MT - Research

Seam Carving for Content-Aware Image Resizing

Ran Chen
email: ranchen@gatech.edu

Abstract

This project reproduces the work done by S. Avidan and A. Shamir, *Seam Carving for Content-Aware Image Resizing* (referred to as "*the paper*" in this report). A video presentation of this project can be accessed at `https://youtu.be/yI8ZDSFjZ60`.

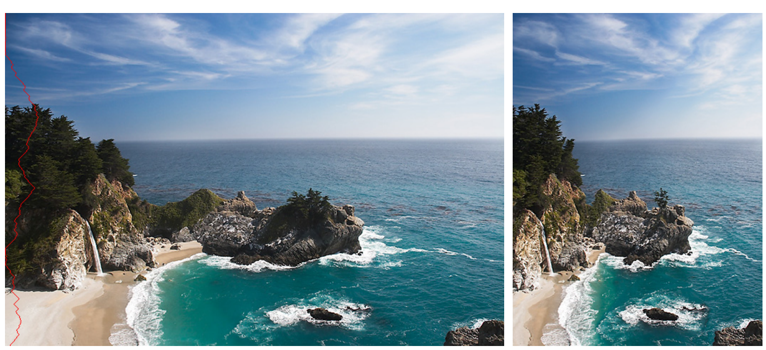
1 Energy Map and Definition of a Seam

The basic idea is to minimize the energy change, or cost, when removing a seam from or add a seam into the original image. The energy of a single pixel should represent how different such pixel is from its neighboring pixels, such that, removing or adding the pixels with the lowest difference would ideally introduce least noticeable artifacts. Here we define the energy (e) of an image $\mathbf{I}$ as a linear combination of the absolute values of gradients: $e(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|$.

Fig 1a shows the energy map of the image presented in *the paper* (Figure 5 in *the paper*).

(a) Energy map. (b) Horizontal seam.

Figure 1: Energy map and one single horizontal seam.



(a) Vertical seam. (b) Shrink.

Figure 2: One vertical seam, and horizontal shrink by removing consecutive vertical seams.

I my implementation, the gradient maps in both directions were obtained using Sobel function directly from OpenCV. Here, a vertical seam is defined as a path of pixels from top to bottom, containing only one pixel in each row, and the horizontal shift between pixels from two consecutive rows must be smaller than or equal to k (in my implementation $k = 1$), and a horizontal seam is defined as a path of pixels from left to right, containing only one pixel in each column, and the vertical shift between pixels from two consecutive columns must be smaller than or equal to k (again, here $k = 1$). Formally, a vertical seam ($\mathbf{s^x}$) and a horizontal seam ($\mathbf{s^y}$) are defined as:

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

$$\mathbf{s^y} = \{s_j^y\}_{j=1}^n = \{(j, y(j))\}_{j=1}^n, \text{ s.t. } \forall i, |y(j) - y(j-1)| \leq k = 1$$

And the cost of removing or adding a seam will be simply the sum of the energy values of all pixels on a seam. The objective is to find the seam, by above definition, with the lowest cost, or cumulative energy. Here, similar to *the paper*, dynamic program was used, Fig 1b shows the horizontal seam with the lowest cost in the image from Figure 5 in *the paper*.

2 Seam Removal

To shrink an image in a content-aware manner, we just need to repeat such process as many times as desired:

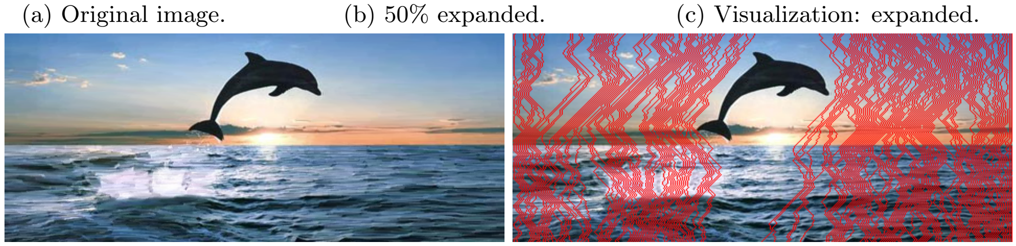
1. Calculate energy map,
2. Find a seam,
3. Remove the seam.

Fig 2a shows one vertical seam found on the original image, and Fig 2b shows the resulting shrinkage in horizontal by removing vertical seams consecutively. This result looks very similar to the one provided in *the paper*.

3 Seam Insertion

To achieve content-aware image enlarging, we just need to insert the seam with lowest cost. However, for multiple seam insertion, in order to avoid inserting the same seam repeatedly, which obviously will introduce noticeable artifacts, we will need to find multiple seams as if we are removing them consecutively, then insert them in the reversed order.

(a) Original image. (b) 50% expanded. (c) Visualization: expanded.



(d) 50% further expanded. (e) Visualization: further expanded.

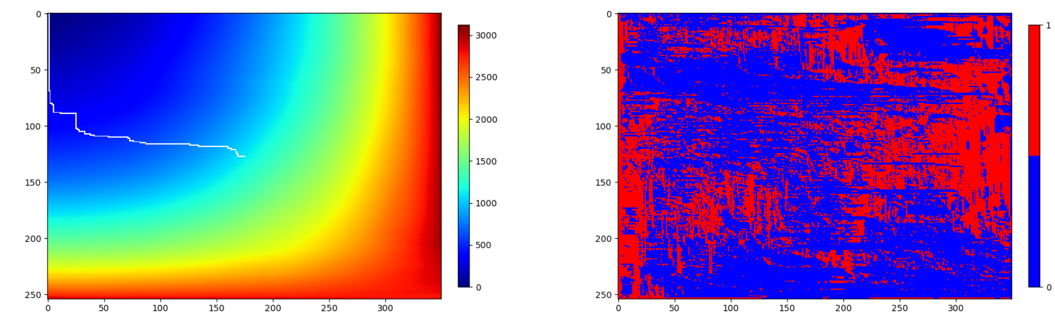
Figure 3: One vertical seam, and horizontal shrink by removing consecutive vertical seams.

Using the strategy described above, image expansion in horizontal direction was performed via insertion of vertical seams repeatedly. Fig 3 shows the 50% expansion from the original image (Fig 3a) to Fig 3b, and Fig 3c shows the seams that have been inserted. Fig 3d and 3e shows another 50% expansion based on Fig 3b. These results also compares quite favorably to the results from *the paper*, the expanded images look quite natural, free of any obvious artifacts.

4 Optimal Retargeting

To achieve content-aware size reduction, we simply need to shrink in both directions, or remove both vertical and horizontal seams. However, the optimal solution requires a specific ordering of vertical and horizontal seam removals to obtain minimal cost. Again, dynamic programing were used to computer a transport map ($\mathbf{T}$) representing the minimal cumulative energy possible for each size reduction, the values in the transport map ($\mathbf{T}(r,c)$) corresponds to the minimal cost required for size reducion of r in height and c in width. A binary (values can be 0 or 1) step map ($\mathbf{S}$) can be computed by comparing $\mathbf{T}(r,c-1)$ and $\mathbf{T}(r-1,c)$:

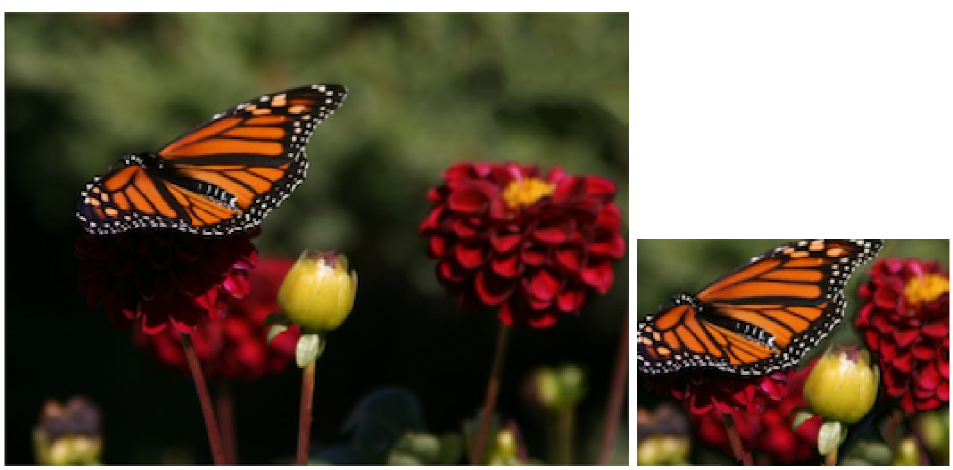
$$\mathbf{S}(r,c) = \begin{cases} 0, \text{ for } \mathbf{T}(r-1,c) \geq \mathbf{T}(r,c-1) \\ 1, \text{ for } \mathbf{T}(r-1,c) < \mathbf{T}(r,c-1) \end{cases}$$



(a) Transport map. (b) Step map.

Figure 4: Transport map and step map (0: vertical seam removal, and 1: horizontal seam removal).

Fig 4 shows the transport and step map computed from the original image in Figure 7 from *the paper*. The white line in the transport map shows the optimal path to take to reduce the image to 50% of the original in both height and width. This path looked different from *the paper* probably because a different target size (r, c) was used.



(a) Original image. (b) 50% reduction.

Figure 5: Reduction in both height and width.

Fig 5 shows the results from size reduction in both directions using the retargeting strategy described above, the reduced image looks quite similar to the results provided in *the paper*.