Content-aware seam carving for image resizing

Liam Kelly
Electrical Engineering
Stanford University
Palo Alto, USA
kellylj@stanford.edu

Zhongjie Li
Electrical Engineering
Stanford University
Palo Alto, USA
jay2015@stanford.edu

***Abstract*—In this paper, we look into Seam Carving as a content-aware image resizing algorithm, along with its shortcomings when it comes to increasing image sizes. Seam carving is used to find the image regions with lowest importance. By inserting seams in those region, we can increase image size without causing distortions. In this project, we implemented 3 different algorithms to change images aspect ratios. The energy function was also optimized by using extracted edges. The background noise has been significantly reduced. Overall the algorithm showed good results with no distortions or artifacts.**

I. Introduction

With the rapid growth of display industry, images are being viewed in displays with very different aspect ratios, ranging from 1:1 to 4:1. The traditional way of simply image scaling and cropping may cause loss of perceivable details and very noticeable defects. Seam Carving [2] is an approach to image resizing that is content-aware and reduces the amount of distortion imposed on the main content of an image. It's a method by which images can be resized by growing/shrinking using the minimum energy seam and interpolating around that. While image enlarging for seam carving has been implemented before, we are interested in looking at how to be even more cognizant of the image content when doing the enlarging step. Seam carving can result in some tearing along edges or shearing along seams that may reduce image quality. By being aware of these behaviors, we hope to design and implement an algorithm that reduces the amount of shear and tear from seam carving methods.

II. Related Work

Shai and Ariel [2] introduced seam carving method in 2007 to do content aware image resizing effectively. By carving out or inserting seams, they were able to change the aspect ratios of images. In addition, this algorithm can also be used for content enhancement and object removal.

Dong, et al [3] further optimized the seam carving by combining seam carving with scaling. It helped reduce the object distortion that seam carving may introduce on images with dense objects.

Seam Carving works by defining an Energy Operator in order to retain pixels that belong to the main content of the image while removing/adding pixels that are not the focus, like backgrounds and other less important features. A simple gradient-based energy function can be defined as:

$$e_1(\mathbf{I}) = \left|\frac{\partial}{\partial x}\mathbf{I}\right| + \left|\frac{\partial}{\partial y}\mathbf{I}\right| \tag{1}$$

where $\mathbf{I}$ is the image and x and y are the columns and rows of the image. A seam is defined by Shai and Ariel to be an 8-connected path in the horizontal or vertical direction with only one pixel in each column or row, respectively.

The energy of a seam is what determines which seam is removed or added. Given a seam s, the energy of a seam is defined as the accumulation of the energies of that seam $E(s) = E(\mathbf{I}(s)) = \sum_{i=1}^{n} e(\mathbf{I}(s_i))$. The goal of seam carving is to find the seam of minimum energy for addition/removal such that $s^* = \min_{\mathbf{s}} E(\mathbf{s})$ where $\mathbf{s}$ is the set of all seams. The original paper uses dynamic programming to find this minimum energy seam in a two-step process. The first step is to compute the minimum cumulative energy function M at each pixel (i, j). This is done for the vertical case (the horizontal case follows immediately from it) by going through each row and computing

$$\begin{aligned} M(i,j) =& e(i,j) + \\ & \min(M(i\text{-}1,j\text{-}1), M(i\text{-}1,j), M(i\text{-}1,j\text{+}1)) \end{aligned} \tag{2}$$

After all the M values have been computed for the last row, the minimum value in the last row will define the minimum seam, which can be found in the second step of the algorithm by traversing up the seam in an 8-connected path from that last pixel and getting the pixel whose cumulative energy function is minimum. With this seam, pixels can be removed to make an image smaller, or they can be added using an interpolation function to grow the image, as we do in our work. To add k rows or columns, the original paper finds the minimum k seams in the respective direction, and then adds interpolated pixels along each of those seams in order.

III. Objective and Scope

Seam carving has limitations, particularly when it comes to aspect ratio changes as a part of growing images. The repeated insertion of seams can cause image shearing. An example of this is shown in Figs. 1 and 2. In this case, we see that the major edges in the enlarged image have sheared, resulting in a distortion in the main part of the image. This shearing is due to the pixel insertions happening irregularly, slicing different parts of the edge in various concentrations. While the aspect

Fig. 1. Original Image, 300w x 230h

Fig. 2. Seam Carving Enlarged Image, 360w x 360h

ratio of the image is indeed changed and the image has been resized directly to the desired scale, our goal is to reduce the shearing that occurs.

In this report, we will implement a seam carving enlargement algorithm to resize an image for different aspect ratios. We will compare existing base seam enlargement implementation and try various methods to reduce subject shearing. These methods include: changing the energy calculation algorithm, manual subject matter indication, and automatic edge detection. We will contrasting these method results to find the algorithm that produces the best images. We will also contrast these images with other resizing algorithms, to see the trade offs of each kind of algorithm. From these comparisons, we hope to discover trade offs based on processing time, image accuracy, sharpness, and other qualitative features that might depend on the type of application.

IV. Methods

We adapted a basic Matlab implementation of Seam Carving [1] to form a skeleton for our modifications. This implementation enlarges images by first adding horizontal seams and then adding vertical seams, based on whatever energy function we choose to utilize.

Vertical seam and horizontal seams are defined as following.

$$S^x = \{s_i^x\}_{i=1}^n = \left\{(x(i), i)\right\}_{i=1}^n s.t. \forall i, |x(i) - x(i-1)| \leq 1 \tag{3}$$

$$S^y = \{s_j^y\}_{j=1}^m = \left\{(y(j), j)\right\}_{j=1}^n s.t. \forall j, |y(j) - y(j-1)| \leq 1 \tag{4}$$

Given an energy function for an image, a seam's energy can be calculated.

$$Energy\left(s\right) = \sum_{i=1}^{n} E\left(I\left(s_i\right)\right) \tag{5}$$

To increase the aspect ratio of an image, we can either add more horizontal seams (suppose k seams to add) or remove vertical seams (suppose j seams to remove). Then the idea is to find the seam location with lowest energy. In our application, we applied 3 different algorithms based on the optimal seam location. The first algorithm looks for the top k horizontal seams with lowest energies and then adds seams at these locations with interpolation. The second algorithm looks for the horizontal seam with lowest energy and adds one seam at its location with interpolation. Update the image and loop this over for k times. The third algorithm looks for the vertical seams with lowest energy and removes it. We update the image and loop this over for j times. Then the output image is scaled to the same size as above 2 algorithms for comparison.

In the seam carving algorithm, the key is to identify important objects in images so as to avoid distortion. We use energy calculation functions to images to quantitatively characterize how important each pixel is. We studied 3 different energy functions as following.

The first energy function is the most classic one. It uses the summation of gradients along horizontal and vertical axis.

$$E_1(\mathbf{I}) = \left|\frac{\partial}{\partial x}\mathbf{I}\right| + \left|\frac{\partial}{\partial y}\mathbf{I}\right| \tag{6}$$

The second energy function uses a high pass filter and thresholding to suppress noises caused by less important objects. The idea is to ignore low value objects and only consider important objects. For example, the tiny edges in grassland may end up having a high energy value when we do sum across certain axis.

$$E_2(\mathbf{I}) = gaussian\,fiter(binarize(E_2(\mathbf{I}))) \tag{7}$$

The third energy function uses Sobel operator to identify edges first and only use detected edges as energy function.

$$E_3(\mathbf{I}) = sobel\,edge(rgb2gray(\mathbf{I})) \tag{8}$$

Fig. 3 visualized the energy functions with the example of castle picture.

One of the issues with the original implementation is that the algorithm doesn't consider the energy that it inserts into the image. The very straightforward approach to computing M and finding the minimum k seams is called a Backward Energy approach. This causes edges that were not in the original image. There is a second approach, called Forward Energy [4], that is used to account for the added energy. This algorithm looks forward at the resulting image rather than backward at

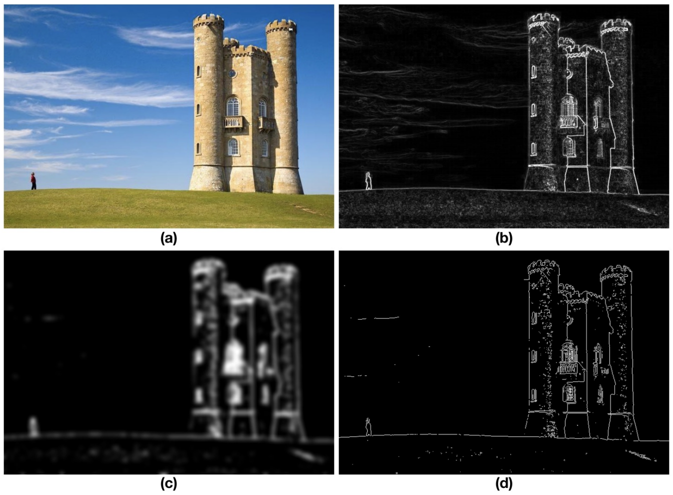

Fig. 3. Comparison of energy functions. a) original image, b)1st energy function, c) 2nd energy function, d)3rd energy function

the current energy. To do this, the local neighborhood requires a new cost function that gets added to the M matrix depending on the direction of the cut, with the cost function defined as follows:

$$\begin{aligned}
&(a)\, C_L(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j-1)| \\
&(b)\, C_U(i,j) = |I(i,j+1) - I(i,j-1)| \\
&(c)\, C_R(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j+1)|
\end{aligned} \tag{9}$$

We then add these costs into a new cumulative cost matrix M through the following steps

$$M(i,j) = P(i,j) + \min \begin{cases} M(i-1,j-1) + C_L(i,j) \\ M(i-1,j) + C_U(i,j) \\ M(i-1,j+1) + C_R(i,j) \end{cases} \tag{10}$$

where P(i,j) is an additional pixel based energy measurement, like the ones we have defined before, that can be used on top of the forward energy cost as defined. We use forward energy as a main way to compute the best seams to be added in order to reduce the amount of edges sheared as a result of the seam carving process.

V. Experimental Results

Fig. 4 shows a comparison between the above 3 energy functions with the first algorithm. The algorithm tried to add seams in the horizontal direction. We can see distortion of the castle and human in the resized images. The 1st energy function has the worst distortion on castle and human. The reason is that the grassland region on the bottom have some low magnitude gradient values. A horizontal seam's energy in grassland region would be the summation across the horizontal axis, which would be comparable to the value of a seam that goes through the castle. Basically the algorithm won't see the grassland as background. The 2nd energy function tried to remove the grassland's noise by introducing thresholding and filtering. It showed some improvement, especially on the human. Overall the 3rd energy function works the best in detecting the most important details. It extracts all the edges and only use edges to calculate energy. It successfully distinguishes important objects from background. In summary, The gradient based method introduced too much noise from the background. Edge detection based method has greatly reduced that noise.

Fig. 5 shows the results of 3 algorithms for increasing aspect ratio to 2:1 from 3:2. The 1st algorithm did very good job in preserving the important objects' aspect ratios with little artifacts. The 2nd algorithm also preserved the castle and human's aspect ratios but introduced artifacts on the right side. The reason is that this algorithm looks for the seam with lowest energy every time. Since the seam with lowest energy is always the same one, the algorithm would be adding interpolated seam at the same position for multiple times. It would introduce artifacts. The 3rd algorithm almost completely removed the grassland in vertical axis. This algorithm would work if the aspect ratio is relatively small. For this case, since the change is significant, the algorithm pretty much remove all background seams, which makes the main objects ("castle") occupies the whole picture.

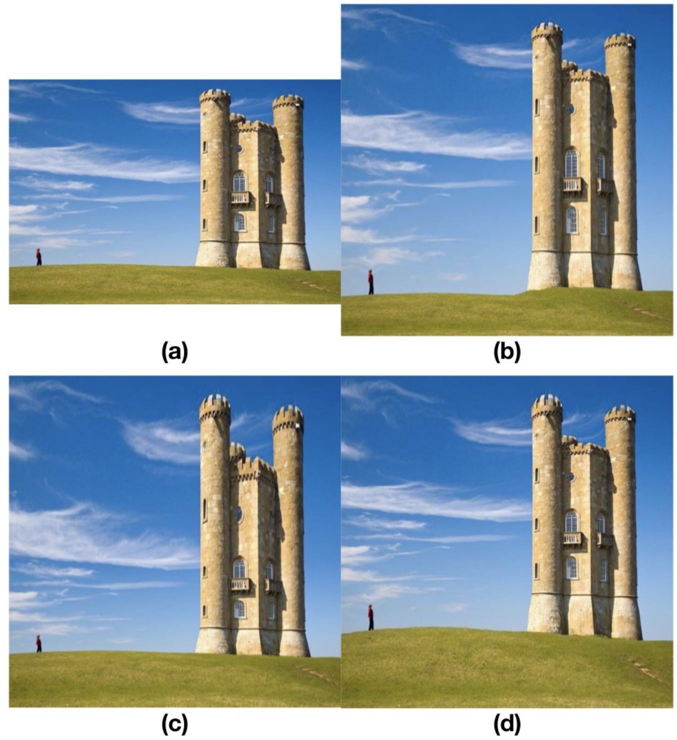

Fig. 4. Output comparison of energy functions. a)original image, b)resized image with 1st energy function, c)resized image with 2nd energy function, d)resized image with 3rd energy function

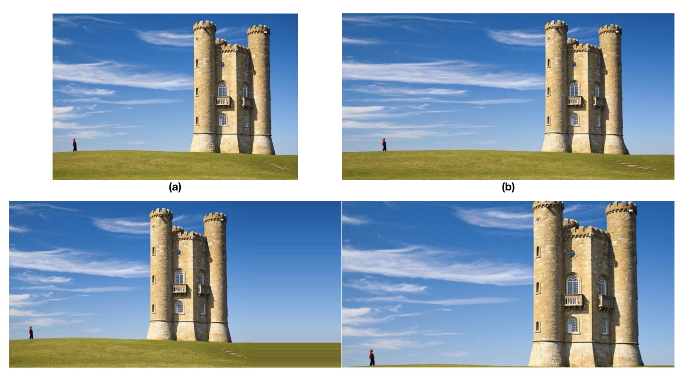

Fig. 5. Output comparison of energy functions. a) original image, b) resized image with 1st algorithm, c) resized image with 2nd algorithm, d) resized image with 3rd algorithm

Fig. 6 shows a comparison between 3 different algorithms on a very difficult task. We want to increase the image size on vertical direction but the main object ("castle") has taken majority of the space and there is not much background region in vertical direction. The first algorithm couldn't find enough seams within the background region. So it added seams within the castle and human regions. As a result, it distorted the castle and human in vertical axis. The second algorithm was able to locate the right position to add seams. It preserved objects' aspect ratios very well but introduced some artifacts in the sky. The 3rd algorithm scaled key objects proportionally so the aspect ratios of objects are well preserved but they become larger in the image.

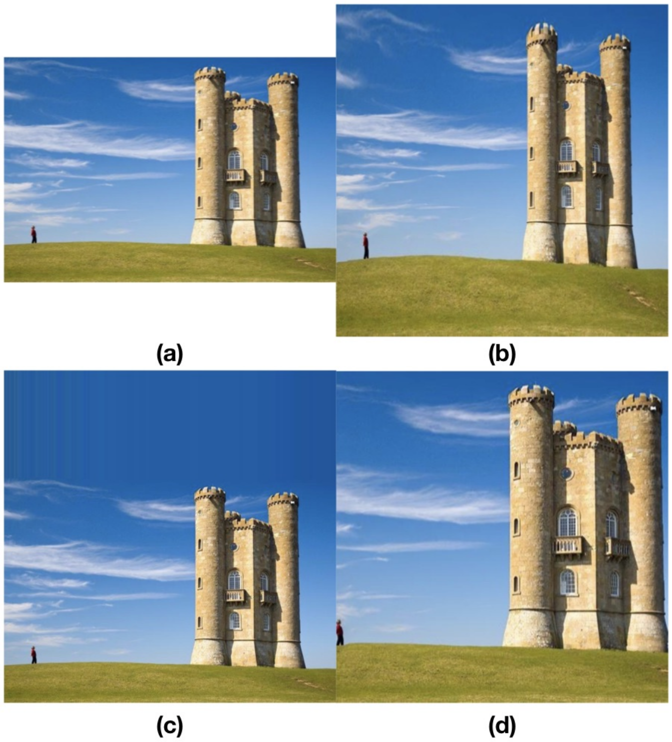

Fig. 6. Comparison of algorithm. a) original image, b) resized image with 1st algorithm, c) resized image with 2nd algorithm, d) resized image with 3rd algorithm

In terms of performance, we look at the time that it took to do each of the three algorithms in Table I. As we can see, the first and second algorithms took roughly the same amount of time, while the third algorithm took around 35% longer to complete.

Algorithm	Length of time (s)
(1)	10.55
(2)	10.84
(3)	14.28

TABLE I
TIME FOR EACH ALGORITHM TO COMPLETE

In summary the first algorithm works well on the images with clear separation between objects of interest and background. The second algorithm works better for the corner cases with very dense objects in images. The third algorithm is not recommended since it tends to remove too much background and makes the output images look strange.

VI. CONCLUSIONS

In this project we have implemented a content aware seam carving method for image resizing. With the optimized energy function, we have got nicely resized images with no visible artifacts or distortions. In addition, to improve its performance in images with very dense objects, we introduced 2 more algorithms to precisely look for the right seam locations. The results showed that our algorithms help reduced the distortions on important objects. Fig. 7 are some results of our image resizing algorithm. It shows that our algorithm was able to successfully change image aspect ratios without introducing visible artifacts or distortions.

Fig. 7. Results of seam carving based image resizing, left: original, Right: resized

VII. FUTURE WORK

There are many possible extensions to this work. The energy function can be further optimized with more complicated feature detection such as face detection. The seam locating algorithm can be improved for high speed processing with low computing power. This way it will enable this content aware image resizing for video displaying on mobile devices. It would also be interesting to combine the seam carving based method with neural networks. We can design a more complicated neural network model with much more parameters and train it with large data set to achieve a more general algorithm for most images.

CONTRIBUTIONS

Liam: Presentation slide creation, algorithm creation, presentations, paper writing and editing
Jay: Presentation slide creation, energy functions, algorithm refinement, paper writing and editing

ACKNOWLEDGMENT

We thank Prof. Bernd Girod and our TA, Jayant Thatte for providing feedback and suggestions on the project.

REFERENCES

[1] Moayad Alnammi. Implementing seam carving for image resizing, 2016.

[2] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3):10, 2007.
[3] Weiming Dong, Ning Zhou, Jean-Claude Paul, and Xiaopeng Zhang. Optimized image resizing using seam carving and scaling. *ACM Trans. Graph.*, 28(5):125:1–125:10, December 2009.
[4] Michael Rubinstein, Ariel Shamir, and Shai Avidan. Improved seam carving for video retargeting. *ACM Trans. Graph.*, 27(3):16:1–16:9, August 2008.