Image deblurring

Matúš Bičanovský, xbican03@fit.vutbr.cz
Diana Maxima Držíková, xdrzik01@fit.vutbr.cz
Mária Nováková, xnovak2w@fit.vutbr.cz

POVa - Project documentation December 2024

1 Introduction

The implementation presented in this work is based on the **Stochastic Deconvolution**[1] framework introduced by Gregson et al. in their 2013 paper[2]. This novel approach addresses the non-blind image deconvolution problem using stochastic random walks, applying arbitrary priors, including non-convex and data-dependent regularizers.

We built on the code snippet provided by the authors. To enhance the method's usability and adaptability, we reimplemented the algorithm using OpenCV, a modern and widely adopted image-processing library.

We expanded the original framework by implementing missing features mentioned in the paper, such as:

- Handling boundary conditions and saturation regions
- Color processing
- Gamma correction
- Integration of three additional regularization terms

To validate our implementation, we designed an evaluation pipeline using the Jupyter Notebook. We evaluated our implementation using traditional metrics such as PSNR and SSIM to measure its performance.

Finally, we created and shared a dataset that includes blurred images, ground truth, and blur kernels. We developed this dataset to address the difficulty of obtaining such datasets for non-blind deblurring.

2 Team Contributions

We divided the work equally among all team members. Each member had their responsibilities, and we maintained constant communication throughout the process, providing feedback to ensure a correct and clean implementation.

- **M. Nováková:** Integrating four regularization techniques (e.g., gamma correction, sparse 1st and 2nd order regularizer, etc.), script for creating a dataset.
- **M. Bičanovský:** Color images, code maintenance, and finalization.
- **D. Držíková:** Rewrite the core algorithm in OpenCV, handling boundary conditions, saturation correction, and evaluation jupyter notebook.

[1] https://www.cs.ubc.ca/labs/imager/tr/2013/StochasticDeconvolution/

3 Implementation

The stochastic deconvolution algorithm operates by iteratively sampling, mutating, and optimizing. Each sample undergoes a mutation process where its position and energy are perturbed based on Gaussian noise or random resets. The energy of a mutation is evaluated by combining data fidelity, which measures the difference from the blurred input image, with the chosen regularizer's energy. Samples that reduce the overall energy are accepted, with a dynamic deposition energy controlling the acceptance rate. This adaptive mechanism ensures convergence towards a sharper and more accurate image. We take specific values from the paper (e.g., the `gamma value` is 2.0).

3.1 OpenCV remake

Our implementation of stochastic deconvolution builds upon the original algorithm by integrating the OpenCV library, enabling efficient handling of image processing tasks. The use of OpenCV simplifies operations such as reading, writing, and manipulating multi-channel (colour) images, which are essential for extending the algorithm to work beyond grayscale inputs.

3.2 Boundary conditions and saturation

To ensure accurate results near the edges of the images, we introduced boundary condition handling using a reflection padding strategy. Additionally, saturation issues, which can lead to inaccuracies in highly chromatic or bright regions, are addressed by generating a mask that excludes these problematic areas from the computation.

Ground Truth Mask of an image

Figure 1: RGB image: Example of a mask for oversaturated regions. The saturation threshold was set to 0.99 after normalization.

3.3 Regularizers

A key enhancement to our algorithm is the incorporation of multiple regularizers, each contributing uniquely to the refinement of the deblurring process. The `Total Variation (TV)` regularizer minimizes the image gradient magnitude to reduce noise while still preserving edges, a critical aspect for maintaining overall sharpness.

The `Gamma`[2] regularizer applies a fractional exponent to pixel intensities, introducing robust penalties that adapt to varying intensity levels. In addition, we implemented a weighted `combination` regularizer that blends both TV and Gamma methods, aiming to balance edge preservation with robust smoothing in experimental scenarios.

We further developed the `Sparse`[3] regularizer, which penalizes first- and second-order derivatives to encourage sparsity and thus avoid over-smoothing.

Then we tried to implement data-dependent and discontinuous regularizers. The `Data-dependent`[4] regularizer incorporates the gradient of the input image (using an exponential edge-weight) to guide the deblurring process, ensuring that fine details are preserved in the reconstructed output. We use the L1 regularizer formula.

Finally, the `Discontinuous`[1] regularizer employs a heavy-tailed Charbonnier function with a fractional exponent to robustly handle image discontinuities and noise. This approach penalizes large gradients less severely, allowing for sharp edges and discontinuities to remain in the final result.

3.4 Color Images

Finally, the algorithm was extended to process colour images by treating each channel consistently during deconvolution and regularization. This enhancement expands the applicability of the method to real-world scenarios.

4 Dataset

We decided to provide a script for generating such datasets because we want to address the difficulty of obtaining such datasets for non-blind deblurring. This script creates blurred versions of input images along with their corresponding blur kernels of size 3x3. We included several examples of generated datasets in the submitted solution.

The input images used in this process are sourced from the dataset provided in the paper *Learning a Non-blind Deblurring Network for Night Blurry Images*[2] by Liang Chen et al. Their dataset includes labels and blur kernels represented as images, which we did not use because we were working with numeric representation in an array. We decided to work with this dataset because its images were meant to be used for deblurring tasks.

[2]https://liangchen527.github.io/

5 Evaluation

The evaluation of regularizers for Stochastic Deconvolution was conducted using PSNR, SSIM, and execution time as metrics to assess performance. The TV regularizer demonstrated the best overall quality, achieving a PSNR of 32.81 dB and an SSIM of 0.91 while requiring a moderate execution time of 27.17 seconds. The data-driven regularizer closely followed, with a PSNR of 32.40 dB and SSIM of 0.90, though it required 54.82 seconds to complete.

In contrast, gamma correction and combination regularizers, while achieving reasonable PSNR and SSIM values (31.45 dB and 31.88 dB, respectively), exhibited significantly higher runtimes, exceeding 200 seconds. These approaches may be better suited for applications where runtime is less critical.

The base OpenCV implementation and the original method from the authors emerged as the fastest, with runtimes of around 10 seconds; however, they produced slightly lower PSNR and SSIM values, highlighting a trade-off between speed and image fidelity. The PSNR and SSIM values were evaluated for grayscale images.

Overall, the TV and data-driven regularizers offer a strong balance between quality and efficiency.

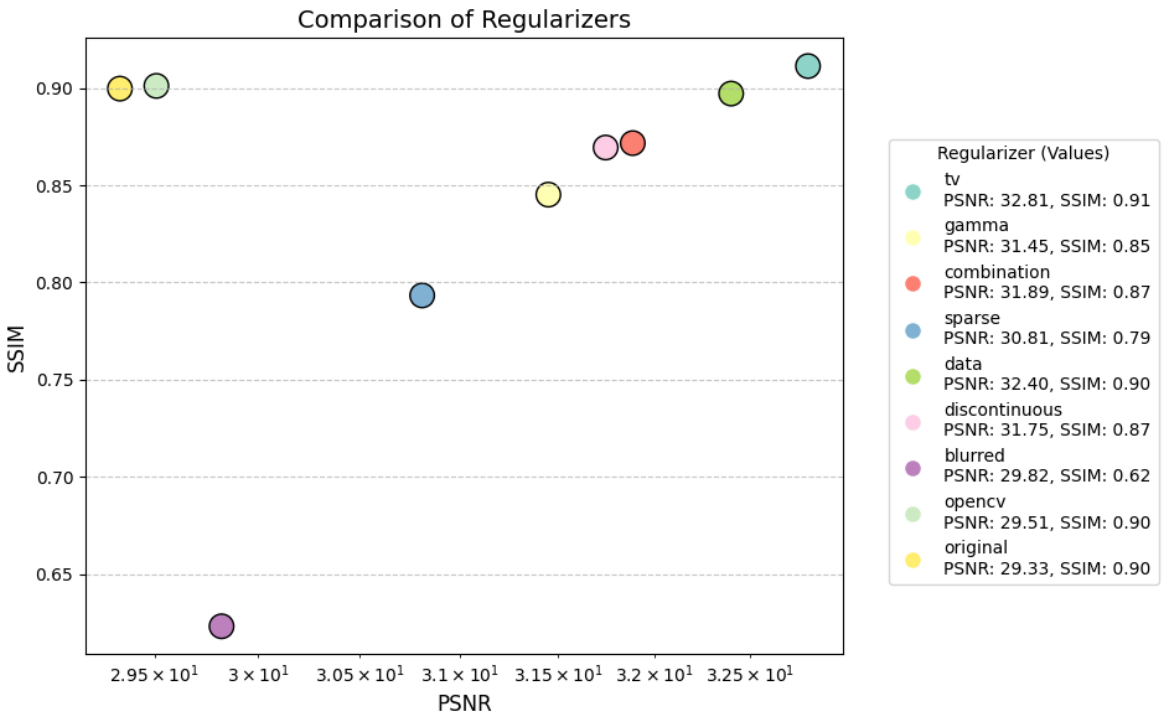

Figure 2: PSNR and SSIM evaluation

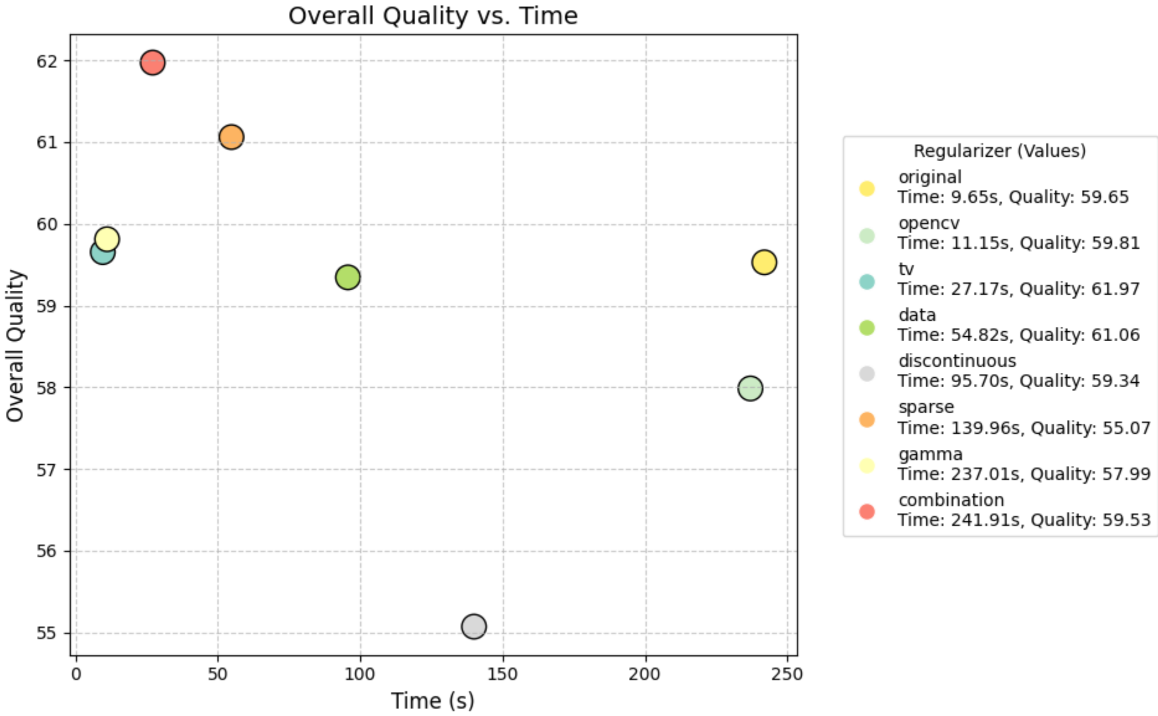

Figure 3: Time and Quality evaluation

6 Images

Ground Truth

Blurred image, input

Figure 4: Ground Truth image and its blurred version (Kernel (3x3 uniform 1/9)).

Total Variation (TV)

Gamma-corrected (GC)

Combination of TV and GC

Data-Dependent

Sparse 1st and 2nd Order Derivatives

Discontinuous

Figure 5: Deblurred images using various regularizers.

7 Summary

The project implements a stochastic deconvolution framework to address non-blind image deblurring using stochastic random walks. The algorithm was reimplemented using OpenCV to enhance usability and efficiency. Key improvements include boundary condition handling, colour image support, and the introduction of regularizers like Gamma, Sparse, Data-dependent, and Discontinuous. These regularizers refine the deblurring process by preserving edges, adapting to intensity variations, encouraging sparsity, and robustly handling noise.

We evaluated our results using traditional approaches such as PSNR and SSIM values. We compared the speed of the regularizers as well, and we deem the implementation to perform best when using Total Variation or Data-Driven regularizers. The overall worst was the Discontinuous Regularizer.

Additionally, we provided a script to generate a dataset for non-blind deblurring tasks.

From this project, we gained a deeper understanding of the stochastic deconvolution framework and its practical implementation for image deblurring. The integration of various regularizers provided insights into their individual and combined effects on image deblurring. Developing a custom dataset and evaluating the algorithm using metrics like PSNR and SSIM emphasized the significance of rigorous testing in understanding trade-offs between quality and computational efficiency.

References

[1] Charbonnier, P., Blanc-Féraud, L., Aubert, G., and Barlaud, M. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of IEEE International Conference on Image Processing (ICIP)* (1994), vol. 2, pp. 168–172.

[2] Gregson, J., Heide, F., Hullin, M. B., Rouf, M., and Heidrich, W. Stochastic Deconvolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2013).

[3] Levin, A., Fergus, R., Durand, F., and Freeman, W. T. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3 (2007).

[4] Perona, P., and Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 7 (1990), 629–639.