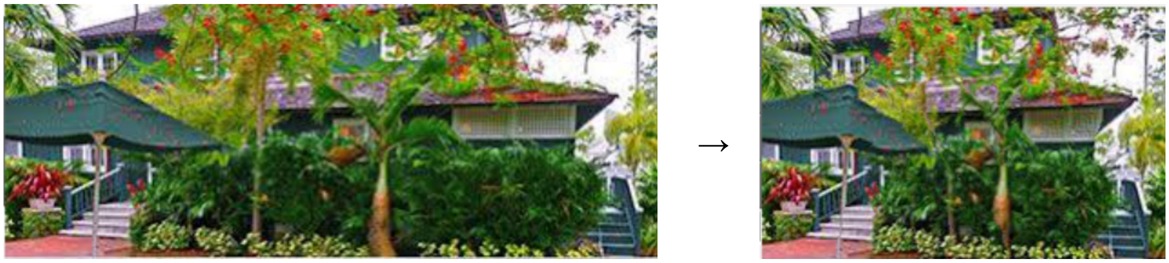
Yuhan Liu

# Algorithms and Data Structures Final Project: **Seam Carving**

## Introduction:

### Content-Aware Image Resizing

While editing images, it is often difficult to retain the crucial content while cropping or resizing. Seam carving, a graphical application of dynamic programming, is an image-resizing algorithm that maintains important elements. The algorithm locates and deletes pixels along paths through the image that have the least content, thereby resulting in resized images that are smooth and complete in subject matter. Seam carving is an effective method of displaying images without distorting them. Thus, the method is useful in resizing images as they are viewed on screens and windows of various sizes. Seam carving, although not as often implemented, goes hand-in-hand with the dynamic programming application of rescaling text to fit the page.

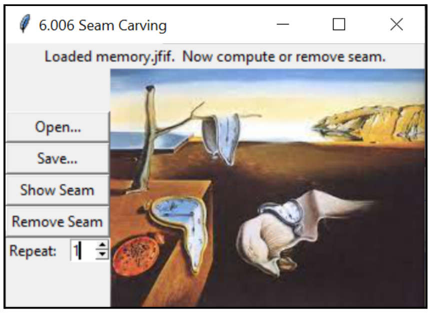


### Dynamic Programming and the Seam Carving Algorithm

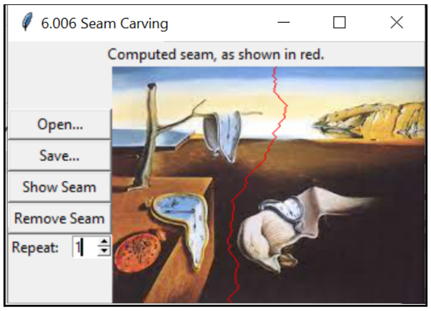
In computer science, dynamic programming is a technique for solving optimization problems. The technique works by efficiently breaking down the larger problem to solve its subproblems. In terms of image-resizing, dynamic programming is used to find the path in which pixels are deleted from. The process begins by calculating the energy map of the given image. The energy map, constructed in a matrix, shows the change in gradient throughout the picture. Dynamic programming is used to select pixels along a path, called a seam, through the energy map that contains the lowest energy. After the lowest-energy seam is located, the algorithm can begin deleting pixels to resize the image.

# Program:

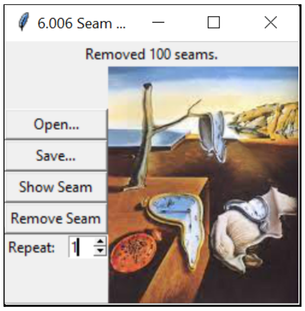
This implementation of seam carving uses the Python Imaging Library (PIL) and Tkinter to manipulate image files and display GUIs. Upon running the program, Tkinter opens a window that allows for the user to load images and explore seam deletion.



Loaded memory.jfif.  Now compute or remove seam.



Computed seam, as shown in red.



Removed 100 seams.

PIL functions are used to create a matrix of pixel values for the uploaded image. Energy() calculates the value of energy for a pixel in the matrix at (i,j) based on the color difference of surrounding pixels. This method calls distance(), which computes the value of that color difference. Energy() is then used in the dynamic programming algorithm to find the entire path down the image with the least energy. In this sense, the energy value is also the cost. To minimize disruptions in the image, dynamic programming finds the path with the lowest cost, or energy, to delete.

The actual dynamic programming algorithm is found in best_seam(). The algorithm begins in the top row by selecting the lowest-energy pixel. Then, the algorithm traverses down through the image by picking the minimum of the three closest pixels below the current one: the pixel directly below, the pixel to the left, and the one the right. This process creates an entire path that runs down the image. As dynamic programming solely looks at the energy in the pixels, a backpointer list is needed to store the locations of the pixels in the lowest seam. This list is used to create the seam[] array and is passed to the remove_best_seam function. This process is repeated for every seam that the user wishes to remove.

Code Disclaimer: The outline for this project as well as the code involving Tkinter and PIL were prepared in Python 2 by the Introduction to Algorithms (2011) class on MIT Opencourseware.

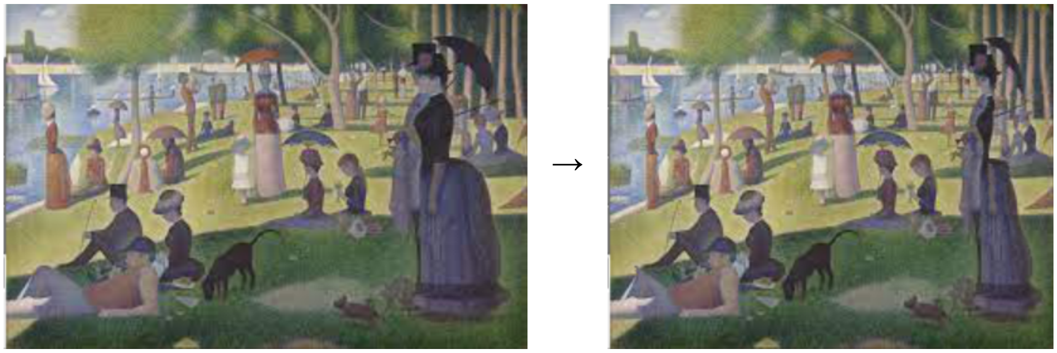
# Analysis and Comparison:

### Time Complexity

The time complexity of a Dynamic Programming algorithm like seam carving depends on the individual complexities for solving the subproblems, which, in this case, is finding the lowest-energy seam that starts at the top row of the image and ends at the pixel (i , j). These subproblems only involve calculating minimums, resulting in a simple time complexity O(1). The subproblems are applied to every pixel in the image. For an image that has a length and width of L and W respectively, the time complexity ends up as O( L x W ). O( Area ), which is linear time, is the generally accepted complexity of the seam carving algorithm.

### Pitfalls of Seam Carving

Seam-Carving is content-aware, as only areas with the lowest energies are removed, but only to an extent. In an image, areas identified as low-energy may be significant to a viewer. This is most apparent in resizing images of people, especially when the background has more variations in color than the human subject. To improve the algorithm, a system could be implemented to manually identify pixels that are important, thereby overriding the algorithm in certain areas of the image. Another improvement would be to allow the algorithm to take user input that specifies an area of the image, with an interface or tool that resembles “cropping”, in which a seam would be calculated. Although the addition of these features would not alter the algorithm or its complexity, they would significantly improve the alterations of specific images.



When resizing A Sunday Afternoon on the Island of La Grande Jatte by Georges Seurat, the seam computed is along the woman's body as her dress is uniform and dark. This results in the woman's distortion.

# References:

Das, A. (2019, May 14). Real-world dynamic programming: Seam carving. Retrieved April 05, 2021, from https://avikdas.com/2019/05/14/real-world-dynamic-programming-seam-carving.html

Demaine, E., & Devadas, S. (2011). *Introduction to Algorithms: 6.006 Problem Set 7*. In *MIT OpenCourseware*. Retrieved April 05, 2021, from https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/assignments/MIT6_006F11_ps7.pdf

Karanth, K. (2018, May 28). Implementing seam carving with python. Retrieved April 05, 2021, from https://karthikkaranth.me/blog/implementing-seam-carving-with-python/

Tao, X. (n.d.). Seam carving. Retrieved April 05, 2021, from http://cs.brown.edu/courses/cs129/results/proj3/taox/