SEAM CARVING

scaling images without losing visual information

Nguyen Huu Thang - 16521102, Giang Phuc Vinh - 16521548, Vu Tran Thanh Nam - 16520783
Intructor: Nguyen Vinh Tiep

***Abstract*—Each pixel in an image covers different information. Therefore, resizing an image, we must consider which pixel to keep and which to delete. Seam carving is an effective algorithm to do that. In this algorithm, it will calculate a seam, which is a connected path of pixels from top to bottom, or left to right in an image. The pixels in the seam must have low content information. By repeatedly removing the seams out of the image, we have a new image containing lots of important information of the old one. Applying this technique horizontal and vertical, we can have a fully-scaled image without losing much content. More over, this technique can be used in adding or removing object. In this project, we implement seam carving with python language.**

***Index Terms*—seam, resizing, scaling, image.**

I. INTRODUCTION

TODAY, image processing is known as an essential in computer science. One of the biggest problems is image resizing. For instance, designers must create different alternatives for different layouts in different devices. But, up to date, images - one of the key elements in digital media, have to change its ratio to fit into various layouts.

Standard image scaling is not sufficient since the object in the picture may be unable to keep its original shape. Another way to do it is cropping. But it can just remove pixels on the edge of the image. We must consider the content of the image to choose which pixel to delete. Only then, we can have an effective resizing.

Talking about such technique, we have seam-carving. seam-carving is an operator that can resize image considering the content of the image. There is an function to calculate the importance/energy of each pixel in the image. Then it creates a seam from top to bottom or left to right. A seam is a connected path of low energy pixels crossing the image. By deleting or adding more seams, we can have a new image that have nearly-same content of the origin.

Seam carving can be applied to solve some other problem, such as aspect ratio change, image retargeting, image content enhancement, and object removal. The energy function is independent so we can change use several types of energy functions such as gradient magnitude, entropy, visual saliency, eye-gaze movement. In our project, we apply seam carving technique and create a program allowing user to resize an image. More over, to have more interaction, our program allow user to mark which area in the image that won't be removed or will be removed first.

II. MATERIALS AND METHODS.

What we try to do is delete the pixels that have less content information. Those are the one that we have less focus than others when we look at the picture. To find those pixels, we have a function. This function apply the following formula to each pixel in the picture.

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

Practically, we apply 2 kernels, horizontal and vertical sobel, on the picture. After that, we have 2 other pictures describing vertical and horizontal energy of the origin. Adding the absolute value of these two, we have an energy map. Then we create a new matrix called M with the first row equal to the first row of energy map. With other rows, we calculate the values base on the following formula.

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

The minimum energy required to traverse from the top of the image to bottom will be present in the last row of M. From this, we backtrack to find the seam line. With several techniques, we manage to remove the seam pixels out of the original picture. Repeating these steps, we have the result we want. But that is just vertical seams. To deal with horizontal seams, we just rotate the picture and apply previous steps and rotate it back to normal.

III. RESULTS

We use the tower image to test our code.

Fig. 1. The original image.

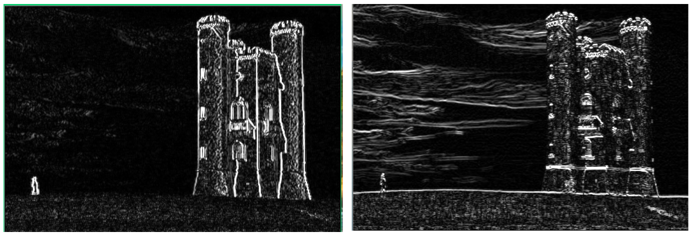

Fig. 2. After apply vertical and horizontal sobel kernel, we have 2 images representing vertical and horizontal energy.

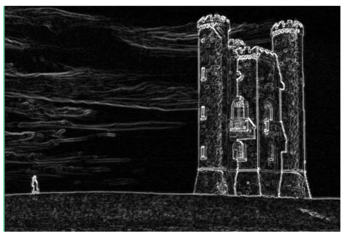

Fig. 3. Energy matrix.

Fig. 4. The result.

Because the result is not good. As we said about the interaction, we can mark some areas so the program won't remove the pixels in those areas.

Fig. 5. Better result when the image is marked by human.

At this point, we notice that the program is quiet slow. So we try to delete the seam lines only after calculating all the seam lines. But the result doesn't seem so good. We figured out that the error is at the seam line finding step. The later seam line may contain the pixels that already belong to the previous seam lines. But we don't know the way to fix it yet.

Fig. 6. The result when we try to fasten the program.

IV. Discussion and Summary

Although we achieve some work, we still have more to do. Applying other energy calculating functions would change the result. But we did not test those yet. More over, if we could manage to calculate all the seam lines at once, the program would be much faster. That is also what we will try to do in the future to improve the program. We think it is promising to have better result than what we have now.

Appendix A
Code.

Python 3.
OpenCV2.
Link to code:
https://github.com/thanhnam1998/SeamCarving.git

References

The original SEAM CARVING paper:
https://perso.crans.org/frenoy/matlab2012/seamcarving.pdf
Some others links that we consult:

- https://vnoi.info/wiki/cs/imageprocessing/Seam-Carving
- http://www.faculty.idc.ac.il/arik/SCWeb/imret/index.html
- https://karthikkaranth.me/blog/implementing-seam-carving-with-python/