The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Nonlocal Patch Based t-SVD for Image Inpainting: Algorithm and Error Analysis

Liangchen Song,[12] Bo Du,[1*] Lefei Zhang,[1*] Liangpei Zhang,[2] Jia Wu,[3] Xuelong Li,[4]
[1]School of Computer Science, Wuhan University, Wuhan 430079, P. R. China.
[2] State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing,
Wuhan University, Wuhan 430079, P. R. China.
[3]Department of Computing, Macquarie University, Sydney, NSW 2109, Australia.
[4]Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, P. R. China.
{liangchensong,remoteking,zhanglefei,zlp62}@whu.edu.cn, jia.wu@mq.edu.au, xuelong_li@opt.ac.cn

Abstract

In this paper, we propose a novel image inpainting framework consisting of an interpolation step and a low-rank tensor completion step. More specifically, we first initial the image with triangulation-based linear interpolation, and then we find similar patches for each missing-entry centered patch. Treating a group of patch matrices as a tensor, we employ the recently proposed effective t-SVD tensor completion algorithm with a warm start strategy to inpaint it. We observe that the interpolation step is such a rough initialization that the similar patch we found may not exactly match with the reference, so we name the problem as Patch Mismatch and analyse the error caused by it thoroughly. Our theoretical analysis shows that the error caused by Patch Mismatch can be decomposed into two components, one of which can be bounded by a reasonable assumption named local patch similarity, and another part is lower than that using matrix. Experiments on real images verify our method's superiority to the state-of-the-art inpainting methods.

Introduction

Image inpainting (Bertalmio et al. 2000) aims to recover missing values in images, and in general, the applications can be classified into two groups by two motivations: the first tries to remove unwanted objects and uses known background of the image to fill in a big missing hole (Criminisi, Perez, and Toyama 2004), and the inpainting result is usually qualitatively evaluated by human visual judgments; the second deals with small scratches or random missing pixels mainly produced during the acquisition process (Liu et al. 2013), where the goal is trying to restore the original image, and quantitative rules such as PSNR are also adopted to evaluate the inpainting result. In this paper, we focus on the second goal that is recovering a degraded image with random missing pixels.

Nearly all the inpainting methods are designed in the light of a key observation that the missing area contains repeated patterns of global geometrical structure and local smoothness of color transforms. Guided by the assumption, the filling-in process is carried out through employing particular mathematical models. Briefly speaking, there are mainly three categories of mature inpainting framework: partial differential equations (PDE) based schemes, exemplar-based schemes and low-rank schemes.

All of the three schemes have their own limitation, while the design philosophies inside them are illuminating. PDE-based schemes (Bertalmio, Bertozzi, and Sapiro 2001) draw an analogy between local smoothness of images and physical phenomenon like liquid diffusion. The simulated diffusion produces a piecewise smoothness but hard to preserve the texture and thus introduces the blur. Exemplar-based schemes (He and Sun 2012) copy the best matching patch from known area to synthesize the background texture and it is usually used to remove objects. Despite the fact that exemplar-based schemes are not suitable for restoring random missing pixels due to the difficulty of finding a complete patch, the nonlocal manner inside the schemes are still heuristic. Low-rank schemes (Liu et al. 2013) stem from matrix and tensor completion which are first used in compressive sensing problems, thus low-rank based inpainting methods are theoretical convincing due to the well-developed guaranteed matrix/tensor completion algorithms. However, some existing low-rank based methods suppose the whole image to be low-rank. And other approaches such as (Dong et al. 2014) which imposes low-rank property on a group of patches lost the theoretical guarantee of recovery results.

Drawing advantages from the schemes mentioned above, we propose a novel inpainting framework consisting of two steps. Since image patch extraction is unattainable when the image contains random missing values, the first step of our framework is simply interpolating intensity values for missing pixels, which is supposed to be faster than PDE based methods, yet producing roughly the same results. Then for each patch we find a set of similar patches to form a third-order tensor, which is inspired by the classic nonlocal denoising methods and exemplar-based inpainting methods. After that tensor is formed, we adopt a recently proposed tensor completion method namely t-SVD (Zhang et al. 2014). It is noteworthy that the similar patches are not

[*]Corresponding authors. This work is supported by the National Natural Science Foundation of China under Grants 41431175, 61471274, 61771349, and 61711530239.

exactly the most accurate match due to the blur effect introduced during interpolation step, which we refer to as Patch Mismatch problem. Fortunately, the error caused by Patch Mismatch can be reduced by t-SVD and t-product system we adopted. In summary, our contributions can be summarized as follows:

- We propose an inpainting framework incorporating interpolation and low-rank tensor completion, which leverage the merits of popular inpainting methods. In the proposed framework, we first estimate an initial image with interpolation and for each patch we group the similar candidates to form a tensor. Then we employ t-SVD and ADMM (Alternating Direction Method of Multipliers) with a warm start strategy to complete the tensor.
- Based on the observation that natural images are piecewise smooth, we introduce a reasonable assumption that if a patch is small enough the patch will contain nearly the same information as its surrounding patches. And we describe the assumption with formal mathematical language, which may be useful for analysing other interpretable image processing techniques.
- We analysis the error caused by Patch Mismatch thoroughly, and we prove that with reasonable assumptions the error bound is lower than performing matrix completion thanks to the algebra properties of t-product system.

Related Work

Tensor Completion. Nowadays many data processing problems are facing high dimensions and multiple classes (Liu, Tsang, and Müller 2017) (Liu and Tsang 2017). Tensor, also known as multi-dimensional array, is a good model to mining the patterns inside complex data, thus has been studied in many areas (Tao et al. 2006)(Tao et al. 2007)(Zhang et al. 2015). When a low-rank constraint is imposed to estimate a tensor with missing entries, it is rare to use the basic definition from multilinear algebra named CANDECOMP/PARAFAC (CP) rank, because CP rank can be neither determined nor approximated by a convex or differentiable function, which is quite different from matrix. To avoid manually specifying the rank, a Bayesian approach (Zhao, Zhang, and Cichocki 2015) combines CP rank with a probability model and imposes sparsity on decomposition factors. Following the idea of dealing with decomposition factors, other decomposition formulas such as Tucker decomposition is also employed (Filipovi and Juki 2015) (Chen, Hsu, and Liao 2014). Besides, the low-rank constraint can also be imposed on other aspects of tensors. For example, (Gandy, Recht, and Yamada 2011) and (Liu et al. 2013) use a linear combination of the unfolded matrix along each mode to guarantee the tensor is low-rank, which needs the coefficients determined by users thus not always produce a stable result. Recently, (Braman 2010), (Kilmer and Martin 2011) and (Kilmer et al. 2013) propose a algebraic framework for tensor namely t-product, in which we can obtain tensor singular value decomposition (t-SVD). By defining new operations among the tensor, matrix and vector, they prove that a free module is constructed. We adopt t-product system in this paper to complete our grouped patches to ensure a solid theoretical analysis.

Low-rank Based Inpainting. A majority of existing low-rank based inpainting methods rely on the low-rank assumption of an entire image, which is sometimes impractical. And the problem has been elaborated in an inpainting review paper (Guillemot and Meur 2014). To overcome the shortcoming, one feasible solution is to impose a regularization of local details on images. (Li, Ye, and Xu 2017) and (Ji et al. 2016) integrate total variation into low-rank completion problems by adding regularization terms into the objective function. However, their methods need to determine beforehand a crucial parameter which balances the low-rank term and total-variation term. Another approach is to exploit the nonlocal sparsity by patch grouping and then a low-rank approximation, and such idea has been studied by (Dong et al. 2014) and (Feng et al. 2016). (Dong et al. 2014) first construct a group of similar patches and then use the logdet$(\cdot)$ as a replacement for the matrix rank. (Feng et al. 2016) make better use of the structured sparsity of similar patches by treating the patches as a tensor, and employing Schatten p-norm as a nonconvex relaxation for the tensor rank. However, both methods estimate an initial image using a standard compressive sensing method, which is slow and ad hoc. Our proposed method advance further by not only formulating a solid theoretical foundation but also providing intensive analysis on real data sets.

Preliminaries

In this paper, scalars are denoted by lowercase letters, e.g., a. Vectors are denoted by lowercase letters with a vector arrow above, e.g., $\vec{a}$. Matrices are denoted by capital letters, e.g., A. Higher-order tensors are denoted by boldface Euler script letters, e.g., $\mathcal{A}$. For a third-order tensor $\mathcal{A} \in \mathbb{R}^{r\times s\times t}$, the ith frontal slice is denoted by A_i. In terms of MATLAB indexing notation, we have $A_i = \mathcal{A}(:,:,i)$. For each entry, we denote $\mathcal{A}(i,j,k) = \mathcal{A}_{ijk}$. And in t-product system, the circulant or block-circulant matrices are used. Specifically, if $\vec{a} = [a_1, a_2, \cdots, a_n]^T$, then

$$\operatorname{circ}(\vec{a}) = \begin{bmatrix} a_1 & a_n & a_{n-1} & \cdots & a_2 \\ a_2 & a_1 & a_n & \cdots & a_3 \\ a_3 & a_2 & a_1 & \cdots & a_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n-1} & a_{n-2} & \cdots & a_1 \end{bmatrix}$$

And the unfold command rearranges the frontal slices of $\mathcal{A}$:

$$\operatorname{unfold}(\mathcal{A}) = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix}, \quad \operatorname{fold}(\operatorname{unfold}(\mathcal{A})) = \mathcal{A}$$

Now we introduce t-product system. At first the multiplication between vectors is defined and then a commutative ring with unity is constructed.

Definition 1. *Let $\vec{a}, \vec{b} \in \mathbb{R}^n$. then $\vec{a} \odot \vec{b} \equiv \operatorname{circ}(\vec{a})\vec{b}$.*

Theorem 1. *$(\mathbb{R}^n, +, \odot)$ is a commutative ring with unity, where $+$ denotes the usual addition of vectors.*

In this paper, this ring is denoted with $\mathbb{K}_n$ just as in (Braman 2010). Then a unitary module is constructed.

Definition 2. *Let $\mathbb{R}^{n\times n}_{\mathbb{K}_n}$ denote the space of $n \times n$ matrices with real entries, together with scalars taken from $\mathbb{K}_n$ and with scalar multiplication $\circ$ defined as follows:*

$$\textit{For} \quad \vec{a} \in \mathbb{K}_n, X \in \mathbb{R}^{n\times n}, \;\; \vec{a} \circ X \equiv X\text{circ}(\vec{a})$$

Theorem 2. *$\mathbb{R}^{n\times n}_{\mathbb{K}_n}$ is a free module over $\mathbb{K}_n$.*

So in $\mathbb{R}^{n\times n}_{\mathbb{K}_n}$ we are able to discuss generating set, which corresponds to the basis in vector space. Moreover, we can define rank on a group of matrices according to module homomorphism in $\mathbb{R}^{n\times n}_{\mathbb{K}_n}$. Also, the multiplication between tensor and matrix is by circulant matrix.

Definition 3 (t-product). *Let $\mathcal{A} \in \mathbb{R}^{r\times s\times t}$ and $\mathcal{B} \in \mathbb{R}^{s\times p\times t}$. Then the t-product $\mathcal{A} * \mathcal{B}$ is a $r \times p \times t$ tensor:*

$$\mathcal{A} * \mathcal{B} = \text{fold}(\text{circ}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B})) \tag{1}$$

And we consider a matrix $X \in \mathbb{R}^{n\times n}$ as a tensor oriented as a single lateral slice: $\mathcal{X}_{i1j} = X(i,j)$ for $i, j = 1, \cdots, n$. So

$$\mathcal{A} * X = \text{fold}(\text{circ}(\mathcal{A}) \cdot \text{unfold}(\mathcal{X})) \tag{2}$$

Being a slightly different from nuclear norm of matrix, the tensor nuclear norm of $\mathcal{A}$ is denoted by the sum of the singular values of all the frontal slice of $\hat{\mathcal{A}}$, and $\hat{\mathcal{A}}$ is obtained by taking the Fourier transform along the third dimension of $\mathcal{A}$. We use $\|\mathcal{A}\|_*$ to denote the nuclear norm of $\mathcal{A}$. Then it is known (Zhang et al. 2014) that

$$\|\mathcal{A}\|_* = \|\text{blkdiag}(\hat{\mathcal{A}})\|_* \tag{3}$$

where blkdiag$(\hat{\mathcal{A}})$ is a block diagonal matrix.

Proposed Framework

In this section, we propose our inpainting framework. We first estimate the image with a interpolation step, then for each patch we find the most similar candidate patches according to their Euclidean distance and group them as a tensor. For each tensor we inpaint them with t-SVD based ADMM iterations which are solved with a warm start. After describing our framework, we analyse the potential inpainting error bound and show that our algorithm can achieve a lower bound thanks to the t-product system.

Inpainting Algorithm

Interpolation. Given a gray image $A \in \mathbb{R}^{r\times s}$ with Ω indicating observations, we inpaint the image with triangulation-based linear interpolation. Specifically, for each missing entry P, we determine where the triangle in Ω that P lies and three apexes of the triangle are denoted by P^1, P^2, P^3. First solve u, v by

$$\begin{cases} P_x = P_x^1 + uP_x^2 + vP_x^3 \\ P_y = P_y^1 + uP_y^2 + vP_y^3 \end{cases} \tag{4}$$

where P_x, P_y are coordinates of the pixel, and then the intensity of P is determined by

$$P = P^1 + uP^2 + vP^3 \tag{5}$$

We denote the image after interpolation as $\hat{A}$ and the missing indices as $\bar{\Omega}$.

ADMM Iterations. After the interpolation step, for each point $P_i \in \bar{\Omega}$, denote $\Psi_{P_i} \in \mathbb{R}^{n\times n}$ as the patch at position P_i. If P_i is so close to the edge that a $n \times n$ patch is impossible, then just use the patch near the edge. For example, if $(P_i)_x < n, n < (P_i)_y < s - n$, then use the patch centered at $(n, (P_i)_y)$. Then we find a set of similar patches of Ψ_{P_i} in $\hat{A}$ by Euclidean distance, i.e., $\|\Psi_{P_i} - \Psi_Q\|_F$ where Ψ_Q is a candidate patch. Specifically, we select the $T-1$ most similar patches in $\{\Psi_Q : |(P_i)_x - Q_x| < N, |(P_i)_y - Q_y| < N\}$ where T is predefined size of the patch group and N is the searching radius. After this patch grouping step, we obtain a set of patches from $\hat{A}$, i.e., a set of matrix. Let Ψ_{P_i} be the first matrix in the group and we rearrange the matrices to a tensor $\mathcal{M}_i$ where $\mathcal{M}_i(:,:,k)(k = 1, ..., T)$ is the kth patch in that group. Also, we take out the patches in Ω to get a tensor indicating known pixels of $\mathcal{M}_i$, and denote it with Ω'. Then for each $\mathcal{M}_i$ we complete it through an optimization problem (Zhang et al. 2014)

$$\begin{aligned} \min_{\mathcal{X}} \quad & \|\mathcal{X}\|_* \\ \text{s.t.} \quad & \mathcal{P}_{\Omega'}(\mathcal{X}) = \mathcal{P}_{\Omega'}(\mathcal{M}_i) \end{aligned} \tag{6}$$

where $\mathcal{P}_{\Omega'}$ is the orthogonal projector onto the span of tensors vanishing outside of Ω'. Let $\Theta = \{\mathcal{X} : \mathcal{P}_{\Omega'}(\mathcal{X}) = \mathcal{P}_{\Omega'}(\mathcal{M}_i)\}$, then problem (6) can be rewritten in ADMM form as

$$\begin{aligned} \min_{\mathcal{X}} \quad & \|\mathcal{X}\|_* + \mathbf{1}_\Theta(\mathcal{Z}) \\ \text{s.t.} \quad & \mathcal{X} = \mathcal{Z} \end{aligned} \tag{7}$$

where $\mathbf{1}_\Theta(\cdot)$ is the indicator function of set Θ. So the augmented Lagrangian is

$$L_\rho(\mathcal{X}, \mathcal{Z}, \mathcal{Q}) = \|\mathcal{X}\|_* + \mathbf{1}_\Theta(\mathcal{Z}) + \frac{1}{\rho}\|\mathcal{X} - \mathcal{Z} + \mathcal{Q}\|_F^2 \tag{8}$$

Then (8) is solved by the following recursion

$$\begin{cases} \mathcal{X}^{k+1} = \arg\min\limits_{\mathcal{X}} \frac{1}{\rho}\|\mathcal{X}\|_* + \frac{1}{2}\|\mathcal{X} - \mathcal{Z}^k + \mathcal{Q}^k\|_F^2 \\ \mathcal{Z}^{k+1} = \arg\min\limits_{\mathcal{Z}\in\Theta} \|\mathcal{Z} - (\mathcal{X}^{k+1} + \mathcal{Q}^k)\|_F^2 \\ \mathcal{Q}^{k+1} = \mathcal{Q}^k + (\mathcal{X}^{k+1} - \mathcal{Z}^{k+1}) \end{cases} \tag{9}$$

where ρ is a parameter controlling the convergence. In (9), the solution to $\mathcal{X}$-update is given by singular value thresholding (Zhang et al. 2014). And $\mathcal{Z}$-update is least-squares projection onto the constraint.

Warm Start Strategy. Since we interpolate the image A and get $\hat{A}$, so $\mathcal{P}_{\bar{\Omega}'}(\mathcal{M}_i)$ is a good initial guess to start the iteration (9), which means $\mathcal{X}^0 = \mathcal{M}_i$. In (Dong et al. 2014), they also estimate the image at first and then just simply set $\mathcal{Q}^0 = 0$, which lacks theoretical motivations and may be not

Algorithm 1: Nonlocal and low-rank tensor based image inpainting

```
Input: an incomplete image matrix $A \in \mathbb{R}^{r\times s}$, index
       of known entries $\Omega$, patch size $n$, search range
       $N$, patch group size $T$, parameters
       $\rho^0, K, \epsilon, \mu_1, \mu_2$ and $\tau$
Initialize $A$ with triangulation-based linear
  interpolation and the result is $\hat{A}$;
for each point $P_i$ in $\bar{\Omega}$(missing area) do
    Take out patch $\Psi_{P_i}$ centered at $P_i$ from $\hat{A}$;
    Find $T-1$ nearest patches of $\Psi_{P_i}$ in $\hat{A}$ by
      Euclidean distance;
    Rearrange the patches as a tensor
      $\mathcal{M}_i \in \mathbb{R}^{n\times n\times T}$;
    Set $\mathcal{X}^0 = \mathcal{M}_i$ and compute $\mathcal{Q}^0$ by (19);
    for $k = 1$ to $K$ do
        Update $\mathcal{X}^k, \mathcal{Z}^k, \mathcal{Q}^k$ by (9);
        Update $\rho^k$ by (23);
        if $\|r^k\|_2 < \epsilon$ and $\|s^k\|_2 < \epsilon$ then
            break;
        end
    end
end
Output: the image $A$ after completion
```

the best choice. In this paper, we initialize the dual variable $\mathcal{Q}^0$ that satisfies dual feasibility as much as possible. Recall that the necessary and sufficient optimally conditions for (8) are primal feasibility,

$$\mathcal{X}^* - \mathcal{Z}^* = 0 \tag{10}$$

and dual feasibility,

$$0 \in \partial\|\mathcal{X}^*\|_* + \mathcal{Q}^* \tag{11}$$

$$0 \in \partial\mathbf{1}_\Theta(\mathcal{Z}^*) - \mathcal{Q}^* \tag{12}$$

where ∂ denotes the subdifferential operator. Note that $\mathbf{1}_\Theta(\mathcal{Z})$ is not continuous, so we only need to find a $\mathcal{Q}^0$ to satisfy (11) as much as possible. More specifically, for our initial points $\mathcal{X}^0, \mathcal{Q}^0$, we have

$$0 \in \partial\|\mathcal{X}^0\|_* + \mathcal{Q}^0 \tag{13}$$

Now we consider how to compute the subdifferential in (13). Recall the SVD of $M \in \mathbb{R}^{n_1\times n_2}$ is

$$M = \sum_k \sigma_k u_k v_k^* \tag{14}$$

where $\sigma_k \geqslant 0$ are the singular values, and the singular vectors $u_k \in \mathbb{R}^{n_1}$ and $v_k \in \mathbb{R}^{n_2}$ are two sets of orthonormal vectors. Define matrix P_U, P_V, E as

$$P_U = \sum_k u_k u_k^*, \quad P_V = \sum_k v_k v_k^*, \quad E = \sum_k u_k v_k^*. \tag{15}$$

Then the subdifferential is (Watson 1992; Lewis 2003)

$$\begin{aligned}\partial\|M\|_* = \{E + W : & W \in \mathbb{R}^{n_1\times n_2}, P_U W = 0,\\ & W P_V = 0, \|W\| \leqslant 1\}\end{aligned} \tag{16}$$

where $\|\cdot\|$ is the spectral norm, which is dual to nuclear norm. As for the subdifferential of tensor nuclear norm, let $\mathcal{A} \in \mathbb{R}^{r\times s\times t}$ and by Equation (3) we have

$$\begin{aligned}\partial\|\text{blkdiag}(\hat{\mathcal{A}})\|_* = \{E + W : & W \in \mathbb{R}^{rt\times st}, P_U W = 0,\\ & W P_V = 0, \|W\| \leqslant 1\}\end{aligned} \tag{17}$$

where E, P_U, P_V are defined by the SVD decomposition of blkdiag$(\hat{\mathcal{A}})$ and (15). Note that E is also a block diagonal matrix, so we can rearrange E as a tensor $\mathcal{E}$ with blkdiag$(\mathcal{E}) = E$. If we set $W = 0$, it is obvious that $E \in \partial\|\text{blkdiag}(\hat{\mathcal{A}})\|_*$. Then rearrange (17) as a third order tensor, we have

$$\mathcal{E} \in \partial\|\mathcal{A}\|_* \tag{18}$$

Now we have our initial guess of $\mathcal{Q}^0$,

$$\mathcal{Q} = -\mathcal{E} \tag{19}$$

then (13) is satisfied with our $\mathcal{Q}^0$.

Note that there is another important penalty parameter ρ controling the convergence of our ADMM iterations. A common trick to improve the convergence is determining ρ varies by iterations and thus can balance the primal and dual residual (He, Yang, and Wang 2000; Wang and Liao 2001). More specifically, define primal residual as

$$r^k = \mathcal{X}^k - \mathcal{Z}^k \tag{20}$$

and dual residual as

$$s^{k+1} = \rho(\mathcal{Z}^{k+1} - \mathcal{Z}^k), \tag{21}$$

then the update scheme of ρ is

$$\rho^{k+1} = \begin{cases}\tau\rho^k & \text{if } \|r^k\|_2 > \mu\|s^k\|_2\\ \rho^k/\tau & \text{if } \|s^k\|_2 > \mu\|r^k\|_2\\ \rho^k & \text{otherwise}\end{cases} \tag{22}$$

where $\mu > 1, \tau > 1$. Since a large value of ρ means a large penalty for violations of primal feasibility, the dynamic ρ update equations (22) try to keep $\|r^k\|_2$ and $\|s^k\|_2$ within a factor of μ and thus both of the residual norms converge to zero. In our algorithm, we also take into consideration that the patch tensor may not be low-rank since the patterns in an image are not always highly repetitive. Since we have a good initial guess, when $\|s^k\|_2$ is much larger than $\|r^k\|_2$, it indicates that the patch tensor may not be low-rank. So at this situation we keep ρ unchanged to avoid overfitting. Specifically, we updata ρ with

$$\rho^{k+1} = \begin{cases}\tau\rho^k & \text{if } \|r^k\|_2 > \mu_1\|s^k\|_2\\ \rho^k/\tau & \text{if } \|s^k\|_2 > \mu_1\|r^k\|_2, \|s^k\|_2 < \mu_2\|r^k\|_2\\ \rho^k & \text{otherwise}\end{cases} \tag{23}$$

where μ_1, μ_2, τ are parameters.

The overall algorithm is summarized as Algoritm 1.

Error Analysis

As described in the previous section, the first interpolation step leads to a noticeable blurring effect. In this part we discuss the error caused by blurred edges.

Patch Mismatch. To clearly illustrate why Patch Mismatch matters, we perform our inpainting algorithm on an image with only black diagonal strips. As shown in Figure 1(a), white points in the middle of the image mean missing entries, and the images from left to right are $A, A_\Omega, \hat{A}$ in our algorithm, respectively. With missing rate being 20%, an interpolation step is performed on it and we get the right image on which edges are vague. And the best similar candidate of the red patch in $\hat{A}$ is the yellow patch, but we can see from Figure 1(b) that they do not actually match with each other. Such problem in our nonlocal algorithm is called "Patch Mismatch".

Local Patch Similarity. We build our error analysis on a key observation which we refer to as local patch similarity. When a patch slides on the image with a short distance and then we have a new patch, obviously the new patch have a large overlapping region with the old patch. Since the image is piecewise smooth, it is reasonable to assume that in most cases the difference of non-overlapping region between old and new patch is small. Before describing the assumption mathematically, we first define two operators on patches, which represent the sliding action on the image. In Figure 1(c), we show that the blue patch, denoted Ψ_b, is the real candidate, and the displacement from Ψ_b to the yellow patch Ψ_y is $\vec{v}$. In fact, Ψ_b moves along $\vec{v}$ on image and it becomes Ψ_y. So we define this movement as $F_{\vec{v}}(\cdot)$ and we have $F_{\vec{v}}(\Psi_b) = \Psi_y$. Then seeing that $\vec{v}$ is a two dimensional vector, we may let $\vec{v} = (x, y)$, where x means moving down x pixels in vertical direction and y means moving left y pixels in horizontal direction and negative x, y means moving up and right respectively. And another operator on patch is defied as $G_{\vec{v}}(\Psi) = P^x \Psi P^y$, where P is a permutation matrix and

$$P = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Then we are able to adjust lines and columns of Ψ_y by $G_{\vec{v}}(\cdot)$ so that the overlapping region between Ψ_b and Ψ_y has the same index in the matrix and then matrix norm can be used to evaluate the difference of non-overlapping areas. Now the local patch similarity assumption can be mathematically defined.

Definition 4. *An area of image is said to be locally patch similar if for any patch Ψ with certain size in this area, when the displacement vector is smaller than a threshold K, i.e., $\|\vec{v}\| < K$, there exists a constant c such that*

$$\|G_{\vec{v}}(\Psi) - F_{\vec{v}}(\Psi)\| \leqslant c\|\vec{v}\| \tag{24}$$

where K only depends on the size of patch and c only depends on the type of norm.

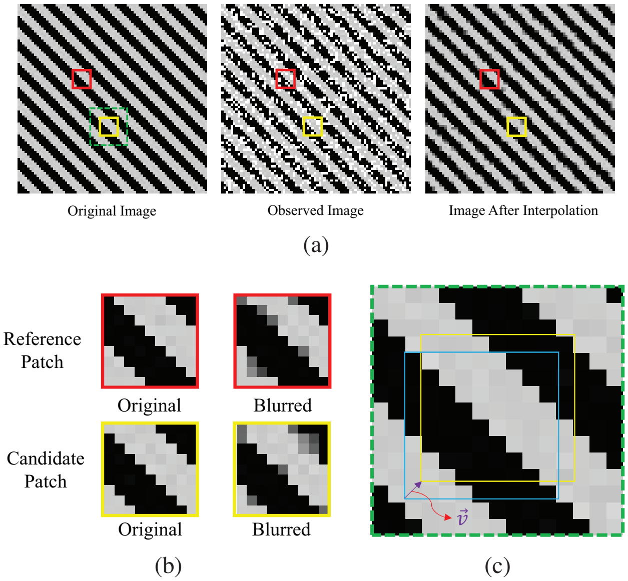

Figure 1: Illustration of the Patch Mismatch problem. (a) The red patch is a reference patch and the yellow patch is selected as candidate according to the image after interpolation. (b) Details about red patch and yellow patch. The yellow patch is the best similar patch in blurred image, but not in original image. (c) The blue patch in original is the best matching. So there is a slight deviation of the candidate in our algorithm and the displacement vector is denoted by $\vec{v}$.

Lower Bound. In this part, we consider the error caused by Patch Mismatch and then we prove that our algorithm has lower error bound than matrix based completion methods. Recall that the tensor $\mathcal{M}$ in our algorithm is composed of a group of patch matrices. Without loss of generality, we consider the error bound when the rank of the recovered $\mathcal{M}$ is 1. (Because when the rank is higher than 1, we just need to repeat the analysis of all bases in the group.) So now the patch matrices are actually an exactly same patch with different missing entries, and in this part for simplification we denote the reference patch by Ψ_Ω and candidate patches by $\Psi_{\Omega_1}, \cdots, \Psi_{\Omega_T}$ where the subscripts means different observations. So the ideal similar patch group should be

$$\mathcal{M} = \text{fold}([\Psi_\Omega, \Psi_{\Omega_1}, \cdots, \Psi_{\Omega_T}]) \tag{25}$$

Unfortunately the real group is affected by Patch Mismatch. In our analysis, we simply let only Ψ_{Ω_1} be affected since we just need to multiply this error by $T - 1$ times if all of the candidates are affected. Then the real similar patch group becomes

$$\mathcal{M}' = \text{fold}([\Psi_\Omega, F_{\vec{v}}(\Psi_{\Omega_1}), \cdots, \Psi_{\Omega_T}]) \tag{26}$$

We denote the recovery result of $\mathcal{M}$ and $\mathcal{M}'$ are Ψ and Ψ' respectively. Also, since we are using t-product, now the front slices of $\mathcal{M}, \mathcal{M}'$ are matrices from $\mathbb{R}^{n\times n}_{\mathbb{K}_n}$. We shall now discuss the bound and for simplicity Frobenius norm is employed. In $\mathbb{R}^{n\times n}_{\mathbb{K}_n}$ let the linear coefficient of $F_{\vec{v}}(\Psi_{\Omega_1})$ be

$\vec{a}$, so have

$$\begin{aligned}\|\Psi - \Psi'\|_F &\leqslant \| \left(\vec{a} \circ F_{\vec{v}}(\Psi_{\Omega_1}) - \Psi\right)_{\Omega_1} \|_F \\ &\leqslant \|\vec{a} \circ F_{\vec{v}}(\Psi) - \Psi\|_F\end{aligned} \tag{27}$$

The first inequality in (27) holds because $\|\Psi - \Psi'\|$ reaches a maximum when all of the wrong information in $F_{\vec{v}}(\Psi_{\Omega_1})$ are used to complete Ψ_Ω. Note that the subscript Ω_1 means that the indices not in Ω_1 are zeros, so the second inequality holds. Then

$$\begin{aligned}&\|\vec{a}\circ F_{\vec{v}}(\Psi) - \Psi\|_F \\ &\quad\leqslant \|\vec{a} \circ F_{\vec{v}}(\Psi) - \vec{a} \circ G_{\vec{v}}(\Psi) + \vec{a} \circ G_{\vec{v}}(\Psi) - \Psi\|_F \\ &\quad\leqslant \|\vec{a} \circ F_{\vec{v}}(\Psi) - \vec{a} \circ G_{\vec{v}}(\Psi)\|_F + \|\vec{a} \circ G_{\vec{v}}(\Psi) - \Psi\|_F\end{aligned} \tag{28}$$

Now we first focus on the second item of (28) and we need another important assumption on image to show the priority of our algorithm. We assume that the change of Ψ is small when the patch is moved in the vertical direction. This is not a strict assumption since it is natural that an image has vertical structures and we will prove that in these area our algorithm is guaranteed to perform better. Mathematically, the assumption means $\|P^x\Psi - \Psi\|_F$ is small and we show the detailed results as the following proposition:

Proposition 1. *If all the entries of $P^x\Psi - \Psi$ satisfy $|2(P^x\Psi - \Psi)_{ij}| < |(P^x\Psi(P^y - I))_{ij}|$, then there exists $\vec{a} \in \mathbb{R}^n$, such that $\|\vec{a} \circ G_{\vec{v}}(\Psi) - \Psi\| < \|G_{\vec{v}}(\Psi) - \Psi\|$.*

Proof. Let $G_{\vec{v}}(\Psi) = P^x \Psi P^y$, then

$$\begin{aligned}\|G_{\vec{v}}(\Psi) - \Psi\|_F &= \|P^x\Psi P^y - \Psi\|_F \\ &= \|P^x\Psi(P^y - I) + (P^x - I)\Psi\|_F\end{aligned} \tag{29}$$

Denote $A \triangleq P^x\Psi(P^y - I), B \triangleq (P^x - I)\Psi$, so

$$\begin{aligned}\|G_{\vec{v}}(\Psi) - \Psi\|_F &= \text{tr}(A^TA + 2A^TB + B^TB) \\ &= \text{tr}(A^TA + 2A^TB) + \text{tr}(B^TB)\end{aligned} \tag{30}$$

And $\vec{a} \circ G_{\vec{v}}(\Psi) = G_{\vec{v}}(\Psi) \cdot \text{circ}(\vec{a})$. If we set the $ny - 1$th element in $\vec{a}$ is 1 and other elements are 0, then it is easy to verify that $\vec{a} = (P^y)^{-1}$, so

$$\begin{aligned}\|\vec{a} \circ G_{\vec{v}}(\Psi) - \Psi\|_F &= \|P^x\Psi P^y(P^y)^{-1} - \Psi\|_F \\ &= \|(P^x - I)\Psi\|_F = \text{tr}(B^TB)\end{aligned} \tag{31}$$

If we denote each entry of A, B as a_{ij}, b_{ij} respectively, then we have $|2b_{ij}| < |a_{ij}|$, so $a_{ij} + 2b_{ij}$ has the same sign as a_{ij}, thus $\text{tr}(A^T(A + 2B)) > 0$. □

Let us emphasize that $\mathbb{R}^{n\times n}_{\mathbb{K}_n}$ is a module, so

$$\vec{a} \circ F_{\vec{v}}(\Psi) - \vec{a} \circ G_{\vec{v}}(\Psi) = \vec{a} \circ (F_{\vec{v}}(\Psi) - G_{\vec{v}}(\Psi)) \tag{32}$$

According to the proof of Proposition 1, $\text{circ}(\vec{a})$ is an orthogonal matrix, then

$$\|\vec{a} \circ (F_{\vec{v}}(\Psi) - G_{\vec{v}}(\Psi))\|_F = \|F_{\vec{v}}(\Psi) - G_{\vec{v}}(\Psi)\|_F \tag{33}$$

So by (28), the first item is bounded by our proposed local patch similarity, and the second item is smaller under our reasonable assumptions when t-product employed.

From the above theoretically analysis, we can see that with the assumption of the local patch similarity, and as long as an image has vertical structures, our t-product based method is guaranteed to have a lower bound than matrix based methods.

(a) Baboon (b) Barbara (c) Facade (d) Lena

Figure 2: RGB-color images used in our experiment.

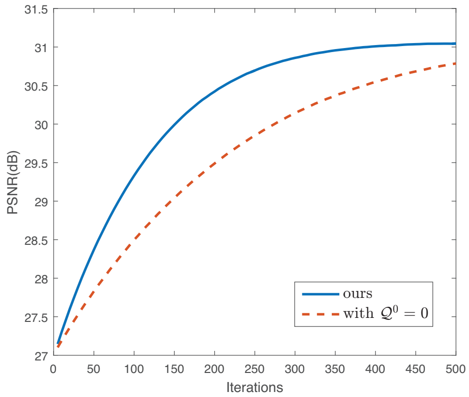

Figure 3: Comparison of convergence speed on grayscale *Facade* image with 70% random missing pixels. The solid line denotes our proposed warm start trick and the dashed line denotes the vanilla version used in (Dong et al. 2014).

Experiments

In this section, we first evaluate the performance of our proposed warm start strategy and then present extensive inpainting results of nature images. The recovered results are evaluated with peak signal-to-noise ratio (PSNR), which is widely used in image processing problems.

Parameter Setting

In the following experiments, we set the patch size as $n = 7$, and search radius as $N = 5n = 35$. And the patch group size is $T = 6$, so $\mathcal{M}_i \in \mathbb{R}^{7\times7\times5}$. Parameters controlling the convergence of iterations are $\rho^0 = 1, K = 100, \epsilon = 10^{-4}, \mu_1 = 10, \mu_2 = 100$ and $\tau = 2$.

Color Image Inpainting

Although our method is designed for grayscale image, the tensor completion methods target at inpainting color image which is a third order tensor. So we perform our algorithm on each color channel separately. Four 256×256 benchmark RGB images are used in our experiments and the images are shown in Figure 2.

Evaluation of Warm Start trick. We first investigate the effectiveness of our proposed warm start trick for ADMM iterations. Specifically, we compare our methods with another warm start trick, which only initializes $\mathcal{X}^0$ by $\mathcal{M}$ and set the dual variable $\mathcal{Q}$ being zero and is the scheme used in (Dong et al. 2014) and (Feng et al. 2016). We conduct the two algorithms on gray scale *Facade* image with different iterations.

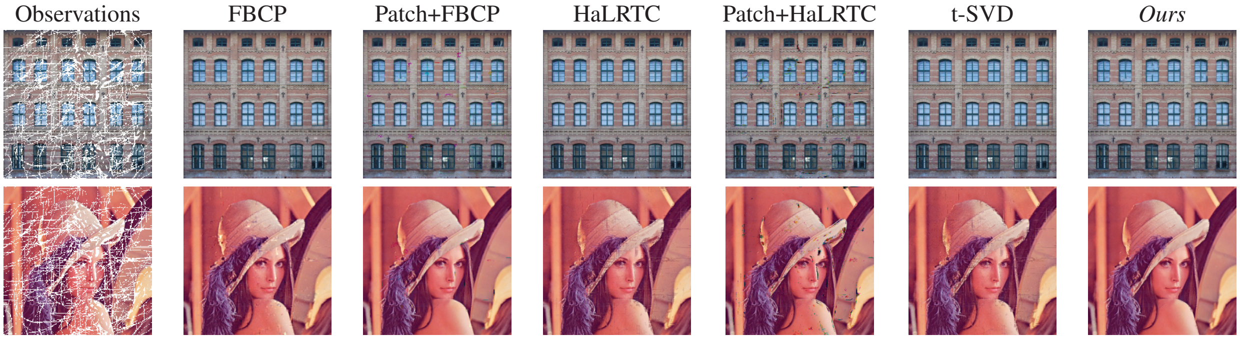

Figure 4: Visual effects of inpainting results with simulated scratches.

Table 1: PSNR of inpainting results with different missing rates on *Lena* image.

Missing Rate	0.8	0.6	0.4	0.2
Ours	**27.28**	**30.50**	**33.60**	**37.39**
t-SVD	20.84	25.30	29.41	34.14
FBCP	23.51	27.84	31.71	36.16
Patch+FBCP	21.65	30.05	32.61	36.93
HaLRTC	20.93	25.66	29.88	34.66
Patch+HaLRTC	8.64	17.29	28.67	35.77

Table 2: PSNR of inpainting results on different image with 50% random missing pixels.

Image	Baboon	Barbara	Facade	Lena
Ours	**25.40**	**30.79**	28.69	**31.85**
t-SVD	23.03	26.63	30.83	27.28
FBCP	23.76	28.51	29.67	29.70
Patch+FBCP	25.03	28.88	28.19	30.98
HaLRTC	23.53	27.11	**31.10**	27.71
Patch+HaLRTC	21.21	23.49	23.43	23.64

The reason why we choose *Facade* image is the global repetitive structures inside it. And we transfer the RGB image to grayscale since our interpolation steps only works for matrices. The comparison of convergence speed is shown in Figure 3, from which we can see that our warm start indeed improves the convergence speed and lead to a better solution. The faster convergence speed may result from our initialization of $\mathcal{Q}$, and the higher PSNR may result from our novel ρ-update equations.

Inpainting Results with Different Missing Rates. We select the comparison algorithms in two aspects: (1) we choose three effective low-rank tensor completion algorithms for image inpainting including HaLRTC, t-SVD and FBCP; (2) since our algorithm consists of two steps, we replace our t-SVD step with the other two tensor completion methods mentioned above. Global inpainting methods are performed on the whole RGB image, and patch based methods are performed on each color channel separately. The above global methods are performed with default parameters. And in order to ensure the convergence, patch based methods are performed with a little different settings: the weight vector α in HaLRTC is $[0.01, 0.01, 1]$; the max rank in FBCP of each grouped patch tensor is the threshold T.

First, we inpaint *Lena* image with randomly missing pixels and inpainting results with different missing rates are shown in Table 1. From Table 1, one can see the following facts. Firstly, our method outperforms the others apparently. Secondly, the patch based versions sometimes are even worse than those using the global completion algorithms, and the reason may be that Patch Mismatch effect leads to serious inpainting errors when the missing rate is high.

Inpainting Results on Different Images. Table 2 shows the inpainting results with different images. Also, the pixels of images are randomly missing with rate 50%. From Table 2, we can see that images such as *Baboon* with more repeated structural details are recovered better since our algorithm benefits from the nonlocal property. And the global image inpainting algorithms perform better on those images with global structures such as *Facade*.

Visual Effects. In Figure 4, we show the inpainted images of the algorithms mentioned above with simulated scratches. As is shown in the figure, we can see that both Patch+FBCP and Patch+HaLRTC have unwanted strange lines and blocks in the results, while this problem does not exist in our algorithm. The reason may be that these places do not have similar patches so the grouped patch tensors are not low-rank. And our results benefit from the ρ-update schema so the existence of similar patches has only a slight impact on our results.

Conclusions

In this paper, we present a image inpainting framework based on a triangulation-based linear interpolation step and a low-rank tensor completion step with t-SVD. Since our interpolation step provides only one good initial guess of variables for the following ADMM iterations, we design a warm start strategy to initialize dual variable effectively. After proposing our framework, we analysis the potential error which we refer to as Patch Mismatch and then we prove that with reasonable assumptions our algorithm has a lower bound than matrix based methods thanks to the t-product system. In future works, we will investigate the error bounds of other tensor completion methods and do more experiments including results on other images and running time comparison.

References

Bertalmio, M.; Bertozzi, A. L.; and Sapiro, G. 2001. Navier-stokes, fluid dynamics, and image and video inpainting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, I–I. IEEE.

Bertalmio, M.; Sapiro, G.; Caselles, V.; and Ballester, C. 2000. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, 417–424. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

Braman, K. 2010. Third-order tensors as linear operators on a space of matrices. *Linear Algebra and its Applications* 433(7):1241–1253.

Chen, Y. L.; Hsu, C. T.; and Liao, H. Y. M. 2014. Simultaneous tensor decomposition and completion using factor priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3):577–591.

Criminisi, A.; Perez, P.; and Toyama, K. 2004. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing* 13(9):1200–1212.

Dong, W.; Shi, G.; Li, X.; Ma, Y.; and Huang, F. 2014. Compressive sensing via nonlocal low-rank regularization. *IEEE Transactions on Image Processing* 23(8):3618–3632.

Feng, L.; Sun, H.; Sun, Q.; and Xia, G. 2016. Compressive sensing via nonlocal low-rank tensor regularization. *Neurocomputing* 216:45 – 60.

Filipovi, M., and Juki, A. 2015. Tucker factorization with missing data with application to low- n -rank tensor completion. *Multidimensional Systems & Signal Processing* 26(3):1–16.

Gandy, S.; Recht, B.; and Yamada, I. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27(2):025010.

Guillemot, C., and Meur, O. L. 2014. Image inpainting : Overview and recent advances. *IEEE Signal Processing Magazine* 31(1):127–144.

He, K., and Sun, J. 2012. Statistics of patch offsets for image completion. In *Computer Vision–ECCV 2012*. Springer. 16–29.

He, B.; Yang, H.; and Wang, S. 2000. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and Applications* 106(2):337–356.

Ji, T.-Y.; Huang, T.-Z.; Zhao, X.-L.; Ma, T.-H.; and Liu, G. 2016. Tensor completion using total variation and low-rank matrix factorization. *Information Sciences* 326:243 – 257.

Kilmer, M. E., and Martin, C. D. 2011. Factorization strategies for third-order tensors. *Linear Algebra and its Applications* 435(3):641–658.

Kilmer, M. E.; Braman, K.; Hao, N.; and Hoover, R. C. 2013. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications* 34(1):148–172.

Lewis, A. S. 2003. The mathematics of eigenvalue optimization. *Mathematical Programming* 97(1-2):155–176.

Li, X.; Ye, Y.; and Xu, X. 2017. Low-rank tensor completion with total variation for visual data inpainting. In *AAAI*, 2210–2216.

Liu, W., and Tsang, I. W. 2017. Making decision trees feasible in ultrahigh feature and label dimensions. *Journal of Machine Learning Research* 18(81):1–36.

Liu, J.; Musialski, P.; Wonka, P.; and Ye, J. 2013. Tensor completion for estimating missing values in visual data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35(1):208–220.

Liu, W.; Tsang, I. W.; and Müller, K.-R. 2017. An easy-to-hard learning paradigm for multiple classes and multiple labels. *Journal of Machine Learning Research* 18(94):1–38.

Tao, D.; Tang, X.; Li, X.; and Wu, X. 2006. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(7):1088.

Tao, D.; Li, X.; Wu, X.; and Maybank, S. J. 2007. General tensor discriminant analysis and gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(10):1700–15.

Wang, S., and Liao, L. 2001. Decomposition method with a variable parameter for a class of monotone variational inequality problems. *Journal of Optimization Theory and Applications* 109(2):415–429.

Watson, G. A. 1992. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications* 170:33–45.

Zhang, Z.; Ely, G.; Aeron, S.; Hao, N.; and Kilmer, M. 2014. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *Computer Vision and Pattern Recognition*, 3842–3849.

Zhang, L.; Zhang, L.; Tao, D.; Huang, X.; and Du, B. 2015. Compression of hyperspectral remote sensing images by tensor approach. *Neurocomputing* 147(1):358–363.

Zhao, Q.; Zhang, L.; and Cichocki, A. 2015. Bayesian cp factorization of incomplete tensors with automatic rank determination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 37(9):1751–1763.