Image Processing - Exercise 3

Shir Hana Stern, shirste, 325945236

Introduction

The exercise is divided into two tasks, each focused on leveraging image processing.
1) Blend two images seamlessly using a mask. The mask controls the transition between images, ensuring smooth blending without visible edges.
2) Create a hybrid image that shows one image up close (using high-frequency details) and another image from afar (using low-frequency content).
The exercise involves two key concepts: image blending and frequency decomposition. Image blending uses spatial masks to combine images smoothly, ensuring seamless transitions without abrupt edges. Frequency decomposition separates an image into low-frequency (broad features) and high-frequency (fine details) components using Gaussian and Laplacian pyramids. These techniques allow for smooth integration of images in blending and dynamic visual effects in hybrid images.

Algorithm

Blend two images seamlessly using a mask (Description):

<u>Laplacian Pyramids Construction:</u> Each image is decomposed into a Laplacian Pyramid to isolate high-frequency details across multiple scales. This involves iterative Gaussian blurring and subtraction. Concept: The Laplacian Pyramid captures image details at different resolutions, enabling multi-scale blending.
<u>Gaussian Mask Pyramid:</u> The blending mask is smoothed into a Gaussian Pyramid to allow gradual blending at all levels. Concept: Gaussian Pyramids smooth the mask to ensure seamless transitions between image regions.
<u>Combining Pyramids:</u> At each level, a weighted sum of the two Laplacian pyramids is computed based on the Gaussian mask: $L_c(i,j) = G_m(i,j)L_a(i,j) + (1 - G_m(i,j))L_b(i,j)$
Concept: The combined pyramid merges details at all scales proportionately.
<u>Reconstruction:</u> The final blended image is reconstructed by summing all levels of the combined pyramid. Concept: This step integrates both high and low-frequency components into the final composite.

Blend two images seamlessly using a mask (Implementation Details):

Algorithm Implementation:
Constructing pyramids and blending at each level was implemented using OpenCV. **Reconstruction** of the blended image involved resizing and summing pyramid levels.
Parts Implemented from Scratch:
Gaussian Pyramid Construction: Iteratively downsamples the input image.
Laplacian Pyramid Construction: Subtracts the upsampled Gaussian levels to isolate high-frequency details. **Pyramid Blending:** Combines corresponding levels of the Laplacian pyramids using Gaussian masks. **Reconstruction:** Upsamples and adds Laplacian pyramid levels to reconstruct the blended image.
Libraries Used:
OpenCV: Used for Gaussian blurring, resizing, and matrix operations due to its efficiency and robust image processing tools.
NumPy: Used for array manipulations and arithmetic operations.
Hyper-parameters:
Number of Pyramid Levels: Set to 5, balancing computational cost and blending quality.
Mask Normalization: Ensures the mask ranges between [0, 1] for correct blending weights.
Challenges and Solutions:
One challenge was understanding how to prepare the blending mask. Initially, it was unclear how the mask's design would influence the blending results. The solution was to experiment with different mask types, such as gradient masks or binary masks, to observe their effects on the blending transition. Normalizing the mask to a [0, 1] range and ensuring smooth gradients were key to achieving seamless blending.
Another challenge was ensuring the input images were compatible for blending. It required careful resizing to ensure both images and the mask had identical dimensions. Additionally, I realized that for successful blending, the images needed complementary content in the blending regions, as abrupt differences would disrupt the transition. Iterative testing helped refine these requirements and improve the final output.

hybrid image (Description):

Low-Pass Filtering for Image 2: A Gaussian blur is applied to Image 2 to retain only its low-frequency components, making it prominent when viewed from afar. Concept: Low-pass filtering smooths an image by removing sharp details and preserving broad patterns.
High-Pass Filtering for Image 1: Subtract a Gaussian-blurred version of Image 1 from the original to isolate its high-frequency details. Concept: High-pass filtering extracts fine details and sharp edges, ensuring Image 1 dominates when viewed closely.
Combining Frequencies: Scale and blend the high frequencies of Image 1 and the low frequencies of Image 2 to create the hybrid image. Concept: Frequency domain

blending allows each image to dominate at its respective viewing distance.
Output: The result is clipped to valid intensity ranges and converted back to an 8-bit image for visualization and saving.

hybrid image (Implementation Details):

Algorithm Implementation:
The algorithm combines high frequencies of one image with the low frequencies of another: Images are resized and normalized. Gaussian blur extracts low frequencies (Image 2), and high frequencies (Image 1) are isolated by subtraction. The components are scaled (alpha, beta) and blended. The result is clipped and converted for saving.
Parts Implemented from Scratch:
High-Frequency Extraction: Subtraction of Gaussian blur from the original image.
Low-Frequency Retention: Application of Gaussian blur to the second image.
Combination of Components: Weighted addition of the filtered components for the hybrid image.
Libraries Used:
OpenCV: Used for Gaussian blur, resizing, and image processing. Its efficient functions simplify complex operations.
NumPy: Facilitates array manipulations and arithmetic for frequency processing.
Hyper-parameters:
Sigma1 and Sigma2: Control the strengths of high-pass and low-pass filters, respectively. **Alpha and Beta:** Adjust the relative weights of high and low frequencies for better balance.
Challenges and Solutions:
Filter Strength Tuning: It was challenging to find appropriate sigma values for clear separation of frequencies. This was addressed by experimenting with different values and observing the results.
Image Alignment: Ensuring the images were resized to the same dimensions was crucial for successful blending.
Balancing Components: Fine-tuning the alpha and beta weights to achieve an even prominence of both images required iterative testing.

Differences between the 2 algorithms:
The blending algorithm combines two images seamlessly using multiresolution pyramids and a mask, ensuring smooth transitions between regions. The hybrid image algorithm merges high frequencies of one image with low frequencies of another, creating a dual-effect image that changes appearance based on viewing distance.

Results

Blending results:

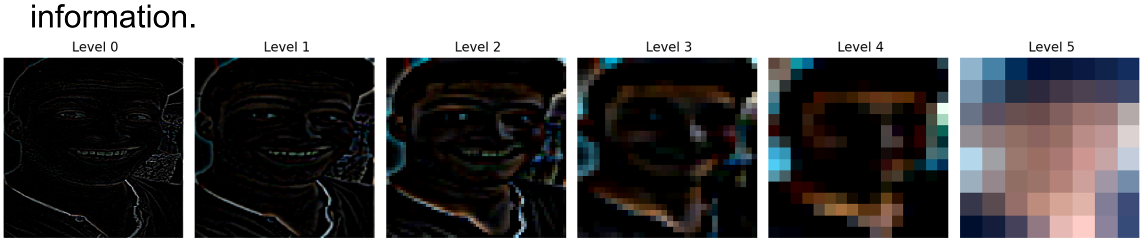
Intermediate results:

1) The failed blending of the first task occurred because a low pyramid level (1 or 2) didn't provide enough resolution for smooth transitions, making the stitching between the two images very noticeable.
2) The blending of the second task failed because the chosen parameters resulted in one image being much more dominant than the other, making it difficult to distinguish the second image.

Pyramids

Gaussian pyramid observations: Each subsequent level of the pyramid is a smaller, blurrier version of the original image. This is achieved by repeatedly applying a Gaussian filter followed by downsampling. As the pyramid levels increase, the image resolution decreases, retaining only the low-frequency details while discarding finer details.

Laplacian pyramid observations: The Laplacian pyramid shows progressively finer details at each level, where the higher levels represent the edges and fine structures, and the lower levels capture broader, smoother components of the image. As we go down the pyramid, the image becomes more abstract, highlighting different scales of information.



Conclusion

Key findings:

- Seamless blending was achieved using multi-resolution pyramids, ensuring smooth transitions between two images based on a mask.
- A hybrid image was successfully created, with clear separation of high-frequency details for up-close viewing and low-frequency features for distant viewing.
- Hyper Parameters were crucial for balancing the visual contributions of each image in both tasks.