Content-Aware Image Resizing with Seam Carving and ResNets

Shane Segal
York University
Toronto ON, Canada
smsegal@eecs.yorku.ca

Abstract

Image resizing that respects the semantic content of images is an improvement on prior methods of Content Aware Image resizing that don't integrate higher-level knowledge of the image into their resizing processes. We present an algorithm that combines the capabilities of modern, state of the art deep learning techniques with the classical Seam Carving algorithm [2]. By combining the ability of [2] and the advanced image segmentation techniques of Mask R-CNN [3], semantically irrelevant information is removed from the image, while regions featuring subjectively important content is retained. In addition, the image enlarging capabilities of the original seam carving algorithm are enhanced by avoiding distortions of higher level features when they are present in the image.

1. Introduction

Image resizing has historically been done by cropping or scaling images, to shrink or enlarge them respectively. The introduction of the seam-carving operator by [2] allowed for the easy resizing of images while preserving spots of the image according to some energy function. The original paper used the gradient magnitude of the image as the energy function in order to eliminate areas of low change, which is a reasonable proxy for "uninteresting" parts of an image.

Content-Aware image resizing, also referred to as image retargetting, is an interesting problem as it provides quick and immediate feedback regarding the success of an approach. In addition, it is one application of a larger problem of exploring what kinds of content and patterns are visually arresting to the human visual system. The heuristic to determine human interest in the original paper is both simple and effective for many classes of image.

Seam Carving works quite well when applied to landscapes or other images where there are large patches of low and high variation. However, when applied to images with features that people are particularly attentive to (such as faces) the results can be noticeably bad and distorted. See Figure 1 for a comparison between our method and the original seam carving algorithm (reimplemented for this paper).

By combining the ability to selectively remove "uninteresting" parts of the image with the ability to detect high level features at an accuracy that can't be matched by a gradient based approach, we demonstrate adaptive image resizing that preserves faces and other high level content in a subjectively more pleasing and natural way.

2. Prior Work

Seam Carving [2] works by selecting seams of the lowest energy in an image, and continuously removing them until a target size is reached. A seam is a connected path leading from one end to the other of an image in a particular dimension. Let $\boldsymbol{I}$ be an $n \times m$ image. Using the notation of the original paper, a vertical seam is:

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

where x is a mapping of: $[1, ..., n] \rightarrow [1, ..., m]$. This is a connected path of pixels running from top to bottom of the image, with exactly one mapping between a row $x(i)$ and the horizontal location of the seam at that row. In the same way, a horizontal path consists of a mapping $y : [1, ..., m] \rightarrow [1, ..., n]$, and:

$$\boldsymbol{s^y} = \{s_j^y\}_{j=1}^n = \{(y(j), j)\}_{j=1}^n, \text{s.t } \forall j, |y(j) - y(j-1)| \leq 1 \tag{2}$$

The pixels of the seam are therefore:

$$\boldsymbol{I}_s = \{\boldsymbol{I}(s_i)\}_{i=1}^n = \{\boldsymbol{I}(x(i), i)\}_{i=1}^n \tag{3}$$

Since seams are always of $1 \times n$ or $m \times 1$, their removal from an image will cause a reduction of exactly one pixel in either the width or height of the target image.

With this idea, the optimal seam to remove from an image will be the one that minimizes the cost, where the cost is defined as the sum of the energy of the seam, where the

(a) Original

(b) Seam-Carving

(c) Our Method

Figure 1: Comparison between the original method and our own. The bottom two images are 80% of the size of the original image.

energy can be an arbitrary function. The optimal seam s^* is the one that minimizes this energy over the range of possible seams for a given dimension of the image:

$$s^* = \min_s(E(s)) = \min_s \sum_{i=1}^{n} e(\boldsymbol{I}(s_i)) \tag{4}$$

The optimal seam can be found through a bottom-up dynamic programming approach, where we can create a scoring matrix M by computing the minimum energy for all possible connected seams at a point by:

(a) Optimal Seam Insertion

(b) Seams Inserted in order of Removal

Figure 2: The result of inserting the optimal seam repeatedly and in order of removal. (Figures taken from [2])

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

After the scoring matrix is constructed, we can find the optimal seam by tracing up from the bottom and picking the minimum value of the three connected elements on the row above. The process is identical for horizontal seams, using the transpose of the original image.

After the seam is found, it can be removed from the image and the whole process can be repeated until the desired image size in the specified dimension $n' \leq n$ is reached. A simplification from the original paper is made here. The authors propose a scheme for finding the optimal removal order of seams, but the qualitative difference in the modified image was negligible. A simpler approach was taken by simply applying horizontal seam removal to an image, and then transposing the resulting image and performing the process again, until the desired size was reached in each dimension.

To enlarge an image from $n \times m$ to $n' \times m'$, it is not sufficient to simply add the optimal seam repeatedly, as this will most often insert the same seam repeatedly, causing obvious banding effects as seen in Figure 2.

To avoid this issue, it suffices to first shrink the image by $(n' - n, m' - m)$ and then save the seams in the order of removal. After the image is fully resized, you then insert the removed seams back into the original image.

The energy function is generally the gradient magnitude of the image, as the original authors experimented with different functions but all provided qualitatively similar results.

2.1. Limitations of Prior Work

The Seam Carving method described above works very well for images with clearly seperated regions of high and low energy, and especially those without faces or other features that human minds are particularly sensitive towards. An example of a good image to perform seam carving on is seen in Figure 3.

However, when seam carving is applied to images with faces or other high level semantic features, the results are often wildly distorted. This is because the idea of gradient magnitude being a perfect measure of human attention in an image is innacurate at best. An example of this can be seen in Figure 4.

Other limitations include only being able to enlarge an image by a factor of 2 as seams have to be removed first before they can be added back to the original.

2.2. Addressing the Limitations

This paper seeks to address the inability to resize images containing human images and other recognizable object categories. By utilizing the neural network defined in [3, 1], the high level feature knowledge embedded in the pretrained network weights allow our algorithm to preserve the semantic content of the images being resized. This not only allows for better shrinking of images, it also allows for better enlarging by the same strategy. We enhance the energy function by introducing the masks generated by Mask R-CNN as a measure of greater importance to the areas belonging to high level features.

(a) Original Image

(b) Shrunk by 20%

(c) Grown by 20%

Figure 3: An image with good results shown with classical seam carving.

3. Datasets

A selection of images have been taken from Flickr, which contain a Creative Commons fair use license. These images have been selected for evaluation as they present the kinds of features the original seam carving algorithm did not perform well on.

4. System Description

The seam carving system described in the original paper was entirely reimplemented in the Julia programming language. This language was chosen for its greater ease of use and speed over Matlab, and chosen over Python for it's increased speed. The metaprogramming and extensive linear algebra and image filtering/manipulation features of Julia made several aspects of the reimplementation easier and faster than an equivalent Matlab or Python implementation. The only high level image filtering features of Julia used was the image gradient calculation. Otherwise, the seam-carving portion of the algorithm has been entirely implemented based on [2]. The Mask-RCNN implementation used is written in Python with TensorFlow, and is communicating with Julia through an interface library called PyCall.

Mask-RCNN is used to calculate the mask overlays of the high level objects in the image. As there is a separate overlay for each specific mask, these are summed and normalized into a single image with pixels inside the mask being α and those outside being 1. The mask sum is computed by:

Figure 4: Seam Carving gone wrong.

$$\boldsymbol{R} = \alpha \sum_{i=1}^{N} R_i + \mathbf{1} \tag{5}$$

where $\mathbf{1}$ is a matrix of ones the same size as R_i, and α is a hyperparameter chosen experimentally to improve the preservation of semantic content in the image. Unless otherwise stated, $\alpha = 20$ for images using our algorithm.

We then element-wise multiply (sometimes known as the "Hadamard Product" denoted by $\oplus$) our gradient magnitude by $\boldsymbol{R}$ to produce our energy matrix:

$$E(\boldsymbol{I}) = |\frac{\delta \boldsymbol{I}}{\delta x}^2 + \frac{\delta \boldsymbol{I}}{\delta y}^2| \oplus \boldsymbol{R} \tag{6}$$

After our modified energy matrix is produced, the classical seam-carving algorithm is followed.

4.1. Running Time

When run without the inclusion of Mask-RCNN, our seam carving implementation runs quite fast, in

$$\boldsymbol{O}\left((n' - n)(m' - m) + \sum_{i=n}^{n'} i + \sum_{j=m}^{m'} j\right) = \boldsymbol{O}(NM) \tag{7}$$

time, where the sums are the cost of computing a seam at each image size, and $(N, M) = (n' - n, m' - m)$ meaning it is linear in the number of seams to remove.

When scaling an $(432, 768)px$ image from $[0.5, ..., 1.5]$ in increments of 0.1, we see that the execution times are what we expect, following a generally parabolic distribution, as M, N are changing at the same rate. In addition, we can expect a constant factor to be added to the execution time of growth images, as they must first remove, and then add back the seams to the image. See Figure 5.

The addition of the RNN to the algorithm contributes greatly to both the subjective quality of the results, and unfortunately the execution time. However, this appears to be a constant factor and not a time complexity increase.

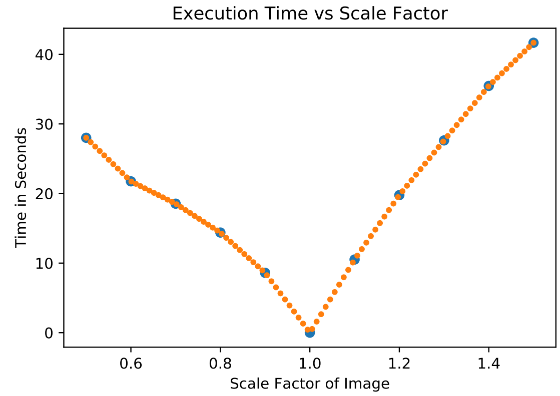

Figure 5: Execution Time Vs Scale Factor for basic Seam Carving

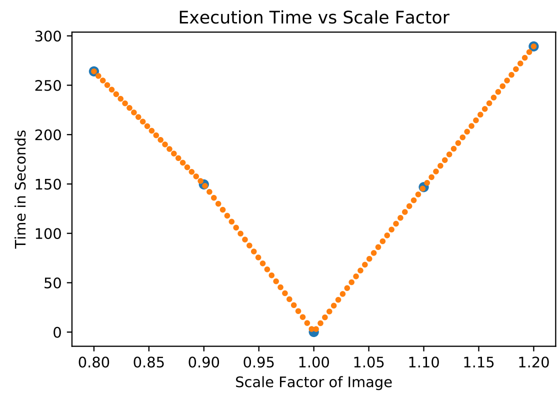

Figure 6: Execution Time Vs Scale Factor for Mask-RNN and Seam Carving

Scale	Execution Time (seconds)
0.5	28.008
0.6	21.751
0.7	18.539
0.8	14.389
0.9	8.6043
1.0	0.0024
1.1	10.512
1.2	19.771
1.3	27.629
1.4	35.464
1.5	41.666

Table 1: Execution time and scale factors for resizing an image without Mask-RCNN

As can be seen in Figure 6, when plotted from a scale of $[0.8, ..., 1.2]$ on the same image, we see that the execution time is much larger.

Scale	Execution Time (Seconds)
0.8	264.049
0.9	149.573
1.0	0.09335
1.1	146.829
1.2	289.340

Table 2: Execution Time and scale factors for resizing an image with Mask-RCNN

5. Conclusion

5.1. What Have We Discovered?

One of the main discoveries of this project was the ability of simple methods to provide complex and interesting results. It is common today to assume that Deep Learning and other Machine Learning methods are required to get good results, but the results of the plain seam carving algorithm are impressive even when held against more modern methods.

However, that is not to say that Deep Learning and associated methods do not have their place. In fact, our combination of classical methods and Deep methods highlights an important synergy that can drive good results in research going forwards.

5.2. Future Work

In the future, the main aspect to be improved is the execution time of the improved content aware image resizing algorithm. A possible way to do this is to eliminate the overhead of having to recompute the masks on each iteration. A possible way to do this is to compute the initial masks, and then combine it with the gradient space of the image, and perform seam-carving in the gradient space entirely, reconstruction the final image using Poisson Reconstruction [5], as was mentioned in [2]. This should offer tremendous speed up, as we would avoid having to recompute the image masks, as well as eliminate most of the overhead of calling a Python Library from within Julia.

Additional testing can also be done with higher values of α. When testing values much beyond 20, there seems to be some kind of instability in the library, causing crashes more often than desired during execution. This is especially unfortunate considering the length of the execution time needed for the integrated RNN-Seam Carving algorithm.

There are additional optimizations that can be made to the original seam-carving application that can greatly increase the speed of its execution such as implementing a neighbour-based approach to calculating seams that was capable of real-time computation back in 2009 [4]. However, as there is no discrete phase of energy computation, it is unclear how to adapt our system to this faster method.

In addition, it can be seen that if the masking of objects in our images was exact, there would be no distortion around the semantic content whatsoever. Knowing this, it seems that as the state of the art of image segmentation/masking improves, so to will further implementations of our algorithm.

6. Additional Images

References

[1] W. Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`, 2017.

[2] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3):10, 2007.

[3] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[4] H. Hua, F. TianNan, R. P. L, and Q. Chun. Real-time content-aware image resizing. *Science in China Press Series F: Information Sciences*, 52(2):172–182, 2009.

[5] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, July 2003.

(a) Original

(b) 20% enlarged (our method)

(c) 20% enlarged (seam carving)

(d) 20% shrunk (our method)

(e) 20% shrunk (seam carving)

(a) Original

(b) 20% enlarged (our method)

(c) 20% enlarged (seam carving)

(d) 20% shrunk (our method)

(e) 20% shrunk (seam carving)