JINNAN CAMPUS

College of Software

Master in Software Engineering

COMPUTER VISION

First Semester Assignment Project

December 26, 2024

Project title: Seam Carving for Content Aware Image Resizing

Group Members

Laggah Julius – ID: 2120246034

Jusu Abdul Karim – ID: 2120246025

Sandar Win – ID: 2120246058

Turay Zidida-Damekhaly – ID: 2120246056

Introduction

1.1 Problem Statement

Resizing images often leads to losing important details or distorting the original look of the image. Traditional methods like cropping or scaling are not ideal because they fail to focus on the most important parts of an image, especially when adjusting the size while keeping the overall appearance intact. Seam carving solves this issue by using a smart technique that focuses on the parts of the image that people find most noticeable, such as the center or main objects. This method ensures that the key details remain untouched while less important areas are either reduced or expanded, depending on the resizing needs. Unlike other resizing techniques, seam carving allows images to be resized intelligently without destroying their main content. It uses an energy map to identify which parts of the image are most important and finds the least noticeable paths, called seams, to remove or expand. For example, if you want to make an image smaller, the algorithm removes the least important seams, which are narrow paths of pixels, to shrink the image. If you need to enlarge an image, new seams are added in less important areas to make it larger. This method makes resizing more effective, preserving the essential content while adjusting the image size. Seam carving is a powerful solution for resizing images intelligently, ensuring the result looks natural and visually appealing.

1.2 Objectives

The primary goal of this project is to implement content-aware image resizing using seam carving techniques. This involves:

Seam removal and insertion to resize images while preserving essential content.

Object removal using seam carving and energy map calculations.

Maintaining the photorealistic quality of resized images.

1.3 Overview of Seam Carving

Seam carving is a content-aware image resizing technique that identifies optimal seams (continuous pixel paths) for removal or insertion. By manipulating these seams, images can be resized or modified without significant loss of visual quality. The algorithm, as proposed by Avidan and Shamir, leverages gradient energy maps to evaluate seam importance, enabling both resizing and object removal functionalities.

2. Background and Literature Review

2.1 What is Seam Carving?

Seam carving is a method of resizing images by adding or removing vertical or horizontal seams. A seam is a connected path of pixels, either from top to bottom or left to right, chosen based on its low energy (importance). Unlike traditional resizing methods, seam carving maintains critical features and the overall composition of the image.

2.2 Traditional Image Resizing Techniques

Standard techniques like cropping or scaling often distort images or cut out essential content. Seam carving overcomes these limitations by analyzing the image's energy map and selectively removing or adding seams, thus preserving the image's important features and aspect ratios.

3. Methodology

3.1 Overview of the Seam Carving Algorithm

The algorithm works by finding and removing the least important seams in an image. A seam is a connected path of pixels that can run vertically or horizontally across the image. The importance of each pixel in the image is measured using an energy map, which calculates the gradients to determine how significant or noticeable each pixel is. Pixels with lower energy are considered less important. The algorithm prioritizes removing these low-energy seams to resize the image while keeping its key elements intact. This method avoids the distortion and loss of important features that are common with traditional resizing techniques like cropping or scaling. By carefully removing or adding these seams, the image maintains its quality and focus, ensuring minimal distortion even after resizing.

3.2 Energy Map Calculation

The energy map is created by summing the absolute values of the image gradients in both the x and y directions for each pixel. This map highlights high-energy areas (important content) and low-energy areas (less critical regions) to guide seam selection.

The horizontal gradient (dx) and the vertical gradient (dy) are calculated using the sobel operator. These gradients highlight the edges in the image by detecting changes in the pixel intensity along the horizontal and vertical directions

Horizontal Sobel Filter (Gx)

```
[[-1, 0, +1],
 [-2, 0, +2],
 [-1, 0, +1]]
```

Vertical Sobel Filter (Gy)

```
[[-1, -2, -1],
 [ 0,  0,  0],
 [+1, +2, +1]]
```

The magnitude of the gradient at each pixel is computed as the Euclidean norm of the horizontal and vertical gradients

3.2 Output Dimensions

For resizing, users can specify the desired dimensions of the output image, such as the new height and width. For example:

```
# Output dimensions
new_height = 500
new_width = 700
```

The user can provide these dimensions interactively or by modifying the script directly. The seam carving algorithm will then resize the image to these specified dimensions while preserving the most important content. This flexibility allows users to customize the resizing process according to their needs, ensuring that the resized image meets their specific requirements.

3.3 Seam Identification

Using dynamic programming, seams are identified as follows:

Initialization: The energy values of the first row are used to initialize the DP table.

Recurrence Relation: For each pixel ‘dp[i][j]’, calculate the minimum cumulative energy by considering the three pixels above it: ‘(i-1, j-1)’, ‘(i-1, j)’, and ‘(i-1, j+1)’.

Backtracking: Trace back from the pixel with the minimum cumulative energy in the last row to retrieve the seam.

3.4 Seam Removal and Insertion

Seam Removal: Deletes pixels along a low-energy seam to reduce image size.

Seam Insertion: Adds artificial pixels along a previously removed seam to enlarge the image.

4. Implementation

4.1 Tools and Libraries Used

NumPy: Efficient numerical operations for handling multidimensional arrays.

SciPy: Provides the Sobel filter for gradient calculation in the energy map.

Pillow: Handles image loading, processing, and saving.

OS: Manages file paths and directories.

Python Standard Libraries:

sys: Enables command-line interactions.

input: Facilitates user-driven operations.

4.2 Code/Program Structure

The project is organized as follows:

```
├── seam_carving/          # Main seam carving module
│   ├── __init__.py        # Package initialization file
│   ├── seam_carving.py    # SeamCarver class implementation
├── main.py                # Entry point for the project
├── in/                    # Input directory
│   ├── images/            # Stores input images
│   ├── masks/             # Stores mask images
├── out/                   # Output directory for processed images
```

Algorithm for Seam Carving

Input:

- Image: The original image to be resized.
- New_Width: Desired width of the resized image.
- New_Height: Desired height of the resized image.

Output:

- Resized image with preserved important content.

Steps:

1. **Preprocessing**:

- Convert the input Image to a numerical matrix representation.
- Initialize the Energy_Map with pixel energy values calculated from gradients.

2. **Energy Map Calculation**:

- For each pixel (i, j) in the image: Compute the gradient magnitude using Sobel filters: Energy[i][j] = |Gradient_x[i][j]| + |Gradient_y[i][j]|
- Store the result in the Energy_Map.

3. **Seam Identification** (Dynamic Programming):

- Initialize a DP_Table for cumulative seam energy:
- Set the first row of DP_Table equal to the first row of Energy_Map.
- For each subsequent row i and column j:

Compute: DP[i][j] = Energy[i][j] + min(DP[i-1][j-1], DP[i-1][j], DP[i-1][j+1])

- Handle edge cases for the first and last columns.

4. **Backtracking to Trace Seam**:

- Start from the pixel in the last row with the minimum energy in DP_Table.
- Trace back the path by selecting the minimum neighbor from the row above.
- Store this path as the identified seam.

5. **Seam Removal or Insertion**:

For Seam Removal:

- Remove the pixels along the identified seam from the image.
- Update the Energy_Map to reflect the change.

For Seam Insertion:

- Duplicate the identified seam.
- Insert new pixels based on neighboring pixel values to create a seamless blend.

6. **Repeat for Desired Dimensions**:

- Continue steps 3 to 5 until the New_Width and New_Height are achieved.

7. **Postprocessing**:

- Convert the modified matrix back to an image format.
- Save or display the resized image.

5. Discussion

5.1 Challenges Faced During Implementation

Masking: Ensuring the alignment and accuracy of masks was a key challenge. Misaligned masks often led to errors in protecting or removing specific regions. To overcome this, we developed a preprocessing step to verify mask dimensions and alignment with the input image, ensuring precise application.

Energy Map: Fine-tuning gradient calculations for optimal seam identification required careful adjustments. Initially, the energy map struggled with accurately highlighting edges in complex images. By implementing advanced Sobel filters and experimenting with different gradient thresholds, we improved the accuracy and reliability of the energy map.

Performance: Managing computational efficiency for large images was difficult due to the high processing demands of dynamic programming and seam calculations. We optimized the code by reducing redundant computations and using efficient data structures, significantly speeding up the algorithm for large-scale images.

Output Quality: Preserving photorealism in resized images required balancing seam removal with maintaining the overall structure. Early results sometimes showed artifacts or distortions. By incorporating advanced interpolation techniques and testing on a wide variety of images, we refined the algorithm to deliver more natural and visually appealing outputs.

6. Conclusion and Future Work

6.1 Summary of Achievements

This project successfully implemented the seam carving technique for intelligent image resizing. It introduced content-aware resizing, which preserves important features and minimizes distortion. The algorithm enables flexible resizing, maintains photorealistic quality, and incorporates user-defined dimensions and masks to achieve accurate and customized results. Additionally, the project tackled several challenges and provided solutions that improved the robustness and efficiency of the implementation.

6.2 Potential Improvements

While the project achieved its primary objectives, several areas could be enhanced:

Advanced Object Removal: Improving the precision of object removal by integrating machine learning techniques to identify and handle complex objects.

Efficiency for Large Images: Optimizing the algorithm further to handle very large images without significant delays, possibly by utilizing parallel processing.

Real-Time Features: Introducing real-time resizing capabilities and a user-friendly graphical interface (GUI) for broader accessibility.
Dynamic Seam Manipulation: Adding functionality for interactive seam editing, where users can manually define or refine seams based on preferences.
In the future, expanding this project to include advanced image analysis and machine learning integration could make it a more powerful tool for modern image processing needs.

Steps to Run the Project

Install Requirements:

Python 3.11

```
pip install numpy scipy pillow
```

Set Up Directory: Organize the project as follows:

```
Project/
├── seam_carving.py
├── main.py
├── in/
│   ├── images/
│   ├── masks/
├── out/
```

Place Input Files:

Input images: in/images/

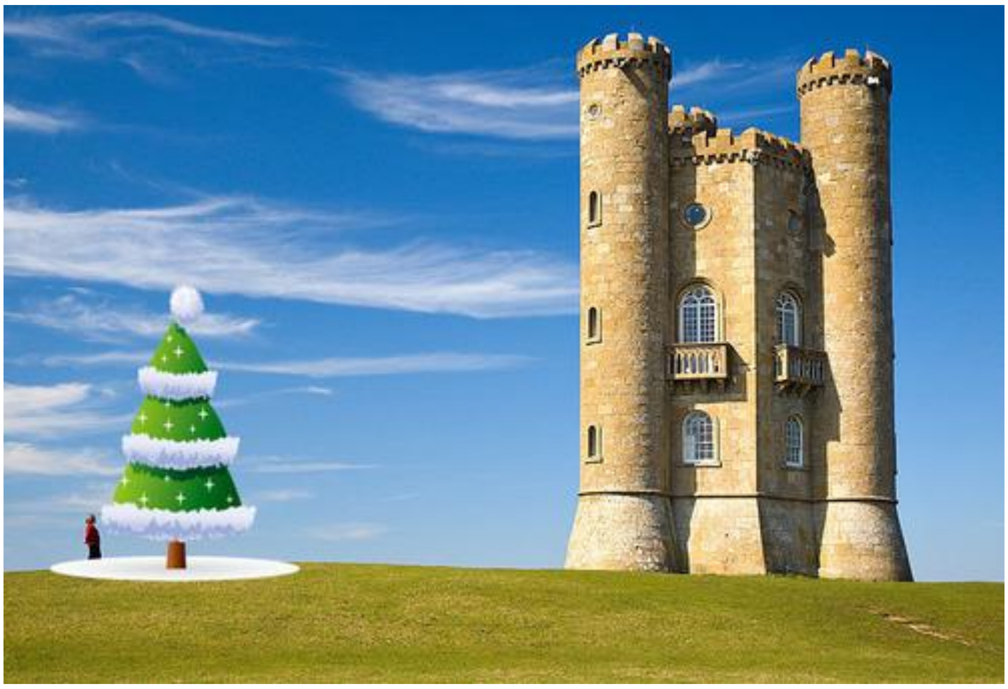
Mask files (optional): in/masks/

Run the Project:

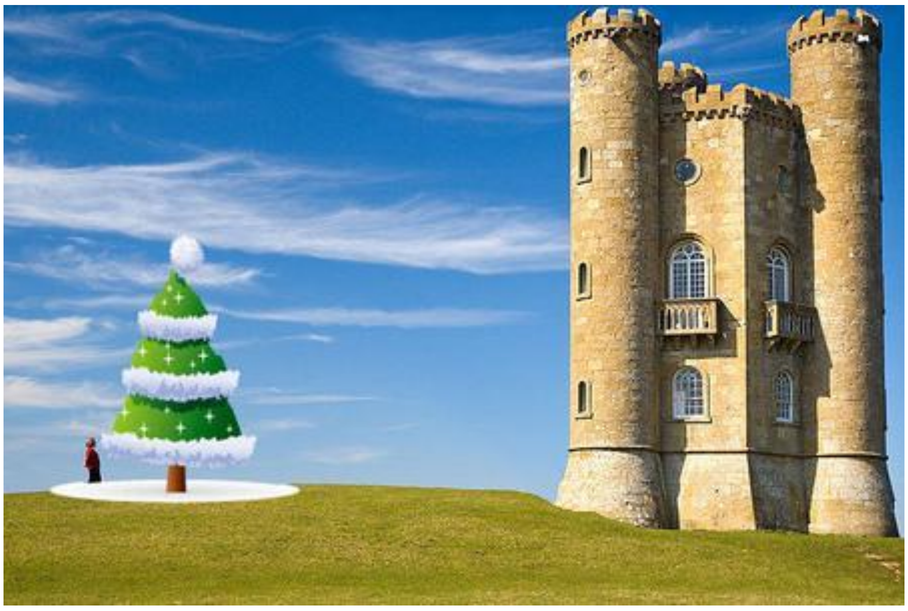
```
python main.py
```

A sample input and output image

Input image 500 X 339



Output Image 450 x 300



References

Avidan, S., & Shamir, A. Seam Carving for Content-Aware Image Resizing.

Medium: Seam Carving with Dynamic Programming.

Brown University CS129 Course: Seam Carving Project.

Princeton Coursera: Seam Carving Algorithm Specifications.

GitHub: Seam Carving by Vivian Lee.

https://www.researchgate.net/publication/336586477_Seam_Carving_for_Enhancing_Image_Usability_on_Mobiles
https://medium.com/@himanshu.sharma.for.work/seam-carving-with-dynamic-programming-and-basic-energy-calculation-7d7888f3dd31#:~:text=A%20seam%20is%20a%20connected,without%20distorting%20critical%20visual%20information.
https://cs.brown.edu/courses/cs129/results/proj3/taox/
https://coursera.cs.princeton.edu/algs4/assignments/seam/specification.php
https://github.com/vivianhylee/seam-carving?tab=readme-ov-file#algorithm-overview