**CSCE 689 Computational Photography**

**Report on Assignment 3**

# Overview

The goal of the project was to perform image blending give a source image and its mask and a target image. This was achieved using two approaches: Pyramid blending and Poisson blending.

# Deliverables

The contents of the submission have the following:

|  |  |
| --- | --- |
| Code\ | Directory containing the python source files |
| Report\_430000753\_RizuJain.pdf | This report. |

# Setup

The assignment was developed in the following environment:

* Host OS: Windows 10
* IDE: Spyder (Python 3.7)

# Task 1: Pyramid Blending

This task is implemented in the function:

def PyramidBlend(source, mask, target)

The overall process and the functions that implement the subtasks are briefed in the following flow diagram:

To reduce the size of the image to build the gaussian pyramid, the following function is implemented:

def myDownsample(img\_in)

To expand the size of the image to build Laplacian pyramids and at the end collapse the images in the pyramid, the following function is implemented:

def myUpsample(img\_in)

Both the above functions deploy the cv2.pyrDown() / cv2.pyrUp() for the resizing and subsequent filtering and sampling to build the pyramid.

The results for this task are given in the [results section](#_Observations_&_Results).

# Task 2: Poisson Blending

The task here is basically to also determine which pixels from source or target in the masked region should blend into the final target image. Rather than naively copying-paste the overall intensity of the pixel, we computer gradient of the source and target image and finally the blended pixel in the target image corresponds to the gradient of the pixel in the source image for the masked region.

The results for this task are given in the [results section](#_Observations_&_Results).

# Extra Credits: Mixing Gradients

# Observations & Results

The following images depict the given source images, the target images and the results obtained in Task 1 (i.e. Pyramid blending of images) and Task 2(i.e. Poisson blending of images.)

|  |  |
| --- | --- |
| Source Image: | Target Image: |
|  |  |
| Pyramid Image Blending: | Poisson Image Blending |
|  |  |

|  |  |
| --- | --- |
| Source Image: | Target Image: |
|  |  |
| Pyramid Image Blending: | Poisson Image Blending |
|  |  |

|  |  |
| --- | --- |
| Surce Image: | Target Image: |
|  |  |
| Pyramid Image Blending: | Poisson Image Blending |
|  |  |

|  |  |
| --- | --- |
| Source Image: | Target Image: |
|  |  |
| Pyramid Image Blending: | Poisson Image Blending |
|  |  |

|  |  |
| --- | --- |
| Source Image: | Target Image: |
|  |  |
| Pyramid Image Blending: | Poisson Image Blending |
|  |  |

|  |  |
| --- | --- |
| Source Image: | Target Image: |
|  |  |
| Pyramid Image Blending: | Poisson Image Blending |
|  |  |

|  |  |
| --- | --- |
| Source Image: | Target Image: |
|  |  |
| Pyramid Image Blending: | Poisson Image Blending: |
|  |  |
| Mixing Gradients: | |
| Source Image: | Target Image: |
|  |  |
| Pyramid Image Blending: | Poisson Image Blending |
|  |  |
| Mixing Gradients: | |
|  | |

## External Example:

The following example is taken externally. The mask image generated using getMask() is also shown:

|  |  |
| --- | --- |
| Source Image: | Target Image: |
|  |  |
| Mask Image: | |
|  | |
| Pyramid Image Blending: | Poisson Image Blending |
|  |  |
| Mixing Gradients: | |

Some inferences were drawn:

# References

1. <https://en.wikipedia.org/wiki/Discrete_Poisson_equation>
2. Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. ACM Trans. Graph. 22, 3 (July 2003), 313–318. DOI:https://doi.org/10.1145/882262.882269

~ End of Report ~