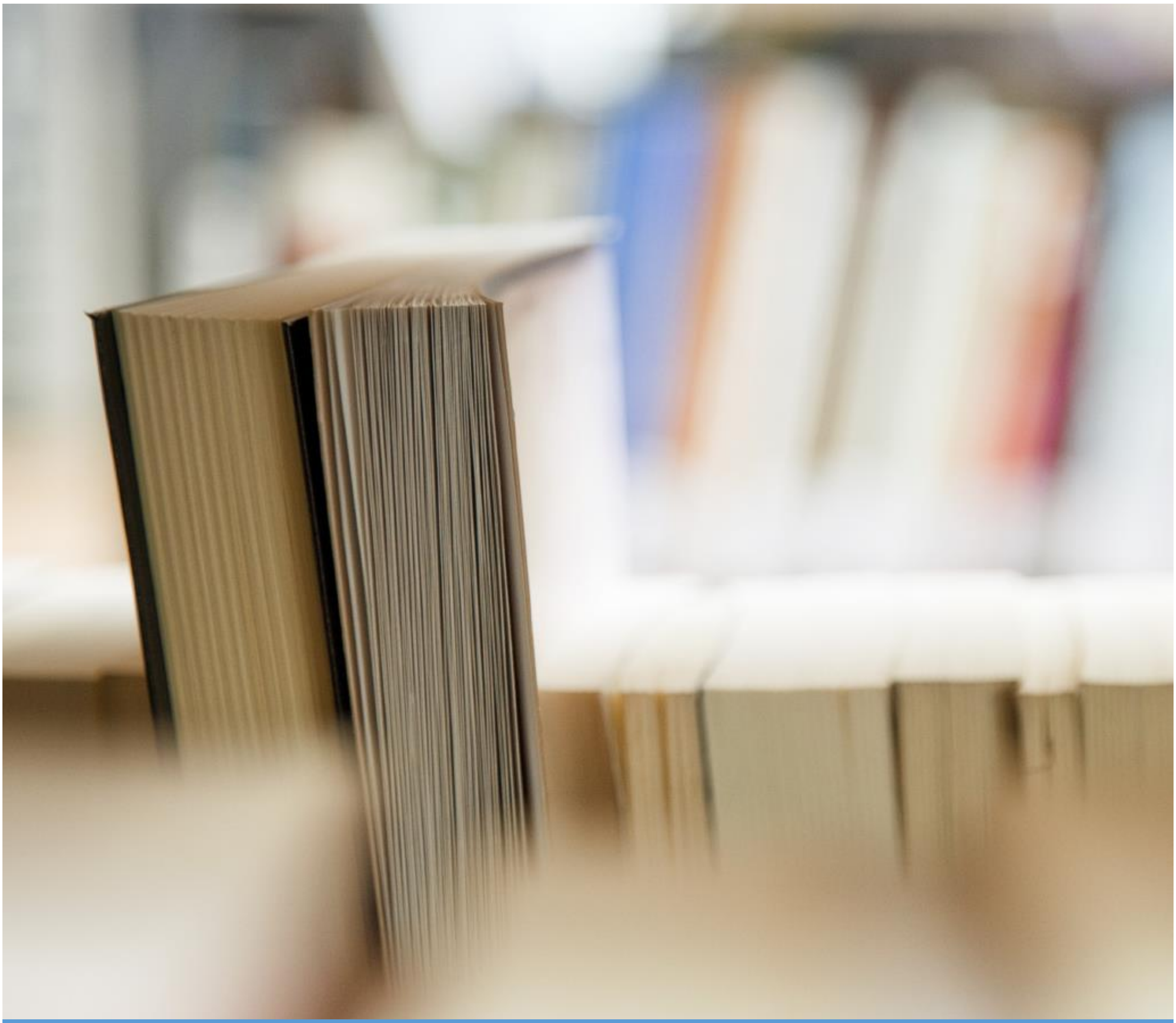


Project on Content Aware Image Resizing

Submitted By

Md. Kamrul Hasan

Fakrul Islam Tushar

Submitted To

Dro Désiré Sidibé

Mojdeh Rastgoo

Guillaume Lemaitre

University of Burgundy 1/9/18 Image processing

Project on Content Aware Image Resizing

Md. Kamrul Hasan[1], Fakrul Islam Tushar[2]

[1,2]Medical Imaging and Applications (M-1)

[1,2]University Center Condorcet - University of Burgundy

[1,2]Le Creusot, France

Email: [1]kamruleeekuet@gmail.com, [2]f.i.tushar.eee@gmail.com

***Abstract*— The availability of sophisticated source attribution techniques raises new concerns about privacy and anonymity of photographers, activists, and human right defenders who need to stay anonymous while spreading their images and videos. An image can be considered to be a combination of both significant (foreground) objects and some less significant (background) objects. Content aware image resizing (CAIR) algorithm uses the different edge detection methods to segregate the useful objects from the background. When applied to an image, CAIR can resize the image to a very different aspect ratio without destroying the aspect ratio of the useful objects in the image. In this project, we simply implement a content aware image resizing (CAIR) in MATLAB environment. The main idea to implement CAIR is to remove or insert the vertical or horizontal seams (paths of pixel) having the lowest energy. After implanted the Seam Carving Algorithm for Content aware image resizing (CAIR), analysis shows that the implemented seam carving for CAIR can generate more desirable resized images than cropping, resampling, and conventional seam carving techniques.**

Keywords*— Energy; Image resizing; Image content; Seam carving; Content aware image resizing (CAIR).

I. INTRODUCTION

Images to be resized without losing meaningful contents is increasingly important for image exchanging and sharing between different electronic platforms. Resizing techniques such as stretching and cropping are not ideal option for this as stretching effectively distort the image which doesn’t looks well and cropping removes meaningful contents [1]. When resizing image, effective means should not only consider geometric constraints, but also handle according to the image content. It is the fact that the image content in the middle is not more attractive than at the edge necessarily, the image resizing means on the basis of content is turning into a new hotspot in the image processing domain. Image resizing is also known as image retargeting. Firstly, this means treats the areas that attract human eyes' attention as more important regions, but the areas that not catch the people's eyes as unimportant districts, then to the greatest extent maintaining the important regions but changing unimportant districts in order to fit target image size. Therefore, this technology is referred to as content-aware image resizing. Among all the means relating to content-aware image resizing, seam carving algorithm's status is beyond all doubt, its advancement and innovation is better than other ways [2]. Seam Carving is a very effective method for content aware image resizing. Seam carving allows resizing the image removing the less important contents and conserving the size and shape of the important contents. The seam carving uses simple data structures, remove or insert lowest energy path of pixel or seams from top to bottom or left to right and preserving the highest energy pixel paths which contain the important contents [3]. In Addition by manipulating this energy map other functionality such as object removal can be achieved. In seam carving method, a seam or pixel path is been selected from the top of the image to the bottom that consist the pixels with less important content and can be removed from the image. Removing the pixels in a connecting path manner makes it sure that same number of pixels removed from each row of pixel in the image and avoids the image to be distorted after deletion. A pixel is considered to be less important if it has a small gradient value. Gradient value indicates the how an image changes. In our project gradient value is calculated by applying a horizontal and vertical *Sobel* filter to each pixel in the image. Compared to a Laplacian edge operator, *Sobel* operator has distinct advantages, although it is slightly more complex. It is less sensitive to isolated high intensity point variations since the local averaging over sets of three pixels tends to reduce this. In effect it is a *small bar* detector, rather than a point detector. Secondly, it gives an estimate of edge direction as well as edge magnitude at a point which is more informative and is of considerable use in later processing. A disadvantage of the *Sobel* edge detector is that the response to a step edge is present over either two or three pixels width, dependent on the precise position of the step relative to the pixel grid. This necessitates the use of post-processing in the form of edge thinning and thresholding in order to reduce the computational complexity of further processing. Although slightly more complex than the Laplacian operator, the Sobel operator is still relatively easy to implement in hardware form, most obviously by a pipeline approach.

II. METHODOLOGY AND PROPOSED RESEARCH

The overall project works are described in several subsections and blocks that are described below step by step-

A. *Image Shrinking*

The workflow for the image shrinking is shown in the Fig. 1. The imported image $I\ (u, v)$ is used to find out the energy which is simply the gradient of the image. To find out the image energy, Sobel operator is used due to some specific reason that mentioned in the introduction sections. The Sobel operator is a discrete differential operator which

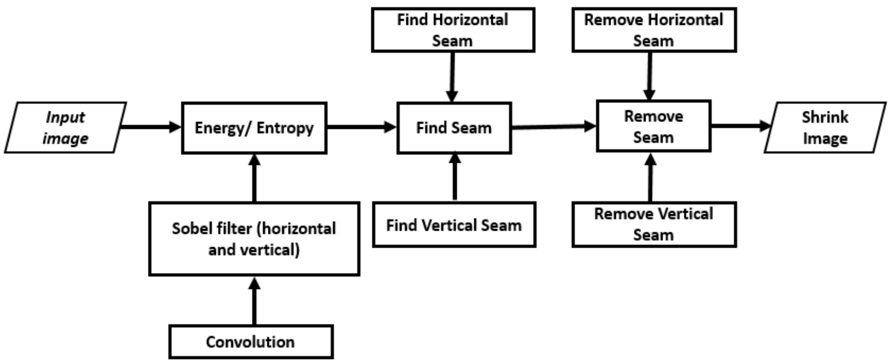


Fig. 1. Seam Removing/ Image Shrinking block diagram

utilize two 3x3 kernels/ mask. Among two mask, one estimates the gradient in the x-direction, while the other one estimates the gradient in the y-direction that are shown in Table-1.

TABLE I. PRESENTATION OF SOBEL KERNEL

Gradient Direction	-1	0	+1
→	-2	0	+2
	-1	0	+1

Gradient Direction	-1	-2	-1
↓	0	0	0
	+1	+2	+1

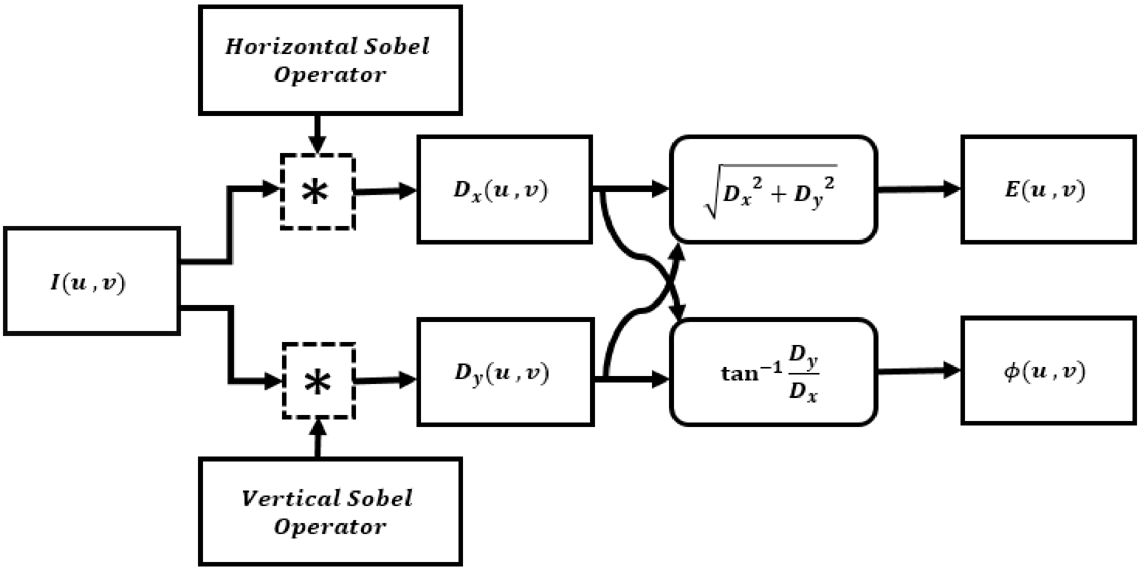


Fig. 2. Computational flow diagram of the Image energy or Image Gradient

The horizontal and vertical Sobel operator provides the gradient $D_x \; and \; D_y$ in the horizontal and vertical direction respectively after Convolution with the Image, $I\ (u, v)$. The gradient of the image $E(u, v)$ will be $\sqrt{D_x^{\ 2} + D_y^{\ 2}}$.

After finding the energy of the image, we need to find the seam of the provided Image, $I\ (u, v)$. A seam is a connected and monotonic path of low energy pixels connecting the two sides of the image either horizontally or vertically. The crucial goal in the seam carving algorithm that is shown in Table 2 is how choose the right seam to remove?

TABLE II. SEAM CARVING ALOGORITHM

Step-1: For each color channel (if RGB), the energy matrix (Taking gradient in both horizontal and vertical direction) is calculated. The energy for all color channels is summed into one 2D image to create the energy map.
Step-2: Find an optimal seam using Dynamic Programming. We have used a "forward energy" formulation in order to decrease the energy added to the image after seam removal.
Step-3: The minimum seam coordinates from *Step-2* are then used to remove the minimum seam. All the pixels in each row after the pixel to be removed are shifted over one column. Finally, the width of the image has been reduced by exactly one pixel.
Step-4: Repeat *Steps-1* to *Step-3* until desired number of seams are removed.
Step-5: Repeat *Steps-1* to *Step-4* for left to right edge.

We define the cost of a seam as the sum of the energy of its pixel $E(s) = E(I_s) = \sum_{i=1}^{n} e\big(I(s_i)\big)$. Then, we can look for the optimal seam s^* that minimizes this cost functions,

$$s^* = \arg\min_s E(s) = arg\min_s \sum_{i=1}^{n} e\big(I(s_i)\big).$$

It is easy to see that there are an exponential number of possible seam, and therefore exhaustive search is impractical. However, it turns out that the characteristics of this optimization problem enable the use if a very efficient algorithm to find the optimal seam called dynamic programming to use dynamic programming, some conditions must be met. First, the solution of the optimal solutions of sub-problems. Second, the spaces of these sub-problems must be small, meaning that the solution of the original problem can reuse solutions of similar sub-problems many times. In our case, if we can find, for every pixel in row $i-1$, the optimal vertical sub-seam from the first row to that pixel, then we can find the optimal vertical seam from the first row to any pixel in row i by extending each sub-seam by one additional pixel. For each pixel in row i, e.g., $I(i,j)$, the optimal seam can be extended from only three pixels because of connectivity : it can be extended from $I(i-1,j-1)$, pixel $I(i-1,j)$, or pixel $I(i-1,j+1)$. Since we know the optimal seams to all these pixels, we can choose the one with the smallest cost. Consequently, the cost of the optimal seam through pixel $I(i,j)$ will be calculated as

$$cost(i,j) = \mathrm{e}(i,j) + \min(cost(i-1,j-1), cost(i-1,j), cost(i-1,j+1))$$

Generalizing the observation, the optimal vertical seam is found by traversing the image from the first row to the last row composing a cost matrix C of size $(n \times m)$. For the first row we define $C(1,j) = e(1,j)$. And for each $i > 1$ we use Equation (up) to fill the matrix. At the end of this process, $C(i,j)$ will contain the commutative minimum energy for all possible connected seams starting at the first row and ending at pixel $I(i,j)$, and the minimum value of the last row in C will indicate the end of the minimal connected vertical seam. Hence, to find the path of the optimal seam we start from the minimum entry of the last row. Then we backtrack and follow, for each entry, the minimum entry of its three neighbors in the row above. For horizontal seams, the matrix C is composed similarly but from the first column to the last.

B. ***Image Enlarging***

The workflow block diagram for the image enlarging is shown in the Fig. 3 which is a similar task to making them shrinking, except now it makes sense to duplicate seams in order of least importance. In this case, the least important areas of the image become larger, and don't disfigure the more important areas (which will ideally remain the same size). Similar to the previous description, we need to find the energy, then horizontal as well as vertical seam of the image using sobel filter followed by convolution. Next step is to find the seams (path of pixel) along horizontal and vertical direction. Then the vertical and horizontal seams is been added to expand the image and finally desired expanded image is given as output. When implementing this, it turns out that we couldn't just order seams as they'd be found in the previous algorithm, because seams ending on different pixels at the bottom of the image would often converge on the same unimportant areas closer to the top of the image (essentially using the same pixels in multiple seams). Thus, when calculating a set of seams for duplication, we assign extra cost to using duplicate pixels (pixels already included in seams calculated earlier), so that the algorithm is more likely to choose mostly independent seams.

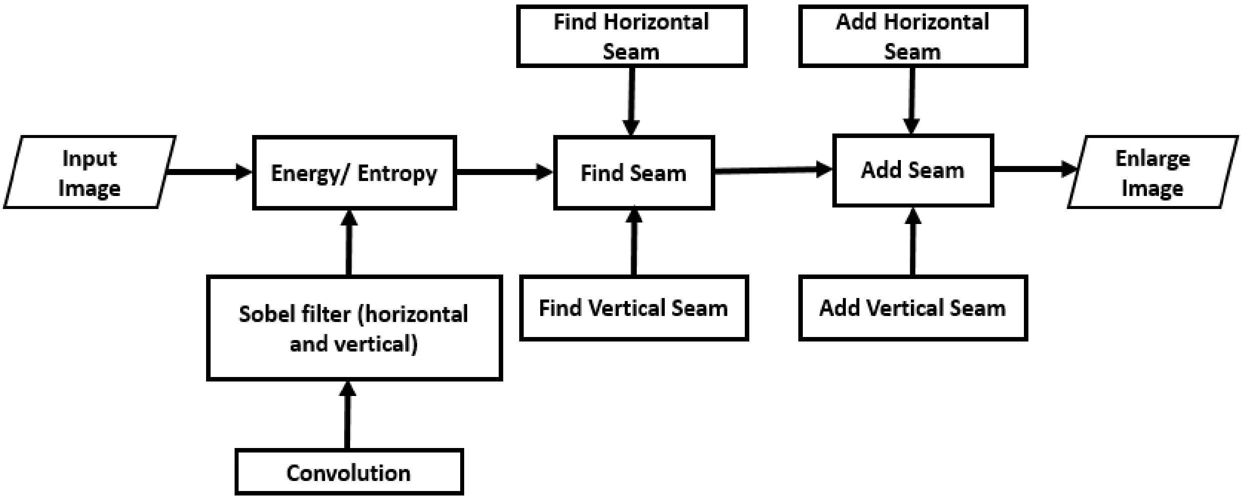


Fig. 3. Seam Adding/ Image Enlarging block diagram

C. *Graphical User Interface (GUI) Design*

The features of the implemented GUI interface is shown in the Fig. 4. to make the seam carving algorithm user friendly.

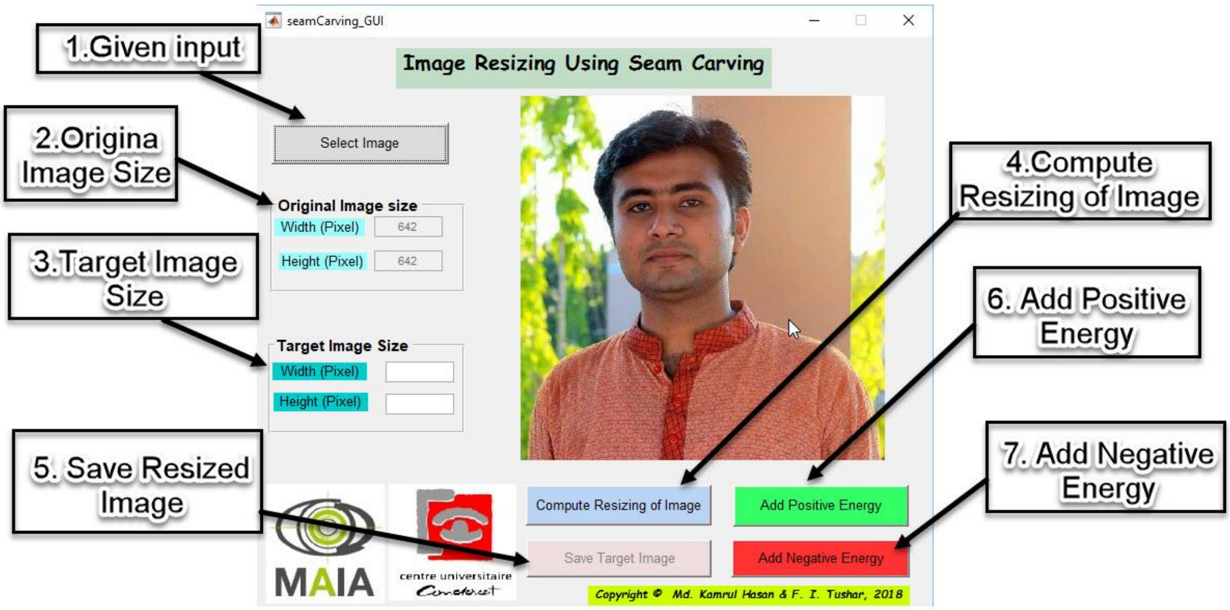


Fig. 4. Different features of Implement GUI for seam carving

From the given input section 1 (shown in Fig .4), anyone can select the image that we want to resize by the Seam Carving technique. When clicking that Icon it will give the following outcome as shown in Fig. 5. to select the image from the directory. Here, it is mentionable that all the desired images should be in the same locations of the GUI file and images format should be *.bmp* (bitmap image file).

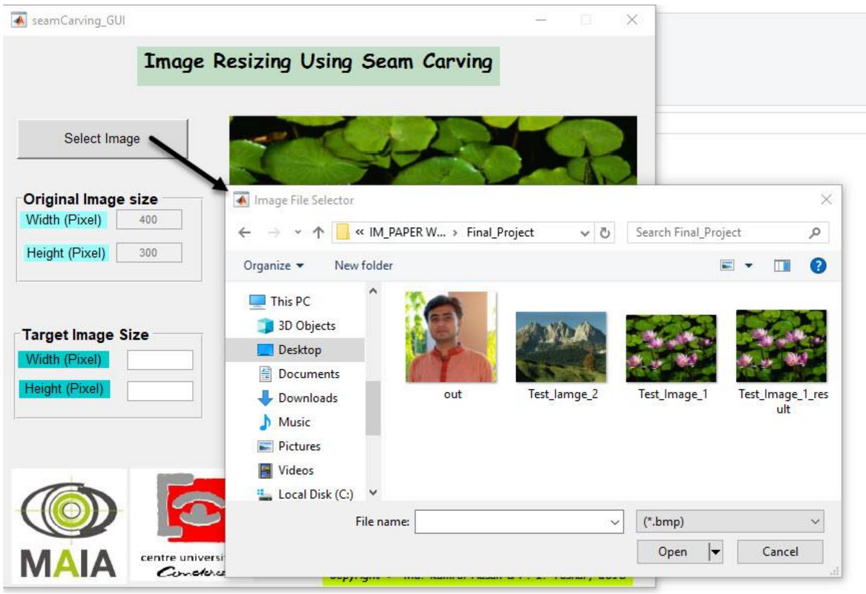


Fig. 5. Selecting the image for resizing using seam carving GUI

In section 2 as indicated in Fig. 4, it will provides original size (Number of pixel) of the image. In section 3 (shown in Fig. 4), user can put desired size to be resized of the original image as shown in Fig. 6. If the given value of pixel in section 3 (shown in Fig. 4) is less than the original size image will be shrink or vice versa.

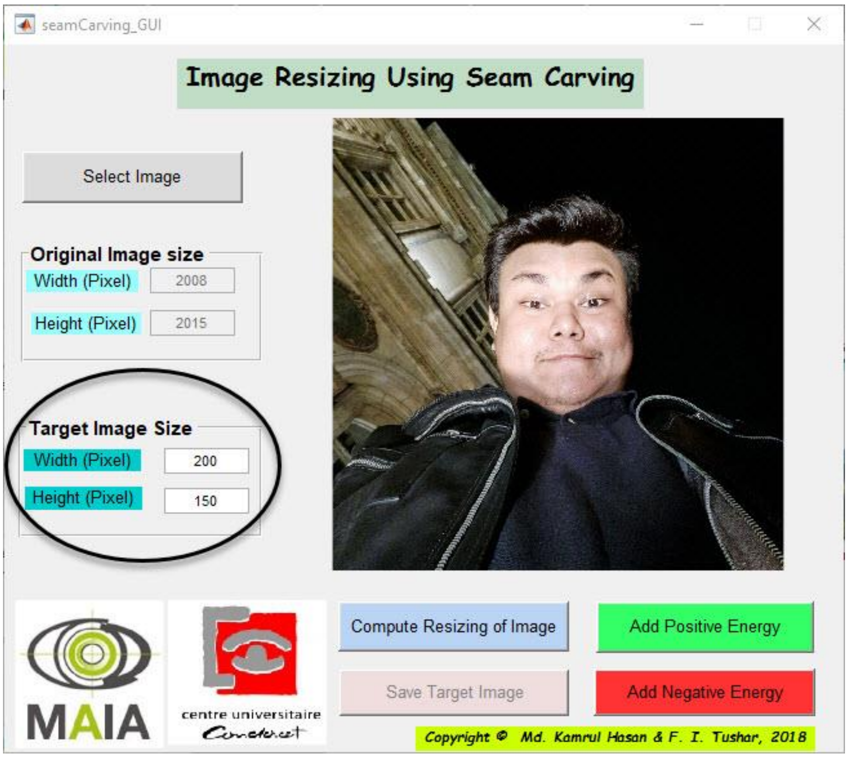


Fig. 6. Providing the target image desired size

After selecting the desired pixel size, we need to go section 4 (shown in Fig. 4). When you will click the section 4 (shown in Fig. 4) button it will takes some times for processing depending of the amount of shrinking and enlarging as shown in Fig. 7.

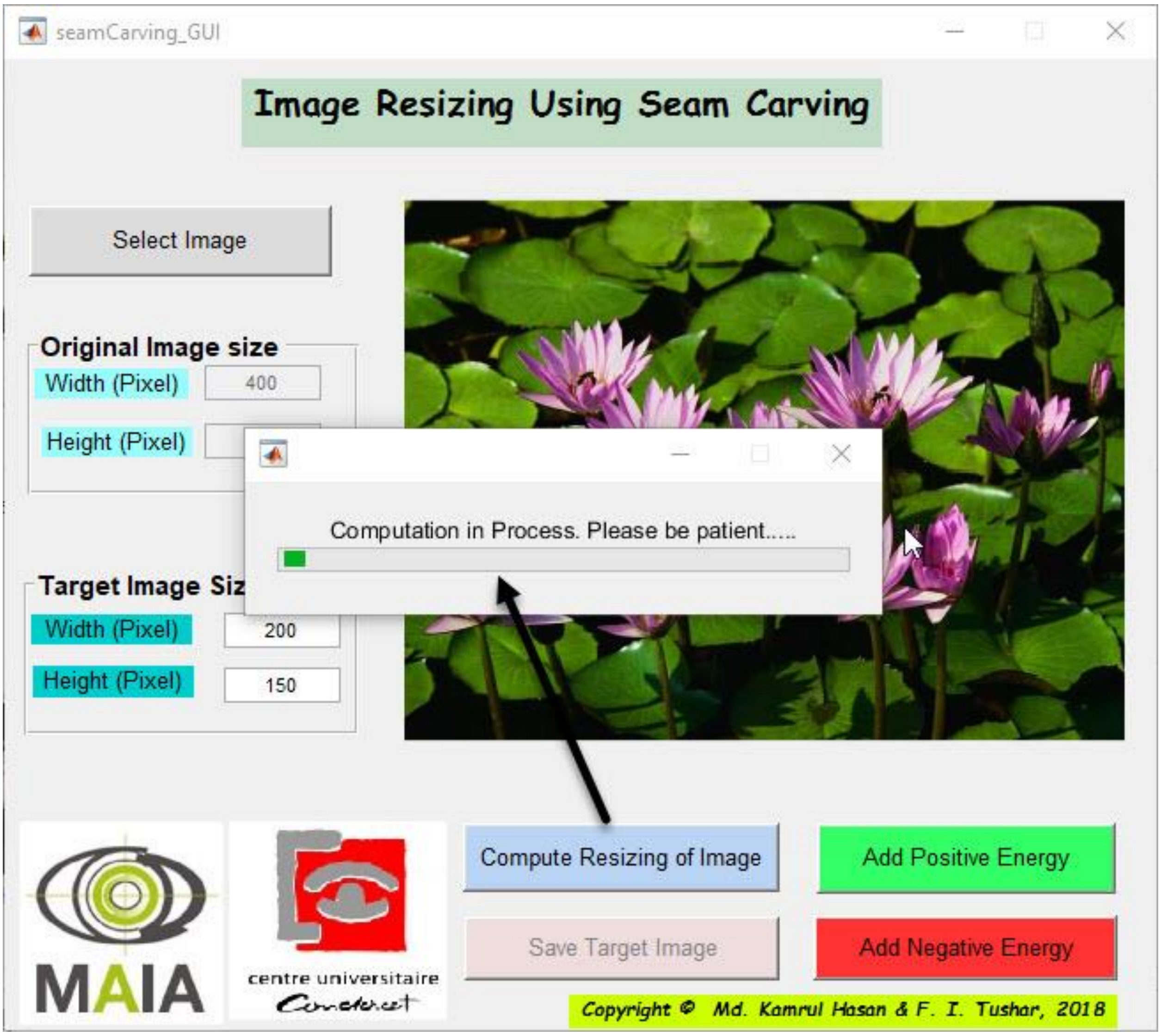


Fig. 7. Computing resizing of the input image

After that, if we want to save desired image, just we need to click on the section 7 (shown in Fig. 4) button as shown in Fig. 8. The output image will be save at the locations of original images.

One interesting feature is that addition or subtraction of some energy in the desired Region of Interest (ROI) of the target image. If we add energy to the ROI of the image no seam can touch that region. So after shrinking this region will be unchanged. Similarly, if we subtract energy from the ROI, after shrinking that region will be vanished. The adding and subtracting of the energy is shown in Fig. 9 (a) and 9 (b).

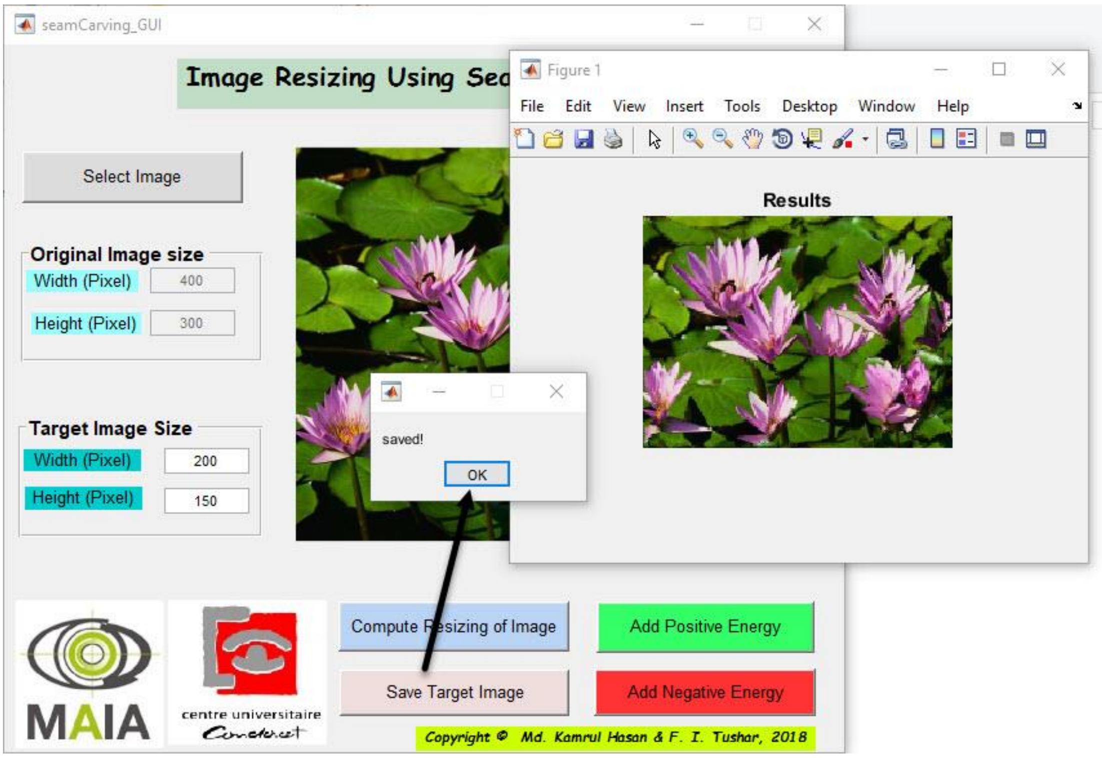
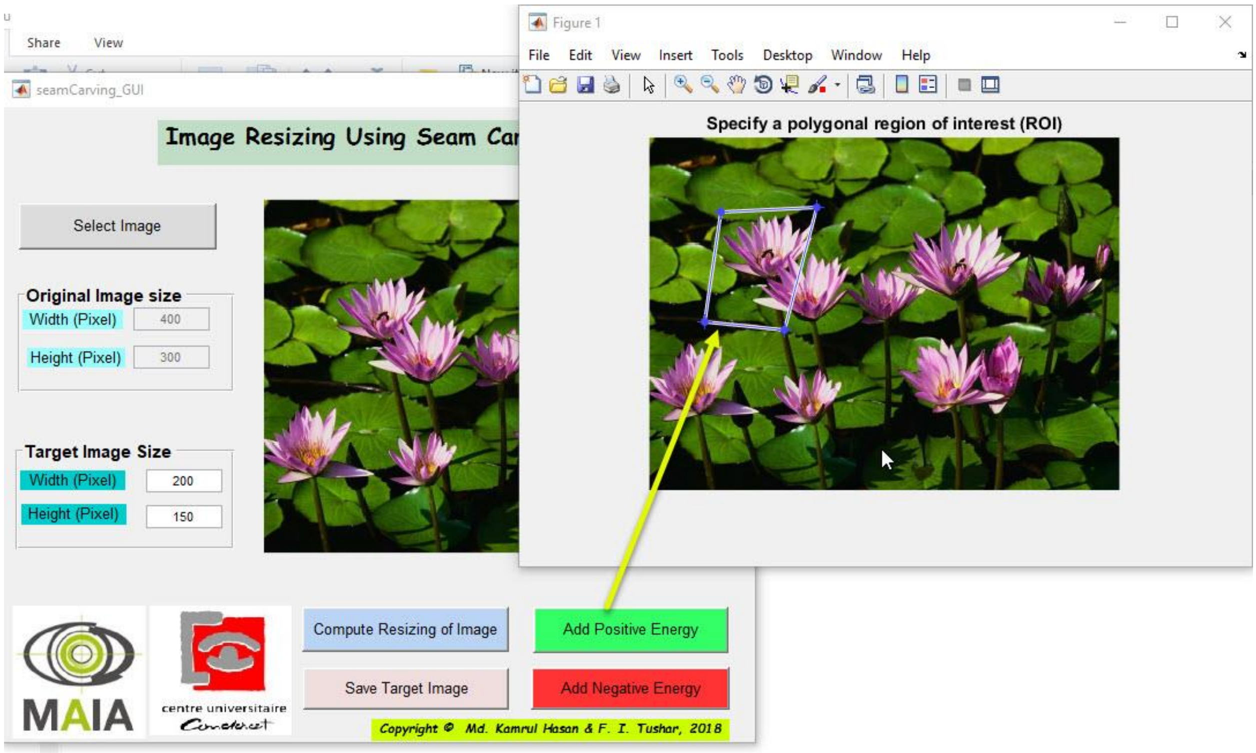
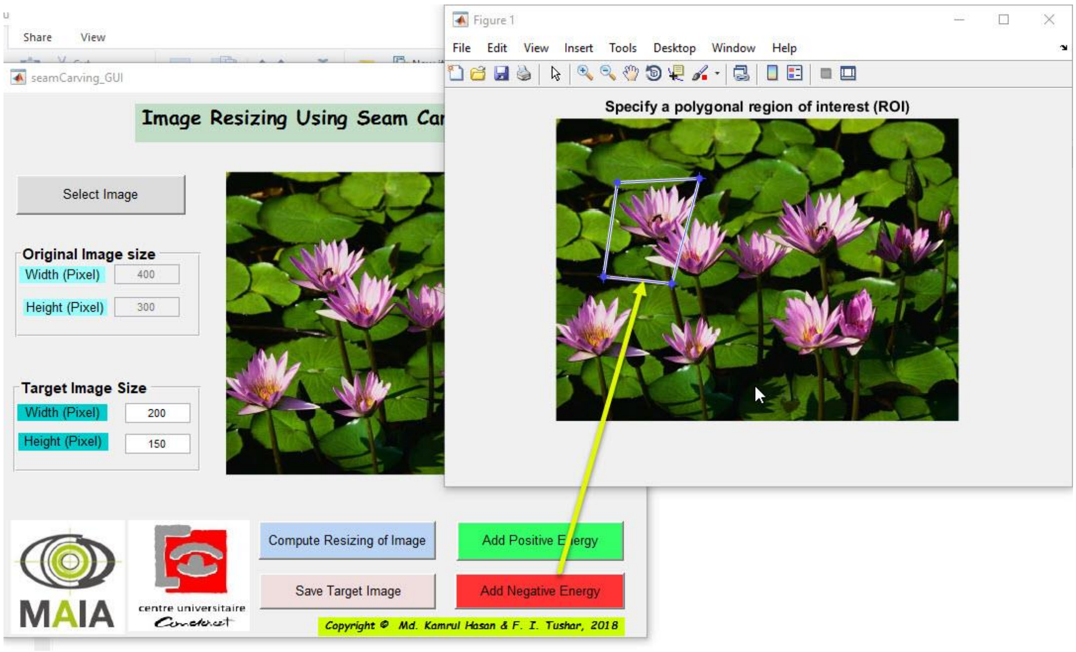


Fig. 8. Saving the resized image



(a)



(b)

Fig. 9. a) Add positive energy to the ROI of the Original Image b) Add negative energy to the ROI of the Original Image

III. RESULTS AND DISCUSSION

The resized image both shrinking and enlarging from the implemented system is shown in Fig. 10. The target image was shrinking and enlarging by 5 pixel in both vertical and horizontal directions. Fig. 10 (a), indicates the original image whose shrink image is Fig. 10 (b) by 5 pixel in both vertical and horizontal directions. On the other hand, we add 5 seam in both vertical and horizontal direction to the image Fig. 10 (b). The addition of seam to the image Fig. 10 (a) shown in Fig. 10 (c). After addition of the seam in Fig. 10 (b), we get the image Fig 10 (d). Fig. 10(a) and Fig. 10 (d), images are same because the removal and addition of seam is same number and in both directions.

(a) (b)

(c)

(d)

Fig. 10. CAIR resizing a) Original image b) Shrinked image (by 5 pixel) c) addition of seam and d) enlarge image (by 5 pixel)

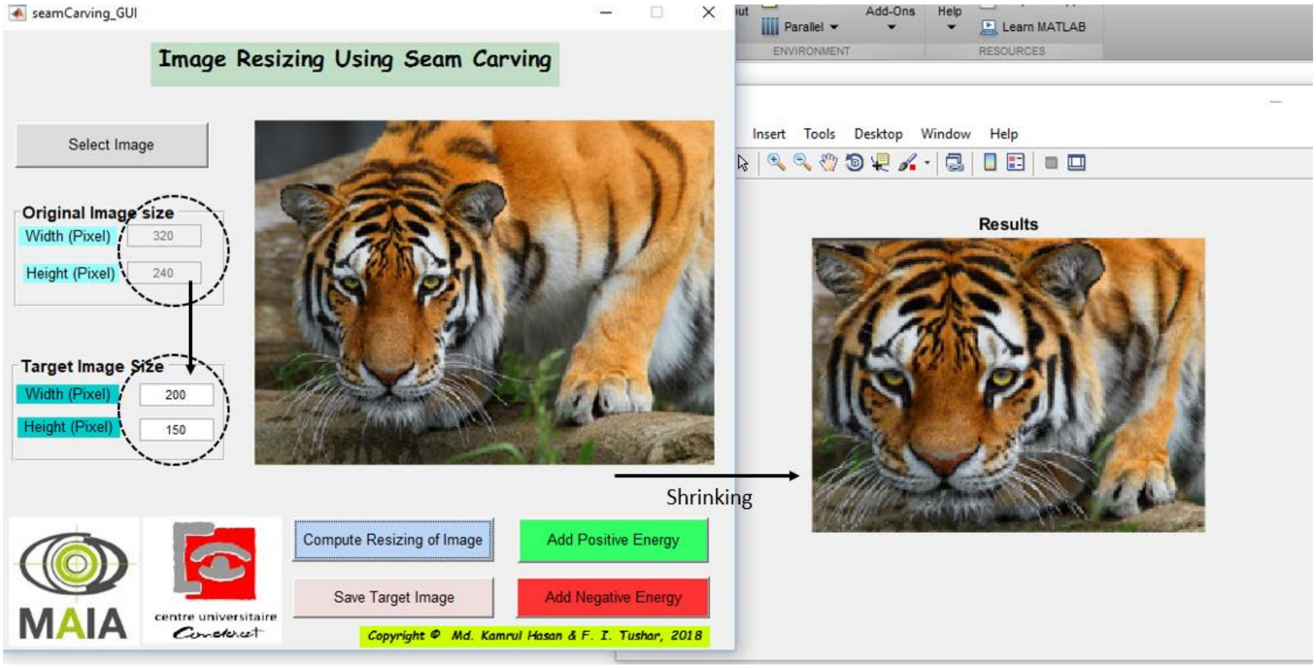


Fig. 11. CAIR Image Shrinking

Fig. 11 indicates the image shrinking using the implemented GUI. The input image has 320x240 pixels but our target image has 200x150. After execting the GUI, it will gives the following shrinking image as shown in Fig. 11. On the other hand, Fig. 12 indicates enlarging of the input image where input image has 395x295 pixels and desired image has 500x400 pixels. The enlarged image from the implemented GUI is shown in Fig. 12. In Fig. 13 and Fig. 14, we add and subtract some energy at the tree as shown in Figure. Due to addition of the energy to the input image, tree remaing unchanged after shrinking from 621x484 to 400x300 pixels. Similarly, due to addition of negative energy, tree is vanished after shrinking from 621x484 to 300x200 pixels.

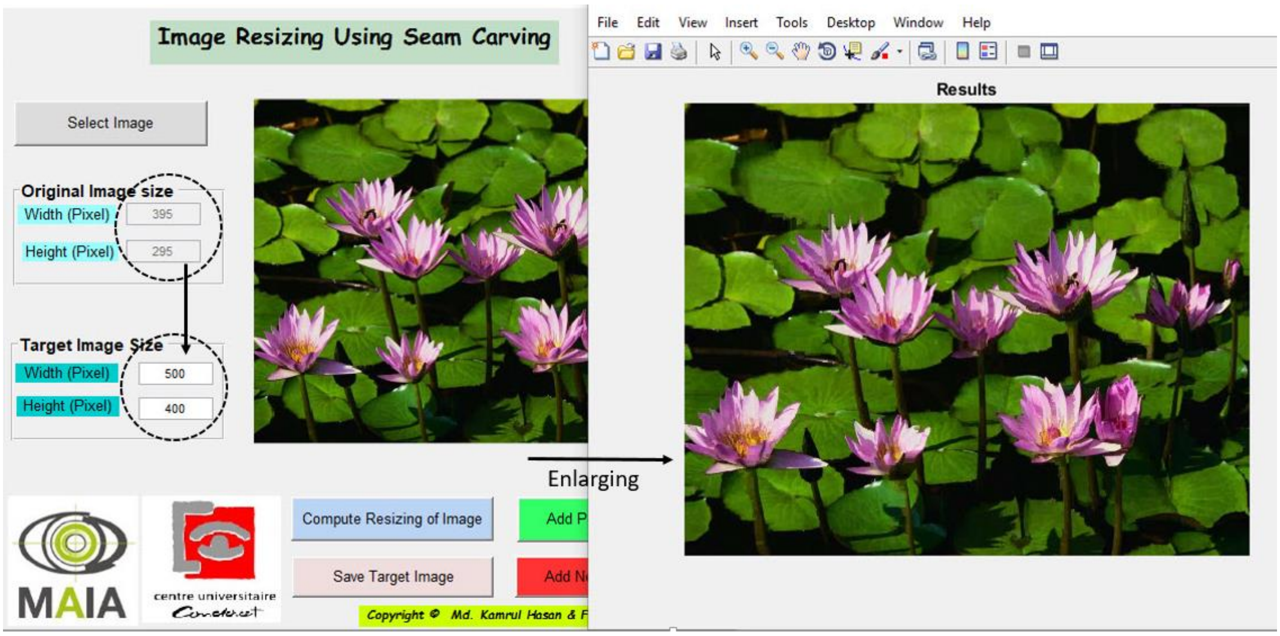


Fig. 12. a) CAIR Image Enlarging

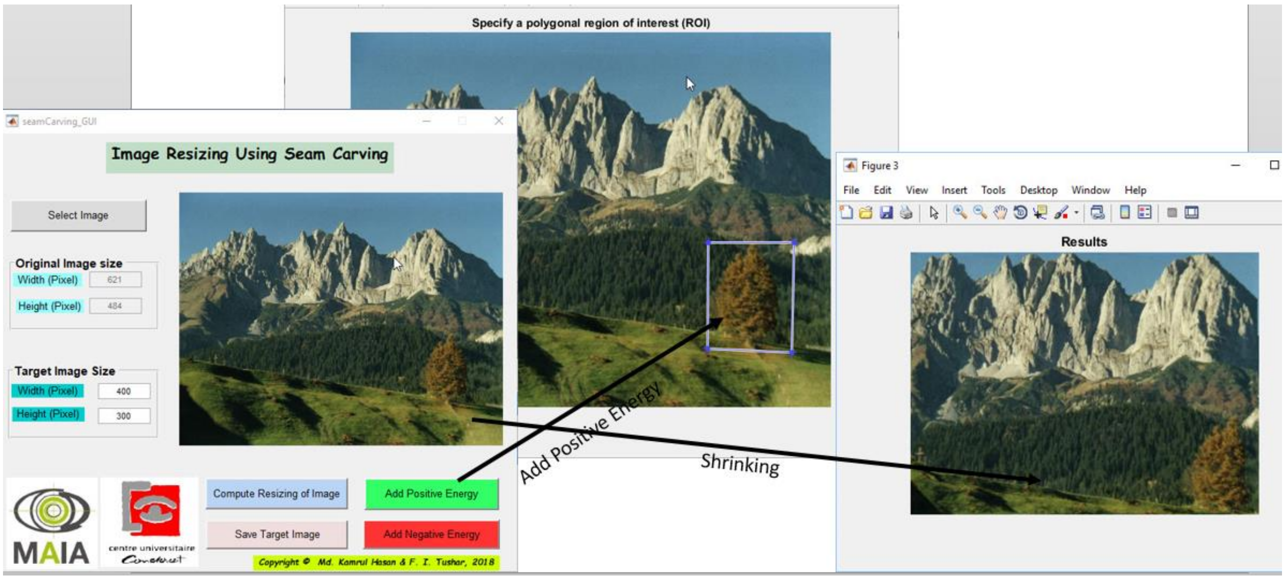


Fig. 13. Add positive Energy and Shrinking

Cropping is the removal of the outer parts of an image to improve framing, accentuate subject matter or change aspect ratio. In Fig. 15, coparison between the croping and implemented seam carving (CAIR) is shown where cropping of the image ignore the content and due to this ignorance some important content may be lost. It is easily noticable that the cropping of image reduce the aspect ratio of the image. On the other hand, Image seam carving algorithms attempt to change the size of images while preserving image salient contents with minimum distortion.

The aspect ratio of the output image from the implemented CAIR is better than the cropping of the image that is shown in Fig. 15.

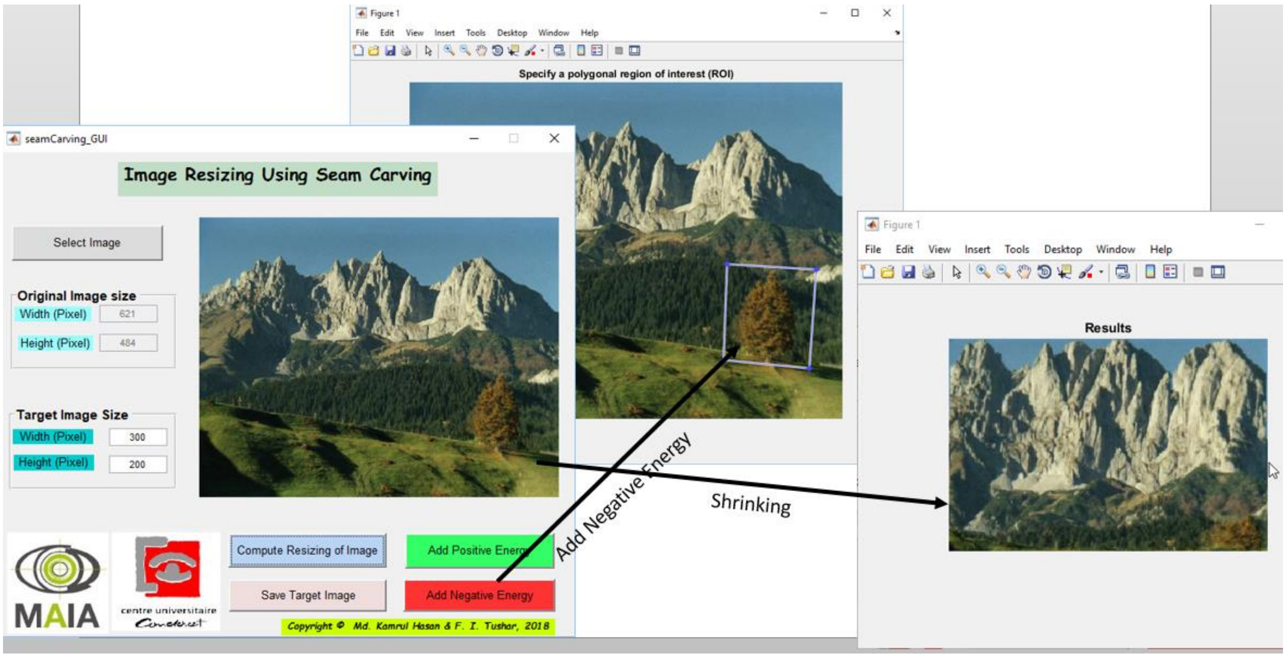


Fig. 14. Add Negative Energy and shrinking

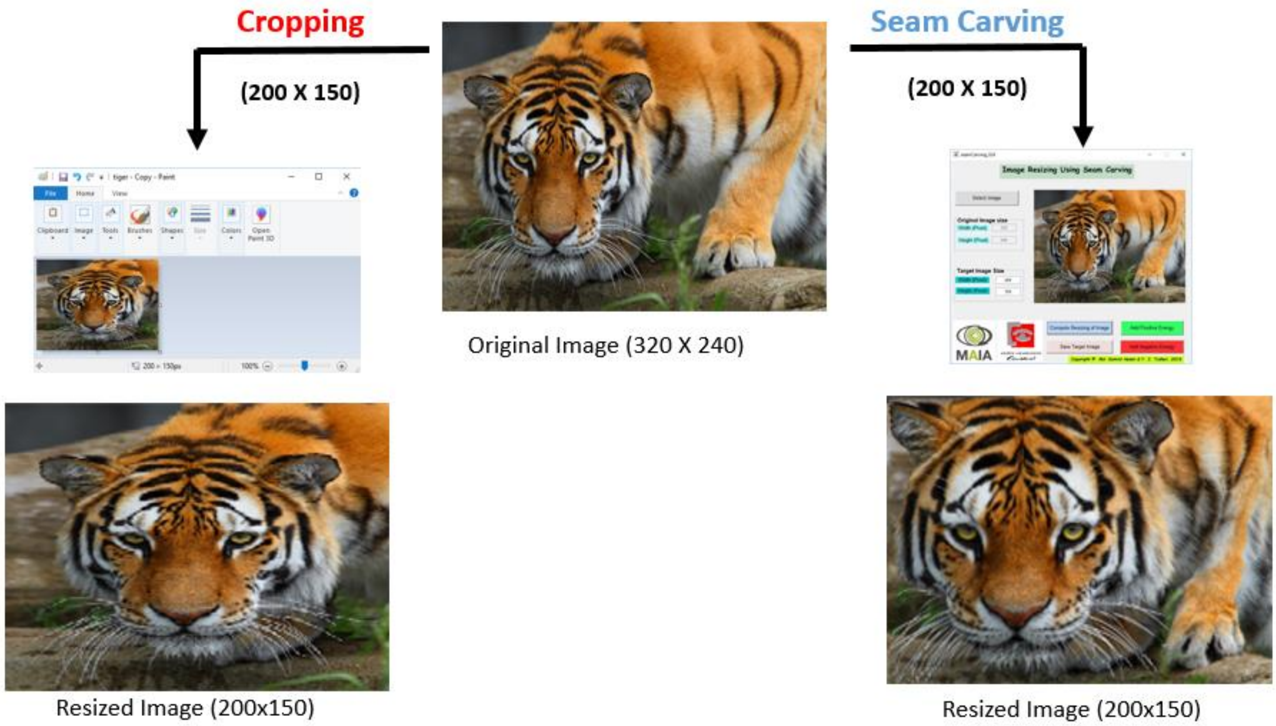


Fig. 15. Comparison between CAIR and cropping for image resizing

IV. Conclusion

Due to the diversity of various electronics devices like computer, mobile phone, pad and television it is great challenges, how to resize an image or video to adjust to different display screens. Seam carving has been a significant method for image resizing. But, if multiple seam is add or remove from the specific region it will make the distorted image which is solved by redistributing energy to connected pixels after adding or removing seam from the original image. As well as to reduce the distortion, the edge information can be added to the image after reducing or adding the pixels. The key problem is to reserve the most attractive regions and useful information, minimize visual distortion, and satisfy user preferences under the constraint of topological relations and the global context.

Acknowledgment

We want to convey special gratitude to our course Teacher DRO DÉSIRÉ SIDIBÉ and Lab Teachers Mojdeh Rastgoo & Guillaume Lemaitre. Without their crucial guideline and teaching methodology, this project might be difficult to implement. We are really grateful to them.

References

[1] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," ACM Transactions on Graphics., vol. 26, no. 3, pp. 1-10, 2007.

[2] Z. Zhang, B. Kang and H. Li, "Improved seam carving for content-aware image retargeting," 2013 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Visakhapatnam, 2013, pp. 254-257. DOI: 10.1109/PrimeAsia.2013.6731216.

[3] Z. Karni, D. Freedman, and C. Gotsman, "Energy-based image deformation," In Computer Graphics Forum, vol. 28, no. 5, pp. 1257-1268, 2009.

[4] D. Cohen-Or, C. Greif, T. Ju, N. J. Mitra, A. Shamir, O. Sorkine-Hornung, H. Zhang, “A Sampler of Useful Computational Tools for Applied Geometry, Computer Graphics, and Image Processing: Foundations for Computer Graphics, Vision, and Image Processing Front Cover,” CRC Press, pp. 246 May 21, 2015.