Patch-based Near-Optimal Image Denoising

Authors: Priyam Chatterjee and Peyman Milanfar

Presenter: Qiang Li

2011-7-7

This paper has been submitted to TIP2011.
Outline

- Contents in brief
- PLOW algorithm diagram
- Geometric clustering
- Motivation
- Derivation and analysis of the estimator
- Parameter estimation for denoising
- PLOW denoising algorithm procedure
- Issues
The framework uses both geometrically as well as photometrically similar patches to estimate the different filter parameters. The authors described how these parameters can be accurately estimated directly from the input noisy image. They analyzed the statistical foundation of the so-called near-optimal algorithm and reported surprising experimental results.
PLOW algorithm diagram

Noisy Image

Photometrically similar patches

Clustering

PLOW

Aggregate estimates

Estimate moments

Denoised Image
Geometric clustering

- Geometric similarity is exploited by clustering similar geometric structures such as edge, corner, flat regions, etc.
- It is different with the photometric similarity which is related to similar image intensities.

Fig. 2. Clustering of a simple image based on geometric similarity. Note how pixels in any particular cluster can have quite different intensities but similar geometric structure (edge, corner, flat regions, etc.)
Motivation (1)

- The performance bounds for the problem of image denoising is derived in [Ref]
- By using a Bayesian Cramer-Rao bound analysis.
- Define an observation model (considering patch intensities) as follows

\[ y_i = z_i + \eta_i \]

Motivation (2)

- MSE for a given image patch is bounded by

\[ E \left[ \| \mathbf{z}_i - \hat{\mathbf{z}}_i \|^2 \right] \geq \text{Tr} \left[ (\mathbf{J}_i + \mathbf{C}_z^{-1})^{-1} \right] \]  \hspace{1cm} (3)

- The first term is the Fisher information matrix (FIM). When AWGN is considered, \( \mathbf{J}_i = N_i \frac{1}{\sigma^2} \) \( \mathbf{N}_i \) is the patch redundancy measured as the number of patches photometrically similar to \( \mathbf{z}_i \)

- The second term is the patch covariance matrix for a particular cluster.

- For details, see Appendix I
The photometrically similar patches are defined as follows.

For noise-free case

\[ \| \varepsilon_{ij} \|^2 \leq \gamma^2 \quad \text{where} \quad \varepsilon_{ij} = z_j - z_i. \]  

(5)

For noisy case

\[ \| \tilde{\varepsilon}_{ij} \|^2 \leq \gamma_n^2 = \gamma^2 + 2\sigma^2 n \quad \text{where} \quad \tilde{\varepsilon}_{ij} = y_j - y_i. \]  

(6)

where, the threshold depends on the number of pixels (n) in each patch.
Motivation (4)

- The Wiener filter is, in fact, the LMMSE estimator that achieves the lower bound.
- Thus, a patch-based Wiener filter, where the parameters are estimated accurately, can lead to near-optimal denoising.
- This forms the basis of this paper.
Derivation and analysis of the estimator (1)

- Basic assumption:
  - Image patches that are geometrically similar (i.e. a particular cluster) were considered to be sampled from the same (unknown) probability density function (pdf) \( p(z) \).
- When the noise is AWGN, the LMMSE estimate is [Ref]

\[
\hat{\mathbf{z}}_i = \mathbf{z} + \mathbf{C}_z \mathbf{C}_y^{-1} (\mathbf{y}_i - \mathbf{z})
\]

\[
\mathbf{C}_y = \mathbf{C}_z + \sigma^2 \mathbf{I}
\]

Derivation and analysis of the estimator (2)

- Data model

\[
y_i = A_i y_i + \tilde{\epsilon}_i = A_i (z_i + \eta_i) + (\varepsilon_i + \eta_i - A_i \eta_i)
\]

\[
= A_i z_i + \varepsilon_i + \eta_i,
\]

Note that \( A_i \) is formed by vertically stacking \( n \times n \) identity matrix.
Thus, the corresponding LMMSE estimator is

\[
\hat{z}_i = \bar{z} + C_z A_i^T \left( A C_z A_i^T + C_{\xi_i} \right)^{-1} \left( y_i - A_i \bar{z} \right)
\]

\[
= \bar{z} + \left( C_z^{-1} + A_i^T C_{\xi_i}^{-1} A_i \right)^{-1} A_i^T C_{\xi_i}^{-1} \left( y_i - A_i \bar{z} \right)
\]

\[
C_{\xi_i} = C_{\varepsilon_i} + C_{\eta_i} = \begin{bmatrix} \cdots & 0 \\ \delta_{ij}^2 I & \cdots \\ 0 & \cdots \end{bmatrix}
\]

\[
\delta_{ij}^2 = \frac{1}{n} E \left[ \| y_i - y_j \|^2 \right] - \sigma^2.
\]

Derivation and analysis of the estimator (4)

- Define \( w_{ij} = \delta_{ij}^{-2} \)

- Then, we get the weighted expression

\[
\hat{z}_i = \bar{z} + \left( C_z^{-1} + \sum_{j=1}^{N_i} w_{ij} I \right)^{-1}\sum_{j=1}^{N_i} w_{ij} (y_j - \bar{z})
\]

\[
= \bar{z} + \left( \frac{C_z^{-1}}{\sum_{j=1}^{N_i} w_{ij}} + I \right)^{-1}\sum_{j=1}^{N_i} \frac{w_{ij}}{\sum_{j=1}^{N_i} w_{ij}} (y_j - \bar{z})
\]

- The estimator achieves the bound formulation of (3) when
Derivation and analysis of the estimator (5)

- The error covariance matrix for such an estimator is approximately

\[
C_e \approx \left( C_z^{-1} + \sum_{j=1}^{N_i} w_{ij} I \right)^{-1}
\]

(16)

- Cz may be rank deficient or ill-conditioned, a practical form of the LMMSE estimator is

\[
\hat{z}_i = \left[ \sum_{j=1}^{N_i} \frac{w_{ij} y_j}{\sum_{j=1}^{N_i} w_{ij}} \right] + \left[ \sum_{j=1}^{N_i} \frac{w_{ij}}{\sum_{j=1}^{N_i} w_{ij}} \left( \sum_{j=1}^{N_i} w_{ij} C_z + I \right)^{-1} (z - y_j) \right]
\]

(17)

- For full derivation, see Appendix III
Parameter estimation for denoising (1)

- Mean estimation
- In the observation model, since noise patches are assumed to be zero mean iid
- The mean of the underlying noise-free image is approximated by the expectation of the noisy patches within each cluster as

\[
\hat{z} = E[y_i \in \Omega_k] \approx \frac{1}{M_k} \sum_{y_i \in \Omega_k} y_i
\]  

(18)
Parameter estimation for denoising (2)

- The covariance matrix is approximated as

\[
\hat{C}_z = \left[ \hat{C}_y - \sigma^2 I \right]_+ \tag{19}
\]

Note that \([X]_+\) means replace its negative eigenvalues (or a very small positive value) by zero [Ref].

Parameter estimation for denoising (3)

- Noise standard deviation estimation
- Gradient-based estimator [Ref]

\[
\hat{\sigma} = 1.4826 \ \text{median} \left( |\nabla Y - \text{median} (\nabla Y)| \right)
\]  
(20)

\[
\nabla Y = \frac{1}{\sqrt{6}} \ \text{vec} \left( Y \ast \begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix} \right)
\]  
(21)

Parameter estimation for denoising (4)

- Weight estimation
- Derivation for noise-free case

Original formation

\[
\delta_{ij}^2 = \frac{1}{n} E \left[ \| z_i - z_j \|^2 \right] + \sigma^2
\]

\[
\Rightarrow \delta_{ij}^2 = w_{ij}^{-1} = \sigma^2 \exp \left\{ \frac{\| z_i - z_j \|^2}{h^2} \right\}
\]

Define \( u \)

\[
e^u = 1 + u + O(u^2) \approx 1 + u \quad \text{[since} \ u < 1]\]

\[
u = \frac{\| z_i - z_j \|^2}{h^2} = 1 + \frac{\| z_i - z_j \|^2}{\sigma^2 n}
\]

\[
\Rightarrow \delta_{ij}^2 = \sigma^2 e^u \approx \sigma^2 + \frac{1}{n} \| z_i - z_j \|^2.
\]
Parameter estimation for denoising (5)

- However, in practice only the noisy patches are observed.
- Note that the distance between noisy patches can be much higher than those between the underlying noise-free patches.
- As a result, a larger smoothing term is needed for denoising.
- In this paper, the authors set $h^2 = 1.75\sigma^2 n$
Parameter estimation for denoising (6)

- Aggregation scheme
- Naive averaging
  - Over-smoothing problem
- LMMSE scheme takes into account the relative confidence

\[ \hat{z}_i = \sum_{r=1}^{R} \frac{v_{rl}^{-1} \hat{z}_{rl}}{\sum_r v_{rl}^{-1}} \]  \hspace{1cm} (26)

The weighted least squares estimate obtained using the error variance from multiple estimates.
Parameter estimation for denoising (7)

- Illustration of aggregation

Figure 5.3: An illustration of how a pixel is estimated multiple times due to overlapping patches. Here we show 3 such overlapping patches. In each estimated patch $\hat{z}_r$ (here $r = 1, 2, 3$), the same pixel is estimated as its $l_r$-th pixel which we denote as $\hat{z}_{r,l}$. These estimates are finally combined to form the final estimate $\hat{z}_i$. 
Parameter estimation for denoising (8)

- Derivation of the Eq.(26)
- Recall the error covariance

\[ C_e \approx \left( C_z^{-1} + \sum_{j=1}^{N_i} w_{ij} I \right)^{-1} \]

\( \hat{z}_{rl} \) is the denoised estimate for the l-th pixel in the r-th patch.

The variance \( \nu_{rl} \) of the error associated with the l-th pixel estimate is given by the l-th diagonal element of \( C_e \).
Parameter estimation for denoising (9)

- Concatenating the multiple estimates in a vector, we get the data model

\[ \hat{z}_{ir} = 1 \zeta_i + \tau_{ir} \]

\( \tau \) is the error vector assumed to be zero mean with covariance \( C_\tau = \text{diag}[...v_{rl}...] \)

- The LMMSE estimate for the i-th pixel of the image is then

\[
\hat{z}_i = \left( \sigma^{-2}_z + 1^T C_\tau^{-1} 1 \right)^{-1} 1^T C_\tau^{-1} \hat{z}_{ir} \\
= \sum_{r=1}^{R} v_{rl}^{-1} \hat{z}_{rl} / \sum_r v_{rl}^{-1} + \sigma^{-2}_z, \]

where

\[
\text{diag} \text{ and } v_{rl}^{-1} \text{ are the diagonal elements of } C_\tau \text{ and } C_\tau^{-1} \text{ respectively.} \]
Parameter estimation for denoising (10)

- The estimated covariance $\hat{C}_z$ does not provide a pixel-wise variance estimate $\sigma_z^2$.
- Because any given pixel $z_i$ can lie in different locations in different patches.
- Moreover, the overlapping patches may lie in different clusters.
- Consider all possible $z_i$ values satisfy the discrete uniform distribution, the variance is

$$\sigma_z^2 = \frac{(256^2 - 1)}{12} \Rightarrow \sigma_z^{-2} = 0$$
PLOW denoising algorithm procedure

- First identify geometrically similar patches (geometric clustering);
- Estimate cluster moments ($\mathbf{z}$ and $\mathbf{Cz}$);
- Identify the photometrically similar patches and Calculating the weights;
- Aggregate multiple pixel estimates (overlapping denoised patches).
Algorithm 1: PLOW denoising

Input: Noisy image: $\mathbf{Y}$

Output: Denoised image: $\hat{\mathbf{Z}}$

1. Set parameters: patch size $n = 11 \times 11$, number of clusters $K = 15$;
2. Estimate noise standard deviation $\hat{\sigma}$ (Eq. 20);
3. Set parameter: $h^2 = 1.75\hat{\sigma}^2 n$;
4. $\mathbf{Y}^0 \leftarrow$ Pre-filter image to obtain pilot estimate;
5. $\{\mathbf{y}_i, \mathbf{y}_i^0\} \leftarrow$ extract overlapping patches of size $n$ from $\mathbf{Y}$ & $\mathbf{Y}^0$;
6. $\mathbf{L} \leftarrow$ compute LARK features for each $\mathbf{y}_i^0$;
7. $\Omega_k \leftarrow$ geometric clustering with $\text{K-Means}(\mathbf{L}, K)$;
foreach Cluster $\Omega_k$ do
  Estimate mean patch $\hat{z}$ from $y_i^0 \in \Omega_k$ (Eq. 18);
  Estimate cluster covariance $\hat{C}_z$ from $y_i^0 \in \Omega_k$ (Eq. 19);
  foreach Patch $y_i^0 \in \Omega_k$ do
    $y_j^0 \leftarrow$ identify photometrically similar patches (Eq. 6);
    $w_{ij} \leftarrow$ compute weights for all $y_j^0$ (Eq. 22);
    $\hat{z}_i \leftarrow$ estimate denoised patch using $y_j$ (Eq. 17);
    $C_{e_i} \leftarrow$ calculate estimate error covariance (Eq. 16);
  end
end

$\hat{Z} \leftarrow$ aggregate multiple estimates from all $\{\hat{z}_i\}$ and $\{C_{e_i}\}$ (Eq. 26);
Issues

- Patch independence assumption. Information shared among overlapping noisy patches are not exploited.
- The moment estimation step is dependent on the ability of the clustering step to classify structurally similar patches.
- Further, even with accurate clustering, noise causes the eigenvalues of the sample covariance matrix $C_y$ to be shifted unequally.
- The weight calculation process of Eq.(22) is quite sensitive to noise.
Appendix I

- Affine bias model
- OB-CRLB (Optimal Bias Bayesian Cramer-Rao Lower Bound)
Appendix II

- Derivation of expression for \( \mathbf{C}_{\xi_i} \)

\[
\mathbf{y}_i = \mathbf{A}_i \mathbf{z}_i + \mathbf{\varepsilon}_i + \mathbf{\eta}_i
\]

\[
\mathbf{C}_{\xi_i} = \mathbf{C}_{\eta_i} + \mathbf{C}_{\varepsilon_i}
\]

\[
\mathbf{C}_{\eta_i} = \sigma^2 \mathbf{I}_q
\]

\[
q = n \mathbf{N}_i
\]

\[
\mathbf{\varepsilon}_{ij} = \mathbf{z}_j - \mathbf{z}_i = (\mathbf{y}_j - \mathbf{y}_i) - (\mathbf{\eta}_j - \mathbf{\eta}_i)
\]

\[
\Rightarrow E[\|\mathbf{\varepsilon}_{ij}\|^2]
\]

\[
= E[\|\mathbf{y}_j - \mathbf{y}_i\|^2] - E[\|\mathbf{\eta}_j - \mathbf{\eta}_i\|^2]
\]

\[
= E[\|\mathbf{y}_j - \mathbf{y}_i\|^2] + 2\sigma^2 n - 2E[(\mathbf{y}_j - \mathbf{y}_i)^T(\mathbf{\eta}_j - \mathbf{\eta}_i)]
\]

\[
= E[\|\mathbf{y}_j - \mathbf{y}_i\|^2] + 2\sigma^2 n - 2 (E[(\mathbf{z}_j - \mathbf{z}_i)^T(\mathbf{\eta}_j - \mathbf{\eta}_i)] + E[(\mathbf{\eta}_j - \mathbf{\eta}_i)^T(\mathbf{\eta}_j - \mathbf{\eta}_i)])
\]

\[
= E[\|\mathbf{y}_j - \mathbf{y}_i\|^2] - 2\sigma^2 n,
\]
\[ C_{\varepsilon_{ij}} = \left( \frac{1}{n} E[\|y_j - y_i\|^2] - 2\sigma^2 \right) I \]

\[ C_{\varepsilon_i} = \begin{bmatrix} \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \varepsilon_{ij} & 0 \end{bmatrix} \]

\[ C_{\zeta_i} = C_{\varepsilon_i} + C_{\eta_i} = \begin{bmatrix} \cdots & 0 \\ \vdots & \delta_{ij}^2 \cdot I \\ 0 & \cdots \end{bmatrix} \]

\[ \delta_{ij}^2 = \frac{1}{n} E[\varepsilon_{ij}^2] + \sigma^2 = \frac{1}{n} E[\|y_j - y_i\|^2] - \sigma^2 \]
Appendix III

- Derivation of Redundancy Exploiting Wiener Filter
\[
\hat{z}_i = \bar{z} + \left( C_z^{-1} + A_i^T C_{\xi_i}^{-1} A_i \right)^{-1} A_i^T C_{\xi_i}^{-1} (y_i - A_i\bar{z})
\]

\[
A_i^T C_{\xi_i}^{-1} (y_i - A_i\bar{z}) = [I \ldots I] \begin{bmatrix}
\cdots & 0 \\
[\delta_{i_j}^{-2} I] & \cdots \\
0 & \cdots
\end{bmatrix}
\begin{bmatrix}
y_1 \\
\vdots \\
y_{N_i}
\end{bmatrix} - \begin{bmatrix}
\bar{z} \\
\vdots \\
\bar{z}
\end{bmatrix}
\]

\[
= \left[ \cdots \delta_{i_j}^{-2} I \cdots \right] (y_j - \bar{z}) = \sum_{j=1}^{N_i} \delta_{i_j}^{-2} (y_j - \bar{z}),
\]

\[
A_i^T C_{\xi_i}^{-1} A_i = \left[ \cdots \delta_{i_j}^{-2} I \cdots \right] = \sum_{j=1}^{N_i} \delta_{i_j}^{-2} I
\]
\( \hat{Z}_i = \bar{Z} + \left( C_Z^{-1} + \sum_{j=1}^{N_i} \delta_{ij}^{-2} I \right)^{-1} \sum_{j=1}^{N_i} \delta_{ij}^{-2} (y_j - \bar{Z}) \)

\( = \bar{Z} + \left( \frac{C_Z^{-1}}{\sum_{j=1}^{N_i} \delta_{ij}^{-2}} + I \right)^{-1} \sum_{j=1}^{N_i} \frac{\delta_{ij}^{-2}}{\sum_{j=1}^{N_i} \delta_{ij}^{-2}} (y_j - \bar{Z}) \)

\[(A + UBV)^{-1} = A^{-1} - A^{-1}U(I + BVA^{-1}U)^{-1}BVA^{-1}\]

\( \left( \frac{C_Z^{-1}}{\sum_{j} \delta_{ij}^{-2}} + I \right)^{-1} = I - \left( \sum_{j} \delta_{ij}^{-2} C_Z + I \right)^{-1} \)
\[
\hat{z}_i = \bar{z} + \left[ I - \left( \sum_j \frac{\delta_{ij}^{-2} C_z + I}{\delta_{ij}^{-2}} \right) \right] \sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} (y_j - \bar{z})
\]

\[
= \bar{z} + \sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} (y_j - \bar{z}) - \left( \sum_j \frac{\delta_{ij}^{-2} C_z + I}{\delta_{ij}^{-2}} \right)^{-1} \sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} (y_j - \bar{z})
\]

\[
= \sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} \left[ y_j - \left( \sum_j \frac{\delta_{ij}^{-2} C_z + I}{\delta_{ij}^{-2}} \right)^{-1} (y_j - \bar{z}) \right]
\]

\[
- \left[ \sum_j \frac{\delta_{ij}^{-2} y_j}{\sum_j \delta_{ij}^{-2}} \right] - \left[ \sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} \left( \sum_j \frac{\delta_{ij}^{-2} C_z + I}{\delta_{ij}^{-2}} \right)^{-1} (y_j - \bar{z}) \right]
\]