RESEARCH PROJECT

-Sudeep Gupta

1. Introduction

Seam Carving is an algorithm for content aware image re-sizing which functions by establishing many seams (paths of least importance) in an image and automatically removes seams to reduce image size or inserts seams to extend it. The purpose of the algorithm is image retargeting, which is used in displaying images on media of different sizes. To reduce the image size, cropping is limited since it can only remove pixels from image periphery. Also, Image scaling is not helpful as it can only be applied uniformly across pixels.

Seam carving uses an energy function defining importance of pixels. A seam is a connected path of low energy pixels either top to bottom or left to right. By successfully removing or inserting seams we can reduce or enlarge an image. In this project, I am going to illustrate the application of seam carving for reduce image size image retargeting and Image Enlarging.

2. Energy and Cost Function

The approach to content aware resizing is to remove pixels in judicious manner. The main goal is to remove un-noticeable pixels. To achieve this, we calculate energy function:

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

Where **I** be an *n* x *m* image. The vertical seam is calculated as:

$$\mathbf{s^x} = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1, \quad (2)$$

The horizontal seam is calculated as:

$$\mathbf{s^y} = \{s_j^y\}_{j=1}^m = \{(j, y(j))\}_{j=1}^m, \text{ s.t. } \forall j |y(j) - y(j-1)| \leq 1. \quad (3)$$

For the energy function, we start with calculating and minimizing the cost function as below:

$$s^* = \min_{\mathbf{s}} E(\mathbf{s}) = \min_{\mathbf{s}} \sum_{i=1}^{n} e(\mathbf{I}(s_i)) \quad (4)$$

The first step is to calculate cumulative minimum energy for all possible connected seams for each pixel entry in the image. We calculate the cumulative minimum energy as below:

$$M(i,j) = e(i,j) + \min(M(i-1,j-1), M(i-1,j), M(i-1,j+1)) \quad (5)$$

At the end of the calculations, the minimum energy at the last row will represent minimum energy needed of a connected vertical seam. From this point, we can backtrack on M to find seam of minimum energy.

3. Image Reduce

We were given an image of size 700 x 466, Figure 1(a). We were asked to reduce the image size horizontally. To do horizontal reduction, first energy of the image was calculated using equation 1. After energy calculation, cumulative minimum energy for all the connected pixels were calculated as per equation 5. From the last row of M matrix, minimum value was taken and backtracked up to the 1st row of M matrix. This gave us vertical seam with minimum energy. Now this vertical seam was removed and the image of size 699 x 466 was obtained. Based on the number of seams to be removed, the same process was followed recursively removing vertical seams and the reduced image was obtained. In this project, I reduced the image to 350 x 466, Figure 1(b). The reduced image looks very similar to the reference reduced image from Figure 5 of Reference 1 (see references).

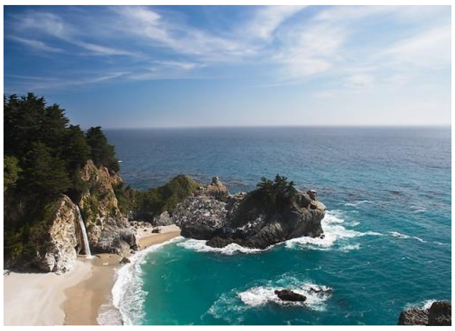
Figure 1(a): Original Image (700 x 466)

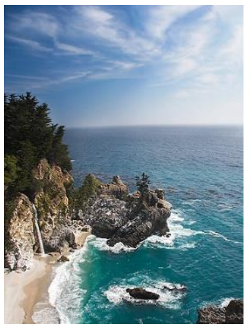
Figure 1(b): Reduced Image (350 x 466)

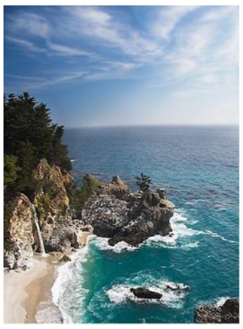
Figure 1(c): Reference Reduced Image (350 x 466)

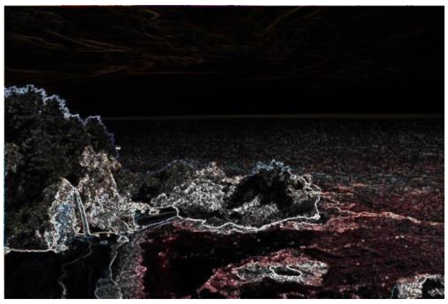
Figure 1(e): Energy of Original Image (700 x 466)

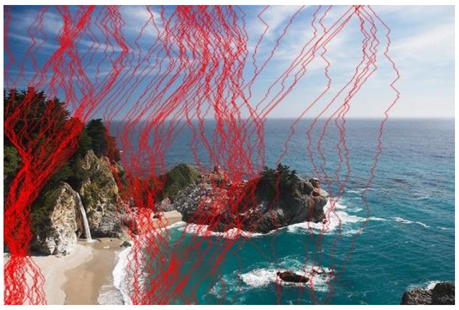
Figure 1(f): 100 Low energy seams in Original Image (700 x 466)

4. Retargeting with Optimal Seams Order

Image retargeting generalizes aspect ratio change from one size to another. To reduce the image both horizontally and vertically, the main challenge is to figure out which seam to remove first horizontal or vertical. To optimize the removal, we followed Transport Map T that specifies, the cost of optimal sequence of horizontal and vertical seam removal. To reduce the image by r rows and c columns, we calculated Transport map using dynamic programming. We started with T(0,0) = 0 and calculated cost of removing horizontal or vertical seam using dynamic programming equation 6.

$$\mathbf{T}(r,c) = \min(\mathbf{T}(r-1,c)+E(\mathbf{s^x}(\mathbf{I_{n-r-1\times m-c}})), \mathbf{T}(r,c-1)+E(\mathbf{s^y}(\mathbf{I_{n-r\times m-c-1}}))) \quad (6)$$

In the above equation $I_{n-r \times m-c}$ denote an image of size n-r x m-c, $E(s^x(\mathbf{I}))$ and $E(s^y(\mathbf{I}))$ are the cost of respective seam removal operation. A one-bit map of size r x c was created which indicates which of the seam removal operation was chosen at each step. We then backtracked from T(r,c) to T(0,0) and applied corresponding seam removal operations. For this task image of size 350 x 254 was taken, figure 2(a) and it was reduced to size 222 x 132, figure 2(b). A transport matrix was also generated which will guide to the optimal seam removal. There is some difference between the retargeted image generated by code and the reference image (figure 2c) .

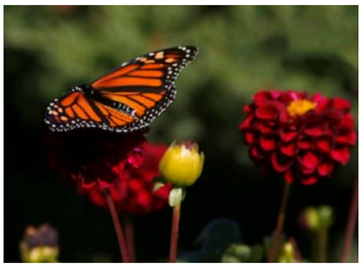
Figure 2(a): Original Image (350 x 254)

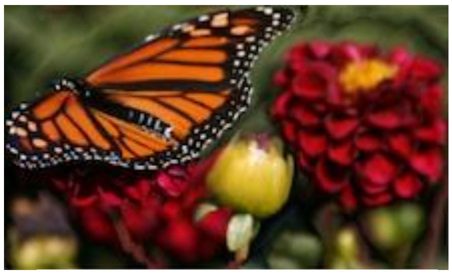
Figure 2(b): Retargeted Image (222 x 132)

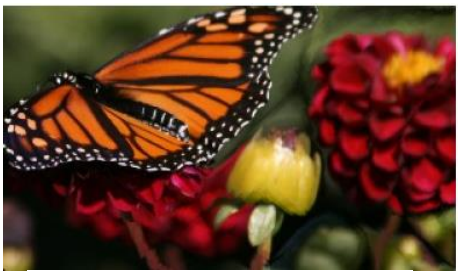
Figure 2(c): Reference retargeted Image

5. Image Enlarging

For Image enlarging by adding seams we follow the process of seam removal in inverse order. To achieve image enlarging, I first calculated cumulative Energy matrix M and generated k seams for removal with lowest energy from the original image. I stored all the indices of k seams in a matrix. While inserting the seams, we should duplicate the pixels by averaging them with their left and right neighbours for horizontal case. For this task, image of size 239 x 200 was taken, figure 3(a). It was enlarged to 358 x 200, figure 3(c). Excessive enlarging to size 479 x 200, figure 3(d), was achieved by two step enlarging process. In first step original image was enlarged to 358 x 200. In second step, the previous enlarged image was given as input and was then enlarged to 479 x 200. From the images we can see that images generated are almost same as the reference images. We can see some difference in figure 3(b), seam inserted image, but visually that difference is hardly noticeable in figure 3(c) and second image of figure 3(e). The main issue faced in image resizing was handling the corners. Earlier, I was using 3 x 3 sobel image gradient function because of which I was getting less energy at image edges. This was resulting in artefacts at the edges in the enlarged image. To overcome this, I used e_1 gradient (equation 1) to calculate the energy. We can see clearly that enlarged image from the code figure 3(c) and figure 3(d) are almost similar to second and third images of reference figure 3(e).

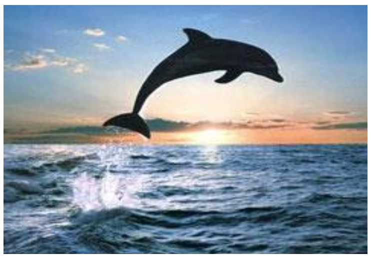
Figure 3(a): Original Image (239 x 200)

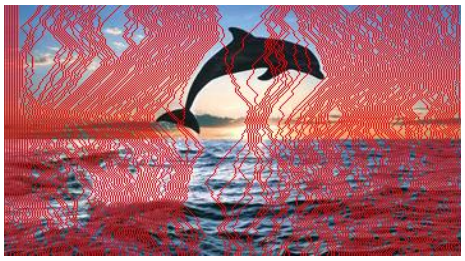
Figure 3(b): Seam insertion in order of removal

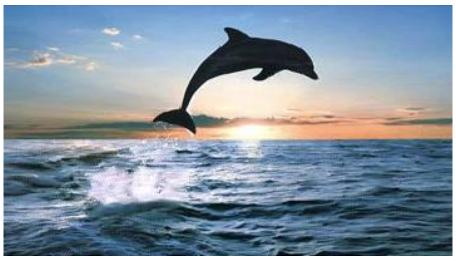
Figure 3(c): Enlarged Image (239 x 200)

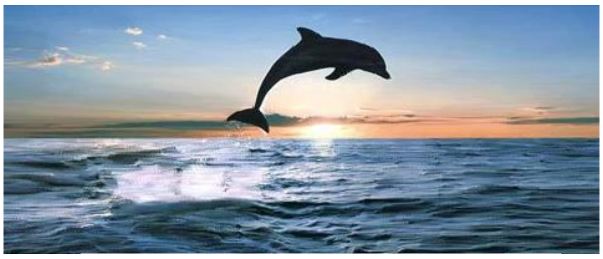
Figure 3(d): Excessively Enlarged Image (479 x 200)

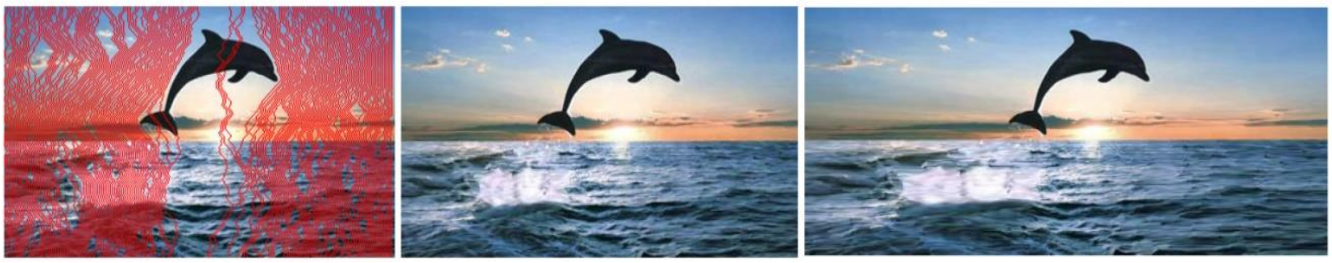
Figure 3(e): Reference Images from the paper

6. Video Link:

https://drive.google.com/file/d/0B_HG5lxe9Xc7MnRXUnVSQ0hoYmM/view?usp=sharing

7. References:

1. Seam Carving for Content-Aware Image Resizing by Shai Avidan and Ariel Shamir.
2. https://en.wikipedia.org/wiki/Seam_carving
3. http://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=scharr#scharr
4. To create gifs: http://gifmaker.org/
5. Bandicam software to create the video

8. Code Instructions:

Instructions to run the code are given in README.md file in resources.zip.