EE5175 - **Image Signal Processing**

Rohith R - EP21B030
Jatin V K - EE22B023

April 25, 2025

Rain Detection

1 Problem statement

Rain streaks in images degrade visibility and affect the performance of various computer vision applications, such as autonomous driving, surveillance, and remote sensing. Detecting rain in a single image is a challenging task, especially when relying only on conventional algorithms without deep learning. Develop a robust algorithm to detect rain streaks in a single image using traditional image processing/ computer vision techniques, **without using deep learning networks**. The goal is to detect rain streaks from the rainy images and get the rain mask layer while ensuring minimal false detections in various lighting and scene conditions.

2 Approach

We have examined two different approaches to rain detection, which are outlined below.

2.1 De-Raining Using Layer Priors [1] – ***Rohith R***

We model the rainy image I in terms of the background layer B and the rain streaks layer R using the following formulation:

$$I = B + R - B \odot R,$$

where $I, B, R \in \mathbb{R}^{m \times n}$, and $\odot$ denotes the element-wise (Hadamard) product. The objective is to estimate B and R given only the observed image I, which is an ill-posed problem.

To resolve this, we adopt a Maximum a Posteriori (MAP) framework that maximizes the posterior probability:

$$p(B, R \mid I) \propto p(I \mid B, R) \cdot p(B) \cdot p(R),$$

assuming that B and R are statistically independent.

This leads to the optimization problem:

$$\max_{B,R} p(I \mid B, R) \cdot p(B) \cdot p(R) \quad \Leftrightarrow \quad \min_{B,R} -\ln p(I \mid B, R) - \ln p(B) - \ln p(R),$$

where $-\ln p(B)$ and $-\ln p(R)$ are regularization terms or priors on the background and rain streak layers, respectively.

Assuming Gaussian noise for the likelihood $p(I \mid B, R)$, we approximate:

$$-\ln p(I \mid B, R) \propto \|I - B - R + B \odot R\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The final optimization problem becomes:

$$\min_{B,R} \|I - B - R + B \odot R\|_F^2 + \Psi(B) + \Phi(R),$$

where $\Psi(B)$ and $\Phi(R)$ are the regularization terms for B and R, respectively.

2.1.1 Priors on B

Natural Image Prior (Sparse Gradients): As B is expected to be a piecewise constant image (like natural images), we impose Total Variation (TV) regularization to encourage spatial smoothness while preserving edges. We use isotropic TV since the anisotropic variant promotes independent sparsity along x and y directions, which may be less natural. The regularization term is:

$$\Psi(B) = \lambda_1 \sum_i \sqrt{(\mathbf{D_x} B)_i^2 + (\mathbf{D_y} B)_i^2},$$

where i indexes each pixel, and $\mathbf{D_x}, \mathbf{D_y}$ denote discrete gradient operators along the x and y directions, respectively.

2.1.2 Priors on R

Sparsity Prior: Rain streaks typically affect only a small subset of image pixels, so we encourage sparsity in R using the ℓ_1 norm:

$$\|R\|_1 = \sum_i |R_i|.$$

Sparse Directional Gradient Prior: If rain falls in a specific direction (say, parallel to vector $\mathbf{a}$, making angle θ with the y-axis), then gradients perpendicular to $\mathbf{a}$ are expected to be sparse, Figure 1 demonstrates this concept.

We estimate θ by computing gradient directions at pixels where the gradient magnitude exceeds the 99^{th} percentile threshold and take the mode of these directions.

The final regularization term on R is:

$$\Phi(R) = \lambda_2 \|R\|_1 + \lambda_3 \|\nabla_\theta R\|_1,$$

where $\nabla_\theta R$ denotes directional gradient of R perpendicular to estimated rain direction θ.

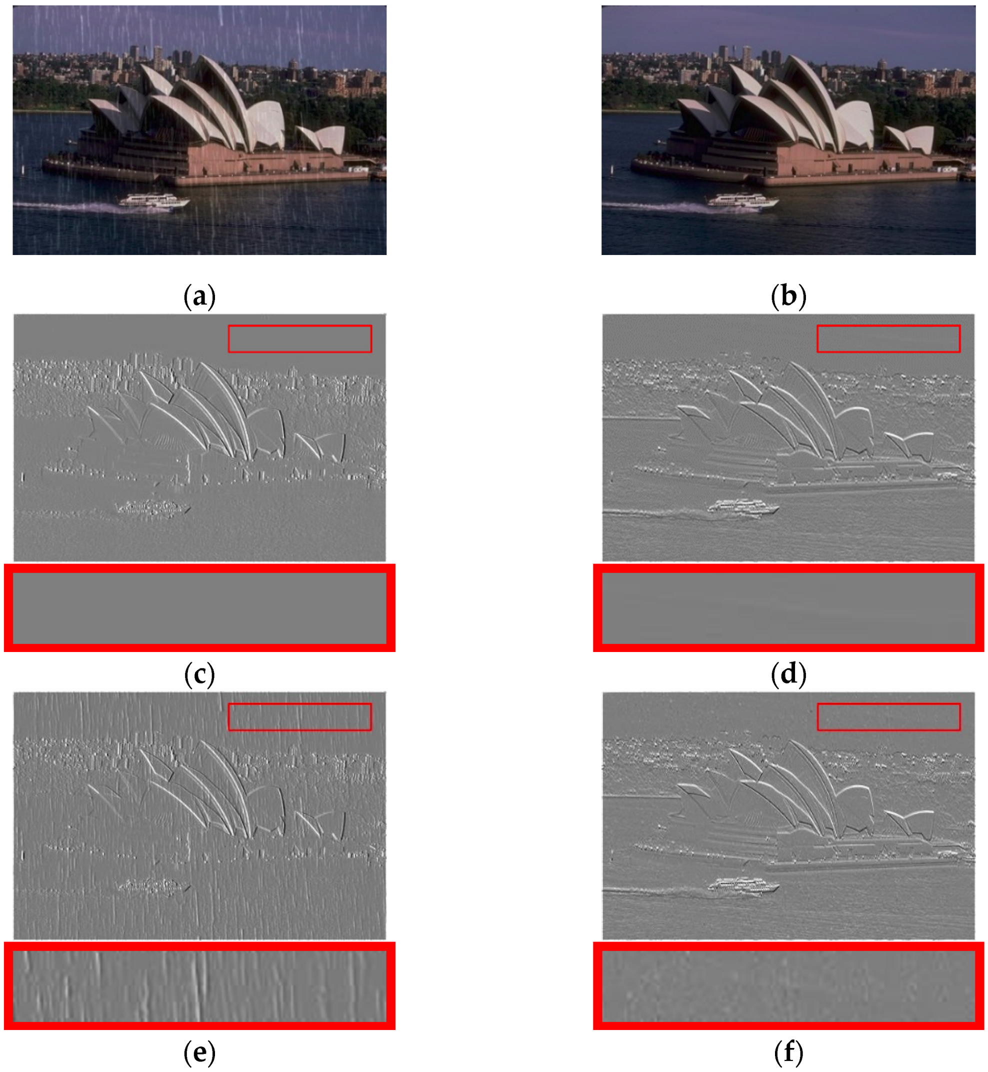


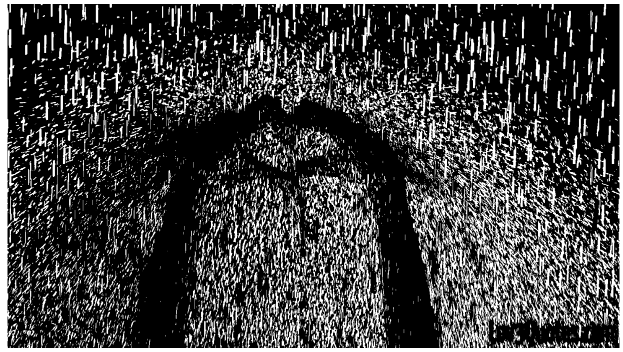
Figure 1: (a,b) Rainy and rainless images. (c,d) Gradient maps of the rainless image along x and y directions. (e,f) Gradient maps of the rainy image along x and y directions, showing directional sparsity in gradients.

2.1.3 Implementation Details

- The optimization problem is solved using the `torch` package, leveraging CUDA for GPU acceleration.
- Regularization weights are set as: $\lambda_1 = 0.02$, $\lambda_2 = 0.005$, and $\lambda_3 = 0.01$.
- The input RGB image is converted to the YUV color space, and the algorithm is applied to the Y (luminance) channel only.
- A binary mask of rain streaks is generated by thresholding R at 0.05 to align with the ground truth annotations in the validation set, even though the unthresholded RR appears more visually appealing on its own.

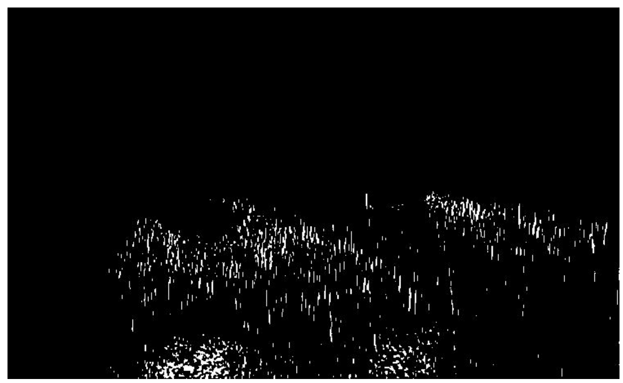
2.1.4 Results

(a) Rainy Image 1



(b) Rain Streaks Layer of Rainy Image 1

(c) Rainy Image 2



(d) Rain Streaks Layer of Rainy Image 2

Figure 2: Few example results using the above described method.

2.2 Morphological Component Analysis [2] – *Jatin V K*

1. **High pass filter**
   We first split the image into high frequency and low frequency part. The rain layer is assumed to have be in high frequency component.

2. **Extracting patches and dictionary learning**
   We extract 500 random patches of size 4 pixels by 4 pixels from the image on which we perform dictionary learning using Orthogonal matching pursuit(OMP).

3. **Feature extraction and Clustering**
   We now use Histogram of oriented gradients to extract features from each atom of the dictionary and then use K means to split the atoms into 2 clusters. The cluster having lesser variance is assumed to be the rain cluster as there is some pattern with respect to the direction of rain streaks.

4. **Rain mask generation using sparse coding**
   We now iterate through the image cells and obtain sparse coding coefficients using OMP. We then check if the highest coefficient corresponds to the rain cluster to mark it as a rain patch.

2.2.1 Parameters

1. Number of patches used in dictionary learning
2. Patch size
3. Number of atoms in dictionary

2.2.2 Current limitations

The assumption of lesser variance for rain components fails sometimes. This leads to an inverted mask(this is sometimes corrected if code is just run again). Another issue arises when uniform sharp objects are present in the image. There is also an issue where the sometimes OMP doesn't converge.

2.2.3 Results

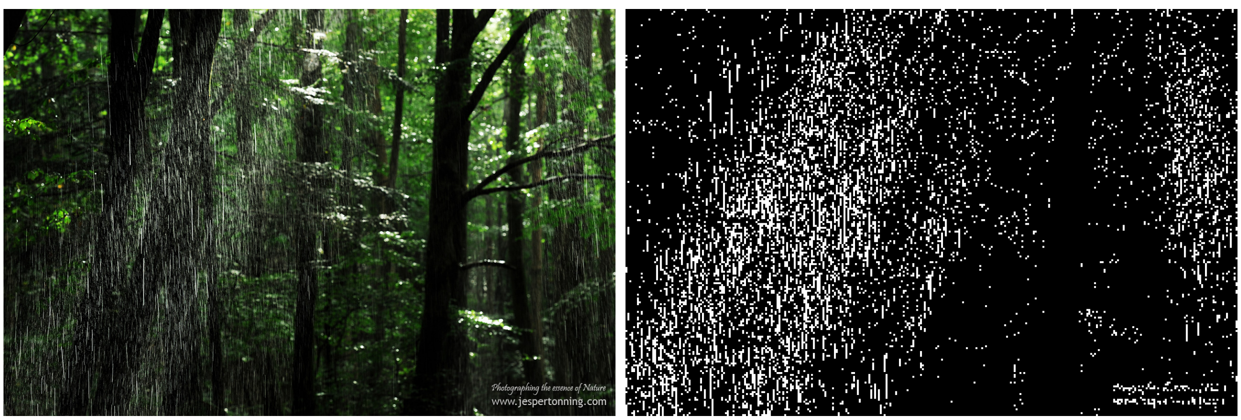


Figure 3: Rain mask of a real image using MCA

References

[1] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Single image rain streak decomposition using layer priors," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3874–3885, 2017.

[2] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, 2012.