Computer Graphics Optimization: Final Report

SHAN LIN, 5122F099
AIWEN SU, 5122F059
JINGYI HUANG, 5122FG14

(a) Original (b) Seam carving (c) Seam carving (with seams) (d) Warping (e) Warping (with mesh)

Fig. 1. **Demonstration of image resizing by different methods.** The content-aware image resizing methods can keep the the aspect ratios of prominent objects similar to the original image.

1 PROJECT TITLE

Content-Aware Image Resizing Using Optimization

2 PROJECT DESCRIPTION

There are always needs to adapt images to different sizes of display in daily lives. Simple uniform resizing may be undesired since it significantly changes aspect ratio and unavoidably loses information, and there are cases where interested objects are on the edges. Generally speaking, it is better to perform resizing while taking its content into consideration. Recent techniques formed either discrete or continuous framework towards the problem, namely *Seam Carving* [Avidan and Shamir 2007] and *Image Warping* respectively, with the objective to preserve important regions while minimize distortions [Asheghi et al. 2022]. Fig.3 demonstrates the resizing process by different methods.

3 PROPOSED APPROACH

3.1 Genetic Seam Carving

Seam carving. We first try the discrete formulation. With the seminal work of *Avidan and Shamir* [Avidan and Shamir 2007], where connected paths of pixels across an image are proposed to be removed to minimize distortions, as is shown in 4a. For an image of size $m \times n$, a vertical seam can be defined as

$$\begin{aligned} &\mathbf{s}^{\mathrm{v}} = \{(i, j) \mid j = \alpha(i)\}_{i=1}^{m} \\ &\text{s.t.} \quad |\alpha(i) - \alpha(i+1)| \leq 1 \quad \forall i \end{aligned} \tag{1}$$

where $\alpha(i)$ is the mapping from row index to column index. And a horizontal seam can be defined the same way.
The optimal seam should be in the lowest energy, that is

$$\mathbf{s}^{\mathrm{v}*} = \arg\min_{\mathbf{s}^{\mathrm{v}}} E(\mathbf{s}^{\mathrm{v}}) \qquad = \arg\min_{i,j} \sum_{i}^{m} e(i, j) \tag{2}$$

where $E(\mathbf{s})$ is the total energy of the seam.
The path can be computed by an exhaustive algorithm since it searches by rows iteratively and keep the energy in each step to be the slowest. This can be given by

$$M(i, j) = e(i, j) + min \begin{Bmatrix} M(i+1, j-1) \\ M(i+1, j) \\ M(i+1, j+1) \end{Bmatrix} \tag{3}$$

where $M(i, j)$ is the cumulative minimum energy at the pixel (i, j). By substituting this, the optimal vertical seam can be formulated as

$$\mathbf{s}^{\mathrm{v}*} = \arg\min M(1, j)_{j=1}^{n} \tag{4}$$

Optimization. Dynamic programming is a standard way to solve the subproblems above [Avidan and Shamir 2007]. And there are also proposals to apply genetic algorithms (GAs) [Oliveira et al. 2015; Oliveira and Neto 2015]. GA [Whitley 1994] is a population-based meta-heuristic algorithm, where solutions are generated iteratively to approximate the optimum based on the fitness. Since GAs are generally used for solving discrete problems and is good at finding a nearly global optimum for problems with large search space, we implement it here.

Genetic representation. Following the representation in [Oliveira et al. 2015], a seam is defined as an individual, each with a single chromosome composed by genes. Mathematically, it is given by

$$g_i \in \begin{cases} [1, n], & \text{if i=p;} \\ [-1, 1], & \text{otherwise.} \end{cases} \tag{5}$$

where p is the pivot point indicating the seam construction starting point. Its value represents the absolute coordinate in the image, while others are relative values of the random walk. The transformation of an individual into a seam is given by converting its gene values into coordinates

$$\alpha(i) = \begin{cases} g_i, & \text{if i=p;} \\ g_i + \alpha(i-1), & \text{if i>p;} \\ g_i + \alpha(i+1), & \text{if i<p.} \end{cases} \tag{6}$$

Genetic operators. We do selection and variations following the GAs based on the above representation. Offsprings are reproduced through variations including crossover and mutation. We do single point crossover here, where parents' genes are exchanged and combined. To avoid the undesired case of pivot being located in

the region of interest, we follow the crossover operator described in [Oliveira and Neto 2015] to give a flexible pivot. This is done by encoding the pivot value into a ternary one and perform again crossover between two of them. In this way, when converting back to integer values, the location of the pivots are changed. And select certain numbers of genes randomly to have a mutation.

Fitness function. Individuals selection is performed by a fitness function, which is based on inversed energy. We apply elitism for each generation to keep a non-degenerate solution. Different choices of energy functions are discussed in 3.4.
This part of the work is carried out based on [Lavender 2019]

3.2 Image Warping

There are many warping-based methods that have achieved good results in the image retarget task [Liu and Gleicher 2005; Niu et al. 2012; Ren et al. 2009; Wang et al. 2008]. We implemented one of the methods based on mesh warping referring to [Wang et al. 2008].
In mesh warping based methods, a mesh $\mathbf{M}$ is created to split the input image into small quad faces $\mathbf{F}$. Quad faces consist of vertices $\mathbf{V}$ and edges $\mathbf{E}$, where $\mathbf{V} = [\boldsymbol{v}_0^T, \boldsymbol{v}_1^T, \cdots, \boldsymbol{v}_N^T]$ and $\boldsymbol{v}_i \in \mathbb{R}^2$ donates the vertex positions. The goal in these methods is to find a optimal vertex positions $\mathbf{V}^*$ to warp the image in order to preserve the important quads as much as possible and allow the distortion of the unimportant quads. The optimization process considers the distortion of quad shapes and the bending of the vertical or horizontal lines.

The distortion of shapes. The shape distortion of a quad face with respect to vertex positions can be measured by:

$$J_s^{(f)}\left(\mathbf{V}^{(f)}\right) = \sum_{\{i,j\}\in\mathbf{E}^{(f)}} \left\| \left(\boldsymbol{v}_i - \boldsymbol{v}_j\right) - s_f\left(\boldsymbol{v}_i^{(0)} - \boldsymbol{v}_j^{(0)}\right)\right\|^2. \quad (7)$$

where $\mathbf{V}^{(f)}$ and $\mathbf{E}^{(f)}$ are the vertices and edges of f, $\boldsymbol{v}_i^{(0)}$ and $\boldsymbol{v}_j^{(0)}$ are the initial positions of the vertices, s_f is the desired scale factor for the face f.
s_f is completely defined by $\boldsymbol{v}_i, \boldsymbol{v}_i^{(0)}$. We can obtain optimal scale factor s_f^* by differentiating $J_s^{(f)}\left(s_f; \mathbf{V}^{(f),*}\right)$ and making it zero. Thus, the optimal scale factor s_f^* can be calculated as:

$$s_f^* = \frac{\sum_{\{i,j\}\in\mathbf{E}^{(f)}} \left(\boldsymbol{v}_i^{(0)} - \boldsymbol{v}_j^{(0)}\right)^T \left(\boldsymbol{v}_i^* - \boldsymbol{v}_j^*\right)}{\sum_{\{i,j\}\in\mathbf{E}^{(f)}} \left\|\boldsymbol{v}_i^{(0)} - \boldsymbol{v}_j^{(0)}\right\|^2}. \quad (8)$$

The bending of lines. Similar to Eq. (7), the bending of lines of a quad face with respect to vertex positions can be measured by:

$$J_l^{(f)}\left(\mathbf{V}^{(f)}\right) = \sum_{\{i,j\}\in\mathbf{E}^{(f)}} \left\| \left(\boldsymbol{v}_i - \boldsymbol{v}_j\right) - l_{ij}\left(\boldsymbol{v}_i^{(0)} - \boldsymbol{v}_j^{(0)}\right)\right\|^2. \quad (9)$$

where l_{ij} is the length ratio of the edges and the optimal length ratio can be calculated by:

$$l_{ij}^* = \frac{\left\|\boldsymbol{v}_i^* - \boldsymbol{v}_j^*\right\|}{\left\|\boldsymbol{v}_i^{(0)} - \boldsymbol{v}_j^{(0)}\right\|}. \quad (10)$$

Objective function. J_s and J_l are combined to obtain the objective function:

$$J\left(\mathbf{V}\right) = \sum_{f\in\mathbf{F}} w^{(f)}\left(J_s^{(f)}\left(\mathbf{V}^{(f)}\right) + \lambda J_l^{(f)}\left(\mathbf{V}^{(f)}\right)\right). \quad (11)$$

where $w^{(f)}$ are the weights of face f, which can be calculated from the energy map of the image. Different from [Wang et al. 2008], we introduced λ to control the effect of J_l, and $w^{(f)}$ is not only used to control the weights of J_s terms, but also used to control the weights of J_l terms.
To ensure that the vertices of the border are not outside the image, the following constraint is used:

$$\boldsymbol{v}_{i,y} = \begin{cases} 0, & \boldsymbol{v}_i \text{ is on the top boundary} \\ h_{\text{target}}, & \boldsymbol{v}_i \text{ is on the bottom boundary} \end{cases}, \quad (12)$$

$$\boldsymbol{v}_{i,x} = \begin{cases} 0, & \boldsymbol{v}_i \text{ is on the left boundary} \\ w_{\text{target}}, & \boldsymbol{v}_i \text{ is on the right boundary} \end{cases}. \quad (13)$$

where h_{target} and w_{target} are the target height and width.

Optimization. By minimizing the objective function Eq. (11), we can obtain the optimal vertex positions. During the optimization calculation, s_f, l_{ij} and the initial vertex positions $\boldsymbol{v}_i^{(0)}$ are fixed, the problem is reduced to a series of linear problems. Two sparse matrix $A_x, A_y \in \mathbb{R}^{N\times N}$ are introduced to represent the coefficients of $\boldsymbol{v}_i$'s, and two vectors $\boldsymbol{b}_x, \boldsymbol{b}_y \in \mathbb{R}^N$ are introduced to represent the constant terms, where N is the number of vertices. By solving $A_x\boldsymbol{v}_x = \boldsymbol{b}_x$ and $A_y\boldsymbol{v}_y = \boldsymbol{b}_y$, we can find the optimal vertex positions $\mathbf{V}^*$ for the current s_f and l_{ij}.
As we cannot the optimal s_f, l_{ij} initially, we need an initial guess for of the vertex positions $\mathbf{V}^{(1)}$. We simply scaled the image to the target size, using its vertex positions as an initial guess to calculate $s_f^{(1)}, l_{ij}^{(1)}$. And use $s_f^{(1)}, l_{ij}^{(1)}$ to compute a better vertex position $\mathbf{V}^{(2)}$. Then, repeat this process over and over again. Since $\mathbf{V}^{(t)}$ will eventually converge, we can find the optimal $\mathbf{V}^*$ iteratively.

3.3 Linear Programming

Since seam carving is setting to find solutions greedily, which would generally be suboptimal. In this section we consider having an attempt to remove all seams at a time to gain a global optimum. A different formulation is presented here as a binary optimization problem by pixel selection.

Objective function. The objective is to find an optimal indication matrix X for selecting pixels with the lowest total energy to guide the resizing process. Constraints are imposed to ensure certain number of pixels selected per row and to restrict the pixels to be consecutive (which is similar to the definition of seam).
Consider the vertical seam case, the objective function is given by

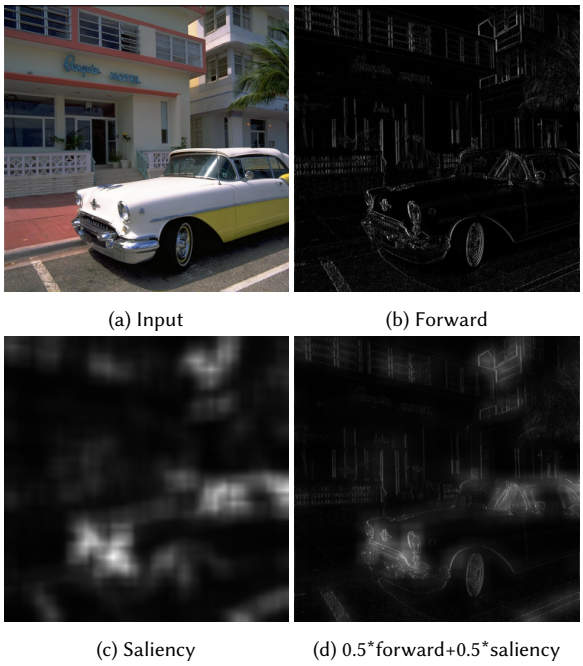
$$\begin{aligned} \arg\min_{X} \quad & \|E \odot X\|_1 \\ & \sum_j x_{ij} = N, \\ & \sum_{j' \in \{j-1, j, j+1\}} x_{i-1,j'} \geq x_{ij}, \\ & \sum_{j' \in \{j-1, j, j+1\}} x_{i+1,j'} \geq x_{ij}. \end{aligned} \tag{14}$$

where $E \in \mathbb{R}_{+}^{H \times W}$ is the energy map for the input image which can be obtained using the energy function, and $X \in \{0,1\}^{H \times W}$ is a binary matrix, with its elements x_{ij} indicating whether the corresponding pixel is selected. $\odot$ donates element-wise product, $||\cdot||_1$ denotes the L1 norm.

Optimization. Since the whole system is linear, we can use linear programming to solve the problem.
In addition to using binary variables $x_{ij} \in \{0,1\}$. We also tried the representation of x_{ij} as a continuous variable, where $0 \leq x_{ij} \leq 1$. In the case of continuous variables, x_{ij} can be considered as the probability when the pixel in row i and column j is selected. We use the same objective function and constraints as in the binary case, with the difference that in the final decision of pixel selection, the N pixels with the highest probability in each row are selected.

3.4 Energy Functions

Energy describes the significance of pixels in the image. It is said to have a great impact on the results of both proposed methods, since removal or deformation in regions of low energy would result in minimal visual impact. As such, we try out several recognized energy functions and compare their effects on the result.
Rubinstein et al. [Rubinstein et al. 2008] adopts a forward energy which considers the effect on the retargeted image energy to replace the original backward energy. It is shown to produces less discontinuities on the results, so we use it as one of the energy functions. Besides, inspired by the choice in [Wang et al. 2008] where a combination of image gradient and saliency map [Itti et al. 1998] is specified to detect both structural significant and attractive objects, and overcome the limitations of either of them, we apply energy functions including gradient magnitudes, forward energy, saliency and their combinations. Examples of energy maps is shown in Fig. 2.



(a) Input (b) Forward

(c) Saliency (d) 0.5*forward+0.5*saliency

Fig. 2. Different energy maps.

4 RESULTS AND DISCUSSIONS

We first have an overall comparison of the results given by genetic seam carving and warping. Subsequently we give some detailed description and analysis for each method individually.

4.1 Overall Comparison

We perform comparisons in terms of visual effects and executing time by reducing 20% of the height and 25% of the width. For the seam carving approach, we set 30 generations and keep a constant population size of 10. The ratio of keeping parents for the next generation is set to be 0.2; For the warping-based approach, the importance weight λ is set to be 10. The determination of part of these hyperparameters is discussed in the following chapters correspondently. In addition, we found that the combination of forward energy benefits the result most, thus we adopt it here and for the following sessions. The comparison results are shown in Fig. 3.

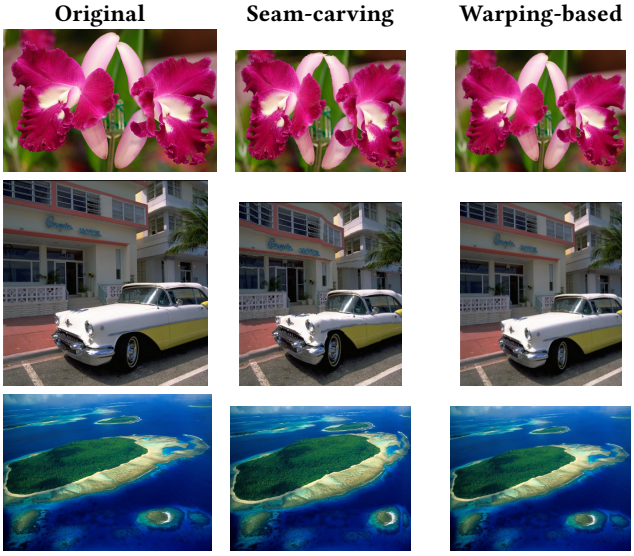


Fig. 3. Comparison of the genetic seam carving and warping. The resizing quality depends on the type of the image.

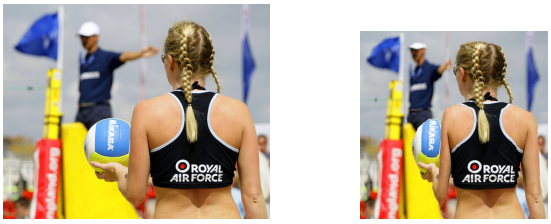
Table 1. Computational time of the two approaches

Approach	orchid	car	islands
Genetic seam-carving	723.3s	563.9s	1369.5s
Warping-based	1.3s	1.1s	2.3s

According to Fig.3, warping-based approach produces smoother results, while seam carving produces noticeable discontinuity, especially in images containing structural objects (eg. the car). This can seriously affect the visual perception, which is caused by the discrete nature of seam-carving methods. Moreover, as shown in Table 1, the computational time for genetic seam carving is far longer than warping, especially when specifying a large number of generations. This is partly due to that the computational cost of warping methods is independent of the resizing dimensions while that of seam carving is proportional to the number of seams [Wang et al. 2008].

4.2 Genetic Seam Carving

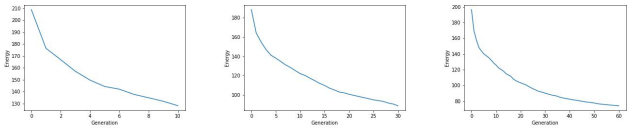
Number of generations. In terms of the genetic seam-carving approach, we looked at the effect of different numbers of generations on the retargeted results (Fig. 4). When the number is small, the head of the subject is shown to be severely distorted; However, this deficiency is alleviated as the number of generations increases. Moreover, as shown in Fig. 5, the loss (i.e. the average total energy of the seam) converges to a lower level as the number of generations increases, aligning with our common knowledge that larger iteration steps generally give a better result. Considering the tradeoff between accuracy and time, we determine the generations to be 30.



(a) Original image (b) Generations = 10

(c) Generations = 30 (d) Generations = 60

Fig. 4. Comparison of the results for different numbers of generations in genetic seam carving. The resized image suffers less distortion as the number of generations increases.

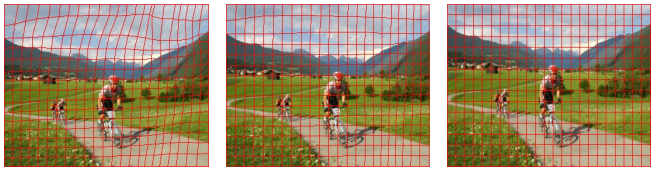


(a) Generations = 10 (b) Generations = 30 (c) Generations = 60

Fig. 5. Comparison of loss convergence for different numbers of generations. The overall trend is roughly the same, but it is clear that candidate solutions with lower energy is able to be found as the number of iterations increases to approximates the global optimum.

4.3 Image Warping

The effect of J_l, J_s. For the warping-based method, we tried different λ in Eq. (11) to investigate the effect of terms J_l, J_s on the results. The output images under different λ are shown in Fig. 6.



(a) $\lambda = 0$ (b) $\lambda = 10$ (c) $\lambda = 100$

Fig. 6. Output images under different λ.

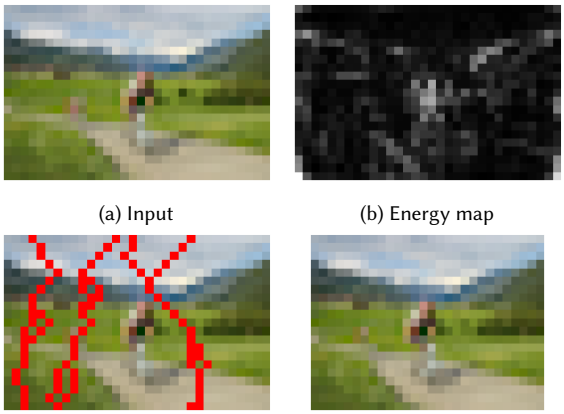
It can be seen that the bending of the lines becomes severe at $\lambda = 0$, which shows the effectiveness of the J_l term in reducing the bending of the lines. At $\lambda = 10$, the effects of J_s and J_l are more balanced and a better result is obtained. At $\lambda = 100$, the effect of J_l increases, and it can be seen that without the J_s term, all quads will be of the same size, which is equivalent to scaling the image directly.

Initial guess. Although warping-based method usually gives results with less distortions, we notice that seam carving results usually look better when not considering distortions. The reason is that seam carving based method does not change the size of the salient object, which makes the salient object look larger in the output image. In contrast, when using the warping based method, although the deformation of the salient object is strongly limited and the aspect ratio of the object is kept, the salient object is scaled according to the corresponding quad size of the initial guess. As a result, the scale of the salient object in the figure becomes relatively small, so it does not look as good as the result of seam carving based method. Due to lack of time, we simply scale the image as initial guess. For improvement, we can try to adjust the initial guess, for example by increasing the size of the higher energy quads and decreasing the size of the lower energy quads. Or we can adjust s_f according to the energy of the corresponding quad face f after s_f's are calculated using Eq. (8). It can be expected that a better initial guess will greatly improve the results of warping based method, which can make the salient objects look larger.

4.4 Linear Programming

We used a python library called PuLP [Mitchell et al. 2011] to implement the method mentioned in subsection 3.3. However, we did not get a good result before the final report. We will show the existing results and discuss ways to improvement.

Binary variable. In the case of using binary matrix X, the biggest problem is the computation time. We tried GLPK [Makhorin 2008] and MOSEK [ApS 2022] solvers and neither could solve the binary linear programming problem in a reasonable time for large input image. As a result, it can only be tested on very small images. An example of resizing a 21×32 size image to 21×28 is shown in Fig. 7.



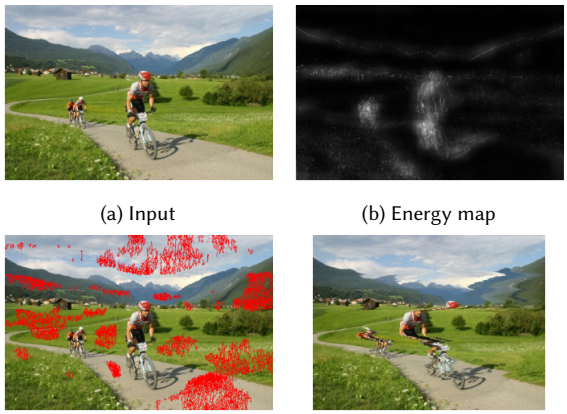
(a) Input (b) Energy map

(c) Selected pixels (red) (d) Output

Fig. 7. Result with binary linear programming method.

From the result in Fig. 7, the binary linear programming based method gave the expected result. It can also be seen from the figure that our consecutive constraint is not guaranteed to find the strict seams, but the same effect as seams can be obtained to reduce the distortion because the selected pixels are still connected.
The computation time of the result in Fig. 7 is 9.8 seconds (MOSEK solver was used), however, when we try an image of size 32×49, it takes a very long time to compute and cannot be finished in 30 minutes. The reason for this problem is most likely the limitation of the algorithm itself used by the solver. If we had more time, we might try other solvers or algorithms to see if we could improve the computation time.

Continuous variable. Unlike the case of binary, when using continuous variables, results can be obtained in a normal time even for a larger images. However, when using constraints as the binary method, we didn't get the seam-like contiguous pixel selection as expected.
In the example of Fig. 8, we used an image of size 300×460 and a target size of 300×400. The computation time is 41.8 seconds.
As can be seen in Fig. 8, the selected pixels are not connected as seam, which leads to poor output image.
The continuous linear programming method has a great potential to obtain better results and faster computational speed than regular



(a) Input (b) Energy map

(c) Selected pixels (red) (d) Output

Fig. 8. Result with continuous linear programming method.

seam carving method. In future work, the following things can be tried:

- Change the consecutive constraints
- Change the final pixel selection strategy: Instead of selecting the largest ones directly, select pixels based on the position of candidate pixels in the previous and next rows

to improve the performance.

5 CONCLUSION

REFERENCES

MOSEK ApS. 2022. *MOSEK Optimizer API for Python 9.3.20.* https://docs.mosek.com/latest/pythonapi/index.html

Bahareh Asheghi, Pedram Salehpour, Abdolhamid Moallemi Khiavi, and Mahdi Hashemzadeh. 2022. A comprehensive review on content-aware image retargeting: From classical to state-of-the-art methods. *Signal Processing* (2022), 108496.

Shai Avidan and Ariel Shamir. 2007. Seam carving for content-aware image resizing. In *ACM SIGGRAPH 2007 papers*. 10–es.

Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* 20, 11 (1998), 1254–1259.

Evan Lavender. 2019. genetic-seam-carving. https://github.com/EvanLavender13/genetic-seam-carving.

Feng Liu and Michael Gleicher. 2005. Automatic image retargeting with fisheye-view warping. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. 153–162.

Andrew Makhorin. 2008. GLPK (GNU linear programming kit). *http://www. gnu. org/s/glpk/glpk. html* (2008).

Stuart Mitchell, Michael OSullivan, and Iain Dunning. 2011. PuLP: a linear programming toolkit for python. *The University of Auckland, Auckland, New Zealand* 65 (2011).

Yuzhen Niu, Feng Liu, Xueqing Li, and Michael Gleicher. 2012. Image resizing via non-homogeneous warping. *Multimedia Tools and Applications* 56, 3 (2012), 485–508.

Saulo AF Oliveira, Francisco N Bezerra, and Ajalmar R Rocha Neto. 2015. Genetic seam carving: A genetic algorithm approach for content-aware image retargeting. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 700–707.

Saulo AF Oliveira and Ajalmar R Rocha Neto. 2015. An improved genetic algorithms-based seam carving method. In *2015 Latin America Congress on Computational Intelligence (LA-CCI)*. IEEE, 1–6.

Tongwei Ren, Yan Liu, and Gangshan Wu. 2009. Image retargeting using multi-map constrained region warping. In *Proceedings of the 17th ACM international conference on Multimedia*. 853–856.

Michael Rubinstein, Ariel Shamir, and Shai Avidan. 2008. Improved seam carving for video retargeting. *ACM transactions on graphics (TOG)* 27, 3 (2008), 1–9.

Yu-Shuen Wang, Chiew-Lan Tai, Olga Sorkine, and Tong-Yee Lee. 2008. Optimized scale-and-stretch for image resizing. In *ACM SIGGRAPH Asia 2008 papers*. 1–8.

Darrell Whitley. 1994. A genetic algorithm tutorial. *Statistics and computing* 4, 2 (1994), 65–85.