A Hyperspectral Image Inpainting Algorithm with Convergence Guarantee via Convex Self-similarity Regularization

Yee-Hsun Lee

October 1, 2020

1 Introduction

- What is hyperspectral image?

  Hyperspectral images (HSIs) usually have more than 100 spectrum bands, which can be used to detect objects, identify materials or analysis ingredients by collecting and processing the information from spectrums and pixels in the HSIs. Most of the above applications can not be done in multispectral image, i.e. RGB image.

- What is image inpainting?

  During the processing or transmission of an image, information in pixels or spectrums may miss or have some defects. Image inpainting is the process of image reconstruction from incomplete observation.

- Motivation

  Convex function guarantees global optimal solution in optimization problems. Therefore, convex function is always our top priority in optimization problems. Moreover, most of the state-of-art denoiser, i.e. BM3D, are non-convex function. In [2], BM3D is served as a regularizer, however, it doesn't guarantee the convergence of algorithm. Therefore, in this project, we will propose a new inpainting algorithm with a convex self-similarity regularizer base on [2] to obtain global optimal solution and convergence guarantee.

2 Problem formulation

The HSI observation problem can be written as

$$\mathbf{Y} = \mathbf{X} + \mathbf{N} \tag{1}$$

where $\mathbf{Y}$, $\mathbf{X}$, $\mathbf{N}$ represent the observed HSI data, the clean HSI data and noise respectively.

In this project, we take advantage of the HSIs' low-rank structure and self-similarity. The clean image can be represented in a subspace by the low-rank structure of HSI [2]. Computing complexity and time can be substantially reduced by processing data in eigenimages.

$$\mathbf{X} = \mathbf{E}\mathbf{Z} \tag{2}$$

where $\mathbf{E}$ is a subspace matrix that can be obtained by hysime algorithm in [2] and $\mathbf{Z}$ is eigenimages.

The observed image $\mathbf{Y}$ may be vectorized as $\mathbf{y} = \text{vec}(\mathbf{Y})$, the clean image $\mathbf{X}$ may be vectorized as $\mathbf{x} = \text{vec}(\mathbf{X})$ and the noise $\mathbf{N}$ may be vectorized as $\mathbf{n} = \text{vec}(\mathbf{N})$. The HSI inpainting model can be written as

$$\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{n} \tag{3}$$

which is also utilized in [2], where the binary matrix $\mathbf{M}$ is a mask that represents incomplete data. Since $\mathbf{X} = \mathbf{E}\mathbf{Z}$ and the vectorization property, we have

$$\mathbf{x} = vec(\mathbf{X}) = vec(\mathbf{E}\mathbf{Z}) = (\mathbf{I} \otimes \mathbf{E})\mathbf{z} \tag{4}$$

where $\otimes$ represents Kronecker product and $\mathbf{z} = \text{vec}(\mathbf{Z})$. The HSI inpainting problem is formulated as [2]

$$\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2^2 + \lambda\phi(\mathbf{z}) \tag{5}$$

and $\mathbf{x} = (\mathbf{I} \otimes \mathbf{E})\mathbf{z}$, we have

$$\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{y} - \mathbf{M}(\mathbf{I} \otimes \mathbf{E})\mathbf{z}\|_2^2 + \lambda\phi(\mathbf{z}) \tag{6}$$

where λ is the regularizer parameter and ϕ is the self-similarity regularizer.

Such an image inverse problem is ill-posed, so we need to add suitable regularizer. Since self-similarity is commonly observed in HSIs, we exploit this property to design a regularizer ϕ. Moreover, this self-similarity regularizer ϕ is a convex function, which can guarantee convergence and global optimal solution.

Instead of solving (6) directly, which involves a large non-diagonal matrix. Large non-diagonal matrix will cost a lot of time when doing matrix operations, i.e. matrix inversion and matrix transpose. We can recover the incomplete data first to simplify the optimization problem. Instead of processing the whole HSI at once, we would rather divide the problem into smaller pieces (pixelwise) [2], and

$$\mathbf{y}_i = \mathbf{M}_i \mathbf{x}_i + \mathbf{n}_i \tag{7}$$

is the subproblem that is observed at the same pixel. Since $\mathbf{x}_i = \mathbf{E}\mathbf{z}_i$, (7) can be written as

$$\mathbf{y}_i = \mathbf{M}_i \mathbf{E} \mathbf{z}_i + \mathbf{n}_i \tag{8}$$

Note that (8) is in the format of linear regression, so we can simply obtain the estimation of $\mathbf{z}_i$ as

$$\hat{\mathbf{z}}_i = (\mathbf{E}^T \mathbf{M}_i^T \mathbf{M}_i \mathbf{E})^{-1} \mathbf{E}^T \mathbf{M}_i^T \mathbf{y}_i \tag{9}$$

by the least square estimator of coefficient.

We can obtain the complete but noisy HSI by computing $\hat{\mathbf{y}}_i = \mathbf{E}\hat{\mathbf{z}}_i$, then our inpainting problem can be simplified as a denoising problem

$$\min_{\mathbf{z}} \frac{1}{2} \|\hat{\mathbf{y}} - \mathbf{x}\|_2^2 + \lambda \phi(\mathbf{z}) \tag{10}$$

and $\mathbf{x} = (\mathbf{I} \otimes \mathbf{E})\mathbf{z}$, we have

$$\min_{\mathbf{z}} \frac{1}{2} \|\hat{\mathbf{y}} - (\mathbf{I} \otimes \mathbf{E})\mathbf{z}\|_2^2 + \lambda \phi(\mathbf{z}) \tag{11}$$

Now, let's move on to ϕ. In this project, we use a self-similarity regularizer ϕ [1] that is a convex function and mitigates the ill-posedness. The general form is defined as follow:

$$\phi(\mathbf{z}) = \frac{1}{2} \sum_{b=1}^{p} \sum_{(i,j)\in\kappa} \alpha_{i,j} \|\mathbf{P}_i \mathbf{z}_b - \mathbf{P}_j \mathbf{z}_b\|_2^2 = \frac{1}{2} \sum_{(i,j)\in\kappa} \alpha_{i,j} \|(\mathbf{P}_i - \mathbf{P}_j)\mathbf{Z}\|_2^2 \tag{12}$$

where p is the dimension of subspace, $\mathbf{z}_b$ is the bth band of the eigenimage $\mathbf{Z}$, $\mathbf{P}_i$ is an operator that shifts on image to find similar patches, κ is the self-similarity graph that collects pairs of similar patches and $\alpha_{i,j}$ is the degree of the similarity. Further details can be found in [1].

At this point, we want to solve (11) by ADMM (alternating direction method of multipliers), a powerful tool that solves convex optimization problems by breaking them into smaller pieces.

3 Algorithm

In this section, we will show you how to solve (11) by ADMM. If we want to solve the optimization problem by ADMM, then we must reformulate our problem into the standard problem form of ADMM. We equivalently reformulate (11) as

$$\begin{aligned} \min_{\mathbf{z}} \quad & \frac{1}{2}\left\|\hat{\mathbf{y}}-\mathbf{v}\right\|_2^2+\frac{\lambda}{2}\sum_{(i,j)\in\kappa}\alpha_{i,j}\left\|(\mathbf{P}_i-\mathbf{P}_j)\mathbf{V}_{i,j}\right\|_2^2 \\ s.t. \quad & \mathbf{v}=(\mathbf{I}\otimes\mathbf{E})\mathbf{z}, \\ & \mathbf{V}_{i,j}=\mathbf{Z}, \end{aligned} \tag{13}$$

Then we rewrite (13) as the augmented Lagrangian by combining the constraints into the equation as

$$\begin{aligned} \mathcal{L}(\mathbf{v},\{\mathbf{V}_{i,j}\},\mathbf{z},\mathbf{d},\{\mathbf{D}_{i,j}\}) = & \frac{1}{2}\left\|\hat{\mathbf{y}}-\mathbf{v}\right\|_2^2+\frac{\mu}{2}\left\|(\mathbf{I}\otimes\mathbf{E})\mathbf{z}-\mathbf{v}-\mathbf{d}\right\|_2^2 \\ & +\frac{\lambda}{2}\sum_{(i,j)\in\kappa}\alpha_{i,j}\left\|(\mathbf{P}_i-\mathbf{P}_j)\mathbf{V}_{i,j}\right\|_2^2 \\ & +\frac{\mu}{2}\sum_{(i,j)\in\kappa}\left\|\mathbf{Z}-\mathbf{V}_{i,j}-\mathbf{D}_{i,j}\right\|_2^2 \end{aligned} \tag{14}$$

where $\mathbf{v}$ and $\{\mathbf{V}_{i,j}\}$ are primal variables in vector and matrix form, $\mathbf{d}$ and $\{\mathbf{D}_{i,j}\}$ are scaled dual variables in vector and matrix form and $\mu > 0$ is the penalty parameter.

In (14), $\{\mathbf{v},\mathbf{V}_{i,j}\}$ and $\{\mathbf{z},\mathbf{Z}\}$ are two "joint" variables in the optimization problem, which means that these two variables must be solved and considered in the same time. However, ADMM allows us to update them sequentially. Therefore, we can divide the whole optimization problem into two convex subproblems. One is updating primal variables $\{\mathbf{v},\mathbf{V}_{i,j}\}$, another is updating eigenimage $\mathbf{z}$.

$$\mathbf{v}^{k+1} = \min_{\mathbf{v}} \frac{1}{2} \|\hat{\mathbf{y}} - \mathbf{v}\|_2^2 + \frac{\mu}{2} \|(\mathbf{I} \otimes \mathbf{E})\mathbf{z} - \mathbf{v} - \mathbf{d}\|_2^2 \tag{15}$$

$$\mathbf{V}_{i,j}^{k+1} = \min_{\mathbf{V}_{i,j}} \frac{\lambda}{2} \sum_{(i,j)\in\kappa} \alpha_{i,j} \|(\mathbf{P}_i - \mathbf{P}_j)\mathbf{V}_{i,j}\|_2^2 + \frac{\mu}{2} \sum_{(i,j)\in\kappa} \|\mathbf{Z} - \mathbf{V}_{i,j} - \mathbf{D}_{i,j}\|_2^2 \tag{16}$$

note that equation (15) and (16) are for updating primal variables.

$$\mathbf{z}^{k+1} = \min_{\mathbf{z}} \frac{\mu}{2} \|(\mathbf{I} \otimes \mathbf{E})\mathbf{z} - \mathbf{v} - \mathbf{d}\|_2^2 + \frac{\mu}{2} \sum_{(i,j)\in\kappa} \|\mathbf{Z} - \mathbf{V}_{i,j} - \mathbf{D}_{i,j}\|_2^2 \tag{17}$$

note that equation (17) is for updating eigenimage, where $\mathbf{Z} = vec^{-1}(\mathbf{z})$

Algorithm 1 ADMM Algorithm for solving (13)

1: **Given** : $\mathbf{y}, \kappa, \alpha_{i,j}, \lambda > 0$ and $\mu > 0$.
2: Initialization : $\mathbf{z}^0$:= zero vector, $\mathbf{d}^0$:= zero vector, $\mathbf{D}_{i,j}^0$:= zero matrix (or warm start for faster computation). Set k := 0
3: **repeat**
4: Update primal variables $\mathbf{v}^{k+1}$ and $\mathbf{V}_{i,j}^{k+1}$ by (15) and (16).
5: Update eigenimage $\mathbf{z}^{k+1}$ by (17).
6: Update dual variables $\mathbf{d}^{k+1} = \mathbf{d}^k + \mathbf{v}^{k+1} - (\mathbf{I} \otimes \mathbf{E})\mathbf{z}^{k+1}$ and $\mathbf{D}_{i,j}^{k+1} = \mathbf{V}_{i,j}^{k+1} - \mathbf{Z}^{k+1}$.
7: **until** the desired result is met (to be designed in experiment)
8: **Output** the clean and complete HSI $\hat{\mathbf{x}} = (\mathbf{I} \otimes \mathbf{E})\mathbf{z}^k$

Next, we will derive the close form solution for the equation (15), (16) and (17) that involves in lines 4 and 5 in Algorithm 1.

Since (15) is a convex function, its optimal solution can be easily found by taking the gradient of (15) as zero, the close-form solution can be derived as

$$\mathbf{v}^{k+1} = \frac{1}{2}(\mathbf{y} + (\mathbf{I} \otimes \mathbf{E})\mathbf{z}^k - \mathbf{d}^k) \tag{18}$$

Although (16) and (17) are also convex function, it takes a lot of time to compute its close-form solution that derived from taking the gradient as zero, because it involves large matrix operations. Therefore, we reference the close-form solution in [1]. Further details and proofs can be found in [1].

4 Experiment

In this section, we test the performance of the proposed method in simulation. The HSI data of Washington DC is used in our simulated experiment, as shown in Fig.1 (a). To simulate the missing information and the defects in data, we add some stripes and noise (Gaussion noise) into groundtruth like Fig.1 (b). The inpainting result of the proposed algorithm is shown in Fig.1 (c).

To further depict the inpainting result, two quantitative picture quality indices are employed for performance evaluation, including peak signal-to-noise ratio (PSNR) and structure similarity (SSIM), PSNR is a measurement for the ratio between the peak signal power and the noise power, and SSIM is a measurement for the similarity between two pictures (Fig.1 (a) and Fig.1 (c) in my project). Furthermore, computational time T in seconds (sec.) is adopted as an index of computational efficiency. The experiment quantitative results are shown in Table 1.

All the experiments in this section are executed on a computer facility equipped with Core-i9-10900X CPU with 3.70-GHz speed and 64-GB RAM, and all the proposed methods are implemented in Mathworks MATLAB R2019a.

(a) (b) (c)

Figure 1: (a)The RGB image of the Washington DC Mall HSI data (RGB = bands 45, 35, and 15) (b)The noisy and stripe-corrupted version (c)The inpainting result of the proposed method.

From the Fig.1, visually, the proposed method is able to inpaint all the stripes and remove most of the noise. From the Table 1, numerically, the proposed method show good quantitative picture quality indices, although Algorithm 1 takes more time than the other state-of-art denoise or inpainting algorithms, however, the main motivation in this project is to propose a new algorithm with methametically provable convergence guarantee and global optimal solution.

Table 1: Quantitative results of the simulated experiment

PSNR	SSIM	T(sec.)
48	0.99692	318

Furthermore, we conduct the experiment to show the convergence guarantee for visual effect, and the result is depicted in Fig.2. From Fig.2, we can find that the algorithm has converged after the 50th iteration.

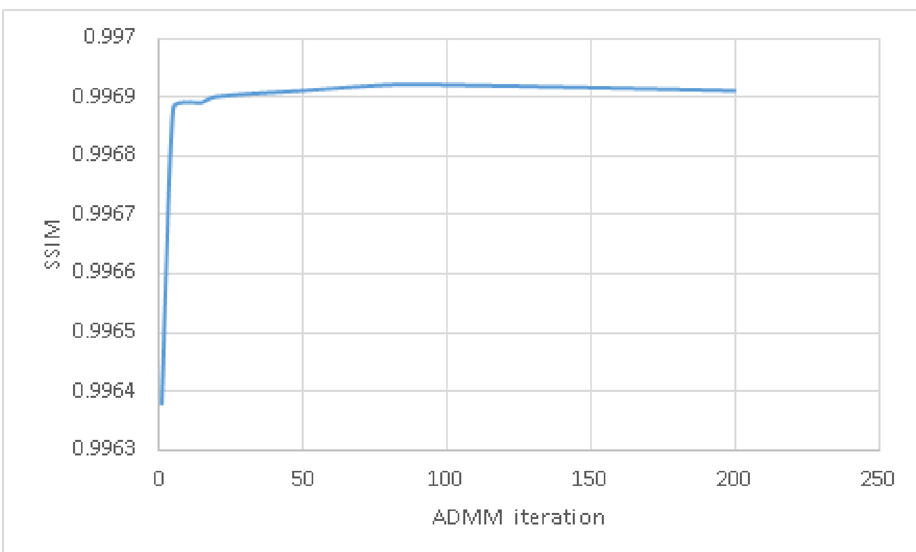

Figure 2: Convergence of SSIM index versus the iteration number in the simulated experiment using Washington DC mall data

5 Conclusion

In this project, we propose a new inpainting algorithm for HSIs with a convex regularizer ϕ, and then transform the inpainting problem into dual space to solve it with ADMM. With ADMM, we can divide the problem into several subproblems without losing equivalence, which can reduce the computational complexity. With convex self-similarity regularizer, we are guaranteed to have provable convergence and global optimal solutions. Therefore, we always pursue convex function in optimization problems. Not only in mathematical base, we also conduct experiment on simulated data to show the feasibility of this algorithm. Although the type of stripes might be complex, the proposed method is still able to reconstruct the HSI from the incomplete observed data.

References

[1] Chia-Hsiang Lin, et al., "An explicit and scene-adapted definition of convex self-similarity prior with application to unsupervised Sentinel-2 super-resolution," *IEEE Trans. Geoscience and Remote Sensing*, 2020

[2] Lina Zhuang, et al., "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, March 2018.