Assignment 03: Seam Carving for Content-Aware Image Resizing

1 Implementation of Seam Carving Algorithm:

- Main Octave Script: (`script.m`)
  First run this script.

```
clear
pkg load image
tic()

I = imread('boat.jpg');

%Removing 200px from X direction
for i = 1:1:200
    I = seam_carving_x_direction(I);
end

%Transpose the color image
I = transpose_color(I);

%Removing 200px from Y direction
for i = 1:1:200
    I = seam_carving_x_direction(I);
end

imwrite(transpose_color(I),'out.jpg');
toc()
```

- Octave Script for Color image Transpose: (`transpose_color.m`)

```
function result = transpose_color(I)
    % The function will do Channel-wise(RGB) matrix transpose for ...
       the image I.
    % The transposed color image will be returned as 'result' image.

    result(:,:,1) = I(:,:,1)';
    result(:,:,2) = I(:,:,2)';
    result(:,:,3) = I(:,:,3)';
end
```

- Octave Script for Seam Carving: (`seam_carving_x_direction.m`)

```
function result = seam_carving_x_direction(I)
%{
Function that finds a Top-down (vertical) seam in the image I and ...
    removes the seam. Reduce one pixel in X direction and return ...
    the image as 'result' matrix.
%}

    I_gray = rgb2gray(I);
    [h w] = size(I_gray);

    %Consider L1 norm as Energy.
    %Calculate energy in the grayscale version of the image.
    dx = conv2(I_gray,[1 -1],'same');
    dy = conv2(I_gray,[1 ;-1],'same');
    Energy = abs(dx) + abs(dy);

    %Initialization
    Min_energy = zeros(h+1,w+2);
    Min_energy(:,1) = Min_energy(:,w+2) = inf;

    %Store the vertical seam directions left(-1), straight (0) and ...
        right (-1)
    direction = zeros(h,w);

    %Dynamic programming for the seam finding
    for i = 1:h
        [min_value, index] = min([Min_energy(i,1:w); ...
            Min_energy(i,2:w+1); Min_energy(i,3:w+2)]);
        Min_energy(i+1,2:w+1) = Energy(i,1:w) + min_value;
        direction(i,:) = index-2;%Shifting 1,2,3 to -1,0,1
    end

    %Find the min energy from the bottom row
    [val, col] = min(Min_energy(h+1,:));
    col = col - 1;

    %{
    BACKTRACKING(Finding the optimal energy path).
    At the same time converting subscript into a row-major linear ...
        index. I am unable to do it using sub2ind function becuase ...
        it gives coloumn-major linear index. But I need row-major ...
        linear index and i do it with a simple equation.
    %}
    linear_ind = zeros(h,1);
    for i = h:-1:1
        linear_ind(i,1) = (i-1)*w+col; %Row-major linear index.
        col = col + direction(i,col);
    end

    %Channel(RGB)-wise seam removing. Iterates 3 times only.
    for i = 1:1:3
        C = I(:,:,i)'(:);
        C(linear_ind) = []; %Deleting the whole seam
        result(:,:,i) = reshape(C,w-1,h)'; %Reshaping (Row major order)
    end
end
```

- Input Image: (`boat.jpg`)

Figure 1: Input image (470 x 640).

- Output Image: (`out.jpg`)

Figure 2: Output image (270 x 440) after removing 200px from both X and Y dimension using Seam Curving algorithm.

2 Questions and Answers:

- **Question 1:** *Discuss the implications of the energy function. For example, what happens if you maximize rather than minimize energy? What are some other functions you could think of?*

  ANSWER:
  The energy function (L1 norm of gradient) is for finding visual salient pixels. Through energy function we can measure which pixel is less noticeable/important (having minimum energy) so that we can remove a seam containing less important points. Figure 3. shows a vertical seam (white color) having minimum energy.

  Figure 3: Image showing a vertical seam (white color) having minimum energy.

  If I maximize the energy, the algorithm will give a seam having maximum energy. That seam will contain maximum number of salient pixels (most of them are edge pixels). Figure 4. shows a vertical seam (white color) having maximum energy.

  Some other energy functions [1] that can be used in seam carving algorithm are given below.

  – L2 norm of gradient (1st order derivative/Sobel/Prewitt)
  – Harries corners measure
  – Entropy
  – Saliency measure
  – Histogram of Gradient(HoG)

  Figure 5. shows the first vertical seam (white color) for different energy functions. Figure 6. shows the output images after removing 200px from both X and Y dimension using Seam Curving algorithm for different energy functions.

Figure 4: Image showing a vertical seam (white color) having maximum energy.

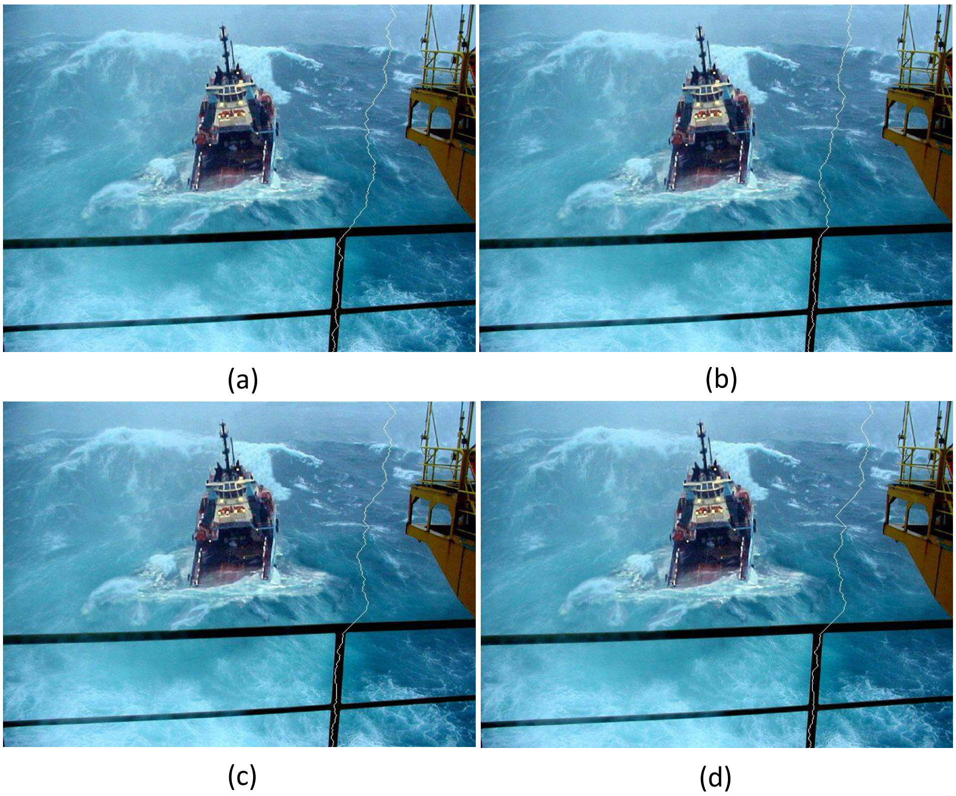


Figure 5: Images showing the vertical seams (white color) for different energy functions (a) L2 norm of 1st order derivative, (2) Harries corner measures, L2 norm of (3) Prewitt gradient and (4) Sobel gradient. We can observe slight differences among the path of the seams.

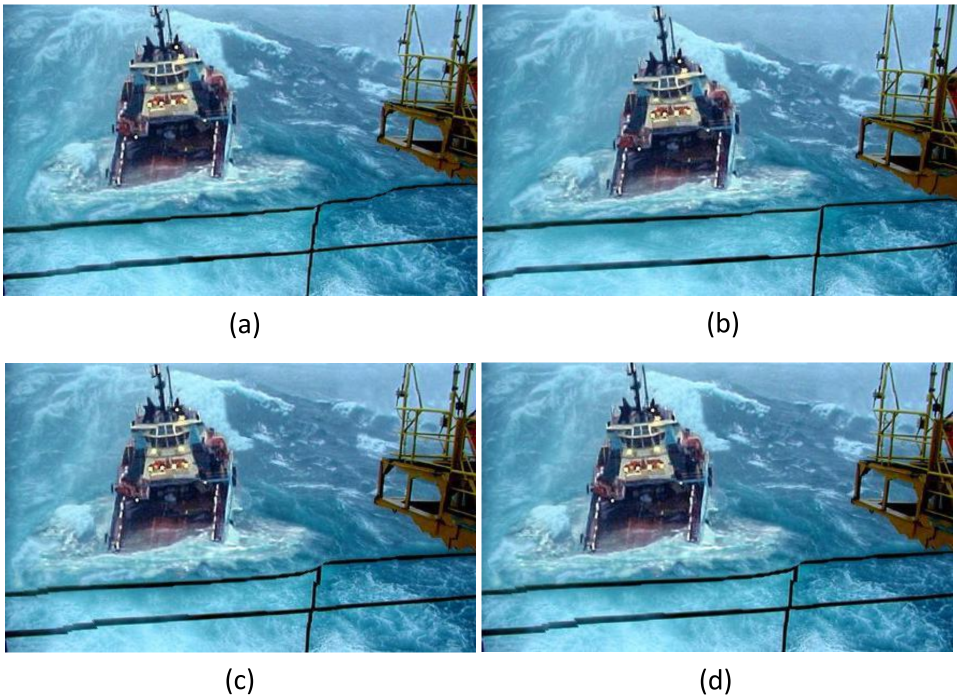
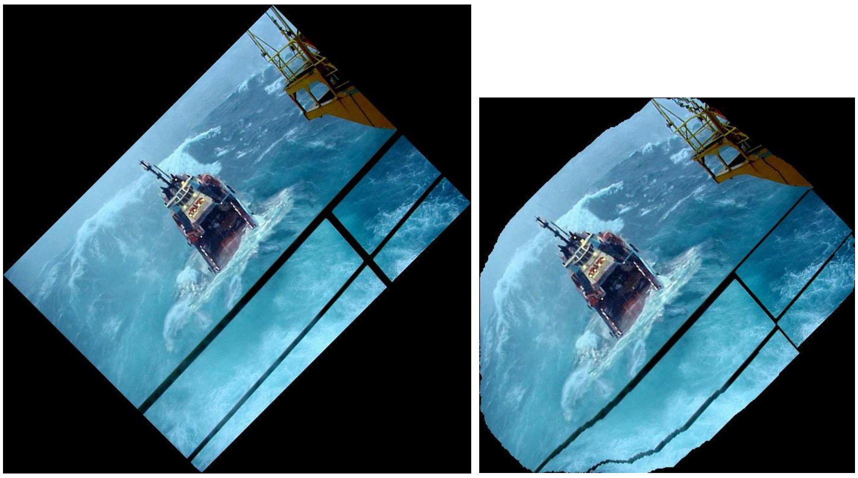


Figure 6: Output images after removing 200px from both X and Y dimension using Seam Curving algorithm for different energy functions (a) L2 norm of 1st order derivative, (2) Harries corner measures, L2 norm of (3) Prewitt gradient and (4) Sobel gradient. We can observe slight differences among the output images.

- **Question 2:** *Reflect on the following questions*
  (a) How could this algorithm be used to grow an image rather than shrink it?
  (b) What happens if we choose non-axis-aligned seams (i.e., rotate the image first or follow a different seam path structure)?
  (c) When would seam carving NOT be a desireable alternative to scaling or cropping?

  ANSWER:
  (a) If we want to grow an image by k pixels then first we need to do k seam carving operations but without deleting the seam. We only need to store the k seam-path's information. Finally duplicate these seams in the same order. In this way, the seam removal algorithm can be used to grow an image by k pixels (in any directions) [1].

  (b) If we choose non-axis-aligned seams (suppose 45 degree), then we will get undesirable output like most of the seams are carved only from the four corners of the image). This is because of the seams near the four corner-region of the image will have less number of pixels on its path which make the path optimal. Moreover, in general four corners of an image does not contain much salient pixels. So almost every iteration, these seams are selected for removal. In Figure 7. image (a) is the 45 degree rotated input image and (b) is the output image after carving the seams from the image (a) by 200px in both X and Y directions. We can see in the image 7(b) that the four corners of the image are carved mostly. Moreover its actually not effecting the height and width of the actual image.



(a) (b)

Figure 7: (a) 45 degree rotated version of the image in Figure 1. (b) The output image after applying 200px seam carving in both X and Y directions.

(c) The seam carving algorithm will not be a desirable alternative to scaling or cropping when the input image is highly dense. That means the image does not contain any less important areas. In this case, there is no way to avoid artifacts. Doing scaling is better in this situation [1].

Another case [1] is that the input image is not densed but has such layout where seams has no way to bypass the important pixels of the image. For example the image in Figure 1 has such layout where vertical seams has no way to bypass the ship's metal railing.

- **Question 3:** *How far does it make sense to carve an image? What happens as the image gets smaller?*

  ANSWER:
  How far we carve an image (that make sense) is actually depends on the content of the image. We can carve as far as the salient content of the image is not affected significantly (not so much artifacts).

  If we carving continuously, the image will get smaller and the artifacts will increases more, because seams are bound to go through the important contents of the image. Figure 8. shows the effect of increasing the number of seam carving by 50 pixels on both directions on the image of Figure 1.

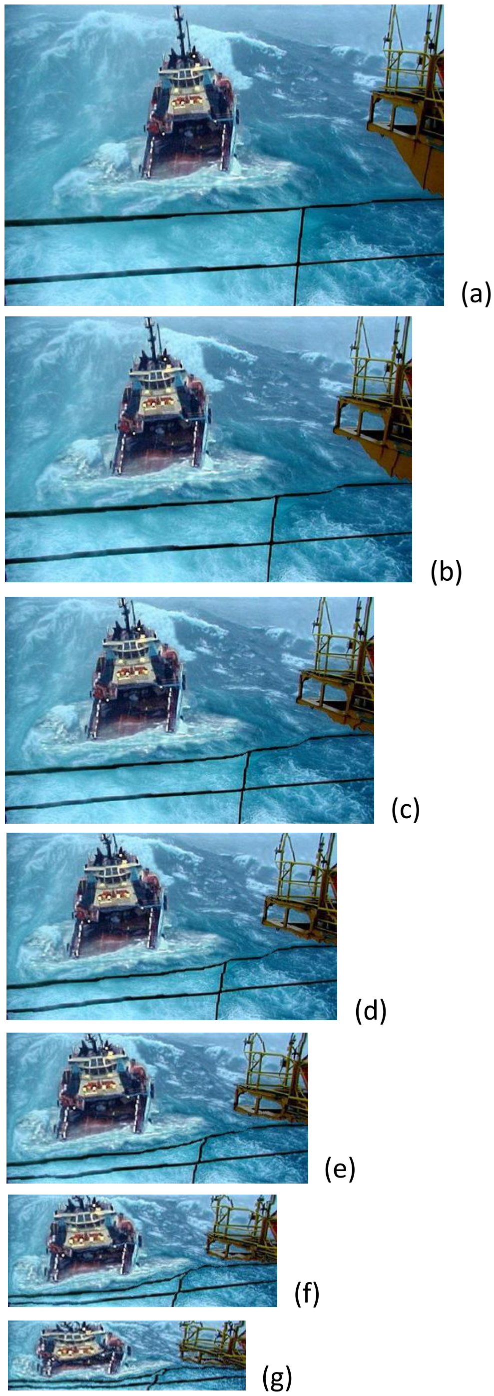


Figure 8: Output images after (a) 100px, (b) 150px, (c) 200px, (d) 250px, (e) 300px, (f) 350px and (g) 400px seam curving on the image in Figure 1 in both X and Y directions. The amount of artifacts are increasing with the amount of seam carving. Here, image (a) is acceptable that contains less amount of artifacts.

- **Question 4:** *Explain any artifacts in your final resulting image. Why did these happen and what can be done about them?*

  ANSWER:
  Their are lots of noticeable artifacts in the output image (after carving the seams from the image in Figure 1 by 200px in both X and Y directions). In Figure 9, some of the significant artifact regions are marked (using red box) as 1, 2 and 3. In region 1, the boat is actually reduced in size in vertical direction. In region 2, the ship's metal railing looks distorted. In region 3, the lower portion of the yellow object is removed/cropped significantly and also the object is shrunk vertically.

  All these artifacts happen due to the fact that some of the seams are passing through the important regions of the image. So after removing those seams, it make the regions distorted or cropped.

  To get rid of this problem, we can block the seams from passing through the region of interest(ROI) by putting some user constraints (manually) over the content of the image [1]. We can easily do it by putting infinite values in the energy matrix based on the pixel locations of ROI. In Figure 10, image (b) is showing the region of interest locations (the boat and the right yellow object) and the image (d) shows the output image after applying the user constraint and the seam carving (by 200px in both X and Y directions). We see that the output image (d) is much better (less artifacts on ROI) than the regular seam carving output image (c).

Figure 9: Some of the significant artifact regions in the output image that are marked (using red box) as 1, 2 and 3.

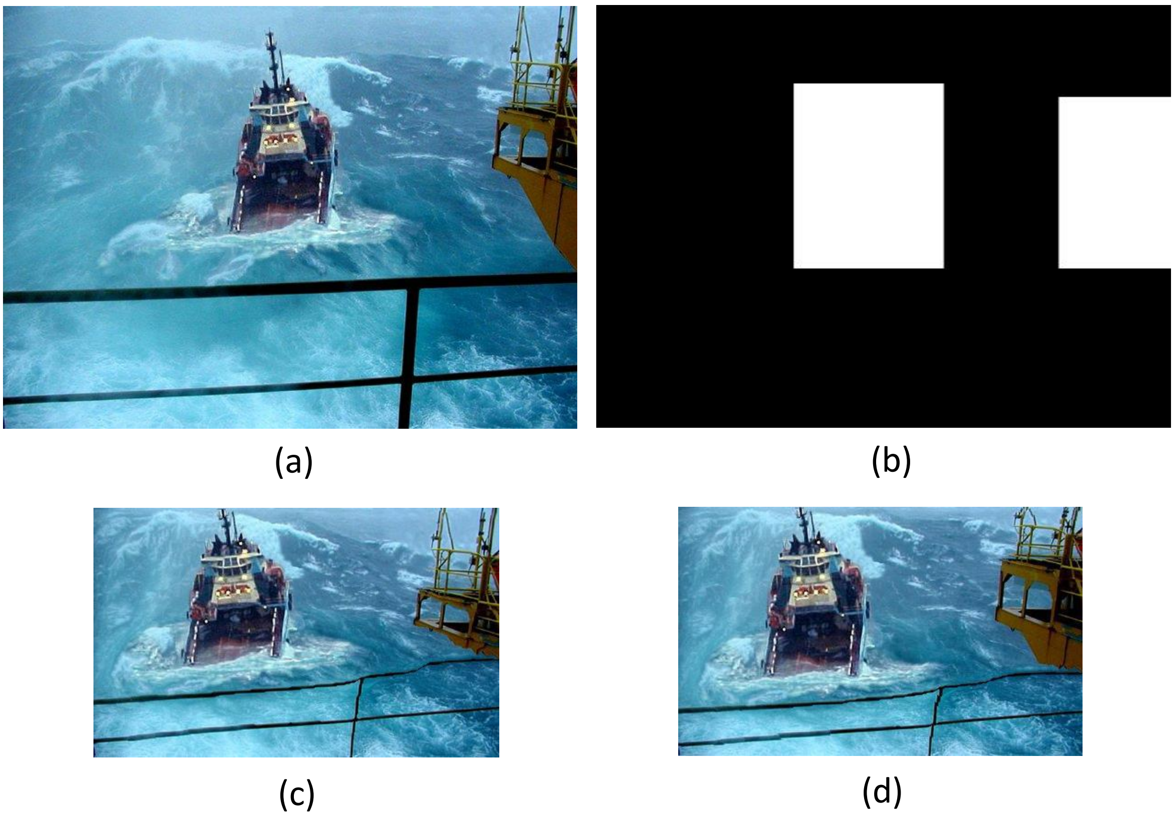


Figure 10: (a) The input image and (b) the region of interest (white area) locations where the user have assigned constraints. (c) The output image without using user constraint and (d) the output image with user constraints. We see that the output image (d) is much better (less artifacts on ROI) than the regular seam carving output image (c).

Bibliography

[1] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM transactions on Graphics, SIGGRAPH, 2007*, vol. 26(3), 2007.