Image Denoising with global structure and local similarity preservations

Gopi Manthena 0030191751, Karthik Sajeev - 0030028666 , Mihir Bhatia - 0029923247

1 Introduction

Images are widely used in various fields, and are usually contaminated by noise during acquisition, transmission and compression. Consequently, real-life images are often degraded with noise and there is often a need for image denoising techniques. Although several techniques have been discussed in the literature, most tend to work well only for specific types of noise removal problems.

In our project, we implement the image denoising method from [2], which is regarded as a highly under-constrained problem. This method removes both additive and multiplicative noise from the image. The method would also allow us to preserve both the global and local structure of the image. This is done by combining reconstruction and learning based approaches using pixel-level and patch-based filtering methods. To capture global structure of the image, Laplacian Schatten p-norm is used. To reduce the multiplicative noise approximately to additive noise, a sparse term is introduced.

Moreover, a new approach to solving the optimization problem of dictionary learning called MOD-AK-SVD (Method of Optimal Directions - Approximate K - SVD) is proposed which is a combination of K-SVD [1] and MOD from [3]

2 Formulation

Image denoising may be regarded as a two-stage optimization problem, which splits the objective function into convex and non-convex parts. The former is solved using alternating optimization and the gradient descent method, while the latter uses the proposed dictionary learning MOD-AK-SVD approach. Our image denoising problem been formulated as:

$$\min \quad G(X,S) + \mu H(X,\Phi,\Omega) \tag{1}$$

G is the function that helps reconstruct global structure, H is the function that preserve local structure, μ is a penalty parameter. In the global function G we represent our noisy image Y as:

$$Y = X + S + E \tag{2}$$

X = Clean image, S = Matrix containing globally sparse noise, E = Remaining noise

$$G(X,S) = ||Y - X - S||_F^2 + \lambda_1 ||LX||_{S_p}^p + \lambda_2 ||S||_1 \tag{3}$$

L = Laplacian operator (high pass filter), λ_1, λ_2 = penalty parameters, $||.||_{S_p}$ = Schatten p-norm

$$H(X,\Phi,\Omega) = \sum_{i=1}^{N} ||R_i X - \Phi\omega_i||_2^2 \quad s.t. \quad ||\omega_i||_0 \leq T \tag{4}$$

Ω = sparse coefficient matrix, Φ = dictionary, R_i extracts the i^{th} path from X such that column vector $x_i = R_i X$, $\omega_i = i^{th}$ column of Ω, T = parameter that controls sparsity of representation

We minimize the two functions G and H separately using two different methods (described in section 3.1 and 3.2)

3 Approach

From [2] we use the following approaches to solve the formulated optimization problem

- **Global Structure Reconstruction Stage**- where gradient descent is used to reconstruct the global features of the image while denoising
- **Dictionary Learning Stage** - where the newly proposed MOD-AK-SVD is used to preserve local structure

3.1 Global Structure Reconstruction Stage

This method pertains to $G(X, S)$ in the formulation above, and can be considered as a pixel-level filtering technique. As the objective function is convex, the gradient descent method is used for optimization. An alternating optimization strategy is used for X and S, where one is fixed and the other updated until convergence is attained.

$$S_t^{s+1} \leftarrow \arg\min_S \quad ||Y - X_t^{(s)} - S||_F^2 + \lambda_2 ||S||_1 \tag{5}$$

$$X_t^{s+1} \leftarrow \arg\min_X ||Y - X - S_t^{(s+1)}||_F^2 + \lambda_1 \mathbf{Tr}[(LX)^T (LX) + \delta^2 I]^{p/2} + \mu ||X - X_2^t||_F^2 \tag{6}$$

A smoothing parameter δ is used to smooth the Laplacian Schatten norm term. X_1^t and X_2^t are obtained at the t^{th} step my minimizing the modified G and H functions given by:

$$\tilde{G} = G + \mu ||X - X_2^t||_F^2 \tag{7}$$

$$\tilde{H} = H + \frac{1}{\mu} ||X - X_1^{t+1}||_F^2 \tag{8}$$

The norm terms are added to have a connection between $\tilde{G}$ and $\tilde{H}$ while optimizing them.

3.2 Dictionary Learning Stage

In the dictionary learning stage, we attempt to minimize $H(X, \Phi, \Omega)$. This is non-convex part of our minimization problem. Here we use a patch based filtering technique. For this stage too, we use a alternating optimization strategy where we fix the value of 2 variables and update the 3^{rd} one. Moreover, since the l_0 minimization is an NP-Hard problem methods such as basis pursuit (BP), matching pursuit (MP), orthogonal matching pursuit (OMP) and focal underdetermined system solver(FOCUSS) can be used. In our project we use OMP which greedily selects dictionary atoms sequentially and is faster than MPA.

In [2], the above mentioned techniques are combined into an algorithm called Laplacian Schatten p-norm and Learning Algorithm (LSLA-p). There are 2 implementations of LSLA depending on the value of p which gives us LSLA-1 and LSLA-2 for $p = 1$ and $p = 2$ respectively.

4 Algorithm

The LSLA algorithm from [2] consists of the following steps:

Algorithm 1 The Laplacian Schatten p-norm and Learning Algorithm (LSLA-p).

Require: Noisy Image: Y;
Penalty parameter: $\lambda_1, \lambda_2, \mu$;
Smoothing parameter: δ;
Stopping tolerence: ϵ_1, ϵ_2;
Clearn Image X;

Initialize Φ, $X_2^0 = 0, S = 0$
$t = 0$; a = 1; $j = 1; \Delta_1 = \epsilon_1 + 1; \Delta_2 = \epsilon_2 + 1$
repeat
  $s = 0$
  while $\Delta_2 \geq \epsilon_2$ & $s \leq s_{\max}$ **do**
  $S_t^{s+1} \leftarrow \arg\min_S \|Y - X_t^s - S\|_F^2 + \lambda_2 \|S\|_1$ (17)
  $X_t^{s+1} \leftarrow \arg\min_X \|Y - X - S_t^{s+1}\|_F^2 + \lambda_1 Tr[(\mathbb{L}X)^T(\mathbb{L}X) + \delta^2 I]^{p/2} + \mu\|X - X_2^t\|_F^2$ (18)
  $\Delta_2 = \min\left\{\|X_t^{s+1} - X_t^s\|_F^2, \|S_t^{s+1} - S_t^s\|_F^2\right\}$
  $s = s + 1$
  end while
  $X_1^{t+1} = X_t^{(s)}$
  $X_2^{t+1} \leftarrow \arg\min_X \frac{1}{\mu}\|X - X_1^{t+1}\|_F^2 + \sum_{i=1}^N \|\mathbf{R}_i X - \Phi\omega_i\|_2^2 \quad s.t. \|\omega_i\|_0 \leq T$ (21)
  $\Delta_1 = \min\left\{\|X_1^{t+1} - X_1^t\|_F^2, \|X_2^{t+1} - X_2^t\|_F^2\right\}$
  $t = t + 1$
until $\Delta_1 \leq \epsilon_1$ or $t \geq t_{\max}$
return $X = X_2^t$
Sparse Coding Stage: Ω = OMP(X, Φ, k_0)
Update Dictionary Φ
repeat
  $X = \left(\frac{1}{\mu}I + \sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i\right)^{-1}\left(\frac{1}{\mu}X_1^t + \sum_{i=1}^N \mathbf{R}_i^T \Phi\omega_i\right)$ (27)
  $E = X - \Phi\Omega$
  repeat
    $E_j = E + \varphi_j\omega_{\tilde{j}}$
    $\varphi_j = E_j(:, t_j)\omega_{\tilde{j}}^T(t_j)$ where $t_j = \{i : \omega_{\tilde{j}}(i) \neq 0\}$
    $\omega_{\tilde{j}}(t_j) = \varphi_j^T E_j(:, t_j)$
    $E = E_j - \varphi_j\omega_{\tilde{j}}$
    j++
  until j = K
  a++
until a = A

Figure 1: LSLA algorithm

4.1 Description

Our images $Y, X \in \mathbb{R}^{h*w}$. $R_i X \in \mathbb{R}^n$ where R_i is the patch extraction operator that extracts patch i from the image X and vectorizes it. R_i^T takes the patch and places it in an blank image at the correct

patch location. ω_i,ϕ_i refers column i of Ω , Φ respectively. $\mathbf{L}$ is the Laplacian operator that acts as a high pass filter for the image X. We perform convolutions of $\mathbf{L}$ over X to get $\mathbf{L}X$.

From **line 1-2** we initialize X_2, S and the dictionary Φ. To initialize the dictionary we use a 2D separable Discrete Cosine Transform (DCT) dictionary of size $\mathbb{R}^{n*k}$ where $n = h$ and $k > n$. First we produce a 1D-DCT matrix Φ_{1D} of size $\sqrt{n} * \sqrt{k}$ such that each atom $\phi_{1D} = \cos((i-1)(j-1)\pi/11)$, $i = 1...\sqrt{n}, j = 1...\sqrt{k}$. To get Φ we use the Kronecker-product $\Phi = \Phi_{1D} \otimes \Phi_{1D}$.

From **line 3-10** we work on the global structure stage as described in equations (5) and (6). For equation (6) we convert the $\mathbf{Tr}(\mathbf{L}X)^T(\mathbf{L}X)$ term into $|| \mathbf{L}X||_F^2$. We adjust the values of s_{max} and t_{max} as to make sure that convergence occurs but also such that run time is not too high. On **line 11-13**, we begin the dictionary learning stage by optimizing the modified version of function H (from equation (8)) until the convergence condition from **line 13** is satisfied. From **line 17** is the Sparse Coding Stage to update Ω using OMP as described in Section **3.2**. **Line 18 - 28** describe a repeated updating process where we alternate between X and Φ and update using the alternating optimization strategy. **Line 22 - 27** describe the update algorithm from [3] where updates to the dictionary are made in an iterative manner, atom by atom.

4.2 Critical Observations

- [2] doesn't mention how X is initialized. We observed that initializing X with the noisy image Y led to faster convergence (i.e. we could set lower values of s_{max},t_{max} and still observe fast convergence.
- [2] has no description of how the patch extraction operator R_i works, and how we need to vectorize the image as well have an output that is vectorized. Other sources describe the operator differently where the patch extracted is a matrix. We figured how R_i and R_i^T worked as per the equations and coded general function to extract patches of size $\sqrt{n} * \sqrt{n}$ where $n * n$ is the image size.
- s_{max}, t_{max} values are not mentioned even for their case (as different images would take different times with the global structure algorithm to converge) in [2]. We tweaked the values s_{max}, t_{max} to balance out time taken and convergence of the algorithm.

5 Parameters

We explain the different parameters and the significance of their values here:

We explored different values of λ_1 and noticed that the output image X seemed to be quite responsive to changes in λ_1. We tried out six levels of λ_1: 20, 10, 5, 1, 0.1, and 0.01. The observed results in the same order are shown below:

Figure 2: Varying levels of λ_1

Next, we fixed a particular value of λ_1 ($\lambda_1 = 0.1$) corresponding to the best result above, and varied λ_2. Again, six levels of λ_2 were tested: 10, 1, 0.1, 0.01, 0.05, and 0.001. The output in the same order is shown below:

Figure 3: Varying levels of λ_2

As seen above, the output image does not seem to vary for different values of λ_2. We thus fix the value of λ_2 to the default value specified in the paper ($\lambda_2 = 0.1$).

Values for many other parameters are set according to the values suggested in the paper. σ is set to 3, δ is set to 0 (corresponding to p=2 for LSLA-2), ϵ_1 and ϵ_2 are both set to 0.001, and the same Laplacian operator $\mathbf{L}$ is used. A is set to 3, in line with its usage in [3].

Different values of t_{max} and s_{max} are tried out, but it was seen that they did not change the output. This may be reasoned as follows: t_{max} and s_{max} are merely used to provide an upper limit to the number of iterations, if convergence does not occur soon enough. In our case, as we initialized X with the noisy image Y, it is seen that convergence always occurs within a few iterations. If we had initialized X to some other image, convergence would take longer and hence the values of t_{max} and s_{max} would be of significance.

K is fixed to the number of atoms in the dictionary, as we need to update each atom of the dictionary in the dictionary learning stage. T represents the upper limit of the maximum number of non-zero elements for each sparse representation vector ω_i, and is set to 20. The sparsity level k_0, which is passed as a parameter to the OMP algorithm, is set to 15.

6 Results

After fixing the values of all parameters as explained above, we tested out the algorithm on five images with varying noise levels. The five noise levels correspond to different values of variance of Gaussian noise applied to the original image: 0.001, 0.005, 0.01, 0.02 and 0.05. The output is shown below, where the images on the left are the noisy images with increasing noise levels, and the images on the right are the corresponding denoised images.

Figure 4: Denoising images of different noise levels

6.1 Performance

We check the performance of our algorithm with various parameters using two indicators

1. **PSNR** (Peak Signal to Noise Ratio) which is defined as $PSNR = 10\log_{10}(\frac{MAX_I^2}{MSE})$ where MSE is the mean squared error between the noise free and approximated image.

2. **SSIM** (Structural Similarity Index Method) is used to check the similarity between the noisy/denoised image & original noise-free image and can be a good measure for checking performance of denoising algorithms.

The performance is checked for all five noise levels, and the results in terms of both PSNR and SSIM are summarized in the tables below:

Table 1: PSNR of noisy and denoised images at different initial noise levels

Noise Variance	**0.001**	**0.005**	**0.01**	**0.02**	**0.05**
PSNR - noisy image	-18.21	-25.13	-28.21	-30.95	-34.49
PSNR - denoised image	-26.25	-26.52	-29.227	-27.22	-29.56

Table 2: SSIM of noisy and denoised images at different initial noise levels

Noise Variance	**0.001**	**0.005**	**0.01**	**0.02**	**0.05**
SSIM - noisy image	0.7765	0.5733	0.4621	0.3596	0.2359
SSIM - denoised image	0.6989	0.651	0.5906	0.5589	0.4864

Table 1 shows the PSNR values for the noisy and denoised images. A higher noise level corresponds to a lower value of PSNR, indicating a larger deviation from the original image. We can thus say that, a higher PSNR value after denoising indicates a successful denoising operation. In our case, this is seen to

be the case for higher levels of noise (0.02 and 0.05). For lower levels, the denoised images either have comparable or lower PSNR values compared to the noisy images.

Table 2 shows the SSIM values for the noisy and denoised images. Again, it is clear from the values of the noisy images that a higher noise level corresponds to a lower value of SSIM, indicating lower similarity with the original image. A higher SSIM value of the denoised image compared to the noisy image shows successful denoising. This is true in our case for all levels except 0.001.

Based on both the performance indicators above, it can be concluded that while the algorithm does not work too well for low levels of noise, performance is significantly improved for higher noise levels.

References

[1] Michal Aharon, Michael Elad, Alfred Bruckstein, et al. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311, 2006.

[2] Shuting Cai, Zhao Kang, Ming Yang, Xiaoming Xiong, Chong Peng, and Mingqing Xiao. Image denoising via improved dictionary learning with global structure and local similarity preservations. *Symmetry*, 10(5):167, 2018.

[3] Shuting Cai, Shaojia Weng, Binling Luo, Daolin Hu, Simin Yu, and Shuqiong Xu. A dictionary-learning algorithm based on method of optimal directions and approximate k-svd. In *Control Conference (CCC), 2016 35th Chinese*, pages 6957–6961. IEEE, 2016.