City University of Hong Kong

CS5187 Vision and Image course project

Image resizing: Seam Carving

Wei Yihuan, Zhang Rui

Obejective

The image resizing technique Seam Carving functions by establishing a number of seams (paths of least importance) in an image and automatically removes seams to reduce image size or inserts seams to extend it. It is useful to display images without distortion on difference sizes. Our project aims at researching some novel approaches about image resizing(mainly on Seam Carving) and combining with other techniques, trying to improve the performance of it.

Scope

Our project aims implementing this algorithms by Python and OpenCV. We would make comparison about the performance of the following aspects, based on former implementation:

- Downsample and upsample.
- Vertical and horizontal resizing.
- Seam carving with objects mask and face detection.

Related Work

Nowadays there are many devices to display a same images on different size. To fit with the screen size, standard image scaling is not satisfied for distorting image and degrading resolution. Here we followed the definition, *seam-carving*, which resizes images by deleting or inserting seams formed by pixels of the unnoticeable parts of an images. A seam is defined by a path connecting low energy pixels through a whole image along one direction, such as from top to bottom, or from left to right. The images is processed seam by seam gradually to keep pixels continuous. And it could also be applied to protect or remove the object in images to be processed which scaling could do not work.

Energy Function

The importance of seam is defined by energy function. There are several types of energy functions based on gradient magnitude, visual saliency and so on. To start with a simple approach, here we use the e_1 energy function which is showed effective before. The energy of a pixel (x, y) is computed as:

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

Where the partial derivative is summed up by RBG channels R(x,y) , G(x,y), and B(x,y). Gathering with the whole pixels' energy, we obtain an energy map for a picture(See Figure1). The darkest part indicates the lowest energy value, whereas the white part indicates high energy.


(a) Original image

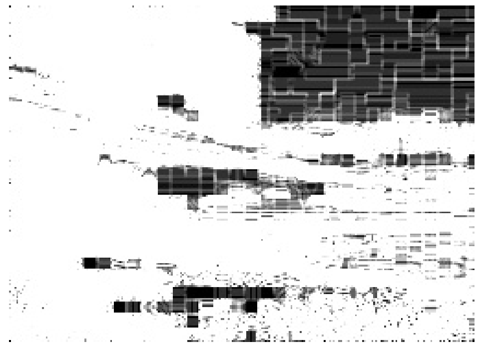

(b) Energy map

Figure 1: *Energy map of an image*

If we compute and track each pixel's energy and simply remove the pixels in an ascending order, due to different amount of pixel removal in every row or column, the shape of the image could not be preserved. Thus, we construct seams(e.g.vertical seam) by selecting the minimum energy pixel in each row, such that in each round just removing only one pixel in one row, keeping the shape of the image. In addition, the cost of removing a seam is defined as:

$$E(s) = \sum_{i} e(I(s_i)) \tag{2}$$

In each round, an optimal seam is found by minimizing the cost:

$$s^* = E(s) = min \sum_{i} e(I(s_i)) \tag{3}$$

For instance, a vertical seam is shown below. And it could be found by dynamic programming. Firstly search the image and record the minimum energy pixel of each row and add them together, then trace back to find a path connecting by the minimum energy pixel and show the seam.

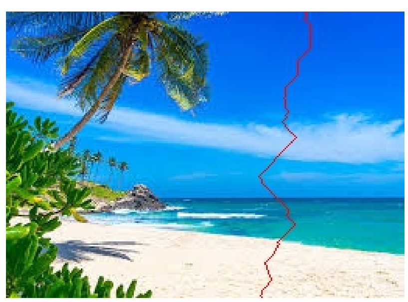


Figure 2: *Example of a vertical seam*

Reduce versus Enlarge

Reduce. Suppose the original size of an image is m∗n, and the target size is m', n'. If it is reducing the size vertically, $n > n'$ and/or $m = m'$. $k = n - n'$. Compute the seams' energy and remove the k seams in k rounds following the minimal cost rule.

Enlarge. It can be considered as an inverse progress for reducing. It is enlarging that $n' > n$ and $n' - n = k$. Search and track the k seams as they were to be removed. And then duplicate them by ascending order until reach the target size m'. If we only remove the minimal cost seam and compensate with the average of the left and right neighbour, it would look like a stretching artifact. Nor duplicate some seams in the image is like scaling some part of it. The comparison is shown by previous work[1].

(a) Compensate by neighbors

(b) Scaling

(c) Seam carving

Figure 3: *Image enlarge*

Object Removal and Protection

If we manually mask an object in an image like the following picture, making the region as largest or smallest intensity, thus the seams went around or traverse the region. Then the object could be left behind or removed as desired.

Figure 4: *Mask an object in an image*

Face Detection

Face is always important part in an image. If a face is detected in an image, it requires to be protected from removing when doing resizing.

Here we apply the Haar feature-based cascade classifiers to detect face in images which is an effective method for face detection. It was trained with positive and negative samples(face and non-face). Haar features are like convolutional kernel computed by subtracting sum of pixels from one user defined area to another. In faces, the features extracted from the region around eyes and the bridge of the nose are considered as good features. Nevertheless, if there were large amount of features, the selection of the best features is achieved by Adaboost.

At first all the features are used for training and their best threshold were found. Classification are applied by them. Select the features with minimum error rate. At the beginning all the images have the same weight while in the iteration, add weights to the misclassified images. The iteration went on until the desired accuracy is reached.

Since the most parts of an image is non-face area, Cascade of Classifiers is introduced. It examine groups of features stage by stage instead of whole features at the same window. If a window could not pass the examination, discard it as well as the remaining features on it, otherwise continue with the next stage of features. Pick the most meaningful and as fewer as possible features in the earlier stage, like the region around eyes, such that the process would be more efficient.

We implement this face detection by OpenCV.

Methodology

Content-aware Image Resizing

Build a function to calculate energy value of every (residual) pixels in the image. Cumulate minimum energy by traversing the image from the second row to the last row (Note: enlarge is inverse). When completing the traversing process, backtrack from this minimum value on every row(column for horizontal seam) to find the path as the optimal seam. Repeat the above steps until resizing is finished. We have k times loop, where $k = |n - n'|$.

We use the function to process the vertically resizing. For horizontal resizing, the image will be rotated 90 degree before and apply this function. However, because of the limitation of the enlarge step (enlarge an image by k, we find the first k seams for removal, and duplicate them), we can not enlarge the image to the size larger than as twice as the original one.

Removal and Protection of an object

Manually mark the desired object at first. Pre-process with the mark by Gaussian filter to smooth the region contour. Due to the mask applied, the marked region would be assigned rather low or high energy values to be removed or protected.

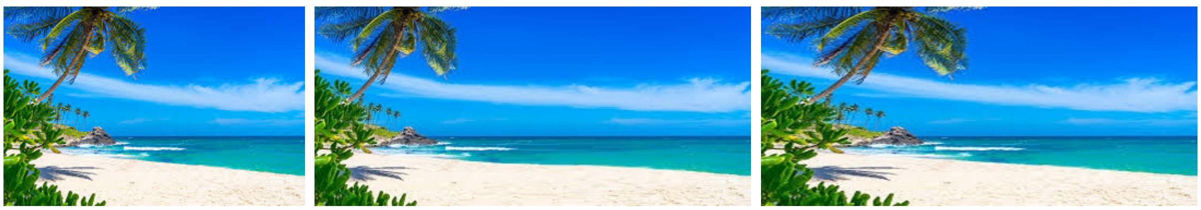
Face Detection and Protection

Apply the Cascade Classifier to detect face in images. If a face is detected, generate a mask automatically for protection. Thus it is a constrain preventing the face from distortion or destruction through resizing.

Result Analysis

Seam carving of enlargement

The original image has a size of 183*276 and target size is 183*400. The difference between normal scaling and seam carving is obvious. In scaling, every object in the image are enlarged. Seam carving is aware of the importance of objects in the image, like coconut tree, stone, leaves, preventing them from distortion. While the less important part of the image, such as sky, sea, beach have been scaled up.



(a) Origin image (b) Seam carving (c) Scaling

Figure 5: *Image resizing*

Object Protection and Removal

Protection

Let's see the result of mask protection. Without mask, the people are partially removed too. The reason can be figured out from the energy map. The removed parts of the people in the energy map are darker which indicates lower energy.

(a) Origin image (b) Seam carving result (c) Energy map

Figure 6: *Seam carving without object protection*

This could be handled with manually adding mask to the important object and apply the mask in the processing. In the energy map of mask applied, the energy at the regions where three people stand are completely white. Since the marked areas have been assigned a high energy value, the seams depends on energy would go around the people instead of going through them. As a result, the people are protected from removing.

(a) Mask of the image (b) Seam carving result (c) Energy map with mask

Figure 7: *Seam carving with object protection*

Removal

Then follows the object removal. In this picture we need to remove the man stand far away. If we only marked the man to be removed with dark color, the result looks weird since the shape of the helmet has been lengthened.

(a) Origin image (b) Removal without protection (c) Energy map

Figure 8: *Object removal without protection*

In the intuition, the man's helmet is white that is close to the color of sky. Thus when compensating the area after removal, the helmet has been duplicated. If we applied protecting mask for the helmet at the same time, the result is better.

(a) Object mask (b) Removal with protection (c) Energy map

Figure 9: *Object removal with protection*

Face detection

The seam carving sometimes would not be aware of the faces. If we do nothing before and apply seam carving to reduce the size of an image directly, the face is distorted. The manually added mask might be able to solve the problem. However, if there were many faces to be labeled, the marking workload is very heavy. This is achieved by using the face detection which can automatically generate a mask for faces.

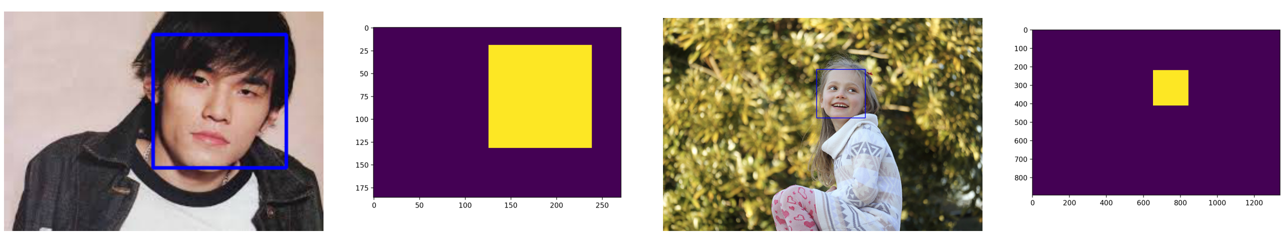


Figure 10: *Face Detection and mask generation*

Also compare the result with/without mask when reducing the image size.

(a) Original image (b) Seam carving with face detection (c) Seam carving without face detection

Figure 11: *Seam carving with/without face detection*

The difference is apparent between the two. Face detection works well along with seam carving.

Conclusion

Thus we implement seam carving of downsample and upsample along vertical and horizontal direction. It is an content aware algorithm by defining the importance of image parts depends on energy function. Thus it achieve advanced effect rather than scaling or cropping images. In addition, we explored object mask and face detection to enhance the importance of objects in images, acquiring better result for seam carving.

Roles

We have 2 team members, Wei Yihuan and Zhang Rui. Hereby is the roles and problems.

- **Wei:** Research relevant papers and materials, suggest improvement of the methods, compare, analyze, and organize the results.
- **Zhang:** Mainly implement and deploy the algorithms.

References

[1] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3), July 2007.

[2] Nguyen Thi Nhat Anh, Wenxian Yang, and Jianfei Cai. Seam carving extension: A compression perspective. In *Proceedings of the 17th ACM International Conference on Multimedia*, MM '09, pages 825–828, New York, NY, USA, 2009. ACM.

[3] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. pages 511–518, 2001.