Seam Carving
CS766 Project Midterm Report

April 2, 2018

Name	Net ID
Qian Zhang	`qzhang348`
Wolong Yuan	`wyuan32`

Table 1: Group Members

Abstract

We have implemented seam carving operator that supports content-aware image resizing for both reduction and expansion, which can prevent the important content of the image from being distorted when resizing and outperforms traditional methods such as cropping and scaling. Also, we have used seam carving for other applications such as content amplification and object removal. Further, we will implement improved seam carving operator with a forward-looking energy function that measures the effect of the seam, and use the new measure in the dynamic programming algorithm for image resizing, and in graph cut algorithm for video retargeting.

1 Introduction

1.1 Seam Carving

Seam Carving is effectively used in image resizing, image content enhancement and object removal. The traditional image resizing method generally only considers geometric constraints of the image and can't meet new demands. We want to keep the image's notable part still significant after image resizing, so we need to take the image content into consideration. Avidan and Shamir's paper[1] presented a simple image operator called seam carving that supports content-aware image resizing for both reduction and expansion. In our project, we have re-implemented this paper and explore different applications of seam carving. Besides, we would like to extend this to video or other kinds of data, such as webpages.

1.2 Motivation

Seam Carving can be widely used in our real life. For example, many mobile apps choose to display thumbnail images before users click and see the full-size images. A thumbnail image owns all image information but usually make it hard for users to perceive the key part of the image because of the reduced size. With seam carving, the significance of key part of an image can be maintained and users can get that more easily. Besides, the results of this project are intuitive and comparable, which will help us examine the performance of our implementation. Finally, this project is extensible. We would like to extend our implementation to other kinds of data in future work.

2 Related Work

2.1 Seams Carving for Image Editing

Avidan and Shamir's work[1] presented an operator for content-aware resizing of images using seam carving. Seams are computed as the optimal paths on a single image and are either removed or inserted from an image, which can minimize the seam cost or energy lost. The optimal seam can be found using dynamic programming and they have shown seam carving is more effective than other strategies of resizing, such as cropping, scaling etc. This operator can be used for a variety of image manipulations including: aspect ratio change, image retargeting, content amplification and object removal, and can be easily integrated with various saliency measures, as well as user input, to guide the resizing process.

2.2 Seams Carving for Video Retargeting

Michael, Avidan and Shamir's paper[2] extend the work of Avidan and Shamir[1] and proposed a graph based approach to seam carving, which can handle video retargeting. This extension defines 2D surfaces to be removed from the 3D video cube. The intersection of the surface with each frame defines one seam in this frame. Hence, removing this manifold removes, in effect, one seam from each video frame. On the one hand, because the surface is flexible, the seams can change adaptively over time in each frame. On the other hand, because the surface is connected, the seams preserve temporal coherency. They employed graph cut to obtain the optimal manifold to remove and demonstrated the effectiveness of this method on several images and video sequences.

3 Algorithms

3.1 Definition of seam and image reducing

The main idea is to remove unnoticeable pixels to blend with surroundings. To do so, we firstly need to define energy function of image $\mathbf{I}$, which is defined as follows.

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

But we can't just globally remove pixels with lowest energy regardless of their positions, or remove pixel with least amount of energy in each row to resize image, because that will destroy the rectangle shape of the image or destroy the image content. So instead, to resize the image without artifacts mentioned above, we define and remove seams from image. Take vertical seam for example, let $\mathbf{I}$ to be n×m image and define a vertical seam to be:

$$s^x = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n,\ s.t.\ \forall i, |x(i) - x(i-1)| \leq 1$$

where x is a mapping $x : [1, ..., n] \to [1, ..., m]$. That is, a vertical seam is an 8-connected path of pixels in the image from top to bottom, containing one, and only one, pixel in each row of image. Denote seam s^x as s, the pixels of the path of seam s will be $\mathbf{I}_s = \{\mathbf{I}(s_i)\}_{i=1}^n = \{\mathbf{I}(x(i), i)\}_{i=1}^n$. Given an energy function e, we can define the cost of a seam as $E(s) = E(\mathbf{I}_s) = \sum_{i=1}^n e(\mathbf{I}(s_i))$. Then if we want to reduce width of image, we just look for optimal seam $s^* = \min_s E(s)$ that minimizes this seam cost to remove.
The optimal seam can be found using dynamic programming. The first step is to traverse the image from second row to last row and compute the cumulative minimum energy M for all possible connected seams for each entry (i, j):

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

At the end of this process, the minimum value of the last row in M will indicate the end of the minimal connected vertical seam. Hence, in the second step we backtrack from this minimum entry on M in last row to find the path of optimal seam, and then remove the optimal vertical seams iteratively to reduce width of image, and the process to reduce height of image is similar.
This algorithm can be integrated with various saliency measures, for example, user can mask image to weight target region with large positive value, and prevent pixels in target region to be removed.

3.2 Image enlarging

Inserting seam to enlarge image can be thought of as inversion of seam removal. For example, to enlarge width of image by k, we find k vertical seams for removal, and duplicate them in the same order when removing. The inserted 'artificial' vertical seams are derived by averaging them with their left and right neighbors. The process to enlarge height of image is similar. And also we can use mask image to guide image enlarging process.

3.3 Content Amplification

Seam carving can be used to amplify the content of image while preserving its size. This can be achieved by combining seam carving and scaling. To preserve the image content as much as possible, we first use standard scaling to enlarge the image and use seam removal to carve the image back to its original size.

3.4 Object Removing

To use seam carving to remove object, we need a mask image of target object and weight the energy value of pixels covered by mask image with a negative value, and then we can calculate the smaller of the vertical or horizontal diametersof the target removel region and perform vertical or horizontal removels accordingly. And then enlarge the image to its original size by inserting seams.

4 Experiments and Conclusions

We have created a website (backed by Django framework) that provides an online tool to do seam carving. Users can upload images and select the operation they want, then the server will handle the request and respond with the result. The user interface of this tool is shown in Figure 1. Some experimental results are also given below.

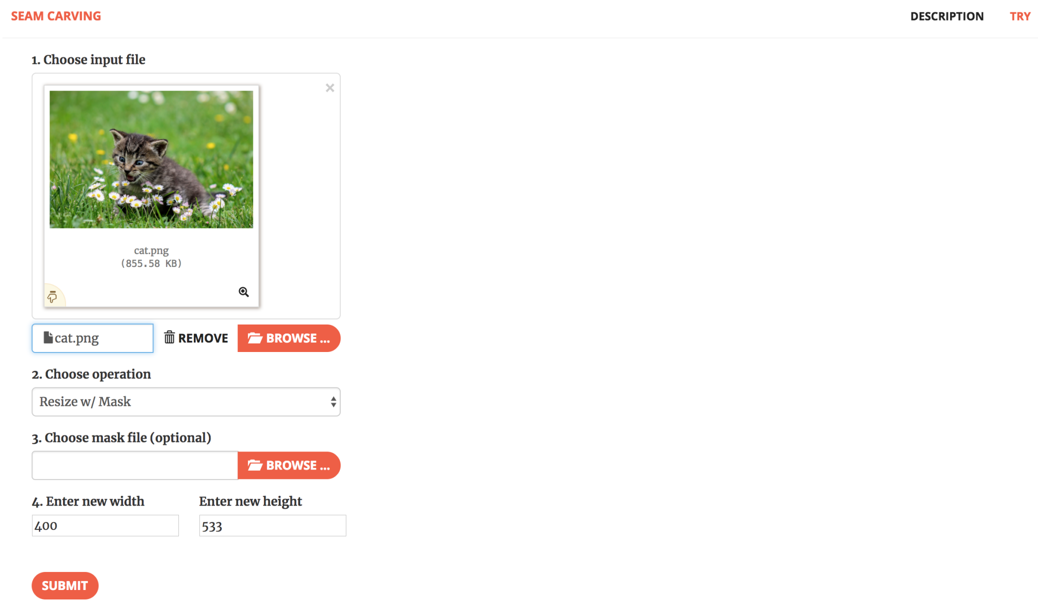

Figure 1: Website user interface

4.1 Resize without Mask

In this test, Figure 2 is the original image, Figure 3 is the reduced image and Figure 4 is the expanded image. We can find that the resizing operation only influences the parts that are not very significant (sky and cloud) and the buildings and skyline, which are more important, are kept well.

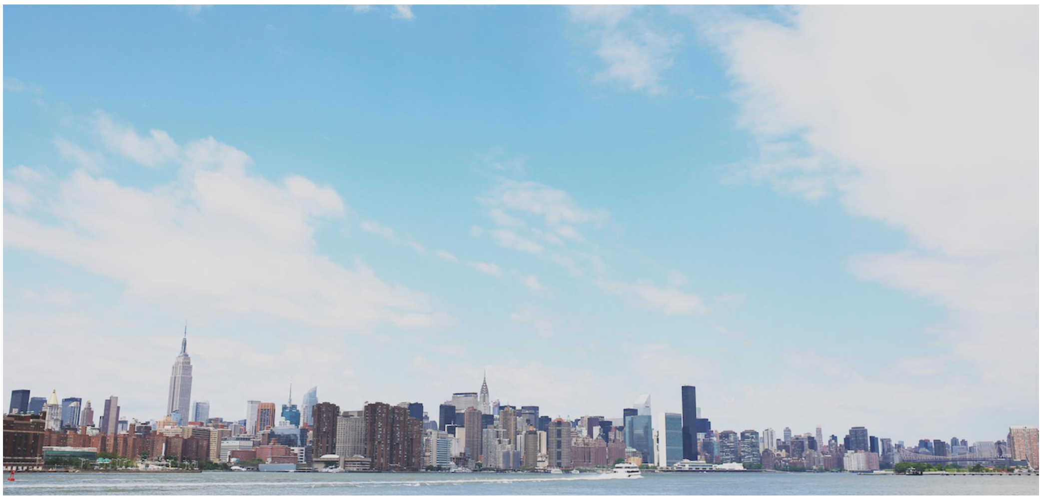

Figure 2: Skyline(original): 1280*607

The resizing operation supports the change of both width and height. Figure 5 shows the result of resizing on two dimensions.

Figure 3: Skyline(expanded): 1280*800

Figure 4: Skyline(reduced): 1280*450

Figure 5: Flower. Left(original): 800*533, Right(resized): 600*500

4.2 Content Amplification

Content amplification operation can be achieved by combining seam carving and scaling. The original image and corresponding amplified result are shown as Figure 6 and Figure 7. We can find that the main parts of the image (cat and flowers) are amplified while the image size doesn't change.

Figure 6: Cat(original): 800*541

Figure 7: Cat(content amplified): 800*541

4.3 Resize with Mask

We can also resize with a protect mask to avoid distoring the important part of the image. Figure 8 is the original image. Figure 9 is the mask we get using grabcut. Figure 10 is the result of the resizing operation with protect mask. We can find that the bus (mask) is kept well because of the protect mask.

Figure 8: Bus(original): 800*533

Figure 9: Bus(mask obtained with grabcut[3])

Figure 10: Bus(resized with protect mask): 400*533

4.4 Object Removal

We can do object removal with an object mask. In this experiment, the original image is the same as Figure 8 and the object mask is the 'bus' shown in Figure 9. The result of our object removal operation is as Figure 11 and the bus is removed completely.

Figure 11: Bus(object removed): 800*533

5 Further Milestones

Milestone	From	To
Implement improved (parallel) seam carving operator	4/2	4/8
Implement seam carving for video retargeting	4/9	4/15
Complete and improve project website	4/16	4/22
Prepare for the presentation	4/23	4/29

Table 2: Plan and Time Table

References

[1] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *ACM Transactions on graphics (TOG)*, volume 26, page 10. ACM, 2007.

[2] Michael Rubinstein, Ariel Shamir, and Shai Avidan. Improved seam carving for video retargeting. *ACM transactions on graphics (TOG)*, 27(3):16, 2008.

[3] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.