Assignment 04

Image Debluring and Poission Stitching

Strict Deadline: 23:59PM, 14th, Nov, 2024

This assignment covers 20% of your final grades.

Late submission policy:

For the whole semester, you totally have 5 exempt days for late submission days. After expert days running out, you totally have 5 slip days, 20% of the grades will deducted per day for the relative assignment.

Introduction:

Due to the physical size limitations of smartphone camera modules, it's inevitable that images will suffer from significant blurring and noise issues. Consequently, enhancing the photo quality of smartphones has become a hot topic in the industry. Current edge computing capabilities in smartphones are constrained by size and power consumption, typically offering limited computing power and memory. However, smartphone photography often involves large image sizes, some even exceeding 12 megapixels, which surpasses the processing and memory capabilities of smartphones for image enhancement on such a large scale. Therefore, this assignment addresses the issue by dividing images into no-large-than 512x512 blocks for deblurring, followed by using Poisson blending to stitch these image blocks together, thereby achieving the processing of large complete images.

Part 1. The "Constrained Least Squares Filtering" (CLSF) (50 points).

Step 1. Problem Formulation

Understand the problem that CLSF aims to solve: restoring an image (x) from its blurred and noisy version (b). The relationship is given by:

$$b = h * x + n$$

- h represents the blurring function.(use GaussianBlur with kernel size 17*17 $\sigma = 3$)
- n represents the noise. (use gaussian noise with $\sigma = 0.01, 0.03, 0.1$)
- $*$ denotes convolution.
- The goal is to estimate x from b.

- Use the reference image in Homework 1 as x to simulated your observation b

Step 2 Define the Objective Function

The objective function for CLSF balances fitting the observed data against smoothing to reduce noise. It is defined as:

$$E(f) = ||h * x - b||^2 + \lambda||\nabla^2 x||^2$$

- $||h * x - b||^2$ is the data fidelity term.
- $||\nabla^2 x||^2$ is the regularization term for noise suppression.
- λ is the regularization parameter.

Step 2 **Define the Objective Function

Step 2 **Define the Objective Function

Step 2 **Define the Objective Function

Step 3: Frequency Domain Transformation

Transform the problem into the frequency domain for computational efficiency, turning convolutions into multiplications:

$$B = H \cdot X + N$$

The frequency domain form of the objective function:

$$E(X) = ||H \cdot X - B||^2 + \lambda||L \cdot X||^2$$

- B, H, and X are the Fourier Transforms of b, h, and x, respectively.
- L is the Fourier Transform of the Laplacian operator.

Step 4: Solve for (F)

Find X by minimizing the objective function, leading to:

$$X = \frac{H^* \cdot B}{|H|^2 + \lambda|L|^2}$$

- H^* is the complex conjugate of H.

Step 5: Inverse Fourier Transform

Apply the Inverse Fourier Transform to X to get the restored image x in the spatial domain:

$$x = \mathcal{F}^{-1}(X)$$

Step 6: Post-Processing

Post-process (f) as needed to enhance visual quality or reduce artifacts. Techniques may include:

- Histogram equalization
- Sharpening
- Noise reduction

Step 7: Parameter Tuning

Experiment with different values of λ to find the optimal balance for your specific case.

Implementation Note

When implementing CLSF, libraries like NumPy and OpenCV can facilitate the computational aspects, particularly for Fourier Transforms.

Part 2: Image Stitching with Poission Blending (50 points)

Image stitching with Poisson blending is a sophisticated technique used to seamlessly combine multiple overlapping images into a single, large composite image. This process is particularly useful in creating panoramic images or in situations where a larger field of view is desired. Poisson blending is used to smoothly blend the overlapping regions by solving a Poisson equation, which helps in maintaining the gradient (color and intensity changes) consistency across the images. Here's a simplified algorithmic outline for image stitching with Poisson blending:

Step 1: Preparation of Image Blocks

- **Input**: A set of pre-segmented, overlapping image blocks(results of Part 1) ready to be stitched together.
- **Process**:
    1. Identify the overlapping regions among the image blocks. This can be predefined or calculated based on the input configuration.
    2. Create masks for the overlapping regions to be used in the blending process

Step 2: Poisson Blending

- **Process**:
    1. For each overlapping region, set up a Poisson equation. The goal is to adjust the pixel values in the overlapping areas so that the gradient (the difference between a pixel and

its immediate neighbors) in the stitched image is similar to the gradients in the original image blocks.
2. Solve the Poisson equations for all overlapping regions. This involves solving a large system of linear equations to find pixel values that best meet the gradient conditions while respecting the pixel values at the boundaries of the overlapping regions.
3. Efficient solving techniques like the conjugate gradient method or multigrid solvers are typically employed to handle the computational complexity.

Step 3. Final Composition

- **Process**:
    1. Combine the non-overlapping parts of the image blocks with the blended results in the overlapping areas. This step should be straightforward since the locations of the image blocks and their overlapping regions are predefined.
    2. Ensure that the transitions between the blended overlapping regions and the non-overlapping parts of the image blocks are seamless.

Step 4. Output

- **Output**: A single composite image that seamlessly integrates the pre-segmented image blocks.

Notes:

- This algorithm assumes that the pre-segmented image blocks are perfectly aligned and accurately represent sections of a larger scene that need to be stitched together.
- The quality of the final stitched image heavily depends on the accuracy of the overlapping region identification and the effectiveness of the Poisson blending in those regions.
- Tools and libraries like OpenCV for image operations and NumPy/SciPy for solving the Poisson equation can facilitate the implementation of this algorithm.

This simplified approach focuses on the core aspect of blending pre-segmented and aligned image blocks, eliminating the need for complex image alignment and transformation steps.

References

[1] A. Agrawal, R. Raskar, S. K. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling.ACM Transactions on Graphics (TOG), 24(3):828–835, 2005.

[2] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless. Gradientshop: A gradient-domain optimization framework for image and video filtering.ACM Transactions on Graphics (TOG), 29(2):10, 2010.

[3] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH'02, pages 257–266, New York, NY, USA, 2002. ACM.

[4] International Electrotechnical Commission and others. IEC 61966-2-1:1999. Multimedia systems and equipment–Colour measurements and management–Part 2-1: Colour management –Default RGB colour space–sRGB, 1999.

[5] P. P ́erez, M. Gangnet, and A. Blake. Poisson image editing. InACM Transactions on graphics (TOG), volume 22, pages 313–318. ACM, 2003.

[6] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs.ACM transactions on graphics (TOG), 23(3):664 –672, 2004.

[7] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. school of computer science.Carnegie Mellon University, Pittsburgh, PA, 15213:10, 1994.