Adaptive Image Denoising by EM-Adaptation

Enming Luo
Feb. 22, 2016

Committee:
Truong Nguyen, Chair
Ery Arias-Castro
Joseph Ford
Bhaskar Rao
Zhuowen Tu
Is Denoising Dead?

Priyam Chatterjee, Student Member, IEEE, and Peyman Milanfar, Fellow, IEEE

Abstract—Image denoising has been a well studied problem in the field of image processing. Yet researchers continue to focus attention on it to better the current state-of-the-art. Recently proposed methods take different approaches to the problem and yet literature on such performance limits exists for some of the more complex image processing problems such as image registration [7], [8] and super-resolution [9]–[12]. Performance limits to object or feature recovery in images in the presence of point-
A Typical Imaging Pipeline

Output Image → Gamma Correction → Color Correction → Noise Suppression → Demosaicking → White Balance (Channel Equalization)
Sources of Noise

(1) Shot Noise
- Result of random photon arrival
- Poisson distributed
- Serious in low-light condition
- Not so bad under good light

(2) Electronic Noise
- Instability of voltage/current
- Temperature fluctuation
- Analog to digital error
- Gaussian distributed

Simplified diagram illustrating the two sources of noise
Noise!

[Image of noise examples at different ISO settings]

My work: Gaussian Noise!

[Link: http://photo.net/equipment/canon/eos6D/review]
Adaptive Image Denoising for my PhD

Single image denoising
- EUSIPCO’12

Multi-view denoising
- ICIP’13

Guided image denoising
- GlobalSIP’15
- TIP’16

Targeted image denoising
- ICASSP’14
- TIP’15

work after PhD qualifying exam

External and Internal
Image Denoising

Consider an additive i.i.d. Gaussian noise model:

\[ y = x + \varepsilon \in \mathbb{R}^n \]

where \( \varepsilon \sim \mathcal{N}(0, \sigma^2 I) \)

Our goal is to estimate \( x \) from \( y \)

Our Approach: Maximum-a-Posteriori

\[ \hat{x} = \underset{x}{\text{argmax}} \ f(x|y) \]
Since the noise i.i.d. is Gaussian, the conditional distribution is

\[ f(y \mid x) \propto \exp\left(-\frac{\|y - x\|^2}{2\sigma^2}\right) \]

Therefore, the MAP is

\[
\arg\max_x f(x \mid y) = \arg\max_x f(y \mid x) f(x) \\
= \arg\min_x \{- \log f(y \mid x) - \log f(x)\} \\
= \arg\min_x \left\{ \frac{1}{2\sigma^2} \|y - x\|^2 - \log f(x) \right\}
\]
Image Priors

• Markov Random Field (80s)
• Gradients (80s)
• Total Variation (90s)
• X-lets (wavelet, contourlet, curvelet, ..., 90s)
• $L^p$ norm (00s)
• Dictionary (KSVD, 00s)
• Example (00s)
• Non-local (BM3D, nonlocal means, 2005, 2007)
• Shotgun! (2011)
• Graph Laplacian (2012)
Patch-based Priors

What is a patch?

A patch is a small block of pixels in an image

Why patch?

\( x \) is extremely high-dimensional and thus \( f(x) \) is intractable.

What is patch-based prior?

\[
\log f(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \log f(P_i x)
\]

\( P_i x \) extracts the \( i \)th patch from \( x \).

Define \( p_i = P_i x \)
Training a Patch-based Prior

Typically, we train a patch-based prior from a large collection of images

\[ f(p) = \sum_{j=1}^{K} \pi_j N(p | \mu_j, \Sigma_j) \]

E.g., Gaussian mixture:

\[ \log f(x) \]

EM Algorithm
Good Training Set

noisy elephant
database: images of clean elephants

noisy Bill Clinton
database: images of clean Bill Clintons
How good?

Example: Text Image

Taking photographs with a poor camera and focus cause motion blurred exposure time miss technical passivity for true. Each image

noisy image

was motivated by modeling at a circuit level, it disagrees with [19]. In the latter work, however, the noise measurements are made not on the image sensors themselves, but rather on the output images from a digital camera that has already undergone less noise. Where as the observed image, is the ideal noiseless image with variance $\sigma^2$, and $\lambda_1$ and $\lambda_2$ are constants, and $\gamma$ represents white Gaussian noise with variance 1. Through experimentation, we verified that (1) was a good noise model for Agilent Technologies camera evaluating a function board HDPC 2000 equipped with 500 K pixel CMOS. Additional tests of sensor and capturing raw sensor data. Left graph of Fig. 1 shows an example, an adverse relationship between the noise standard deviation plotted against the sensor value. As predicted in (1), the graph clearly implies an affine line. While in the second line, the white Gaussian noise in (2), removes the signal-dependent noise. See Appendix I for a review of general noise models. This approach is also important to note that the noise patches in the vicinity of $x_0$. This approach is cited than 50, (relative in [26])

than 50, (relative in [26]) and from the noisy environment, correctly cor.

procedure determine the determine the

means algorithm means algorithm

experimental results

clean image

noisy image

BM3D

[ Luo-Chan-Nguyen, 15 ]

(single image method) (use targeted training)
Challenge:

(1) Finding good examples is HARD.
(2) Finding a lot of good examples is EVEN HARDER.

My work: Can priors be learned adaptively?

Generic database
[Zoran-Weiss ‘11]
2 million 8x8 image patches

Image of interest
update

Gaussian mixture model
\[ f(p) = \sum_{j=1}^{K} \pi_j \mathcal{N}(p | \mu_j, \Sigma_j) \]
Our Proposed Idea
This is what we are going to do:

$$\arg\max_x f(x|y) = \arg\min_x \left\{ \frac{1}{2\sigma^2} \| y - x \|^2 - \frac{1}{n} \sum_{i=1}^{n} \log f(P_i x) \right\}$$

**Question 1 : How to SOLVE this optimization problem?**

(If we cannot solve this problem, then there is no point of continuing.)

**Question 2 : How to ADAPTIVELY learn a prior?**

Generic prior (from an arbitrary database)

\[\downarrow\]

Specific prior (match the image of interest)
This is what we are going to do:

\[
\underset{x}{\text{argmax}} \ f(x|y) = \underset{x}{\text{argmin}} \ \left\{ \frac{1}{2\sigma^2} \|y - x\|^2 - \frac{1}{n} \sum_{i=1}^{n} \log f(P_i x) \right\}
\]

**Question 1 : How to SOLVE this optimization problem?**

(If we cannot solve this problem, then there is no point of continuing.)

**Question 2 : How to ADAPTIVELY learn a prior?**

Generic prior (from an arbitrary database)

\[\downarrow\]

Specific prior (match the image of interest)
Half Quadratic Splitting

General Principle [Geman-Yang, T-IP, 1995]

\[
\arg\min_x \left\{ \frac{1}{2\sigma^2} \|y - x\|^2 - \frac{1}{n} \sum_{i=1}^n \log f(P_i x) \right\}
\]

\[\Rightarrow \arg\min_{x, \{v_i\}_1^n} \left\{ \frac{1}{2\sigma^2} \|y - x\|^2 + \sum_{i=1}^n \left( \frac{\beta}{2} \|P_i x - v_i\|^2 \right) - \frac{1}{n} \sum_{i=1}^n \log f(v_i) \right\}
\]

The Algorithm:

\[
v_i^{(m+1)} = \arg\min_{v_i} \left\{ -\log f(v_i) + \frac{\beta^{(m)}}{2} \|P_i x^{(m)} - v_i\|^2 \right\} \quad (1)
\]

\[
x^{(m+1)} = \arg\min_x \left\{ \frac{n}{2\sigma^2} \|y - x\|^2 + \frac{\beta^{(m)}}{2} \sum_{i=1}^n \|P_i x - v_i^{(m+1)}\|^2 \right\} \quad (2)
\]

penalizer: it increases as \( m \) increases
\[ \mathbf{v}_i^{(m+1)} = \arg\min_{\mathbf{v}_i} \left\{ -\log f(\mathbf{v}_i) + \frac{\beta(m)}{2} \|P_i \mathbf{x}^{(m)} - \mathbf{v}_i\|^2 \right\} \]  \hspace{1cm} (1) \\
\[ \mathbf{x}^{(m+1)} = \arg\min_{\mathbf{x}} \left\{ \frac{n}{2\sigma^2} \|\mathbf{y} - \mathbf{x}\|^2 + \frac{\beta(m)}{2} \sum_{i=1}^{n} \|P_i \mathbf{x} - \mathbf{v}_i^{(m+1)}\|^2 \right\} \]  \hspace{1cm} (2)

**Solution to Problem (1):** depends on \( f(\mathbf{v}_i) \)

**Example**  Gaussian Mixture Model

[Zoran-Weiss ‘11]

If \( f(\mathbf{v}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{v}_i | \mu_k, \Sigma_k) \), assuming \( \mathbf{v}_i \) is dominated by its mode, then the solution to (1) is:

\[ \mathbf{v}_i^{(m+1)} = \left( \beta(m) \Sigma_{k_i^*} + I \right)^{-1} \left( \mu_{k_i^*} + \beta(m) \Sigma_{k_i^*} P_i \mathbf{x}^{(m)} \right) \]

where \( k_i^* \overset{\text{def}}{=} \arg\max_k \pi_k \mathcal{N}(\mathbf{v}_i | \mu_k, \Sigma_k) \).
The solution to (2) is

\[
\mathbf{x}^{(m+1)} = \arg\min_{\mathbf{x}} \left\{ \frac{n}{2\sigma^2} \| \mathbf{y} - \mathbf{x} \|^2 + \frac{\beta^{(m)}}{2} \sum_{i=1}^{n} \| \mathbf{P}_i \mathbf{x} - \mathbf{v}_i^{(m+1)} \|^2 \right\}
\]
Question 1: How to SOLVE this optimization problem?

\[ \arg\max_x f(x|y) = \arg\min_x \left\{ \frac{1}{2\sigma^2} \| y - x \|^2 - \frac{1}{n} \sum_{i=1}^{n} \log f(P_i x) \right\} \]

For Gaussian Mixture:

\[
\boldsymbol{v}_i^{(m+1)} = \left( \beta^{(m)} \Sigma_{k_i}^* + I \right)^{-1} \left( \mu_{k_i}^* + \beta^{(m)} \Sigma_{k_i}^* P_i x^{(m)} \right)
\]

\[
x^{(m+1)} = \left( n\sigma^{-2} I + \beta^{(m)} \sum_{i=1}^{n} P_i^T P_i \right)^{-1} \left( n\sigma^{-2} y + \beta^{(m)} \sum_{i=1}^{n} P_i^T \boldsymbol{v}_i^{(m+1)} \right)
\]

The prior affects the final estimate much!
This is what we are going to do:

$$\arg\max_x f(x|y) = \arg\min_x \left\{ \frac{1}{2\sigma^2} \|y - x\|^2 - \frac{1}{n} \sum_{i=1}^{n} \log f(P_i x) \right\}$$

Question 1: How to SOLVE this optimization problem?
(If we cannot solve this problem, then there is no point of continuing.)

Question 2: How to ADAPTIVELY learn a prior?
- Generic prior (from an arbitrary database)
- Specific prior (match the image of interest)
Goal: Update the generic GMM from a targeted image

Generic database [Zoran-Weiss ‘11]
2 million 8x8 image patches

Image of interest

update

Gaussian mixture model

\[ f(p) = \sum_{j=1}^{K} \pi_j \mathcal{N}(p \mid \mu_j, \Sigma_j) \]

EM Adaptation
Imagine that:

(a) Original generic database (A LOT of samples)
(b) Ideal targeted database (A LOT of samples)
(c) In reality, samples from targeted database are FEW!!!
(a) GMM 1: 400 points

(b) GMM 2: 400 points

(c) GMM2: 20 points

(d) Adapted: 20 points
\( \rho = 100 \)

(e) Adapted: 20 points
\( \rho = 10 \)

(f) Adapted: 20 points
\( \rho = 1 \)

From (d) to (f): The extent of adaptation increases.
**Algorithm 1** EM Algorithm for GMM

Input: sample patches $p_1, \ldots, p_n$, number of clusters $K$.
Output: Parameters $\Theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$.
Initialize $\pi_k^{(0)}, \mu_k^{(0)}$ and $\Sigma_k^{(0)}$ for $k = 1, \ldots, K$.

while not converge do

E-step: Compute, for $k = 1, \ldots, K$ and $i = 1, \ldots, n$

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(p_i | \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(p_i | \mu_l, \Sigma_l)}, \quad n_k = \sum_{i=1}^n \gamma_{ki}. \quad \gamma_{ki} = \frac{n_k \mathcal{N}(\tilde{p}_i | \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(\tilde{p}_i | \mu_l, \Sigma_l)}$$

M-step: Compute, for $k = 1, \ldots, K$

$$\pi_k = \frac{n_k}{n}, \quad \mu_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} p_i,$$

$$\Sigma_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (p_i - \mu_k)(p_i - \mu_k)^T.$$

Update Counter: $m \leftarrow m + 1$

end while

---

**Algorithm 1** EM-adaptation Algorithm

Input: Old parameters $\Theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$, new sample patches $\tilde{p}_1, \ldots, \tilde{p}_n$.
Output: Adapted parameters $\tilde{\Theta} = (\tilde{\pi}_k, \tilde{\mu}_k, \tilde{\Sigma}_k)_{k=1}^K$.

E-step: Compute, for $k = 1, \ldots, K$ and $i = 1, \ldots, n$

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(\tilde{p}_i | \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(\tilde{p}_i | \mu_l, \Sigma_l)}, \quad n_k = \sum_{i=1}^n \gamma_{ki}.$$

M-step: Compute, for $k = 1, \ldots, K$

$$\tilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1 - \alpha_k) \pi_k,$$

$$\tilde{\mu}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \tilde{p}_i + (1 - \alpha_k) \mu_k,$$

$$\tilde{\Sigma}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (\tilde{p}_i \tilde{p}_i^T) - \tilde{\mu}_k \tilde{\mu}_k^T + (1 - \alpha_k) (\Sigma_k + \mu_k \mu_k^T).$$
**Algorithm 1 EM Algorithm for GMM**

Input: sample patches $p_1, \ldots, p_n$, number of clusters $K$.
Output: Parameters $\Theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$.

Initialize $\pi_k^{(0)}, \mu_k^{(0)}$, and $\Sigma_k^{(0)}$ for $k = 1, \ldots, K$.

while not converge do

E-step : Compute, for $k = 1, \ldots, K$ and $i = 1, \ldots, n$

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(p_i | \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(p_i | \mu_l, \Sigma_l)}, \quad n_k = \sum_{i=1}^n \gamma_{ki}.$$ 

M-step : Compute, for $k = 1, \ldots, K$

$$\pi_k = \frac{n_k}{n}, \quad \mu_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} p_i,$$

$$\Sigma_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (p_i - \mu_k)(p_i - \mu_k)^T.$$ 

Update Counter: $m \leftarrow m + 1$

end while

**Algorithm 1 EM-adaptation Algorithm**

Input: Old parameters $\Theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$, new sample patches $\tilde{p}_1, \ldots, \tilde{p}_n$.

Output: Adapted parameters $\tilde{\Theta} = (\tilde{\pi}_k, \tilde{\mu}_k, \tilde{\Sigma}_k)_{k=1}^K$.

E-step : Compute, for $k = 1, \ldots, K$ and $i = 1, \ldots, n$

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(\tilde{p}_i | \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(\tilde{p}_i | \mu_l, \Sigma_l)}, \quad n_k = \sum_{i=1}^n \gamma_{ki}.$$ 

M-step : Compute, for $k = 1, \ldots, K$

$$\tilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1 - \alpha_k) \pi_k,$$

$$\tilde{\mu}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \tilde{p}_i + (1 - \alpha_k) \mu_k,$$

$$\tilde{\Sigma}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (\tilde{p}_i - \tilde{\mu}_k)(\tilde{p}_i - \tilde{\mu}_k)^T + (1 - \alpha_k) (\Sigma_k + \mu_k \mu_k^T)$$
EM Adaptation

Classical EM:

**Input:** sample patches $p_1, \ldots, p_n$

**Output:** A GMM with $\Theta \overset{\text{def}}{=} \{(\pi_k, \mu_k, \Sigma_k)\}_{k=1}^K$

EM Adaptation:

**Input:** some new patches $\{\tilde{p}_1, \ldots, \tilde{p}_n\}$,
and a generic GMM with $\Theta \overset{\text{def}}{=} \{(\pi_k, \mu_k, \Sigma_k)\}_{k=1}^K$

**Output:** A specific GMM with $\tilde{\Theta} \overset{\text{def}}{=} \{\tilde{\pi}_k, \tilde{\mu}_k, \tilde{\Sigma}_k\}_{k=1}^K$
Algorithm 1 EM Algorithm for GMM

Input: sample patches $p_1, \ldots, p_n$, number of clusters $K$.
Output: Parameters $\Theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$
Initialize $\pi_k^{(0)}$, $\mu_k^{(0)}$ and $\Sigma_k^{(0)}$ for $k = 1, \ldots, K$.
while not converge do
  E-step: Compute, for $k = 1, \ldots, K$ and $i = 1, \ldots, n$
    $\gamma_{ki} = \frac{\pi_k \mathcal{N}(p_i \mid \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(p_i \mid \mu_l, \Sigma_l)}$, $n_k = \sum_{i=1}^n \gamma_{ki}$.
  M-step: Compute, for $k = 1, \ldots, K$
    $\pi_k = \frac{n_k}{n}$, $\mu_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} p_i$,
    $\Sigma_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (p_i - \mu_k)(p_i - \mu_k)^T$.
Update Counter: $m \leftarrow m + 1$
end while

Algorithm 1 EM-adaptation Algorithm

Input: Old parameters $\Theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$,
new sample patches $\tilde{p}_1, \ldots, \tilde{p}_n$.
Output: Adapted parameters $\tilde{\Theta} = (\tilde{\pi}_k, \tilde{\mu}_k, \tilde{\Sigma}_k)_{k=1}^K$.

E-step: Compute, for $k = 1, \ldots, K$ and $i = 1, \ldots, n$
    $\gamma_{ki} = \frac{\pi_k \mathcal{N}(\tilde{p}_i \mid \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(\tilde{p}_i \mid \mu_l, \Sigma_l)}$, $n_k = \sum_{i=1}^n \gamma_{ki}$.

M-step: Compute, for $k = 1, \ldots, K$
    $\tilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1 - \alpha_k) \pi_k$,
    $\tilde{\mu}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \tilde{p}_i + (1 - \alpha_k) \mu_k$,
    $\tilde{\Sigma}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (\tilde{p}_i \tilde{p}_i^T) - \tilde{\mu}_k \tilde{\mu}_k^T$
    $+ (1 - \alpha_k) (\Sigma_k + \mu_k \mu_k^T)$
EM Adaptation

Classical EM:

**E-Step:** Compute the likelihood of $p_i$

$$
\gamma_{ki} = \frac{\pi_k \mathcal{N}(p_i | \mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(p_i | \mu_l, \Sigma_l)}, \quad n_k = \sum_{i=1}^{n} \gamma_{ki}
$$

EM Adaptation:

**E-Step:** Compute the likelihood of $\tilde{p}_i$

$$
\gamma_{ki} = \frac{\pi_k \mathcal{N}(\tilde{p}_i | \mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\tilde{p}_i | \mu_l, \Sigma_l)}, \quad n_k = \sum_{i=1}^{n} \gamma_{ki}
$$
Algorithm 1 EM Algorithm for GMM

Input: sample patches \( p_1, \ldots, p_n \), number of clusters \( K \).
Output: Parameters \( \Theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K \).

Initialize \( \pi_k^{(0)}, \mu_k^{(0)} \) and \( \Sigma_k^{(0)} \) for \( k = 1, \ldots, K \).

while not converge do

E-step: Compute, for \( k = 1, \ldots, K \) and \( i = 1, \ldots, n \)

\[
\gamma_{ki} = \frac{\pi_k \mathcal{N}(p_i | \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(p_i | \mu_l, \Sigma_l)}, \quad n_k = \sum_{i=1}^n \gamma_{ki}.
\]

M-step: Compute, for \( k = 1, \ldots, K \)

\[
\pi_k = \frac{n_k}{n}, \quad \mu_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} p_i,
\]

\[
\Sigma_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (p_i - \mu_k)(p_i - \mu_k)^T.
\]

Update Counter: \( m \leftarrow m + 1 \)
end while

Algorithm 1 EM-adaptation Algorithm

Input: Old parameters \( \Theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K \), new sample patches \( \tilde{p}_1, \ldots, \tilde{p}_n \).

Output: Adapted parameters \( \tilde{\Theta} = (\tilde{\pi}_k, \tilde{\mu}_k, \tilde{\Sigma}_k)_{k=1}^K \).

E-step: Compute, for \( k = 1, \ldots, K \) and \( i = 1, \ldots, n \)

\[
\gamma_{ki} = \frac{\pi_k \mathcal{N}(\tilde{p}_i | \mu_k, \Sigma_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(\tilde{p}_i | \mu_l, \Sigma_l)}, \quad n_k = \sum_{i=1}^n \gamma_{ki}.
\]

M-step: Compute, for \( k = 1, \ldots, K \)

\[
\tilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1 - \alpha_k) \pi_k,
\]

\[
\tilde{\mu}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \tilde{p}_i + (1 - \alpha_k) \mu_k,
\]

\[
\tilde{\Sigma}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (\tilde{p}_i \tilde{p}_i^T) - \tilde{\mu}_k \tilde{\mu}_k^T
\]

\[
+ (1 - \alpha_k)(\Sigma_k + \mu_k \mu_k^T)
\]
EM Adaptation

Classical EM:

**M-Step:** (Mean) Update the mean.

\[ \mu_k = \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \tilde{p}_i \]

EM Adaptation:

**M-Step:** (Mean) Linear combination of old and new mean.

\[ \tilde{\mu}_k = \alpha_k \left( \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \tilde{p}_i \right) + (1 - \alpha_k) \mu_k, \]

\[ \alpha_k \rightarrow 1: \text{more emphasis on the new data.} \]
\[ \alpha_k \rightarrow 0: \text{keep the generic parameter.} \]
\[ \alpha_k \overset{\text{def}}{=} \frac{n_k}{n_k + \rho} \text{ is the combination weight in our derivation.} \]

from some hyper-parameters
EM Adaptation in the literature

Theory of EM Adaptaion:


Speech/speaker verification:


Image classification:


Image denoising: ?
Goal: Update the generic GMM from a targeted image

Generic database
[Zoran-Weiss ‘11]
2 million 8x8 image patches

Image of interest
update

Gaussian mixture model

\[ f(p) = \sum_{j=1}^{K} \pi_j \mathcal{N}(p | \mu_j, \Sigma_j) \]
Goal: Update the generic GMM from a targeted image

Generic database
[Zoran-Weiss ‘11]
2 million 8x8 image patches

Image of interest
update

Gaussian mixture model

\[ f(p) = \sum_{j=1}^{K} \pi_j \mathcal{N}(p | \mu_j, \Sigma_j) \]
EM Adaptation for Noisy Images

If the data is itself noisy, first obtain a \textbf{pre-filtered} image.

Assume the pre-filtered image satisfies

\[ \tilde{P}_i = P_i + \epsilon_i, \]

\[ \epsilon_i \sim \mathcal{N}(0, \sigma^2 I) \]

In this case, the adaptation process becomes

\begin{align*}
\text{E-step:} & \quad \gamma_{ki} = \frac{\pi_k \mathcal{N}(\tilde{P}_i | \mu_k, \Sigma_k + \sigma^2 I)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\tilde{P}_i | \mu_l, \Sigma_l + \sigma^2 I)} \\
\text{M-step:} & \quad \tilde{\mu}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \tilde{P}_i + (1 - \alpha_k) \mu_k, \\
& \quad \tilde{\Sigma}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \left( \tilde{P}_i \tilde{P}_i^T - \sigma^2 I \right) - \tilde{\mu}_k \tilde{\mu}_k^T \\
& \quad \quad \quad \quad \quad + (1 - \alpha_k) (\Sigma_k + \mu_k \mu_k^T) \end{align*}
Stein’s Unbiased Risk Estimator (SURE)

What is the difference?

Clean:

\[ \tilde{\Sigma}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \left( \tilde{p}_i \tilde{p}_i^T \right) - \tilde{\mu}_k \tilde{\mu}_k^T \]

\[ + (1 - \alpha_k) \left( \Sigma_k + \mu_k \mu_k^T \right) \]

Pre-filtered:

\[ \tilde{\Sigma}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \left( \tilde{p}_i \tilde{p}_i^T - \tilde{\sigma}^2 I \right) - \tilde{\mu}_k \tilde{\mu}_k^T \]

\[ + (1 - \alpha_k) \left( \Sigma_k + \mu_k \mu_k^T \right) \]

How to estimate \( \tilde{\sigma}^2? \)

**Algorithm 1** Monte-Carlo SURE for Estimating \( \tilde{\sigma}^2 \)

- **Input:** noisy image \( y \in \mathbb{R}^n \), noise variance \( \sigma^2 \), a small \( \delta = 0.01 \).
- **Output:** \( \tilde{\sigma}^2 \).
- Generate \( b \sim \mathcal{N}(0, I) \in \mathbb{R}^n \).
- Construct \( y' = y + \delta b \).
- Apply MAP denoising with the generic GMM on \( y \) and \( y' \) to get two denoised images \( \bar{x} \) and \( \bar{x}' \), respectively.
- Compute \( \text{div} = \frac{1}{\delta} b^T (\bar{x}' - \bar{x}) \).
- Compute \( \tilde{\sigma}^2 = \text{MSE}(\bar{x}) = \frac{1}{n} \| y - \bar{x} \|^2 - \sigma^2 + \frac{2\sigma^2}{n} \text{div} \).
Monte-Carlo SURE Performance

Comparison between the true MSE and Monte-Carlo SURE when estimating $\tilde{\sigma}/\sigma$ over a large range of noise levels.
Overall Denoising with a Noisy Image

Noisy image \xrightarrow{\text{prefiltering}} \text{Prefiltered image} \rightarrow \text{SURE} \rightarrow \tilde{\sigma}^2

EM adaptation \rightarrow \text{Adapted GMM}

MAP denoising \rightarrow \text{Denoised image}

\text{Generic GMM} \rightarrow \text{classic EM}

\text{Image database}
Results
Single Image Denoising

noisy: $\sigma = 20$

Generic GMM: 31.47 dB

Adapted GMM: 31.82 dB

noisy: $\sigma = 20$

Generic GMM: 32.93 dB

Adapted GMM: 33.52 dB
Single Image Denoising

noisy: $\sigma = 40$

Generic GMM: 28.74 dB

Adapted GMM: 28.97 dB
Single Image Denoising

Generic GMM: 31.48 dB

Adapted GMM: 31.80 dB
Comparison with other state-of-the-art

<table>
<thead>
<tr>
<th></th>
<th>BM3D</th>
<th>aGMM -BM3D</th>
<th>EPLL</th>
<th>aGMM -EPLL</th>
<th>aGMM -clean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Boat</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma = 20$</td>
<td>29.59</td>
<td><strong>29.87</strong></td>
<td>29.78</td>
<td><strong>29.94</strong></td>
<td>30.47</td>
</tr>
<tr>
<td>$\sigma = 40$</td>
<td>26.19</td>
<td><strong>26.66</strong></td>
<td>26.54</td>
<td><strong>26.70</strong></td>
<td>27.01</td>
</tr>
<tr>
<td>$\sigma = 60$</td>
<td>24.65</td>
<td><strong>24.72</strong></td>
<td>24.74</td>
<td><strong>24.85</strong></td>
<td>25.11</td>
</tr>
<tr>
<td>$\sigma = 80$</td>
<td>23.39</td>
<td>23.27</td>
<td>23.38</td>
<td>23.40</td>
<td>23.62</td>
</tr>
<tr>
<td>$\sigma = 100$</td>
<td><strong>22.60</strong></td>
<td>22.41</td>
<td>22.60</td>
<td><strong>22.58</strong></td>
<td>22.77</td>
</tr>
<tr>
<td><strong>Cameraman</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma = 20$</td>
<td>30.24</td>
<td><strong>30.34</strong></td>
<td>30.24</td>
<td><strong>30.40</strong></td>
<td>31.33</td>
</tr>
<tr>
<td>$\sigma = 40$</td>
<td>26.83</td>
<td><strong>27.35</strong></td>
<td>26.98</td>
<td><strong>27.31</strong></td>
<td>27.97</td>
</tr>
<tr>
<td>$\sigma = 60$</td>
<td>25.31</td>
<td><strong>25.36</strong></td>
<td>25.30</td>
<td><strong>25.54</strong></td>
<td>26.25</td>
</tr>
<tr>
<td>$\sigma = 80$</td>
<td>23.88</td>
<td>23.77</td>
<td>23.70</td>
<td>23.88</td>
<td>24.53</td>
</tr>
<tr>
<td>$\sigma = 100$</td>
<td><strong>22.95</strong></td>
<td>22.83</td>
<td>22.75</td>
<td><strong>22.90</strong></td>
<td>23.48</td>
</tr>
<tr>
<td><strong>House</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma = 20$</td>
<td>33.65</td>
<td><strong>33.77</strong></td>
<td>32.98</td>
<td><strong>33.56</strong></td>
<td>34.58</td>
</tr>
<tr>
<td>$\sigma = 40$</td>
<td>30.47</td>
<td><strong>30.84</strong></td>
<td>29.90</td>
<td><strong>30.71</strong></td>
<td>31.54</td>
</tr>
<tr>
<td>$\sigma = 60$</td>
<td><strong>28.60</strong></td>
<td>28.36</td>
<td>27.64</td>
<td>28.32</td>
<td>29.02</td>
</tr>
<tr>
<td>$\sigma = 80$</td>
<td>27.36</td>
<td>27.18</td>
<td>26.61</td>
<td><strong>27.18</strong></td>
<td>27.75</td>
</tr>
<tr>
<td>$\sigma = 100$</td>
<td><strong>25.75</strong></td>
<td>25.68</td>
<td>25.18</td>
<td><strong>25.64</strong></td>
<td>25.96</td>
</tr>
<tr>
<td><strong>Lena</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma = 20$</td>
<td>31.61</td>
<td><strong>31.76</strong></td>
<td>31.44</td>
<td><strong>31.79</strong></td>
<td>32.68</td>
</tr>
<tr>
<td>$\sigma = 40$</td>
<td>27.83</td>
<td><strong>28.23</strong></td>
<td>28.07</td>
<td>28.36</td>
<td>28.81</td>
</tr>
<tr>
<td>$\sigma = 60$</td>
<td>26.38</td>
<td>26.22</td>
<td>26.04</td>
<td>26.31</td>
<td>26.59</td>
</tr>
<tr>
<td>$\sigma = 80$</td>
<td><strong>24.93</strong></td>
<td>24.83</td>
<td>24.53</td>
<td><strong>24.83</strong></td>
<td>25.04</td>
</tr>
<tr>
<td>$\sigma = 100$</td>
<td><strong>23.99</strong></td>
<td>23.89</td>
<td>23.63</td>
<td><strong>23.91</strong></td>
<td>24.03</td>
</tr>
<tr>
<td><strong>Montage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma = 20$</td>
<td>33.57</td>
<td>33.52</td>
<td>32.55</td>
<td>33.23</td>
<td>34.63</td>
</tr>
<tr>
<td>$\sigma = 40$</td>
<td>29.13</td>
<td><strong>29.39</strong></td>
<td>28.37</td>
<td>29.06</td>
<td>30.21</td>
</tr>
<tr>
<td>$\sigma = 60$</td>
<td>26.56</td>
<td><strong>26.66</strong></td>
<td>25.97</td>
<td><strong>26.59</strong></td>
<td>27.60</td>
</tr>
<tr>
<td>$\sigma = 80$</td>
<td>24.96</td>
<td><strong>24.98</strong></td>
<td>24.38</td>
<td><strong>24.95</strong></td>
<td>25.74</td>
</tr>
<tr>
<td>$\sigma = 100$</td>
<td>23.72</td>
<td><strong>23.76</strong></td>
<td>23.27</td>
<td><strong>23.64</strong></td>
<td>24.48</td>
</tr>
<tr>
<td><strong>Peppers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma = 20$</td>
<td>31.23</td>
<td><strong>31.50</strong></td>
<td>31.23</td>
<td><strong>31.52</strong></td>
<td>32.40</td>
</tr>
<tr>
<td>$\sigma = 40$</td>
<td>27.43</td>
<td><strong>27.96</strong></td>
<td>27.72</td>
<td>28.03</td>
<td>28.52</td>
</tr>
<tr>
<td>$\sigma = 60$</td>
<td>25.78</td>
<td><strong>25.87</strong></td>
<td>25.68</td>
<td><strong>25.99</strong></td>
<td>26.38</td>
</tr>
<tr>
<td>$\sigma = 80$</td>
<td>24.27</td>
<td><strong>24.45</strong></td>
<td>24.11</td>
<td><strong>24.50</strong></td>
<td>24.70</td>
</tr>
<tr>
<td>$\sigma = 100$</td>
<td>23.05</td>
<td><strong>23.28</strong></td>
<td>22.91</td>
<td><strong>23.30</strong></td>
<td>23.58</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26.86</td>
<td><strong>26.96</strong></td>
<td>26.61</td>
<td><strong>26.96</strong></td>
<td>27.56</td>
</tr>
</tbody>
</table>
Example Image Denoising - text denoising
denosing

model corresponds to
image fine structures
experimental method,
to propose a non
1 a digital image. The
d as the difference be
oven to be asymptotici
ance of all consider-
clean image

model corresponds to
image fine structures
experimental method,
to propose a non
1 a digital image. The
d as the difference be
oven to be asymptotici
ance of all consider-
noisy image

model corresponds to
image fine structures
experimental method,
to propose a non
1 a digital image. The
d as the difference be
oven to be asymptotici
ance of all consider-
example image

BM3D (16.70 dB)
aGMM-example (18.38 dB)
aGMM-clean (19.65 dB)
Example Image Denoising - face denoising

noisy: $\sigma = 50$

Generic GMM: 29.98 dB  Adapted GMM: 30.76 dB

noisy: $\sigma = 50$

Generic GMM: 29.34 dB  Adapted GMM: 30.10 dB
Conclusion
EM adaptation is

- a method to combine generic database and the noisy image

EM adaptation swings between

- Generic database
  - When noise is extremely high
  - When patches are relatively smooth
  - Where there are insufficient training samples

- Noisy image
  - When there are sharp edges in a patch
  - When there are enough training samples
Publication

Journal:


Conference:

Is Denoising Dead?

Priyam Chatterjee, Student Member, IEEE, and Peyman Milanfar, Fellow, IEEE

Abstract—Image denoising has been a well studied problem in the field of image processing. Yet researchers continue to focus attention on it to better the current state-of-the-art. Recently proposed methods take different approaches to the problem and yet...
Questions?

Thank you!
Appendix
Bayesian Inference

\( \{\tilde{p}_1, \ldots, \tilde{p}_n\} \)

MAP framework:

\[
\tilde{\Theta} = \arg \max \frac{1}{\tilde{\Theta}} \log f(\tilde{\Theta} | \tilde{p}_1, \ldots, \tilde{p}_n)
\]

\[
= \arg \max \frac{1}{\tilde{\Theta}} \left( \log f(\tilde{p}_1, \ldots, \tilde{p}_n | \tilde{\Theta}) + \log f(\tilde{\Theta}) \right)
\]

**Gaussian mixture model**

\[
f(\tilde{p}) = \sum_{j=1}^{m} \tilde{\pi}_j N(\tilde{p} | \tilde{\mu}_j, \tilde{\Sigma}_j)
\]

\[
\tilde{\Theta} \overset{\text{def}}{=} \{(\tilde{\pi}_k, \tilde{\mu}_k, \tilde{\Sigma}_k)\}_{k=1}^{K}
\]
\( f(\Theta) \) for GMM

Dirichlet:  \( \tilde{\pi}_1, \ldots, \tilde{\pi}_K \sim \text{Dir}(\nu_1, \ldots, \nu_k) \)

normal-inverse-Wishart:  \( (\tilde{\mu}_k, \tilde{\Sigma}_k) \sim \text{NIW}(\vartheta_k, \tau_k, \Psi_k, \varphi_k) \)

\[
f(\Theta) \propto \prod_{k=1}^{K} \left\{ \tilde{\pi}_k^{\nu_k-1} |\tilde{\Sigma}_k|^{-(\varphi_k+d+2)/2} \exp \left( -\frac{\tau_k}{2} (\tilde{\mu}_k - \vartheta_k)^T \tilde{\Sigma}_k^{-1} (\tilde{\mu}_k - \vartheta_k) - \frac{1}{2} \text{tr}(\Psi_k \tilde{\Sigma}_k^{-1}) \right) \right\}
\]

**Proposition 1:** Given the prior above, the posterior \( f(\Theta | \tilde{p}_1, \ldots, \tilde{p}_n) \) is given by

\[
f(\Theta | \tilde{p}_1, \ldots, \tilde{p}_n) \propto \prod_{k=1}^{K} \left\{ \tilde{\pi}_k^{\nu_k'-1} |\tilde{\Sigma}_k|^{-(\varphi_k'+d+2)/2} \exp \left( -\frac{\tau_k'}{2} (\tilde{\mu}_k - \vartheta_k')^T \tilde{\Sigma}_k^{-1} (\tilde{\mu}_k - \vartheta_k') - \frac{1}{2} \text{tr}(\Psi_k' \tilde{\Sigma}_k^{-1}) \right) \right\}.
\]
Solution to $\tilde{\Theta}$

**optimization:**

$$\begin{align*}
\text{maximize}_{\tilde{\Theta}} & \quad L(\tilde{\Theta}) \overset{\text{def}}{=} \log f(\tilde{\Theta}|\tilde{p}_1, \ldots, \tilde{p}_n) \\
\text{subject to} & \quad \sum_{k=1}^{K} \tilde{\pi}_k = 1.
\end{align*}$$

**Proposition 2** [Luo, Chan, and Nguyen 2016]: The solution $(\tilde{\pi}_k, \tilde{\mu}_k, \tilde{\Sigma}_k)$ are:

$$\begin{align*}
\tilde{\pi}_k &= \frac{n}{(\sum_{k=1}^{K} v_k - K) + n} \cdot \frac{n_k}{n} + \frac{\sum_{k=1}^{K} v_k - K}{(\sum_{k=1}^{K} v_k - K) + n} \cdot \frac{v_k - 1}{\sum_{k=1}^{K} v_k - K}, \\
\tilde{\mu}_k &= \frac{1}{\tau_k + n_k} \sum_{i=1}^{n} \gamma_{ki} \tilde{p}_i + \frac{\tau_k}{\tau_k + n_k} \vartheta_k, \\
\tilde{\Sigma}_k &= \frac{n_k}{\varphi_k + d + 2 + n_k} \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} (\tilde{p}_i - \tilde{\mu}_k)(\tilde{p}_i - \tilde{\mu}_k)^T \\
&\quad + \frac{1}{\varphi_k + d + 2 + n_k} \left( \Psi_k + \tau_k (\vartheta_k - \tilde{\mu}_k)(\vartheta_k - \tilde{\mu}_k)^T \right).
\end{align*}$$
Simplification of $\tilde{\Theta}$

**Proposition 3** [Luo, Chan, and Nguyen 2016]: Define $\rho \overset{\text{def}}{=} \frac{n_k}{n} \left( \sum_{k=1}^{K} v_k - K \right) = \tau_k = \varphi_k + d + 2$. Let

$v_k = \mu_k$, $\Psi_k = (\varphi_k + d + 2) \Sigma_k$, $\frac{v_k - 1}{\sum_{k=1}^{K} v_k - K} = \pi_k$,

and $\alpha_k = \frac{n_k}{\rho + n_k}$, then we become

$$\tilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1 - \alpha_k) \pi_k,$$  \hspace{1cm} (1)

$$\tilde{\mu}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \tilde{p}_i + (1 - \alpha_k) \mu_k,$$  \hspace{1cm} (2)

$$\tilde{\Sigma}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} (\tilde{p}_i - \tilde{\mu}_k) (\tilde{p}_i - \tilde{\mu}_k)^T + (1 - \alpha_k) \left( \Sigma_k + (\mu_k - \tilde{\mu}_k)(\mu_k - \tilde{\mu}_k)^T \right).$$  \hspace{1cm} (3)
Convergence of EM Adaptation Algorithm

The PSNR only improves marginally after the first iteration.
Single Image Denoising

noisy: $\sigma = 40$

Generic GMM: 29.47 dB

Adapted GMM: 29.72 dB