CS 6476 - Computer Vision

Problem Set 1

Harsh Bhate (903424029)

Problem 1B (Using Python)

1.

The associative property of convolution helps in efficiently computing filter outputs especially in case of multiple filtering operations. Let us assume an image, I of size $M \times N$, being filters by a bank of filters, $[f_1, f_2]$ of size $m \times n$ such that $M >> m$ and $N >> n$. The filtering follows the order:

$$I_{out} = (f_1 * I) * f_2$$

However, because of the associative property of convolution,

$$I_{out} = I * (f_1 * f_2)$$

In this case, the number of steps required to compute the convolutions $f1 * f2$ (say, N_1) is much less than the number of steps required to compute $(f_1 * I)$ (say, N_2). In other words, $N_1 << N_2$. Thus, the operation as expressed by the topmost equation would take $2N_2$ steps while the operation expressed by the latter equation would take $N_2 + N_1$ steps.

2.

$f[n] = [00110011]$ when dilated with a structuring element [111] results in the following signal:

$$\boxed{h[n] = [01111111]}$$

3.

We have, $f' = [0, -\frac{1}{2}, 0, \frac{1}{2}, 0]$. The second derivative, g'', of a signal g can be computed by:

$$g'' = f' * (f' * g) = (f' * f') * g$$

Thus, $f'' = f' * f'$ by associative property.

$$f'' = \left[0, 0, \frac{1}{4}, 0, -\frac{1}{2}, 0, \frac{1}{4}, 0, 0\right]$$

ALITER

Another tedious but worthwhile effort to find the second derivative is to use a First order finite difference method on the Taylor series expansion resulting in the following:

$$g'' = \frac{g(x+1) - 2g(x) + g(x-1)}{1^2}$$

This results in the following filter:

$$f'' = [1, -2, 1]$$

4.

- The presence of edges indicates a high concentration of high frequency components in the image. Thus, to reduce the number of fine, detailed edges a blurring filter such as Gaussian filter can help reduce the high frequency components.
- Theshold the values of the edge to eliminate weak edges as weak edges are usually the finer detailed components in the image.
- Implement a strong thinning technique such as Non-Maximal Suppression to reduce the number of edges.

5.

An additive white noise has the following properties:

$$\eta = \eta_1 + \eta_2 + \eta_3 + \dots$$

where η refers to the total noise and η_n refers to a noise components. Let us now assume an image, I, being corrupted by a noise component, η_1 represented by a Gaussian distribution with standard deviation σ_1. Next, let us assume that another noise component, η_2 represented by a Gaussian distribution with standard deviation σ_2 is added. In this case,

$$\eta = \eta_1 + \eta_2$$

By the additive property of Gaussian noise, the new noise signal η is also a Gaussian with a standard deviation, η, given by:

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$$

It is evident that $\sqrt{\sigma_1^2 + \sigma_2^2} \leq \sigma_1 + \sigma_2$. Thus, the new noise signal ends up having a shorter deviation as compared to the individual noises. This presents a problem as the addition of new noise may end up reducing the noise profile in the signal.

6.

While the design of the system is an open-ended question, the following steps are recommended:

Imaging Sensor and Design

- Calibrate the camera to ensure consistent and respectable image capture.
- Apply camera noise correction algorithms such as color correction, lens distortion correction, Gaussian filter/Shot noise correction, median filter etc.

Problem Space Analysis and Design Decisions

- Perform an image analysis to find metrics such as Peak Signal to Noise Ratio, mean intensity, color-space analysis.
- Find the color, intensity characteristics of the conveyor belt. Subsequently subtract or threshold the hues corresponding to the conveyor belt from the image to extract the object pixels with greater contrast.

Algorithms based on Specific problems

- **Texture Detection.** Using gradient filters such as sobel, prewitt to find the fine edges in the image. Subsequently, applying erosion and dilation to better characterize the texture space. Finally counting the number of individual lines/pixels to ensure the product is within tolerance of normal sample.
- **Structure Detection.** Using edge detection algorithms such as Canny edge detection followed by Non-maximal suppression for structure estimation. Finally, subtracting the Canny Edge output with a bank of images comprising of Canny edge output of a normal sample across various orientations. In case any of the samples in the filter banks show a minimal level of difference, label the sample as normal.
- **Area/Volume Estimation.** In case of volume/area estimation, apply erosion followed by dilation to better assemble the areas corresponding to the object. Subsequently, count the number of pixels in the blobs obtained. Classify the sample as normal if the number of pixels are within tolerance of an ideal sample.
- **Color Accuracy Detection.** Convert the image space to HSV instead of RGB. Next, collect samples of HSV values of a large samples of "acceptable" products. Label the image as faulty if the HSV value of the sample is far off from the bank of "acceptable" HSV values.

Problem 2 (Using Python)

1.

(c).

Figure 1: Test image 1 with 100 pixel width reduction

(d).

Figure 2: Test image 2 with 100 pixel width reduction

2.

(c).

Figure 3: Test image 1 with 100 pixel height reduction

(d).

Figure 4: Test image 2 with 100 pixel height reduction

3.

(a).

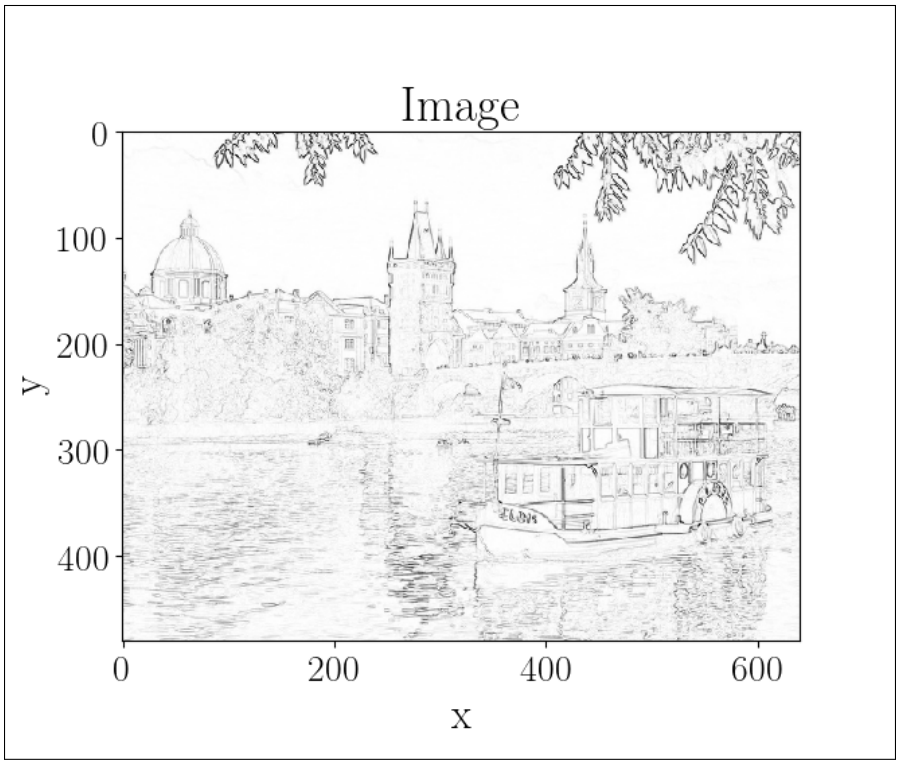

Figure 5: The Energy Function Output of the Test Image

The output of the gradient mainly shows contrast between the edges. This is because the image is mostly comprised of closed edges enclosing an almost uniform texture. However, the water shows a difference because of the texture.

(b).

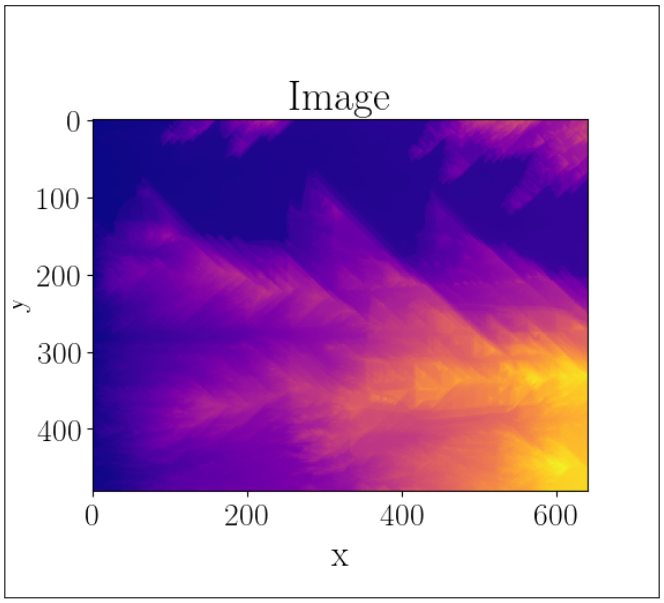

Figure 6: The Cumulative Energy Map along the Horizontal Direction

The cumulative energy map along the top direction shows little energy. This is because the top area of the image is sky which almost no texture and thus, no gradient. In the bottom, the presence of highly textured object like the ship results in a "hot" heatmap along the bottom-right.

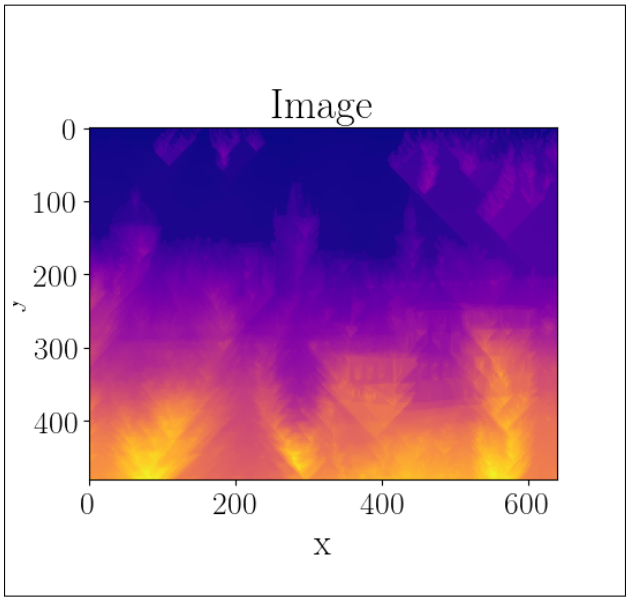

Figure 7: The Cumulative Energy Map along the Vertical Direction

The cumulative vertical energy map shows little variation. This is because the image is almost consistent in the texture variation along the columns.

4.

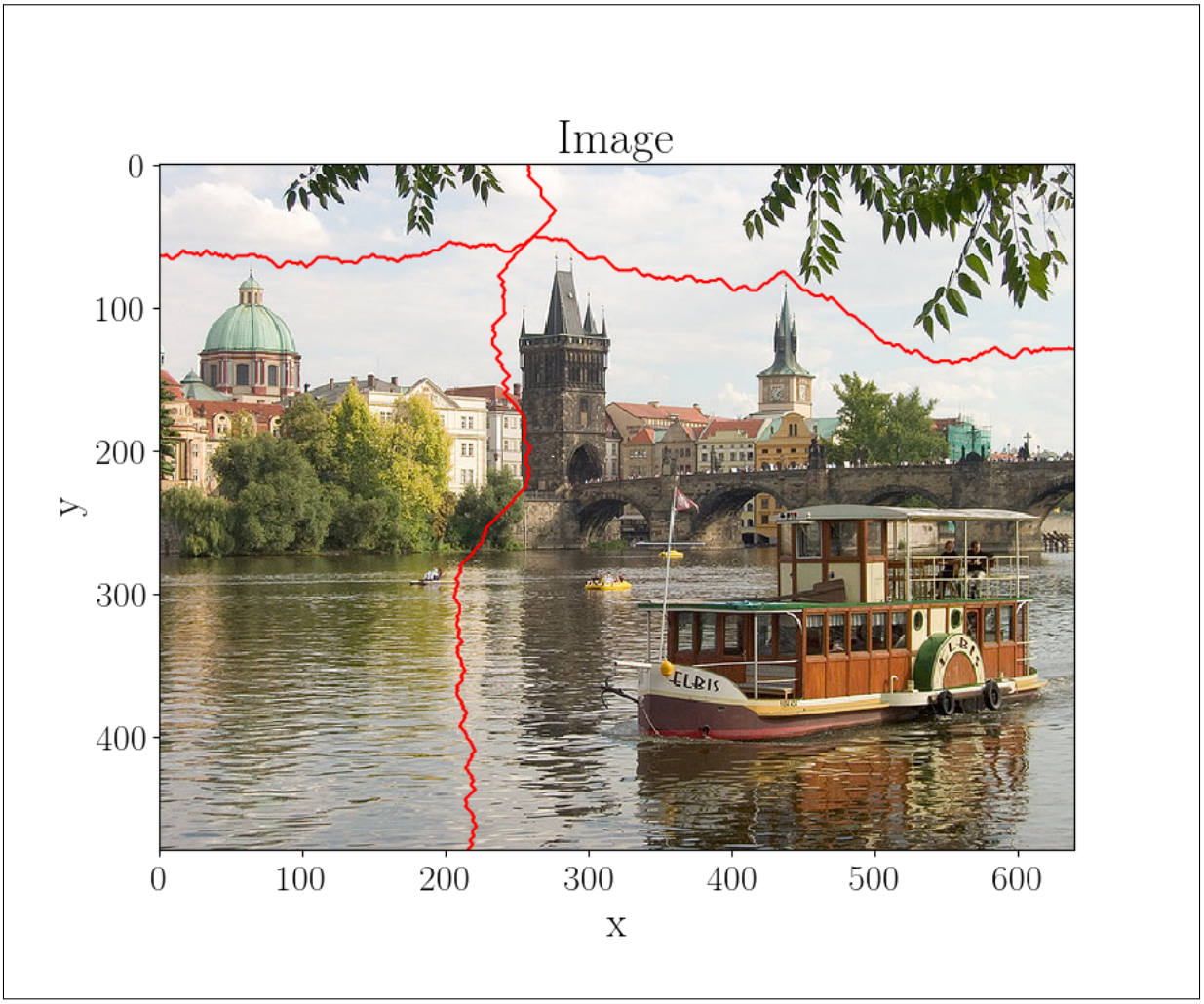

Figure 8: The Optimal Seam

The vertical seam is the optimal vertical seam because it traverses through a region of low texture. From a visual analysis, the seam passes from the sky to the region of uniform texture near the solid building and goes on to terminate through water.

The horizontal seam traverses through the sky and thus shows the minimal change in texture. Thus, the path traced by the horizontal seam appears optimal.

5.

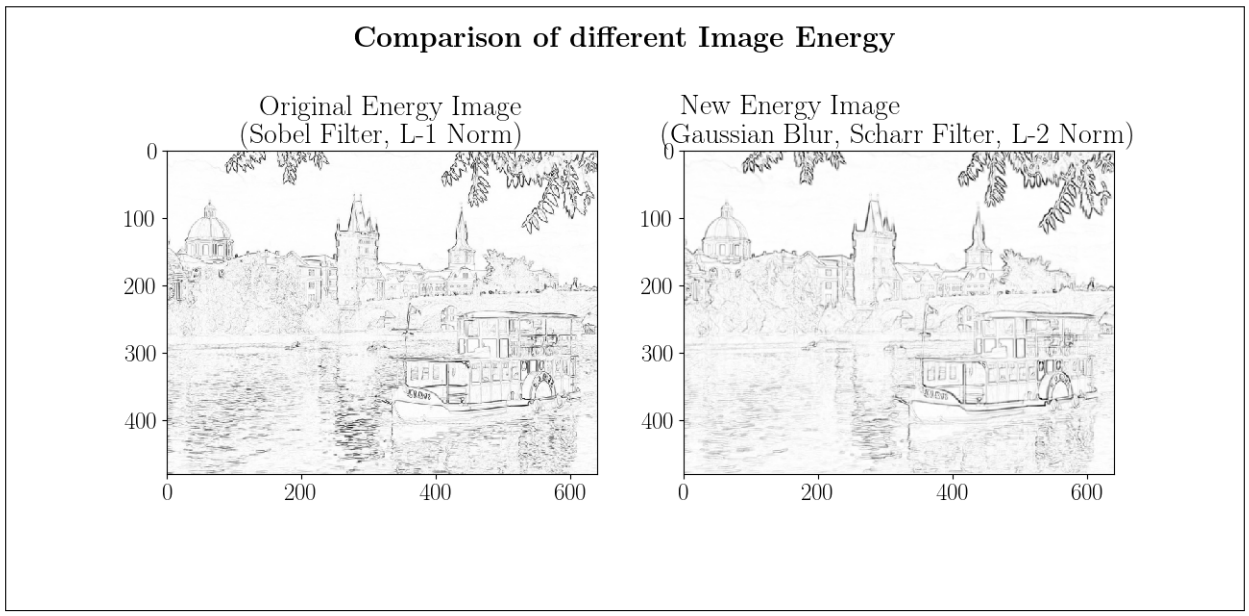

Figure 9: Comparisons of different method of energy calculation

Figure 10: Output of the reduced height using new image energy

Upon observation of the image and its energy, it is evident that the sharp contrasts offered by the waves in the water and buildings result in incorrect contextual seam generation. To avoid that, a Gaussian filter was applied to the image to reduce the fine details in the image. Following that, a Scharr filter was used to better highlight the contrast in the gradient. Finally, to further smoothen out the effects of both x and y gradient, L-2 norm was implemented. The final result through such a computation scheme results in a better image geometry with minimal distortion.

6.

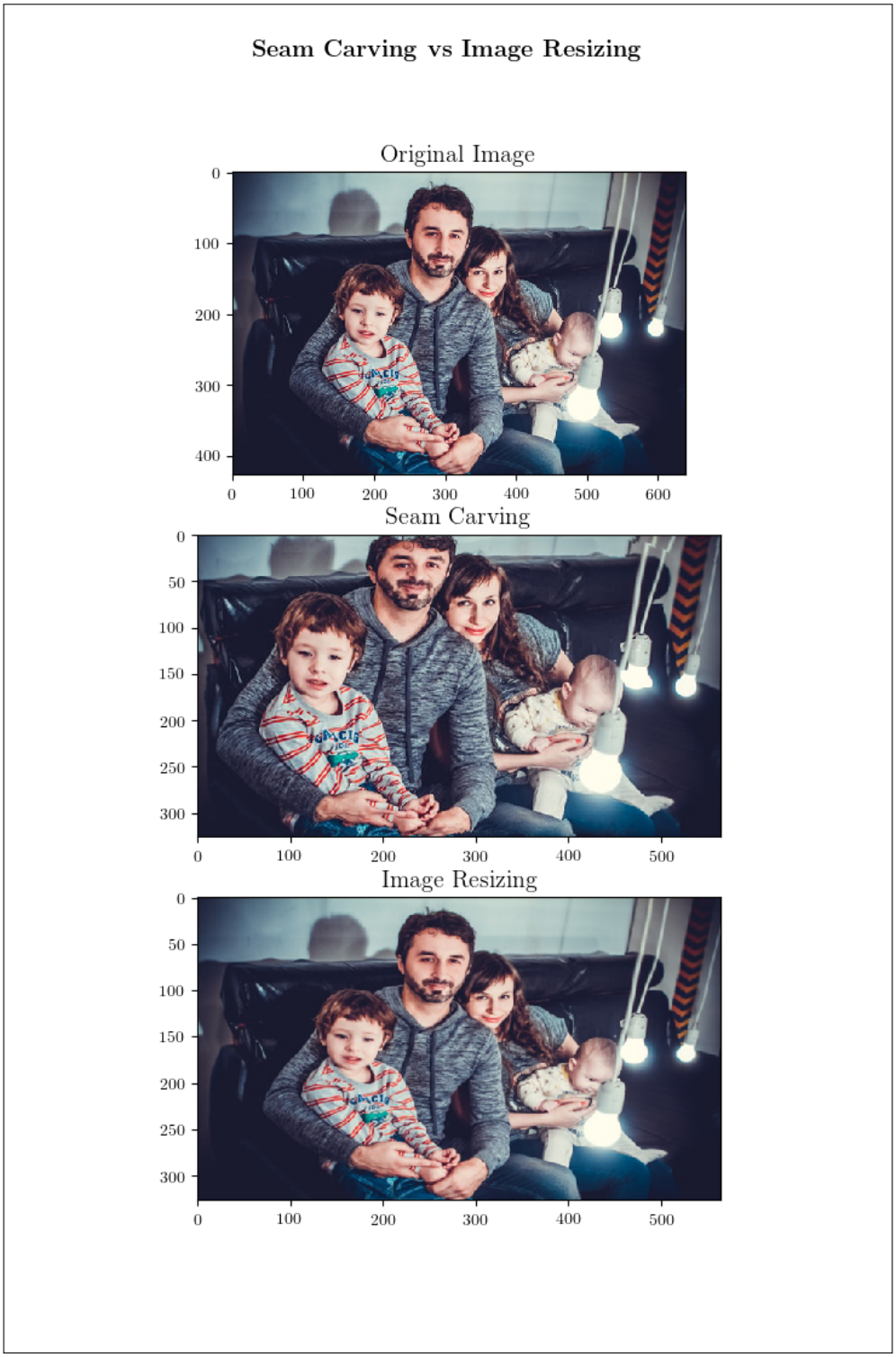

Figure 11: Seam Carving vs Image Resizing [Original Image ($425 \times 640 \times 3$) and Reduced Image($350 \times 540 \times 3$)]

In this example, the seam carving algorithm does a poor job of reducing width. In this case, the sequence of removal was height removal followed by width removal. As evident in the photo, the seam carving algorithm failed to maintain the aspect ratio of the face resulting in a squished face. The standard resize function does a good job of maintaining structure in this case.

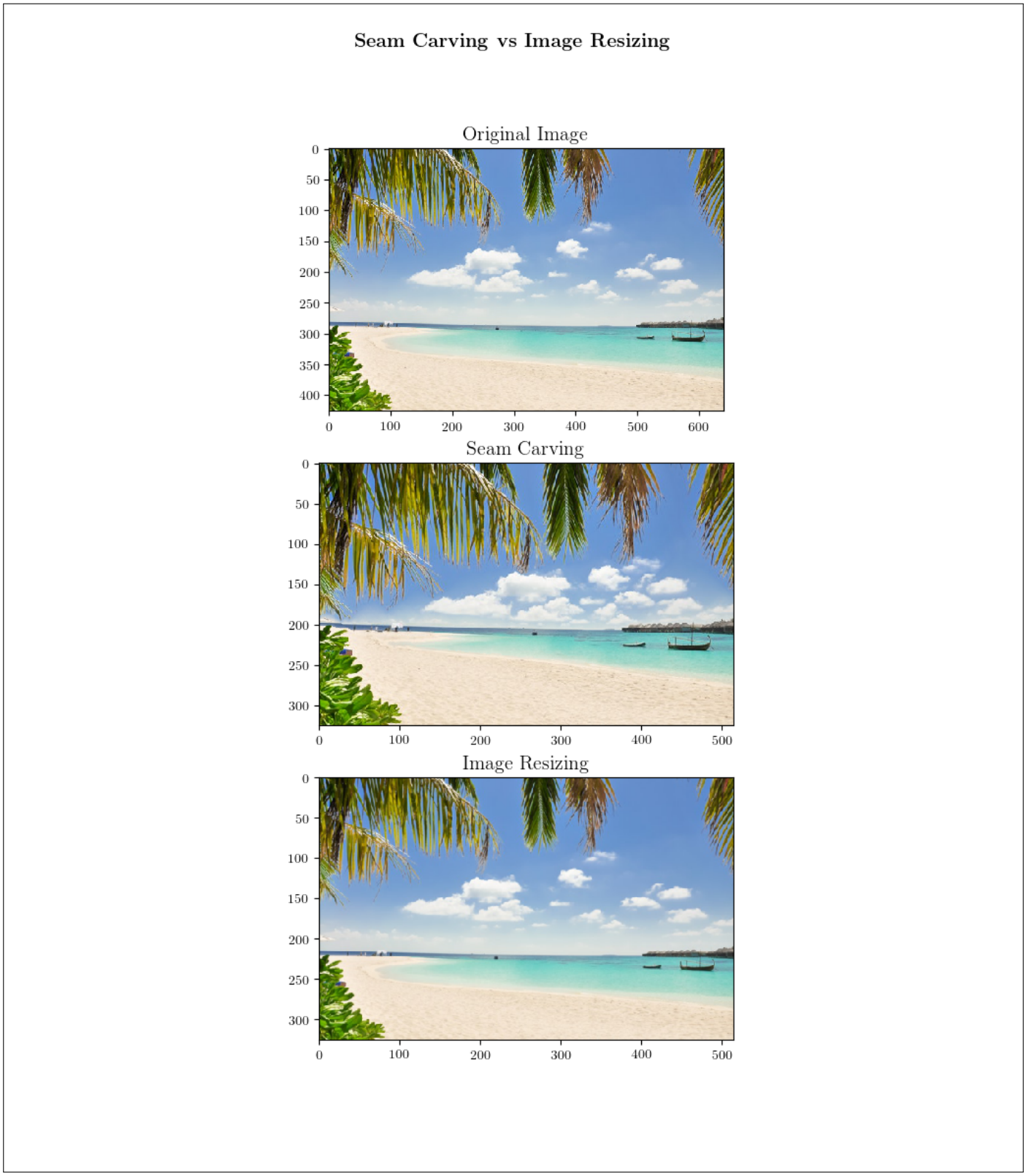

Figure 12: Seam Carving vs Image Resizing [Original Image ($425 \times 640 \times 3$) and Reduced Image($325 \times 515 \times 3$)]

In this example, the seam carving algorithm does a decent job of reducing width. In this case, the sequence of removal was width reduction followed by height reduction. As evident in the photo, the seam carving algorithm failed to maintain the geometry structure of the beach front. On the other hand, the seam carving algorithm did a good job of maintaining the boat and leaf size.

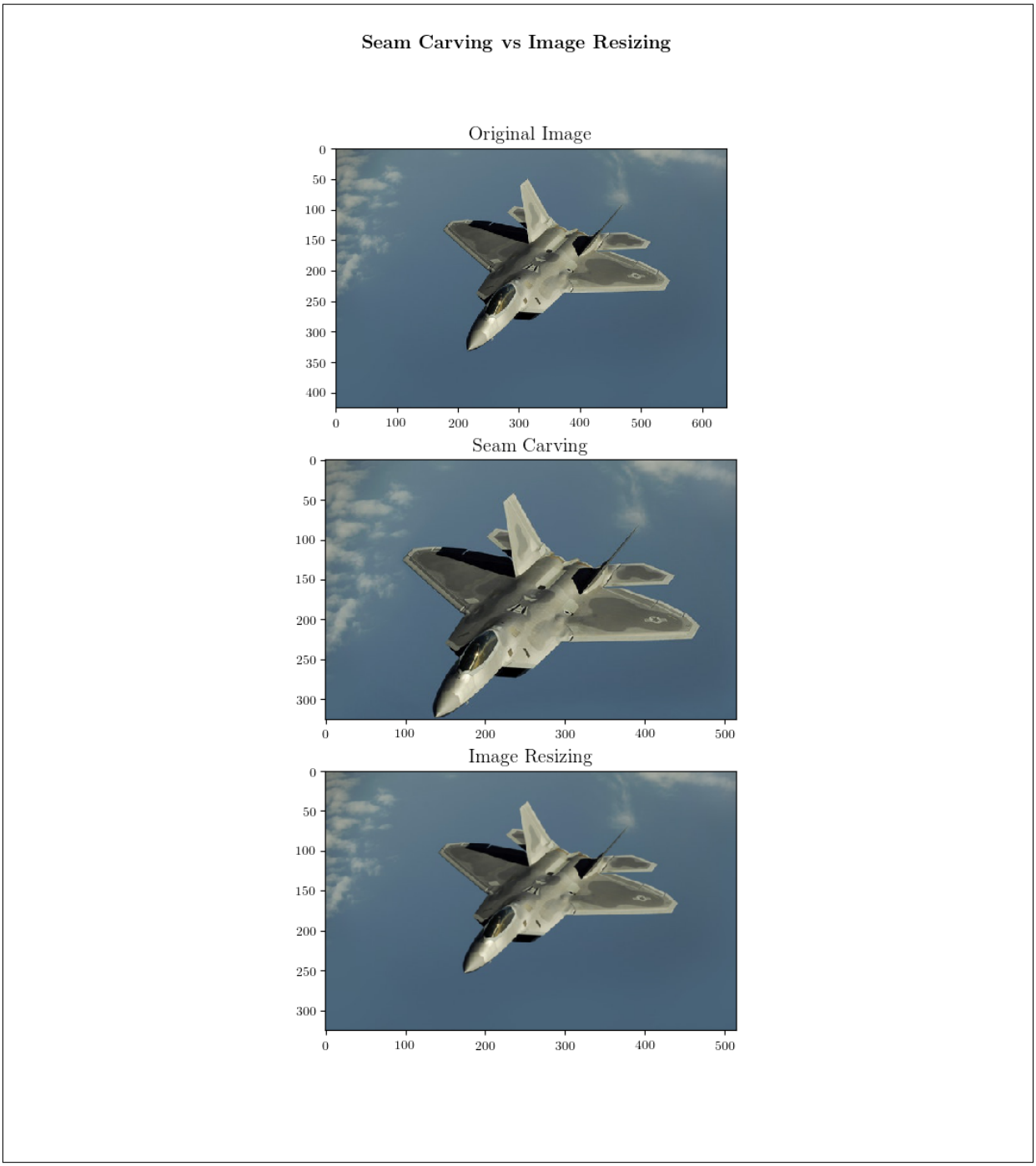

Figure 13: Seam Carving vs Image Resizing [Original Image ($425 \times 640 \times 3$) and Reduced Image($325 \times 515 \times 3$)]

In this example, the seam carving algorithm does an excellent job of resizing. The sequence of removal was height reduction followed by width reduction. As evident, the seam carving alogirthm excellently removed the sky from the picture while maintaining the details in the airplane. The standard resizing algorithm, on the other hand, modified the aspect ratio of the fighter jet and reduced the size of the object too.