Seam Carving for Content-Aware Image Resizing

Project Report

Computer Vision 2019-20

Pranjali Jain
16110119

Naman Jain
16110100

Paper: Seam Carving for Content-Aware Image Resizing (pdf)
Published: ACM SIGGRAPH 2007
Authors: Shai Avidan, Ariel Shamir

I. Introduction

Effective resizing of images should consider the image content and not just geometric constraints. Standard image scaling is not sufficient as they forget about the content of the image and it can only be applied uniformly. Cropping is limited since it can only remove pixels from the image periphery, we may lose important information with it.

Seam carving is a simple operator that supports content-aware image resizing. A seam is an optimal 8-connected path of pixels on a single image from top to bottom or left to right. Here, optimality is defined by an image energy function that defines the importance of pixels in an image. Hence, a seam is a connected path of low energy pixels in an image.

Repeatedly carving out or inserting seams in one direction can change the aspect ratio of an image, and by doing the same in both directions we can retarget the image to a new size. The selection and order of seams are defined by the energy function.

Image reduction removes seams with more of the low energy pixels and fewer of the high energy ones. Image enlarging inserts pixels maintaining a balance between the original image content and the artificially inserted pixels.

Seam carving can also be used for image content enhancement and object removal.

II. Previous works and their limitations:

Image retargeting seeks to change the size of the image while maintaining the important features intact. These features can be either detected top-down or bottom-up. Top-down methods use tools like face detectors[1] to detect important regions in the image. Bottom-up methods rely on visual saliency methods[2]. They construct a visual saliency map of the image and then use cropping to display the most important region of the image. However, all these methods rely on traditional image resizing and cropping operations to change the size of the image.

Non-linear, data-dependent scaling has been used for image retargeting[3,4] by finding the Region-Of-Interest(ROI) and constructing a warp that applies a piecewise linear scaling function in each dimension to the image. Another retargeting algorithm[5] segments an image into regions, identifies important regions, removes them, fills the resulting gaps, resizes the remaining image, and re-inserts the important region. These methods do not use seams.

The use of seams for image editing is prevalent. Digital Photomontage system[6], Drag-and-Drop Pasting[7], and AutoCollage[8] compute optimal seams between the source and target images. But none of these methods discuss image retargeting.

One work[9], comes close to image retargeting using seams, where the authors discuss simultaneously solving matting and compositing, by allowing the user to scale the size of the foreground object and paste it back on the original background.

Computing the seam can be done in a variety of ways, including Dijkstra's shortest path algorithm, dynamic programming, or graph cuts. Several cost functions for seamless image-stitching have been evaluated in [10]. For object removal, image inpainting method[11] and patch-based approaches [12,13] have been proposed.

III. Seam carving background

Content-aware resizing using seams requires removing unnoticeable pixels that blend with their surroundings. This is described by the following energy function:

$$e_1(I) = |\tfrac{d}{dx}I| + |\tfrac{d}{dy}I|$$

There are several methods to reduce the image width keeping energy into consideration -

- Optimal strategy → remove the pixels with the lowest energy in ascending order. This destroys the rectangular shape of the image, because we may remove a different number of pixels from each row.
- Pixel strategy → removes an equal number of low energy pixels from every row. This preserves the rectangular shape of the image but destroys the image content by creating a zigzag effect
- Cropping → finds a sub-window, the size of the target image, that contains the highest energy. This preserves both the shape and the visual coherence of the image.
- Column strategy → removes whole columns with the lowest energy. This can lead to distortion in the image.

Seam carving is a better option compared to all these strategies, as it is less restrictive and preserves the image content better than simple pixel removals. Removing the pixels of a seam from an image has only a local effect: all the pixels of the image are shifted left (or up) to compensate for the missing path. The visual impact is noticeable only along the path of the seam, leaving the rest of the image intact.

Let I be an $n \times m$ image,

- Vertical seam:
  $s^x = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n \;\; s.t. \;\; \forall i, |x(i) - x(i-1)| \leq k$
  where x is a mapping $x : [1, ..., n] \rightarrow [1, ..., m]$

- Horizontal seam:
  $s^y = \{s_j^y\}_{j=1}^m = \{(j, y(j))\}_{j=1}^m \;\; s.t. \;\; \forall j, |y(j) - y(j-1)| \leq k$
  where y is a mapping $y : [1, ..., m] \rightarrow [1, ..., n]$

We can get either a simple column(row) for $k = 0$, a piecewise connected or a completely disconnected set of pixels for any value $i \leq k \leq m(n)$.

Hence, a vertical seam is an 8-connected path of pixels in the image from top to bottom, containing only one pixel in each row of the image and similarly for the horizontal seam. The pixels of the path of a vertical seam will be: $I_s = \{I(s_i)\}_{j=1}^n = \{I(x(i), i)\}_{i=1}^n$

Given an energy function e, the cost of a seam is: $E(s) = E(I_s) = \Sigma_{i=1}^n e(I(s_i))$

and the optimal seam that minimizes this seam cost :
$s^* = min_s\ E(s) = min_s\ \Sigma_{i=1}^{n} e(I(s_i))$

The optimal vertical/horizontal seam can be found using dynamic programming.
For vertical seam,

- traverse the image from the second row to the last row and compute the cumulative minimum energy M for all possible connected seams for each entry (i,j) :
  $M(i,j) = e(i,\ j) + \ min(M(i-1,\ j-1), M(i-1,\ j), M(i-1,\ j\ +1))$
- the minimum value of the last row in M will indicate the end of the minimal connected vertical seam.
- backtrack from this minimum entry on M to find the path of the optimal seam.

A similar case is true for horizontal seams which can be achieved by rotating the images by 90 degrees counter-clockwise.

Content-aware resizing raises the average energy of all pixels in an image as it removes low energy pixels and keeps the high energy ones. Compared to all other strategies, seam carving strikes the best balance between energy preservation and visual coherency.

Also, no single energy function performs well across all images for seam carving. They vary in the parts of the image they affect. The paper reports e1 and eHoG(Histogram of Gradients) to work quite well.

IV. Applications of Seam Carving

1. ***Aspect Ratio:*** *changing one dimension to create smaller image*

The aspect ratio of an image I can be changed from $n\ x\ m$ to $n\ x\ m'$, where $m -\ m' = c$, by successively removing c vertical seams from I.

It can also be done by increasing the number of rows in I by a factor of m/m'. This approach does not remove any information from the image.

2. ***Image Retargeting:*** *changing image size using two dimensions*

Image retargeting generalizes aspect ratio change from one dimension to two dimensions such that an image I can be changed from $n\,x\,m$ to $n'\,x\,m'$, where $m - m' = c$ and $n - n' = r$.

We find the optimal order using a transport map T that specifies, for each desired target image size $n'\,x\,m'$, the cost of the optimal sequence of horizontal and vertical seam removal operations. T is computed using dynamic programming.

- Store a simple $n\,x\,m$ 1-bit map indicating which of the two options was chosen in each step of the dynamic programming.
- Choosing a left neighbor corresponds to a vertical seam removal while choosing the top neighbor corresponds to a horizontal seam removal.
- Backtrack from $T(r,c)$ to $T(0,0)$ and apply the corresponding removal operations.

3. ***Image Enlarging:*** *Increasing image size*

Enlarging an image can be seen as the inversion of seam removal, where we insert new 'artificial' seams to the image.

Hence, to enlarge the size of an image I by one we compute the optimal vertical (horizontal) seam s on I and duplicate the pixels of s by averaging them with their left and right neighbors (top and bottom in the horizontal case). To enlarge an image by k, we find the first k seams for removal, and duplicate them.

We break the process of enlarging into several steps where each step does not enlarge the size of the image in more than a fraction of its size from the previous step, so ensure content-aware resizing.

4. ***Object Removal***

Object removal from a target region is done by removing the seams from the image until all pixels in the target region are gone. We can calculate the smaller of the vertical or horizontal diameters (in pixels) of the target removal region and perform vertical or horizontal removals accordingly. The original size of the image can be regained by seam insertion.

V. Limitations of the algorithm

Seam carving cannot work automatically for all images.

- The error function can fail for certain images.
- User constraints about preserving parts of the image may be very necessary for some images.

The method can also fail if the image is too condensed, that is the image does not contain 'less important' areas.
While there can also be images where the layout of the image content is such that it prevents the seams from bypassing important parts.

In these cases, seam carving can fail and give undesirable results.

VI. Our Implementation of the Algorithm

1. Seam Removal - ***Reducing image size in one dimension:*** *Height/Width shortening* [Section IV-1.]

Horizontal resizing: *removing vertical seams*

While (target width not achieved):

Energy map calculation

- Energy map = sum of absolute values of gradient in x and y direction for R,G,B channels
- Size of energy map = size of the image

Cumulative energy cost matrix calculation(M) [refer Section III.]

- M is found using DP.
- $M(i,j) = e(i,\ j) +$ minimum new neighbor energy introduced by removing one of the three top neighbors of (i,j).

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

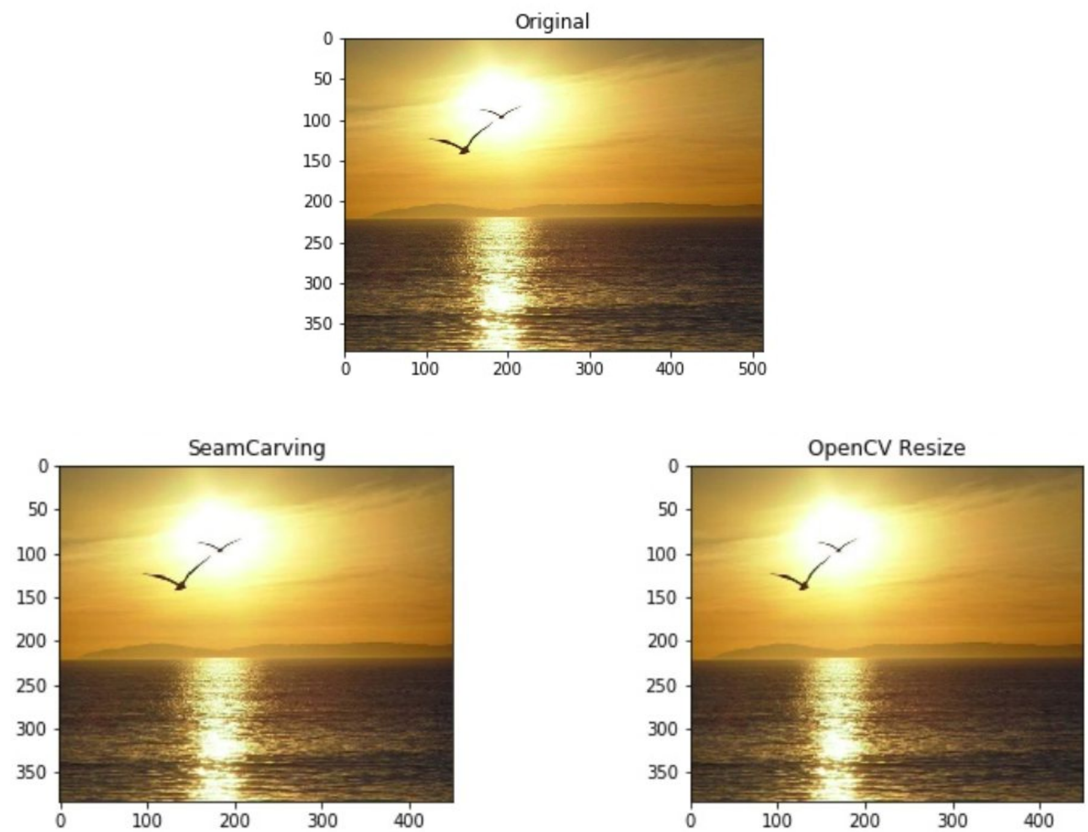
Finding the minimum seam:

- the minimum value of the last row in M will indicate the end of the minimal seam
- backtrack from this minimum entry on M to find the path of the optimal seam

Removing the minimum seam:

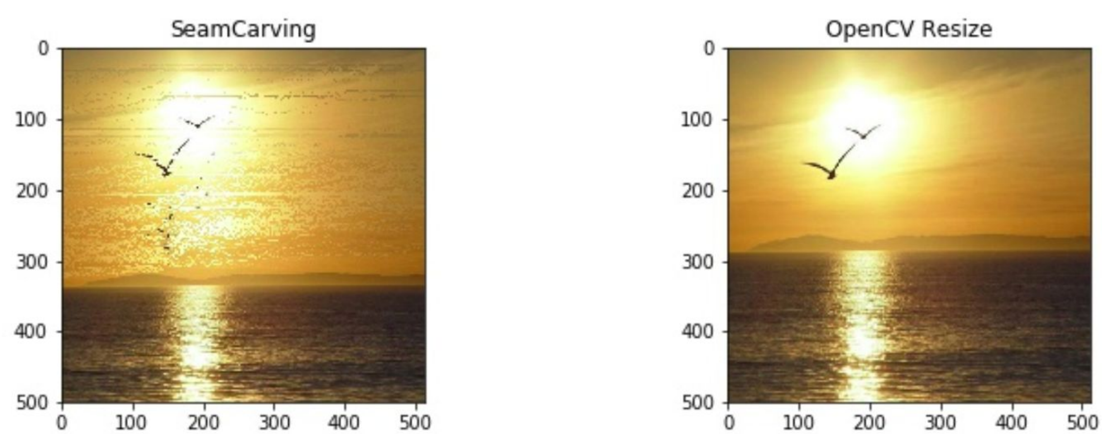
- Pixels in each row after the removed pixel are shifted over one column to the left if it has an index greater than the minimum seam.

End while



Vertical resizing: *removing horizontal seams*

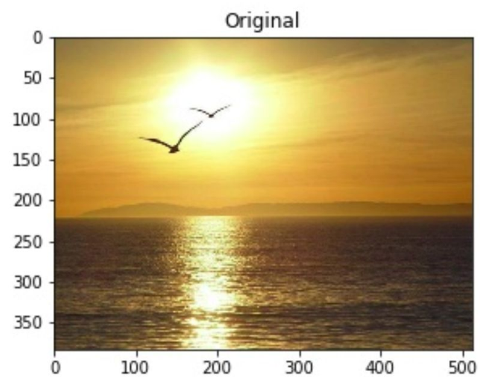
Rotate the target image 90 degrees counter-clockwise and follow the same steps as horizontal resizing.

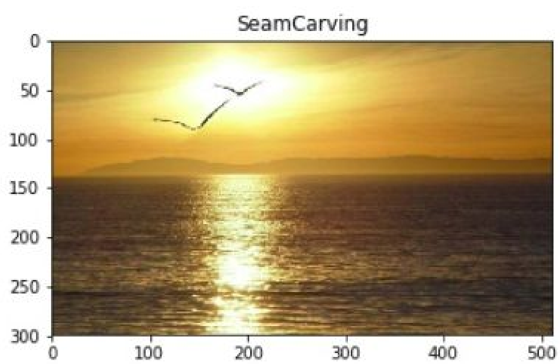


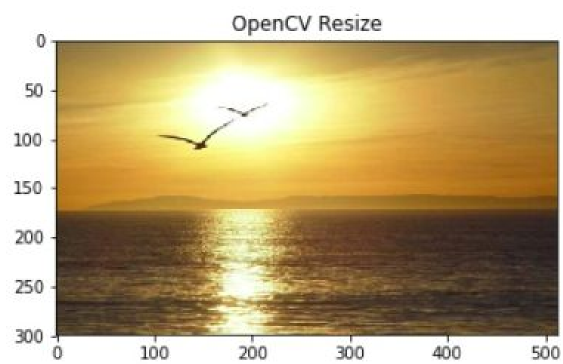
2. **Seam Insertion -** ***Increasing image size in one dimension:*** *Height/Width elongation* [Section IV-3.]

Seam insertion is like the inversion operation of seam removal, where we insert artificial pixels in the image.

- Duplicate the input image
- Perform seam removal for a given number of seams, and record coordinates in order of removal
- Insert new seams in the original input image in same order as that of removal.
- Pixel values for inserted seams = average pixel value of left and right neighbours



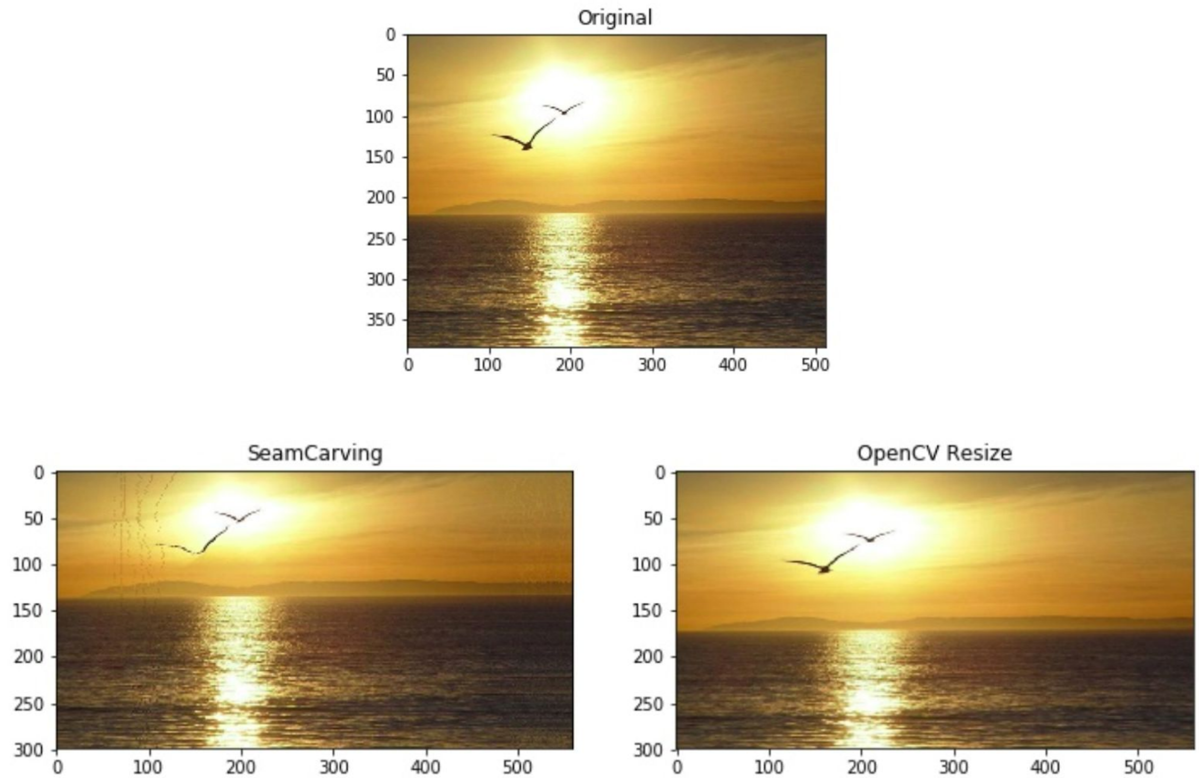




3. **Image retargeting -** *changing both height and width* [Section IV-2.]

For changing both dimensions of the image, we perform seam insertion or removal, based on the target size of the image.

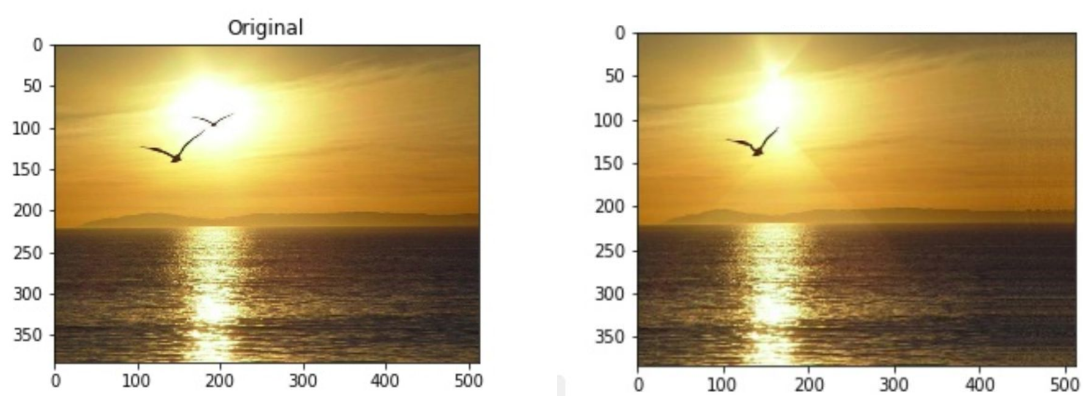
- First, we perform horizontal resizing for the target number of columns
- Then we perform vertical resizing for the target number of rows



4. **Object Removal** [Section IV-4.]

For object removal,

- Remove object by seam removal
    - Generate the energy map such that the region to be removed is weighed with very high negative energy value.
    - Thus, the minimum seam will pass through this target region which is to be removed.
    - Perform seam removal till the target region is completely removed
- Seam insertion to return image back to its original dimension



VII. Results

We have presented some results of our implementation here. We compare the results of resizing with the OpenCV function in python for resizing.

Image Retargeting

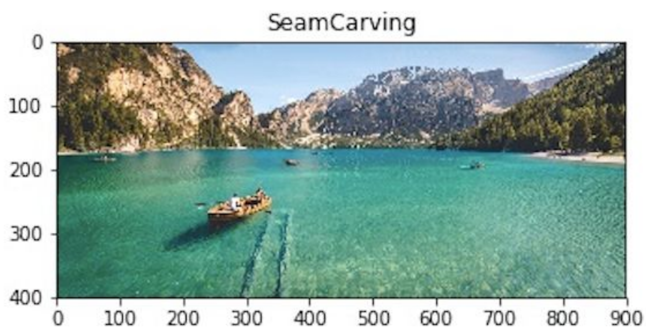
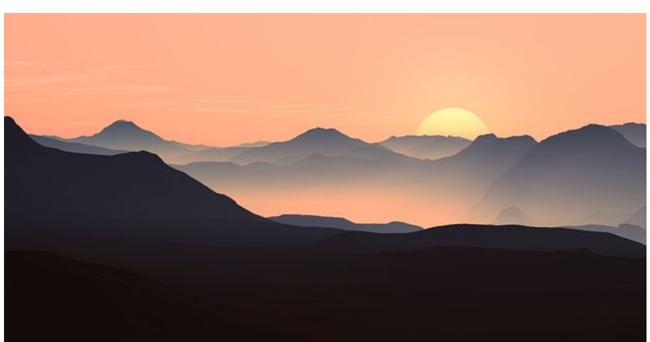
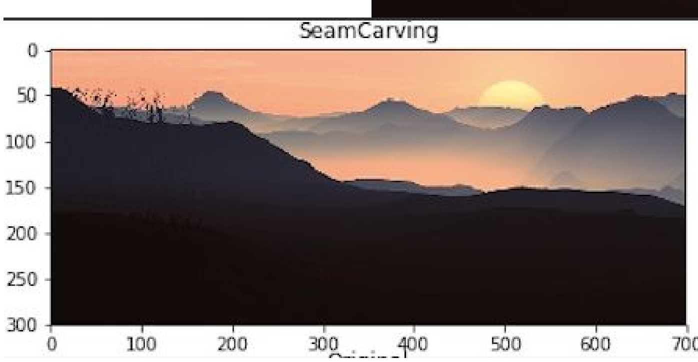
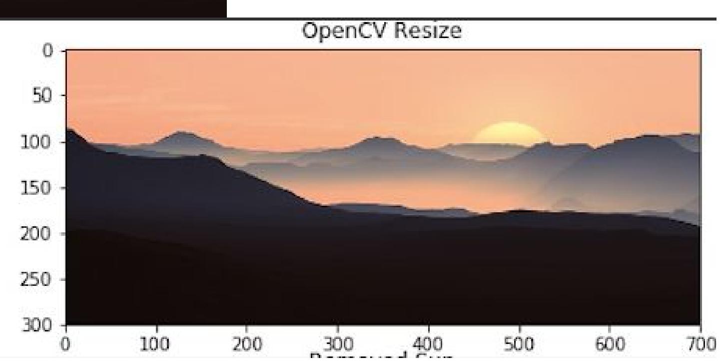


Image Enlargement

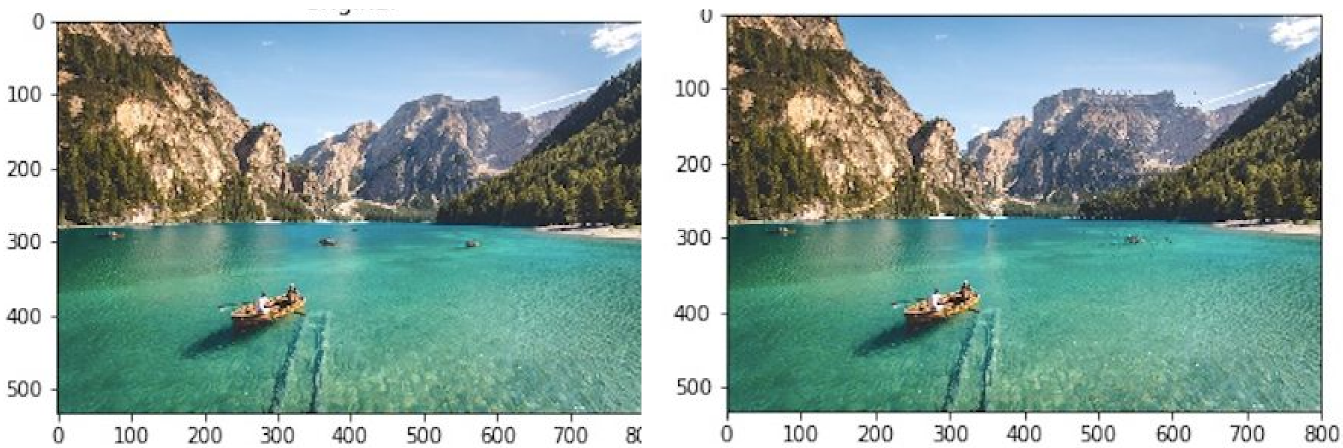




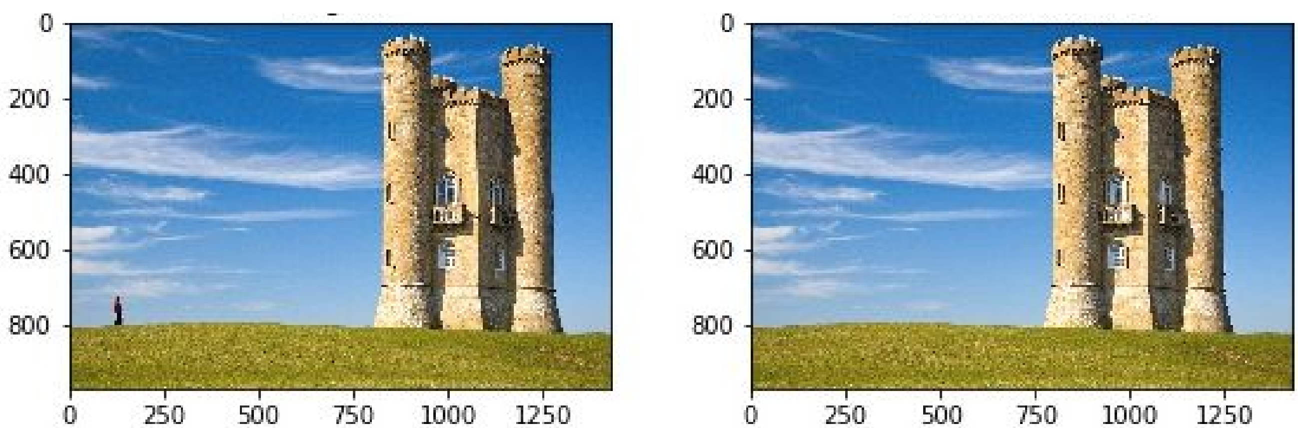


Object Removal

Removed smaller boats



Removed human



Removed pigeon

VIII. Limitations in the implementation

Image resizing: If the image is too condensed, that is the image does not contain 'less important' areas, then seam carving can fail. In the given image, image retargeting to a bigger size leads to a distorted image, because the image has important areas through with seams that introduce artificial pixels are passing.

Object removal: the seam removal process can lead to the removal of important parts of the image sometimes. For example, in the given image, on removing the hot air balloon's reflection from the water through object removal, the seams removed also pass through the hot air balloon in the air, giving a distorted result image. This can be taken care of by effectively selecting seams that do not pass through any 'object of importance' in the image.

IX. Conclusion and Future optimizations

Content-aware resizing of images can be performed using seam carving. It can be used for aspect ratio change, image retargeting, and object removal.

For our implementation, we can make several improvements. The implementation can also be integrated more extensively with user input, to guide the resizing process. We can also find ways to run the object removal algorithm faster. Also, better ways can be used to give input for object removal in our implementation.

X. References

1. VIOLA, P., AND JONES, M. 2001. Rapid object detection using a boosted cascade of simple features. In Conference on Computer Vision and Pattern Recognition (CVPR).
2. ITTI, L., KOCH, C., AND NEIBUR, E. 1999. A model of saliency-based visual attention for rapid scene analysis. PAMI 20, 11, 1254–1259.
3. LIU, F., AND GLEICHER, M. 2005. Automatic Image Retargeting with Fisheye-View Warping. In ACM UIST, 153–162.
4. LIU, F., AND GLEICHER, M. 2006. Video Retargeting: Automating Pan and Scan. In ACM international conference on Multimedia, 241–250.
5. SETLUR, V., TAKAGI, S., RASKAR, R., GLEICHER, M., AND GOOCH, B. 2005. Automatic Image Retargeting. In In the Mobile and Ubiquitous Multimedia (MUM), ACM Press.
6. AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. ACM Trans. Graph. 23, 3, 294–302.
7. JIA, J., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2006. Drag and-drop pasting. In Proceedings of SIGGRAPH.
8. ROTHER, C., BORDEAUX, L., HAMADI, Y., AND BLAKE, A. 2006. Autocollage. In Proceedings of SIGGRAPH 2006.
9. WANG, J., AND COHEN, M. 2006. Simultaneous Matting and Compositing. Microsoft Research Technical Report, MSR-TR2006-63 (May).
10. ZOMET, A., LEVIN, A., PELEG, S., AND WEISS, Y. 2005. Seamless image stitching by minimizing false edges. IEEE Transactions on Image Processing 15, 4, 969–977.
11. BERTALMIO, M., VESE, L., SAPIRO, G., AND OSHER, S. 2003. Simultaneous structure and texture image inpainting. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 707–714.
12. DRORI, I., COHEN-OR, D., AND YESHURUN, Y. 2003. Fragment-based image completion. In Proceedings of ACM SIGGRAPH, 303–312.

13. CRIMINISI, A., PEREZ, P., AND TOYAMA, K. 2003. Object removal by exemplar-based inpainting. In IEEE Conference on Computer Vision and Pattern Recognition, 417–424.
14. Avidan, S. and Shamir, A., 2007. Seam carving for content-aware image resizing. In ACM SIGGRAPH 2007 papers (pp. 10-es).
15. GitHub: https://github.com/vivianhylee/seam-carving