Problem formulation
- Our goal: reconstruct corrupted images when likelihood is known
- \( X \) and \( Y \) – clean and corrupted images
- Likelihood \( P(Y|X) \) is known

MAP approach
- Reconstruct \( X \) by maximizing the posterior \( P(X|Y) \)
- Equivalent problem

\[
\min_{X} \log P(Y|X) - \log P(X) \tag{fidelity} \quad \text{regularizer}
\]

Main challenge:
- Choosing a good prior \( P(X) \)

Example: Gaussian denoising with L1 Laplacian regularizer

\[
\min_{X} \frac{1}{2} \|Y - X\|_2^2 + \mu \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{n} |P_{ij}|} \|X_{ij}\|_1
\]

Intuition via unfair denoising
- Sort \( X \) and store the permutation matrix \( P \)
- Apply \( P \) on the noisy \( Y \), filter the \( P_Y \), and reconstruct the image:

\[
\hat{X} = P^{-1}F\{P_Y\}
\]

Intuition via unfair denoising (continued)

- Sorting (finding the shortest possible route) is replaced with TSP (Travelling Salesman Problem)

\[
\min_{P} \sum_{i=2}^{N} \|x_i^P - x_{i-1}^P\|_2 \equiv \min_{P} \sum_{i=2}^{N} \|x_i - x_{i-1}\|_2
\]

- Clean pixel \( x_i \) is represented by \( \sqrt{N} \times \sqrt{N} \) noisy patch \( z_i \)
- Similar patches \( \rightarrow \) close central pixels
- Patches \( z_i \) are points in \( \mathbb{R}^n \)
- TSP seeks the shortest path that visits every point once

Previous work – denoising [1]

- Noisy image, \( \sigma = 75 \)
- Output, PSNR = 24.82 dB

Outline of the algorithm
- Extract all possible patches with overlaps from the \( Y \), order them to form the shortest possible path by solving TSP problem, and find the permutation \( P \)
- Apply \( P \) on the \( Y \), filter the \( P_Y \) and reconstruct the image:

\[
\hat{X} = P^{-1}F\{P_Y\}
\]

* This result is obtained with: (i) cycle-sieving, (ii) sub-image averaging, (ii) learned filters, and (iv) two iterations.

Patch-ordering based regularizer

\[
\min_{X} \log P(Y|X) + \mu \sum_{i=1}^{n} \sum_{j=1}^{n} |MLPS_{ij}| \|X_{ij}\|_1
\]

- Main idea – apply Laplacian on the permuted image \( P \) – permutation matrix, \( L \) – 1D Laplacian

- Enhancements:
  - \( S_I \) – sum over \( S_I \) matrices accumulates penalties over all pixels within the patches
  - \( M = \text{diag}(m_k) \) – diagonal matrix that contains weights
  - \( m_k = \min \{ \frac{1}{L_{k-1}}, m_{\max} \} \)
  - \( \beta_k \frac{1}{2} \|x_k - \frac{1}{2} \|_2 \|x_{k+1} - 2x_k + x_{k-1}\|_2 \)
  - \( \gamma_k = \{ Y_{\text{edge}} \sum_{i=1}^{3} |P_{i,j}| \} > \text{thr} \) else

- We initialize our algorithm with an output of a recent method which handles the problem

Results

Benefits of the subimage accumulation
- Increasing number of orderings from 1 to \( n \)
- Introducing an implicit spatial prior

Severe Poisson denoising

Patch-Ordering Concept
- Sorting (finding the shortest possible route) is replaced with TSP (Travelling Salesman Problem)

Constructing permutation
- \( P \) found by too good TSP solution influenced by artifacts and cause artifact magnification

Randomized Nearest Neighbor
- Start from a random patch
- Randomly choose one of the two unvisited nearest neighbors
- Stop after visiting the last patch

References