.

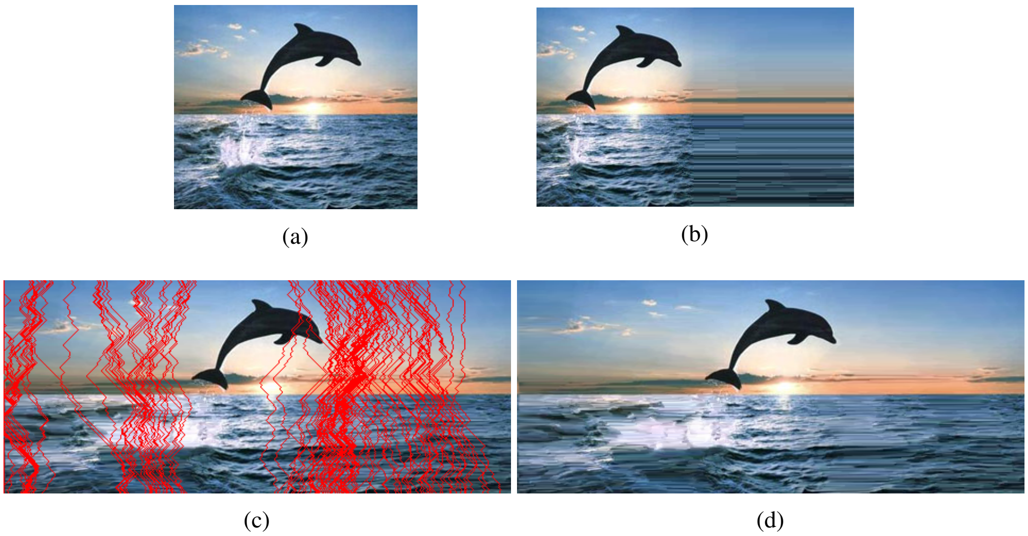


Figure 2: Comparing the original image (a) with the expanded image (d). Figure (b) shows what happens you add the lowest energy seam repeatedly and figure (d) shows the lowest energy seams that were removed from the original image.

3 Image Enlarging

Similarly, when expanding the size of the image, we want to only expand the image by the lowest energy seams to prevent altering high energy contents of the image. However, when expanding you cannot just simply add the lowest energy seam to an image, and repeat as we did when reducing an image. Doing so, you will repeatedly find the same seam in the image creating a stretch artifact you can see in Figured 2(b). Instead, we need to create a temporary copy of the original image, perform an image reduction of that image (following the process in Section 2) and while doing so collect all of lowest energy seams that were removed. Then, we added that collection of low-energy seams to our original image. It is important to note that when increasing an image by more than 50% you may create unwanted artifacts. Figure 2(d) shows two steps of 50% expansion resulting an image that is twice as wide as the original.

Unlike Avidan and Shamir, our expansion did not work as smoothly as we had hoped. The seams that were used in the expansion, as seen in Figure 2(c) do not match the seams from their work. This is likely due to some miscalculation in determining where to add the lowest energy seams in the image due to the fact that you need to replace existing seams when adding new ones.

4 Conclusion

Overall, I was unable to replicate the exact results that Avidan and Shamir were able to produce, but I did get close. The time in which it took to resize my images was drastically longer (up to 10-15 seconds) and the seams I identified do not exactly match the ones they shared. While their research was incredibly helpful, I did find their explanations and implementation techniques to be fairly vague particularly when inserting seams. I was also able to resize the height of an image by first rotating the image, then applying the same techniques described earlier, and rotating the image back upon completion.

References

[1] Shai Avidan , Ariel Shamir, (2007) Seam carving for content-aware image resizing, *ACM Transactions on Graphics (TOG), v.26 n.3*, July 2007