Image Inpainting by Patch Propagation Using Patch Sparsity

Zongben Xu and Jian Sun

***Abstract*—This paper introduces a novel examplar-based inpainting algorithm through investigating the sparsity of natural image patches. Two novel concepts of sparsity at the patch level are proposed for modeling the patch priority and patch representation, which are two crucial steps for patch propagation in the examplar-based inpainting approach. First, patch structure sparsity is designed to measure the confidence of a patch located at the image structure (e.g., the edge or corner) by the sparseness of its nonzero similarities to the neighboring patches. The patch with larger structure sparsity will be assigned higher priority for further inpainting. Second, it is assumed that the patch to be filled can be represented by the sparse linear combination of candidate patches under the local patch consistency constraint in a framework of sparse representation. Compared with the traditional examplar-based inpainting approach, structure sparsity enables better discrimination of structure and texture, and the patch sparse representation forces the newly inpainted regions to be sharp and consistent with the surrounding textures. Experiments on synthetic and natural images show the advantages of the proposed approach.**

***Index Terms*—Image inpainting, patch propagation, patch sparsity, sparse representation, texture synthesis.**

I. Introduction

The filling-in of missing region in an image, which is called image inpainting, is an important topic in the field of computer vision and image processing. Image inpainting has been widely investigated in the applications of digital effect (e.g., object removal), image restoration (e.g., scratch or text removal in photograph), image coding and transmission (e.g., recovery of the missing blocks), etc.

The most fundamental inpainting approach is the diffusion-based approach [1]–[3], in which the missing region is filled by diffusing the image information from the known region into the missing region at the pixel level. These algorithms are well founded on the theory of partial differential equation (PDE) and variational method. Bertalmio *et al.* [1] filled in holes by continuously propagating the isophote (i.e., lines of equal gray values) into the missing region. They further introduced the Navier–Strokes equation in fluid dynamics into the task of inpainting [2]. Chan and Shen [3] proposed a variational framework based on total variation (TV) to recover the missing information. Then a curvature-driven diffusion equation was proposed to realize the connectivity principle which does not hold in the TV model [4]. A joint interpolation of isophote directions and gray-levels was also designed to incorporate the principle of continuity in a variational framework [5]. Recently, image statistics learned from the natural images are applied to the task of image inpainting [6]–[8]. The diffusion-based inpainting algorithms have achieved convincingly excellent results for filling the nontextured or relatively smaller missing region. However, they tend to introduce smooth effect in the textured region or larger missing region.

The second category of approaches is the examplar-based inpainting algorithm. This approach propagates the image information from the known region into the missing region at the patch level. This idea stems from the texture synthesis technique proposed in [9], in which the texture is synthesized by sampling the best match patch from the known region. However, natural images are composed of structures and textures, in which the structures constitute the primal sketches of an image (e.g., the edges, corners, etc.) and the textures are image regions with homogenous patterns or feature statistics (including the flat patterns). Pure texture synthesis technique cannot handle the missing region with composite textures and structures. Bertalmio *et al.* [10] proposed to decompose the image into structure and texture layers, then inpaint the structure layer using diffusion-based method and texture layer using texture synthesis technique [9]. It overcomes the smooth effect of the diffusion-based inpainting algorithm; however, it is still hard to recover larger missing structures. Criminisi *et al.* [11] designed an examplar-based inpainting algorithm by propagating the known patches (i.e., examplars) into the missing patches gradually. To handle the missing region with composite textures and structures, patch priority is defined to encourage the filling-in of patches on the structure. Wu [12] proposed a cross-isophotes examplar-based inpainting algorithm, in which a cross-isophotes patch priority term was designed based on the analysis of anisotropic diffusion. Wong [13] proposed a nonlocal means approach for the examplar-based inpainting algorithm. The image patch is inferred by the nonlocal means of a set of candidate patches in the known region instead of a single best match patch. More examplar-based inpainting algorithms [14]–[16]were also proposed for image completion. Compared with the diffusion-based inpainting algorithm, the examplar-based inpainting algorithms have performed plausible results for inpainting the large missing region.

Manuscript received April 15, 2009; revised December 20, 2009; accepted December 20, 2009. First published February 02, 2010; current version published April 16, 2010. This work was supported by the National Key Fundamental Research Program (the 973 Program) of China under Grant 2007CB311002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark (Hong-Yuan) Liao.

The authors are with the School of Science, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (e-mail: zbxu@mail.xjtu.edu.cn; jiansun@mail.xjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2042098

Recently, image sparse representation is also introduced to the inpainting problem [17]– [21]. The basic idea of this approach is to represent image by sparse combination of an over-complete set of transforms (e.g., wavelet, contourlet, DCT, etc.), then the missing pixels are inferred by adaptively updating this sparse representation. Guleryuz *et al.* [18]–[20] proposed an image inpainting algorithm using adaptive sparse representation of image. Elad *et al.* [17] improved this approach by separating the image into cartoon and texture layers, and sparsely represented these two layers by two incoherent over-complete transforms. This approach can effectively fill in the region with composite textures and structures, especially in the application of missing block completion. However, similar to the diffusion-based approach, it may fail to recover structure or introduce smooth effect when filling large missing region.

This paper focuses on the examplar-based inpainting algorithm through patch propagation. The two basic procedures of patch propagation are patch selection and patch inpainting. In the patch selection, a patch on the missing region boundary with the highest priority is selected for further inpainting. The priority is defined to encourage the filling-in of patches on structure such that the structures are more quickly filled than the textures, then missing region with composite structures and textures can be better inpainted [11], [22]. Traditionally, the patch priority is defined based on the inner product between isophote direction and the normal direction of the missing region boundary [11], [22]. In the patch inpainting, the selected patch is inpainted by the candidate patches (i.e., examplars) in the known region. The approach in [11] and [22] utilizes the best match candidate patch to inpaint the selected patch. The approach in [13] uses a non-local means of the candidate patches for robust patch inpainting.

To better address the problems of patch selection and patch inpainting, two novel concepts of patch sparsity of natural image, i.e., patch structure sparsity and patch sparse representation, are proposed and applied to the examplar-based inpainting algorithm. First, we define a novel patch priority based on the sparseness of the patch's nonzero similarities to its neighboring patches. This sparseness is called ***structure sparsity*** in this paper. It is based on the observation that a patch on the structure has sparser nonzero similarities with its neighboring patches compared with the patch within a textured region. Compared with the priority defined on isophote, this definition can better distinguish the texture and structure, and be more robust to the orientation of the boundary of missing region.

Second, to inpaint a selected patch on the boundary of missing region, we use a sparse linear combination of examplars to infer the patch in a framework of sparse representation. This linear combination of patches are regularized by the sparseness prior (ℓ^0 regularization) on the combination coefficients. It means that only very few examplars contribute to the linear combination of patches with nonzero coefficients. This representation is called ***patch sparse representation*** in this paper. The patch sparse representation is also constrained by the local patch consistency constraint. This model extends the patch diversity by linear combination and preserves texture without introducing smooth effect by sparseness assumption. More importantly, the inpainted patches are more consistent with their surrounding textures or structures due to the local patch consistency constraint.

In summary, the structure sparsity and patch sparse representation at the patch level constitute the ***patch sparsity*** in this paper. The patch structure sparsity is inspired by the recent progress on the research of sparseness prior of natural image. The previous sparseness prior generally models the sparseness of image's nonzero features, e.g., gradients or filter responses. This kind of sparseness prior has been successfully applied to the image denoising [8], [23], super-resolution [24], [25], inpainting [6]–[8], deblurring [26], [27], and so on. The structure sparsity also models the sparsity of natural image. However, it models the sparseness of nonzero similarities of a patch with its neighboring patches instead of high-frequency features.

The patch sparse representation is inspired by the recent progress on sparse representation [28], which assumes that the image or signal is represented by the sparse linear combination of an over-complete library of bases or transforms under $\ell^p (0 \leq p < 2)$ sparseness regularization. This framework has been widely applied to image denoising [29], [30], edge detection [31], recognition [32], [33], super-resolution [34], texture synthesis [35], etc., and achieved state-of-the-art performance. In this work, the idea of sparse representation is introduced to the examplar-based inpainting algorithm under the assumption that the missing patch can be represented by the sparse linear combination of candidate patches. Then a novel constrained optimization model is designed for patch inpainting.

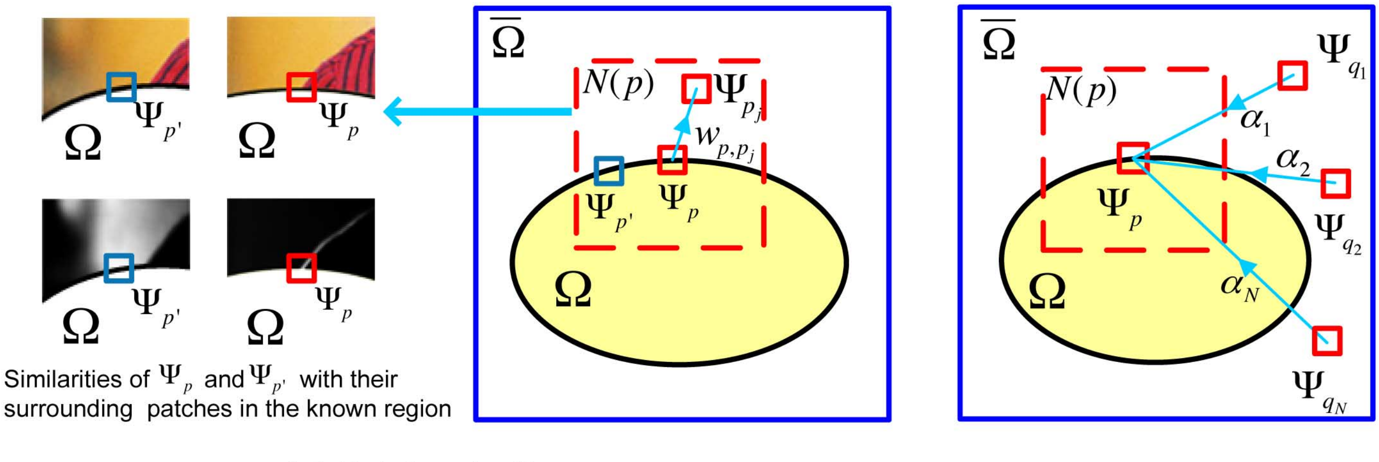
The paper is organized as follows. In Section II, an overview of the proposed examplar-based inpainting algorithm is presented. In Section III, the details of the inpainting algorithm, including the patch sparsity and patch sparse representation, are introduced. The experiments and comparisons with the previous algorithms are performed in the Section IV. Finally, we conclude this work in Section V.

II. Algorithm Overview

Given an image I with the missing region Ω and the known region $\overline{\Omega}$, the task of image inpainting is to fill in the target region (i.e., the missing region Ω) using the image information in the source region (i.e., the known region $\overline{\Omega}$). The boundary of the target region is denoted by $\partial\Omega$, which is called the fill-front in the examplar-based inpainting algorithm. We further denote Ψ_p as a patch centered at a pixel p.

The main procedures of the proposed examplar-based inpainting algorithm are illustrated in Fig. 1. This algorithm is based on patch propagation by inwardly propagating the image patches from the source region into the interior of the target region patch by patch. In each iteration of patch propagation, the algorithm is decomposed into two procedures: patch selection and patch inpainting.

In the procedure of patch selection, patch priority should be defined to encourage the filling-in of patches on the structure with higher priority. We define ***structure sparsity*** by measuring the sparseness of the similarities of a patch with its neighboring patches. Then patch priority is defined using the structure sparsity. In the example shown in Fig. 1(a), the patches Ψ_p and $\Psi_{p'}$ are centered at pixel p and p' which lie in the edge structure



(a) Patch selection

(b) Patch inpainting

Fig. 1. Image inpainting using patch sparsity. Ω is the missing region, $\overline{\Omega}$ is the known region, and $\partial\Omega$ is the fill-front of patch propagation. Iterate the following two steps until completion. (a) Patch selection: The patch Ψ_p on the fill-front with the larger sparseness of nonzero similarities to its neighboring patches is encouraged to be selected with higher priority. The top left of (a) shows two examples of known image parts surrounding patch Ψ_p and $\Psi_{p'}$, which locate at edge and flat texture region respectively. The bottom-left of (a) shows the similarities of Ψ_p and $\Psi_{p'}$ with their surrounding known patches, the higher brightness means larger similarity. Obviously, similarity map of Ψ_p is much sparser than that of $\Psi_{p'}$. (b) Patch inpainting: For the selected patch Ψ_p, sparse linear combination of candidate patches $\{\Psi_{q_1}, \Psi_{q_2}, \ldots, \Psi_{q_N}\}$ is used to infer the missing pixels in patch Ψ_p under the constraint of local patch consistency in $N(p)\bigcap\overline{\Omega}$.

and the flat texture region respectively. The left-down part of Fig. 1(a) shows the maps of their similarities with neighboring known patches. Obviously, the patch Ψ_p has sparser nonzero similarities; therefore, it has larger patch priority. The patch on the fill-front with the highest priority is selected to be inpainted firstly.

In the procedure of patch inpainting, the selected patch on the fill-front should be filled in. Instead of using a single best match examplar or a certain number of examplars in the known region to infer the missing patch, we assume that the selected patch on the fill-front is the sparse linear combination of the patches in the source region regularized by ℓ^0 sparseness prior. In the example shown in Fig. 1(b), the patch Ψ_p on the fill-front is inpainted by the sparse linear combination of candidate patches $\{\Psi_{q_k}\}_{k=1}^N$ weighted by coefficients $\{\alpha_k\}_{k=1}^N$, in which only very sparse nonzero elements exist. The neighboring patches in $N(p)\bigcap\overline{\Omega}$ are also used to constrain the appearance of patch Ψ_p by local patch consistency constraint.

In Section III, we will introduce the definitions of patch priority and patch sparse representation in detail.

III. Inpainting by Patch Sparsity

A. *Patch Priority Using Structure Sparsity*

The natural images are generally composed of structures and textures. A good definition of patch priority should be able to better distinguish the structures and textures, and also be robust to the orientation of the fill-front. In this paper, a novel definition of patch priority is proposed to meet these requirements. We now introduce the key component of our definition of patch priority, i.e., structure sparsity.

1) Structure Sparsity: The structure sparsity is defined to measure the confidence of a patch located at structure instead of texture. Structure sparsity is inspired by the following observations: structures are sparsely distributed in the image domain, e.g., the edges and corners are distributed as 1-D curves or 0-D points in the 2-D image domain. Nevertheless, the textures are distributed in 2-D sub-regions of the image domain, which are less sparsely distributed. On the other hand, for a certain patch, its neighboring patches with larger similarities are also distributed in the same structure or texture as the patch of interest. Therefore, we can model the confidence of structure for a patch by measuring the sparseness of its nonzero similarities to the neighboring patches. The patch with more sparsely distributed nonzero similarities are prone to be located at structure due to the high sparseness of structures.

Suppose Ψ_p is a patch on the fill-front $\partial\Omega$, its neighboring patch Ψ_{p_j} is defined as the patch that is in the known region and with the center p_j in the neighborhood of pixel p, i.e., p_j belongs to the set

$$N_s(p) = \{p_j : p_j \in N(p) \text{ and } \Psi_{p_j} \subset \overline{\Omega}\}. \tag{1}$$

$N(p)$ is a neighborhood window centered at p, which is set to be larger than the size of patch Ψ_p. Suppose P is a matrix to extract the missing pixels of Ψ_p, and $\bar{P}$ extracts the already known pixels of Ψ_p, then the similarity between Ψ_p and Ψ_{p_j} is defined as

$$w_{p,p_j} = \frac{1}{Z(p)}\exp\left(-\frac{d(\bar{P}\Psi_p, \bar{P}\Psi_{p_j})}{\sigma^2}\right) \tag{2}$$

where $d(\cdot,\cdot)$ measures the mean squared distance, $Z(p)$ is a normalization constant such that $\sum_{p_j\in N_s(p)} w_{p,p_j} = 1$, and σ is set to 5.0 in our implementation.

For the patch Ψ_p, we measure the sparseness of its similarities to the neighboring patches in region $N_s(p)$ by

$$\rho(p) = \|\vec{w}_p\|_{L_2}\sqrt{\frac{|N_s(p)|}{|N(p)|}} = \sqrt{\left[\sum_{p_j\in N_s(p)} w_{p,p_j}^2\right]\cdot\frac{|N_s(p)|}{|N(p)|}} \tag{3}$$

where $\vec{w}_p$ is the vector of elements $w_{p,p_j}(p_j \in N_s(p))$, and $|\cdot|$ means the number of elements. $|N(p)|$ is incorporated to restrict

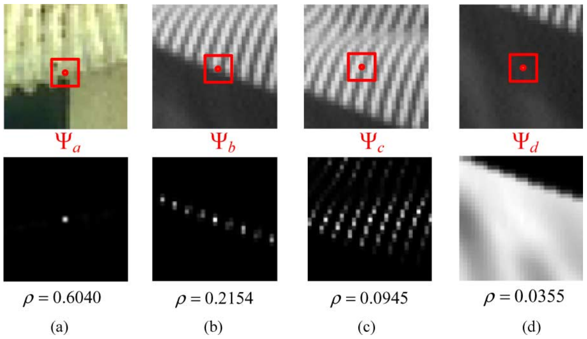


Fig. 2. Structure sparsity values of patches on corner, edge and textures. The first row shows the patches in rectangle and their surrounding regions. The similarities of patches Ψ_a, Ψ_b, Ψ_c, and Ψ_d with their neighboring patches are computed and shown in the second row. The patch Ψ_a and Ψ_b on corner and edge have the larger structure sparsity values compared with the patches Ψ_c and Ψ_d in texture regions. (a) Patch on corner. (b) Patch on edge. (c) Patch in texture. (d) Patch in flat region.

$\rho(p)$ in the interval $(0, 1]$, though its value is same for different patches. This definition embodies the fact that the larger sum of squared similarities in the larger region N_s means larger sparseness. In the following paragraph, we call the sparseness of patch similarities as the ***structure sparsity***.

For the definition of structure sparsity $\rho(p)$, we prove the following conclusion.

Theorem 1: The structure sparsity $\rho(p)$ for patch Ψ_p achieves the maximal value $\sqrt{|N_s(p)|/|N(p)|}$ if a single nonzero similarity exists, and it achieves the minimal value $\sqrt{1/|N(p)|}$ if all the similarities are same and equal to $1/|N_s(p)|$.

Proof: We observe that $\rho(p)$ consistently increases with respect to $W = \sum_{p_j \in N_s(p)} w_{p,p_j}^2$. To find the maximum and minimum values of $\rho(p)$, we only need to compute the maximum and minimum values of W.

Firstly, to find the minimum value of W, we minimize W under the normalization constraint $\sum_{p_j \in N_s(p)} w_{p,p_j} = 1$. This can be achieved by Lagrangian multiplier method, i.e., maximizing: $E(p) = -W + \lambda \left(\sum_{p_j \in N_s(p)} w_{p,p_j}^2 - 1 \right)$. It is easy to prove that W achieves minimal value when $w_{p,p_j} = 1/|N_s(p)|$ for each $p_j \in N_s(p)$.

Then we maximize W. Due to the fact that $0 \leq w_{p,p_j} \leq 1$, so $W \leq 1$. The equality holds when only a single w_{p,p_j} equals to 1, and all the other similarities equal to 0. So W achieves its maximal value 1 when only a single similarity is nonzero and equals to 1.

Finally, it is easy to derive the maximal and minimal values of $\rho(p)$ by inserting the maximum and minimum values of W into (3). ■

This theorem tells us that the structure sparsity achieves its maximum and minimum values when the patch similarities are distributed in the sparsest and smoothest fashion respectively, and the structure sparsity increases with respect to the sparseness of patch's nonzero similarities to its neighboring patches.

We now investigate how the structure sparsity measures the structure confidence for different types of patches in the natural images. For the patch on the 0-D corners [e.g., Fig. 2(a)], it is saliently distributed within the local region; therefore, it has the highest structure sparsity. Due to the sparsity of image edges, the patch on 1-D edge [e.g., Fig. 2(b)] has similar patches sparsely distributed along the same edge; therefore, they have higher structure sparsity. However, for the texture patches [e.g., Fig. 2(c) and (d)], they have similar patches in the 2-D local regions; therefore, they have smaller structure sparsity values. Under the guidance of structure sparsity, the patches located at structures (e.g., edges and corners) have higher priority for patch inpainting compared with the patches in texture regions.

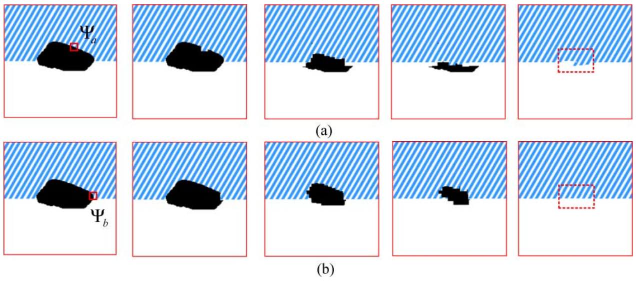


Fig. 3. Comparison of patch priority. (a) Process of inpainting using isophote-based priority [11], [22]. (b) Process of inpainting using structure sparsity based priority. The texture synthesis technique in [11] and [22] is used for both cases. Patch size and neighborhood window size are set to 11 × 11 and 51 × 51. The missing illusory edge is accurately recovered using structure sparsity-based priority; however, it is not perfectly recovered using isophote-based priority.

2) Patch Priority: The final patch priority is defined by multiplying the transformed structure sparsity term with patch confidence term: $P(p) = T_{[\zeta,1]}(\rho(p)) \cdot C(p)$. $C(p)$ is the confidence of patch Ψ_p, which specifies the reliability of color or intensity in the patch. It is same defined as in [11] and [22]: $C(p) = \sum_{q \in \Psi(p) \bigcap \overline{\Omega}} c(q)/|\Psi(p)|$, where $c(q)$ is the confidence of the color or intensity of pixel q, and initialized to 0 in the missing region or 1 in the known region. After each procedure of patch inpainting, the confidence of the newly filled pixels in the patch are updated by the confidence of the patch's central pixel. $T_{[\zeta,1]}$ is a linear transformation of $\rho(p)$ from its original interval $\left[\sqrt{1/|N(p)|}, \sqrt{|N_s(p)|/|N(p)|}\right]$ to the interval $[\zeta, 1]$, where ζ is set to 0.2. This transformation is necessary to make the structure sparsity varies in a comparable scale with $C(p)$. By multiplying these two terms in $P(p)$, the inpainting algorithm is encouraged to firstly inpaint the patch located at image structures (i.e., edges or corners) and with larger confidence of its colors or intensities, then the missing region with composite texture and structure can be more robustly inpainted [11], [22].

B. Comparison With the Isophote-Based Priority

Now we show that structure sparsity based priority is more robust to identify the structure than the isophote-based definition [11], [22], which uses the inner product of isophote direction and the normal direction of the fill-front.

Fig. 3 presents an example of inpainting for an image with composite textures and illusory edge. Fig. 3(a) shows the process of inpainting using isophote-based priority, and Fig. 3(b) shows the process of inpainting using structure sparsity based priority. The texture synthesis technique in [11] and [22] is incorporated for both cases. Using isophote-based priority, the patch at the top-right part of missing region has the larger priority because the isophote direction is nearly same to the orthogonal direction of the fill-front at its central pixel. For example, the patch Ψ_a in the texture region of the first image in Fig. 3(a) is with the highest priority, and the illusory edge is failed to be accurately recovered in the final result. However, structure sparsity based priority is able to robustly identify the structure regardless of the shape of fill-front. For example, the patch Ψ_b in Fig. 3(b) along structure is with the highest priority using structure sparsity based priority, and the missing region is inpainted perfectly in the final result.

C. Patch Inpainting Using Patch Sparse Representation

The patch Ψ_p on the fill-front with the highest patch priority is selected to be filled firstly. In the traditional examplar-based inpainting technique [11], [22], Ψ_p is filled by sampling the best match patch from the known region. Recently, a nonlocal means approach [13] is proposed to fill in patch by the nonlocal means of several top similar patches instead of a single best match patch. Due to multiple samples are utilized, it can more robustly estimate the missing information and produce better result. However, it tends to introduce smooth effect in the recovered image. In this work, we propose a novel model to inpaint patch by the sparse combination of multiple examplars in the framework of sparse representation. This method achieves sharp inpainting result by sparseness prior on the combination coefficients, and achieves consistent inpainting results with the surrounding textures by the constraints on the patch appearance in local neighborhood.

1) Patch Sparse Representation: Given the patch Ψ_p to be inpainted, a set of candidate patches $\{\Psi_q\}_{q=1}^N$ are sampled from the image source region, where N is the number of candidate patches for Ψ_p. Similar to the previous work [13], [36], the candidate patches are selected as the top N most similar patches, and the selection of N will be discussed in Section IV-E. Denote P as a matrix to extract the unknown pixels in patch Ψ_p. Basically, Ψ_p is approximated as the linear combination of $\{\Psi_q\}_{q=1}^N$, i.e.,

$$\hat{\Psi}_p = \sum_{q=1}^{N} \alpha_q \Psi_q. \tag{4}$$

Then the unknown pixels in patch Ψ_p is filled by the corresponding pixels in $\hat{\Psi}_p$, i.e.,

$$P\Psi_p = P\hat{\Psi}_p. \tag{5}$$

The combination coefficients $\vec{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ are inferred by minimizing a constrained optimization problem in the framework of sparse representation. Since we have assumed that the patch Ψ_p is the sparsest linear combination of $\{\Psi_q\}_{q=1}^N$, the objective of this constrained optimization problem is to minimize the ℓ^0 norm of $\vec{\alpha}$, i.e., the number of nonzero elements in vector $\vec{\alpha}$. Next we will introduce the constraints for the linear combination.

We now introduce the first type of constraint on the appearance of $\hat{\Psi}_p$, which is called local patch consistency constraint. Firstly, we constrain that the estimated patch $\hat{\Psi}_p$ should approximate the target patch Ψ_p over the already known pixels, i.e.,

$$\|\bar{P}\hat{\Psi}_p - \bar{P}\Psi_p\|^2 < \epsilon \tag{6}$$

where ϵ is a parameter to control the error tolerance of this approximation. Secondly, we further assume that the newly filled pixels in the estimated patch should be consistent with the neighboring patches in appearance, i.e.,

$$\beta \left\| P\hat{\Psi}_p - P \sum_{p_j \in N_s(p)} w_{p,p_j} \Psi_{p_j} \right\|^2 < \epsilon \tag{7}$$

where w_{p,p_j} is same defined as (2). This constraint measures the similarity between the estimated patch and the weighted mean of the neighboring patches over the missing pixels. β balances the strength of the constraints in (6) and (7). It is set to 0.25 in our implementation.

The local patch consistency constraint can be rewritten in a more compact formulation

$$\mathcal{E} = \|D\hat{\Psi}_p - \Psi_T\|^2 < \epsilon \tag{8}$$

where

$$D = \begin{bmatrix} \bar{P} \\ \sqrt{\beta} P \end{bmatrix} \text{ and } \Psi_T = \begin{bmatrix} \bar{P}\Psi_p \\ \sqrt{\beta} P \sum_{p_j \in N_s(p)} w_{p,p_j} \Psi_{p_j} \end{bmatrix}. \tag{9}$$

The second type of constraint is that the summation of the coefficients vector $\vec{\alpha}$ equals to one: $\sum_i^N \alpha_i = 1$. This constraint is widely used in the local linear embedding literature for invariance to transform when reconstructing the target patch Ψ_T from its neighboring candidate patches [36], [37]. Under this constraint, when only one element in $\vec{\alpha}$ is nonzero, the coefficient of that element must be one. Then the model will degrade to the same fashion as traditional examplar-based inpainting method [22], in which a single best match patch is selected for filling in the patch to be inpainted.

Finally, the linear combination coefficients $\vec{\alpha}$ can be inferred by optimizing the following constrained optimization problem:

$$\min\{\|\vec{\alpha}\|_0\} \quad \text{s.t. } \|D\hat{\Psi}_p - \Psi_T\|^2 < \epsilon \qquad \sum_i^N \alpha_i = 1. \tag{10}$$

This constrained optimization model can also be formulated as an energy minimization problem

$$\vec{\alpha}^* = \min_{\vec{\alpha}} \left\{ \gamma \|\vec{\alpha}\|_0 + \|D\hat{\Psi}_p - \Psi_T\|^2 + \eta \left\| \sum_i^N \alpha_i - 1 \right\|^2 \right\}. \tag{11}$$

It is equivalent to the constrained optimization problem in (10) when proper γ and η are selected.

2) Optimization Algorithm: Generally, the ℓ^0-norm regularized reconstruction model is hard to be solved due to its combinatorial nature. Matching pursuit (MP) or orthogonal matching pursuit (OMP) [38], [39] algorithm and basis pursuit (BP) [40] algorithm can efficiently retrieve the sparse representation and approximate the optimal solution in a greedy fashion. Another method for optimizing the ℓ^0-norm regularized model is to convexify the problem by ℓ^1-norm regularization. The ℓ^1-norm regularized reconstruction model is the well-known Lasso [41] in

Input: Ψ_p is the patch to be filled. $Q = \{\Psi_q\}_{q=1}^{N}$ are the top N most similar patches with Ψ_p (measured by SSD over the already filled pixels) collected from the source region $\bar{\Omega}$.

Initialization: Set $m = 0$, $S^m = \{\}$, $\epsilon = 5$.

Loop: Iterate the following operations:

- For each candidate patch $\Psi \in Q - S^m$, compute the coefficients $\vec{\alpha}$ for the linear combination of patch set $S^m \cup \{\Psi\}$ by Eqn. (15).
- Select the patch $\Psi_{q_{m+1}} \in Q - S^m$ with the minimal $\mathcal{E}$ in Eqn.(8) over the patch set $(S^m \bigcup \{\Psi_{q_{m+1}}\})$.
- If $\mathcal{E}$ decreases, then $S^{m+1} = S^m \bigcup \{\Psi_{q_{m+1}}\}$, $m = m+1$; When $\mathcal{E} < \epsilon$ or $\mathcal{E}$ increases, jump out of the loop and perform the next procedure.

Filling in patch: Filling in the missing pixels of Ψ_p by the corresponding pixels in the sparse linear combination $\sum_{i=1}^{m} \alpha_i \Psi_{q_i}$.

Fig. 4. Inpainting algorithm using patch sparse representation for a patch on the fill front.

the statistical literatures. In applications, due to the simplicity of OMP algorithm, it is widely used in image sparse representation, and applied to image denoising [29], [30], coding [42], edge detection [32], audio source separation [43], and so on.

For this optimization problem in (10), we propose a novel algorithm to derive the sparse linear combination coefficients in a greedy fashion. Similar to the Matching Pursuit Algorithm, we gradually select nonzero elements from the candidate set of patches $Q = \{\Psi_q\}_{q=1}^{N}$ step by step. Suppose we have selected m nonzero candidate patches in the step m (denoted as $S^m = \{\Psi_{q_1}, \Psi_{q_2}, \ldots \Psi_{q_m}\}$), so the sparse representation in this step is

$$\hat{\Psi}_p^m = \sum_{i=1}^{m} \alpha_i \Psi_{q_i}. \tag{12}$$

In the next step $m+1$, we select a new candidate patch $\Psi_{q_{m+1}}$ from the remaining candidate patches in $Q - S^m$. The patch with the best Local Patch Consistency $\mathcal{E}$ in (8) is selected as the newly selected nonzero element, i.e.,

$$\begin{aligned} \{\Psi_{q_{m+1}}\} &= \min_{\Psi \in Q - S^m} \{\mathcal{E}(\Psi_{q_1}, \Psi_{q_2}, \ldots, \Psi_{q_m}, \Psi)\} \\ &= \min_{\Psi \in Q - S^m} \left\{ \| D\left(\sum_{i=1}^{m} \alpha_i \cdot \Psi_{q_i} + \alpha_{m+1} \cdot \Psi \right) \right. \\ &\qquad \left. - \Psi_T \|^2 \right\}. \end{aligned} \tag{13}$$

For each candidate patch $\Psi \in Q - S^m$, the combination coefficients in (13) are derived by optimizing

$$\begin{aligned} \min_{\vec{\alpha}} \quad & \left\{ \left\| \left(\sum_{i=1}^{m} \alpha_i \cdot D\Psi_{q_i} + \alpha_{m+1} \cdot D\Psi \right) - \Psi_T \right\|^2 \right\} \\ \text{s.t.} \quad & \sum_{i=1}^{m+1} \alpha_i = 1. \end{aligned} \tag{14}$$

This optimization problem (14) is well studied in the literature of Locally Linear Embedding (LLE) [37] in manifold learning. We define the Gram matrix $G = (\Psi_T \mathbf{1}^T - X)^T (\Psi_T \mathbf{1}^T - X)$, where X is a matrix with columns $\{D\Psi_{q_1}, \ldots, D\Psi_{q_m}, D\Psi\}$, and $\mathbf{1}$ is a column vector of ones. Then it has a close form solution

$$\vec{\alpha} = \frac{G^{-1}\mathbf{1}}{\mathbf{1}^T G^{-1} \mathbf{1}}. \tag{15}$$

We iterate the procedure of selecting a patch in each step until the Local Patch Consistency Constraint (8) is satisfied or the value of $\mathcal{E}$ increases.

In summary, the algorithm for filling in the missing pixels in patch Ψ_p using patch sparse representation is listed in Fig. 4. The final inpainting algorithm based on patch sparsity is listed in Fig. 5. Fig. 6 presents three examples in which the top corner region of pyramid, the crossing structure of window frame and the curved missing structures are removed. As shown in Fig. 6(b)–(d), the missing regions are gradually completed by the proposed method, and Fig. 6(d) are the inpainting results of our method, in which the removed structures are successfully filled in. In Fig. 6(e), we also present the results of the most related algorithm in [22]. The structures in rectangles are not perfectly recovered compared with our results. The keys to the success of our method in completing the complex structures are that, first, the sparsity-based priority better controls the filling order of patches. Second, the patch sparse representation improves the generalization ability of examplars over the single best match examplar used in [22].

Fig. 7 gives an example to illustrate the number of nonzero coefficients for 200 patches in the missing region of the second example in Fig. 8. It is shown that only sparse number of candidate patches are selected adaptively due to the sparseness constraint on the coefficients. That is the key difference to the previous work that a single best match patch or the combination of constant number of patches are used. The quantitative advantage of this sparse representation over the previous methods will be shown in Section IV-B.

IV. Experiments and Comparisons

In this section, we test the proposed examplar-based patch propagation algorithm on a variety of natural images. We apply our algorithm to the applications of scratch/text removal, object removal and block completion. In these examples, we compare our algorithm with the previous diffusion-based, examplar-based, and sparsity-based inpainting algorithms. In the following

Input: Image I with the unknown (or target) region Ω.

Step 1: For each pixel p on the fill-front $\partial\Omega$, compute the similarities of patch Ψ_p with its neighboring patches $\{\Psi_{p_j}\}_{p_j \in N_s(p)}$ in the source region. Then compute the patch priority using structure sparsity.

Step 2: Iterate the following operations until the unknown region is completed:

- Select the patch Ψ with the highest patch priority on the fill-front.
- Inpaint the selected patch Ψ using patch sparse representation by the algorithm in Fig. 4.
- Update the fill-front, and the missing region Ω. For each newly-apparent pixel on the fill-front, compute its patch similarities with the neighboring patches and its patch priority.

Output: The inpainted image.

Fig. 5. Examplar-based inpainting algorithm by patch sparsity.

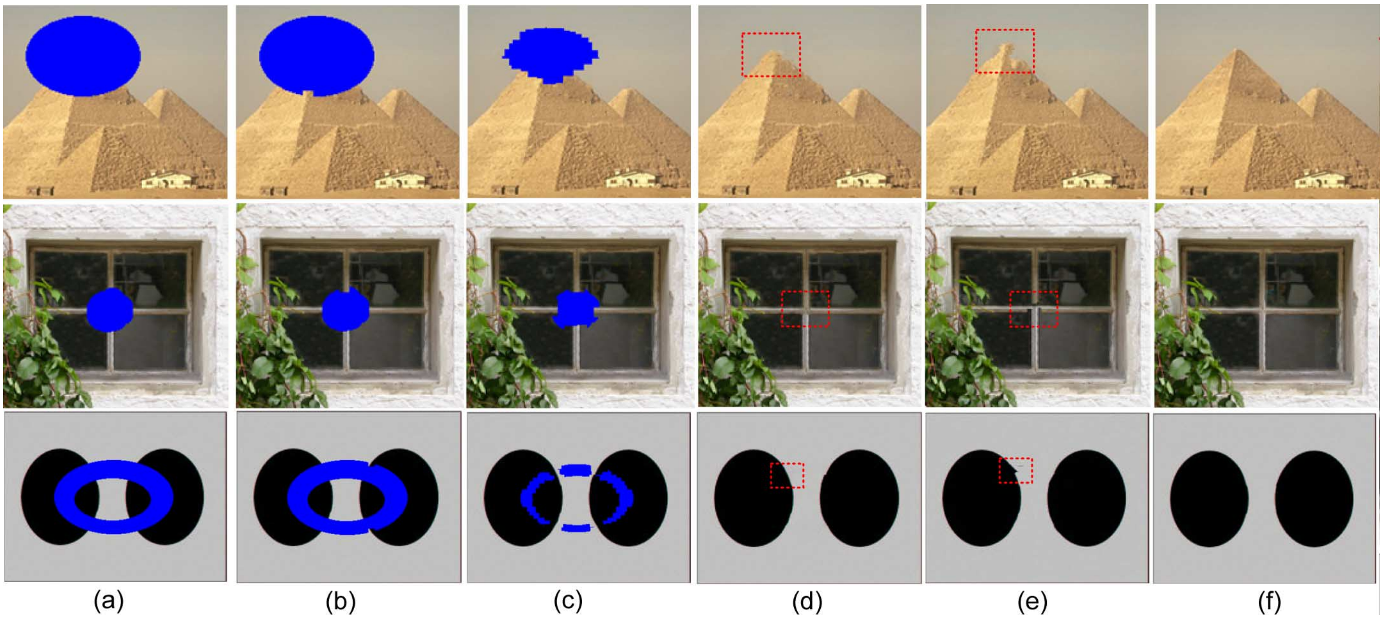


Fig. 6. Examples of inpainting. (a) Input images with unknown regions. (b)–(d) The inpainting process and the final results of our method. (e) The results of method in [22]. (f) Ground-truth images.

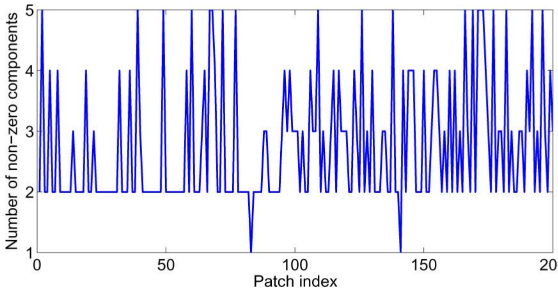


Fig. 7. Number of nonzero components for 200 patches in the second example of Fig. 8.

examples, if not specially stated, the size of patch is set to 7×7, the size of neighborhood (i.e., $N(p)$ around p in (1)) for computing patch similarities is set to 51×51, the error tolerance ϵ in (10) is set to 25 times of the number of pixels in a patch, and the number of candidate patches N is set to 25.

A. Experiments and Comparisons for Scratch and Text Removal

We now compare the proposed inpainting algorithm with the previous diffusion-based and examplar-based inpainting algorithms for scratch and text removal. Due to the over-smoothing effect of diffusion-based approach for texture inpainting, we select the simultanous texture and structure inpainting algorithm [10] as an improved version of the diffusion-based approach for comparison. The simultanous texture and structure inpainting algorithm performs inpainting in the texture layer and structure layer simutanously. In our implementation, the structure/texture decomposition method and the inpainting method in the structure layer are chosen as the same methods in the original paper [10]. However,

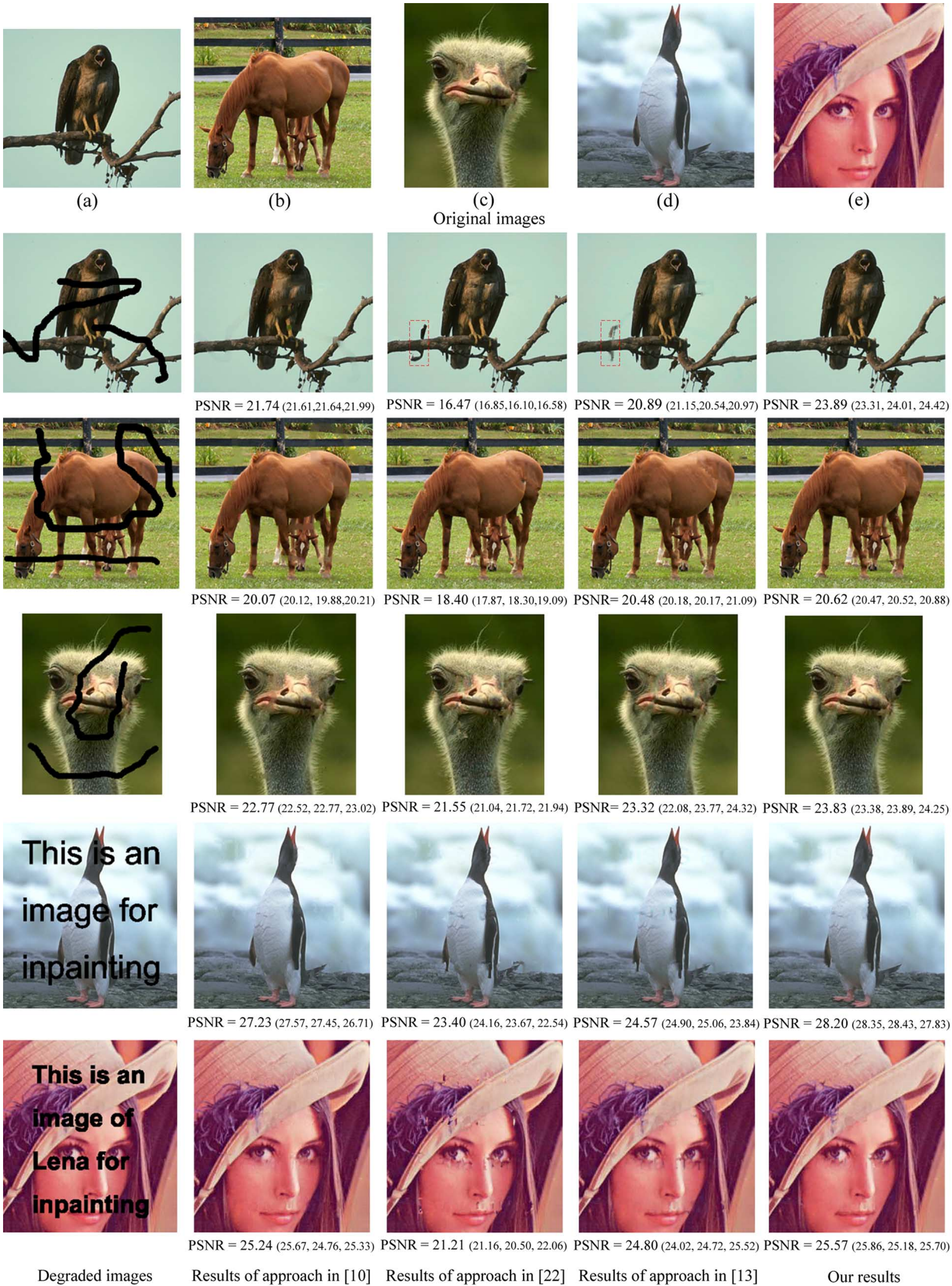

(a)
(b)
(c)
Original images
(d)
(e)
PSNR = 21.74 (21.61,21.64,21.99)
PSNR = 16.47 (16.85,16.10,16.58)
PSNR = 20.89 (21.15,20.54,20.97)
PSNR = 23.89 (23.31, 24.01, 24.42)
PSNR = 20.07 (20.12, 19.88,20.21)
PSNR = 18.40 (17.87, 18.30,19.09)
PSNR= 20.48 (20.18, 20.17, 21.09)
PSNR = 20.62 (20.47, 20.52, 20.88)
PSNR = 22.77 (22.52, 22.77, 23.02)
PSNR = 21.55 (21.04, 21.72, 21.94)
PSNR= 23.32 (22.08, 23.77, 24.32)
PSNR = 23.83 (23.38, 23.89, 24.25)
This is an image for inpainting
PSNR = 27.23 (27.57, 27.45, 26.71)
PSNR = 23.40 (24.16, 23.67, 22.54)
PSNR = 24.57 (24.90, 25.06, 23.84)
PSNR = 28.20 (28.35, 28.43, 27.83)
This is an image of Lena for inpainting
PSNR = 25.24 (25.67, 24.76, 25.33)
PSNR = 21.21 (21.16, 20.50, 22.06)
PSNR = 24.80 (24.02, 24.72, 25.52)
PSNR = 25.57 (25.86, 25.18, 25.70)
Degraded images
Results of approach in [10]
Results of approach in [22]
Results of approach in [13]
Our results

Fig. 8. Comparison for scratch and text removal. The first row shows five original images. In the remaining five rows, the first to the fifth columns show the degraded images, results of simultaneous texture and structure inpainting approach [10], the Criminisi's examplar-based approach [22], Wong's examplar-based approach [13], and our proposed approach. PSNR values of the inpainted image and its each color channel (shown in bracket for R, G, B channels) are presented for each result.

the texture layer is inpainted by Criminisi's examplar-based algorithm [11], which is better for texture inpainting than the texture systhesis method [9] used by the original paper [10]. We will also compare our algorithm with Criminisi's examplar-based al-

TABLE I
EFFECT OF PATCH SPARSITY ON INPAINTING PERFORMANCE MEASURED IN PSNR. THE EXAMPLES ARE FROM FIG. 8

Examples	*Wong*[13]	*Crim*[21]	*Crim_Spar1*	*Crim_Spar2*	*Spar_Crim*	*Spar*
(a)	20.89	16.47	21.67	23.48	18.22	**23.89**
(b)	20.48	18.40	19.92	20.31	18.81	**20.62**
(c)	23.32	21.55	22.63	22.65	21.69	**23.83**
(d)	24.57	23.40	23.37	26.12	25.64	**28.20**
(e)	24.80	21.21	23.50	25.07	23.31	**25.57**
mean	22.81	20.26	22.22	23.53	21.53	**24.42**

gorithm [11], [22] and Wong's examplar-based algorithm [13], which are most related to our work.

Fig. 8 presents five examples for scratch and text removal. The first row are the original nondegraded images. In the remaining rows, from the first to the fifth columns are the degraded images, results of simultanous texture and structure inpainting algorithm [10], examplar-based inpainting algorithm [11], [22], Wong's examplar-based inpainting algorithm [13], and the proposed algorithm. Peak signal-to-noise ratio (PSNR) between the inpainted images and the original images are measured for qualitative comparison. Furthermore, PSNR values in each color channel (R, G, B) are also presented in the brackets.

As shown in Fig. 8, the Criminisi's algorithm produces sharp inpainting results shown in the third column. However, due to the fact that only a single best match patch is used, some unpleasant artifacts are introduced in the results. For example, the unwanted structure appears within the red rectangle of Criminisi's result in the second row of Fig. 8. The Wong's algorithm produces more pleasant results because more candidate patches are combined. For example, the unwanted structure shown in the red rectangle is alleviated in the result of Wong's algorithm. The simultaneous texture and structure inpainting algorithm well recovers the texture and structure of the images, and achieves high PSNR values. However, it introduces blurring effect along structure caused by diffusion, which was also observed in [11]. For our proposed algorithm, the patch priority is defined more robustly, and the candidate patches are adaptively combined in the framework of sparse representation, it achieves sharp and consistently better inpainting results with the best PSNR values.

B. *Effect of Patch Sparsity on Inpainting Performance*

In this section, we quantitatively justify the improvement of inpainting performance caused by structure sparsity and patch sparse representation. To this end, we take the traditional Criminisi's examplar-based inpainting algorithm [22] as the baseline, then measure the performance improvement after replacing isophote-based priority [22] by structure sparsity based priority or(and) replacing the texture synthesis based patch inpainting [22] by the sparse representation based patch inpainting.

Table I presents the performances of six inpainting algorithms: Wong's examplar-based algorithm [13] (*Wong*), Crimimisi's examplar-based algorithm [22] (*Crim*), the approach using isophote-based priority and sparse representation based patch inpainting model in [44] (*Crim_Spar1*), the approach using isophote-based priority and sparse representation based patch inpainting model in (10) (*Crim_Spar2*), the approach using structure sparsity based priority and texture synthesis based patch inpainting (*Spar_Crim*), and the proposed inpainting algorithm using patch sparsity (*Spar*). It is shown that, the mean performances of *Spar_Crim* and *Crim_Spar2* gain 1.27 and 3.27 dB over the baseline respectively. This implies that the patch inpainting using sparse representation model in (10) contributes more to the performance improvement than the structure sparsity based priority. If updating both of patch priority and patch inpaining by the proposed patch sparsity, the performance of *Spar* gains 4.16 dB over the baseline. The results of Wong's algorithm [13] gain 2.55 dB in mean performance over the baseline which is lower than (*Crim_Spar2*). The inpainting model in [44] utilizes the isophote-based priority and the sparse representation model without local patch consistency constraint, so the performances of *Crim_Spar1* are significantly worse than the performance of *Crim_Spar2* and *Spar*.

C. *Experiments and Comparisons for Object Removal*

We now apply the proposed algorithm to inpaint the missing region after object removal. We also compare the proposed algorithm with the related examplar-based inpainting algorithms. See Fig. 9 for examples. The first and second columns are the original images and the degraded images. The third to the fifth columns are the results of Criminisi's algorithm [22], Wong's algorithm [13] and our proposed algorithm. In the results of Criminisi's algorithm, the inpainted patches are not always consistent with the surrounding textures, for example the texture of trees occurs in the texture of water in the first example, and texture of grass appears in the texture of rock in the third example. The Wong's algorithm uses several top best examplars to infer the unknown patch, so the results have less effect of patch inconsistency. However, it introduces smooth effect as shown in the results. As for the proposed algorithm, local patch consistency is constrained, so the inpainted patches are more consistent with the surrounding textures. In addition, the patch is inferred by the sparse combination of candidate patches, so the results have rare smooth effect in appearance.

In Fig. 10, we compare our algorithm with the semi-automatic inpainting algorithm [45] and the Criminisi's algorithm in [22] on the examples from [45]. The semi-automatic inpainting algorithm [45] resolves the structure ambiguity in the missing region by utilizing the human-labeled structures as the guidance, and achieves state-of-the-art results for completing the highly complex structures. In the first example, our method automatically completes the large missing region after removing the car, and the result is comparable to the result in [45], which uses human-labeled structures as guidance. In the second example, the structures hidden by the horse are reasonably recovered by the algorithm in [45] guided by the labeled structures. However,

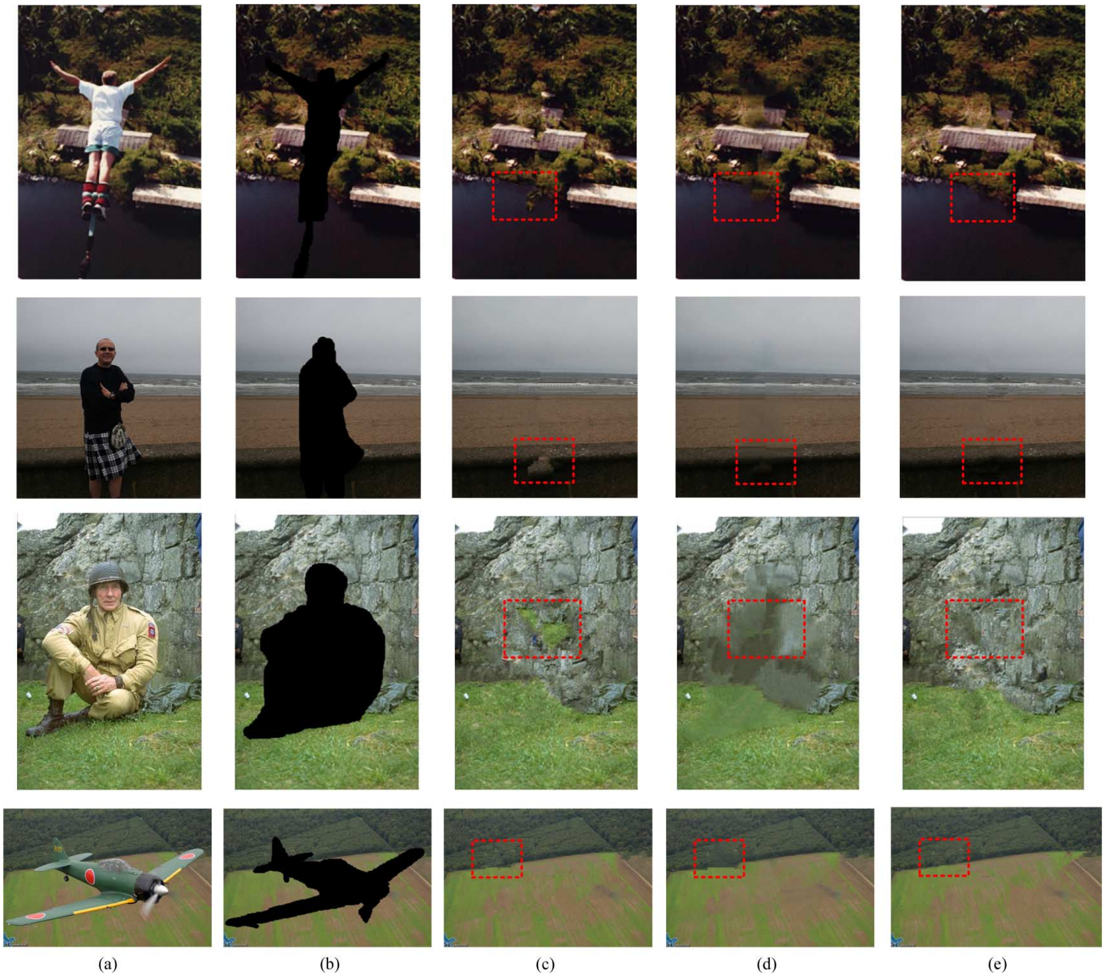


Fig. 9. Comparison for object removal (please refer to the color images in the electronic version for better comparison). (a) The original image. (b) The degraded image. (c)–(e) The results of Criminisi's examplar-based algorithm [22], Wong's examplar-based algorithm [13], and our examplar-based algorithm. Our algorithm achieves sharp inpainting results more consistent with the surrounding textures or structures.

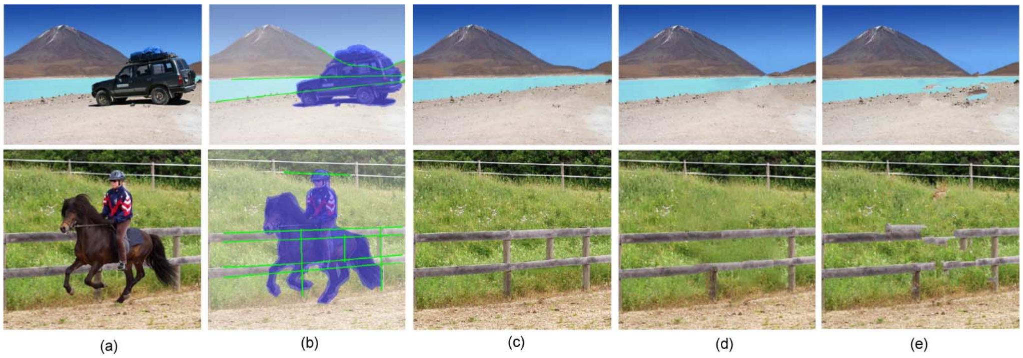


Fig. 10. Comparison on the examples in [45] (please refer to the color images in the electronic version for better comparison). (b), (c) Human labeled structures and the completion results from [45]. (a) Original images. (b) Unknown regions and human provided structures. (c) Results of structure propagation. (d) Our results. (e) Results of algorithm in [22].

our algorithm only recovers the horizontal structures and fails to recover the vertical fence structures which are totally hidden by the object. This shows a limitation of our algorithm, i.e., it cannot recover the missing structures without any structure cue in the known region. In both examples, our results are significantly better than the results of algorithm in [22], in which the completion results are not visually pleasant.

We also compare our method with the fragment-based method [16] on two challenging examples presented in [16], and the results are shown in Fig. 11. Both of the fragment-based method [16] and our proposed method achieve visually pleasant results in the first example. In the second example, the contour of apple in the result of [16] is blurry and not visually pleasant. However, our method recovers sharp and reasonable contour of apple compared with the original image before removal. The reason is that the sparse combination of patch examplars extends the representation ability of examplars and preserves the sharpness of structure at the same time.

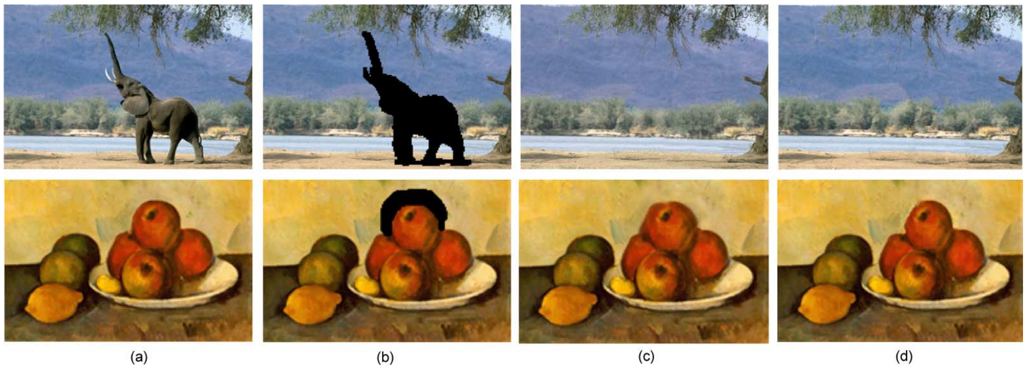


Fig. 11. Image completion results on the examples in [16] using fragment-based method and our method. (a) Input images. (b) Images with unknown region. (c) Results in [16]. (d) Our completion results.

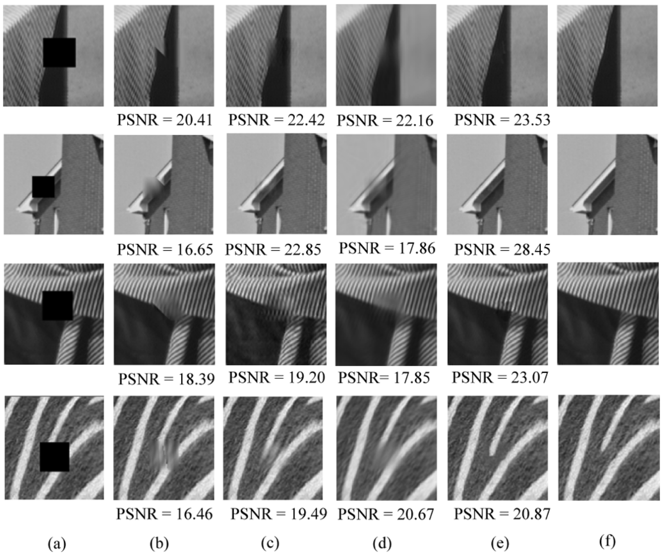


Fig. 12. Comparison with sparsity-based inpainting algorithms. (a) The input images. (b)–(e) The results of Guleryuz's algorithm [20], Elad's algorithm [17], Fadili's algorithm [21], and the proposed algorithm. (f) The ground-truth images.

D. Comparisons With the Sparsity-Based Inpainting Algorithms for Missing Block Completion

We also compare our algorithm with the other sparse representation based inpainting algorithms. Guleryuz [18]–[20] recovered the missing blocks by adaptively updating image sparse representation in the transform domain. Elad *et al.* [17] improved the approach by decomposing image into texture and structure layers [46], and recover each layer by sparse representation. Fadili *et al.* [21] proposed an expectation minimization (EM) based Bayesian model for inpainting using sparse representation. They have achieved excellent inpainiting results for completion of missing blocks, especially the blocks with size not larger than 16×16. For comparison, we downloaded the source codes of Guleryuz's algorithm,[1] Elad's and Fadili's algorithms[2] from their web page and run the algorithms on examples with large missing blocks. For the Guleryuz's algorithm, we use the Discrete Cosine Transform (DCT, with size of 32×32) as the basis for sparse representation. For the Elad's and Fadili algorithms, we run the codes of *MCALab* designed for image decomposition and inpainting. We set the dictionary of texture layer as the DCT with block size 32×32, and the dictionary of cartoon layer as the curvelet transform with 5 resolutions.

Fig. 12 shows four examples for missing block completion. The first column shows the input images, and the second to the fifth columns show the results of Guleryuz's algorithm, Elad's algorithm and Fadili's algorithm and the proposed algorithm. PSNR values are presented for performance comparison. All of these examples are with complicated missing structures, e.g., the structure of eaves in the second example, and the crossing of textures in the third and fourth examples. Generally, Elad's algorithm performs better than Guleryuz's algorithm due to the texture and structure decomposition. Fadili's algorithm well recovers the structures; however, the inpainting results are blurry in appearance. In contrast, the proposed algorithm is an examplar-based approach, which is more powerful for recovering large missing regions with complicated structures, so it better recovers missing regions appeared in these examples.

E. Effect of the Number of Candidate Patches on Inpainting Results

We now test the effect of the number of candidate patches N on inpainting results. We take the five test images in Fig. 8 as examples, and measure the inpainting quality in PSNR with the increasing N from 10 to 55. The inpainting quality results are shown in Fig. 13. We observe that, first, the mean PSNR values vary within a limited range of $[24.28, 24.42]$; therefore, the inpainting quality is relatively stable to the number of candidate patches. Second, the algorithm performs best when $N = 25$, and larger N does not always mean higher performance. The reason might be that the candidate patches provide a subspace for the sparse linear combination in patch inpainting model. Setting larger N will include more irrelevant patches as candidate patches which may decrease the generalization ability of the model when inferring the unknown pixels of patches. Based on the above observations, the number of candidate patches in the inpainting model is set to 25 in our implementation.

F. Implementation Details and Computation Time

In our implementation, the patch similarities defined in (2) are incrementally computed only for the newly introduced patches

[1] http://eeweb.poly.edu/%7Eonur/source.html

[2] http://www.greyc.ensicaen.fr/~jfadili/demos/WaveRestore/downloads /mcalab/Home.html

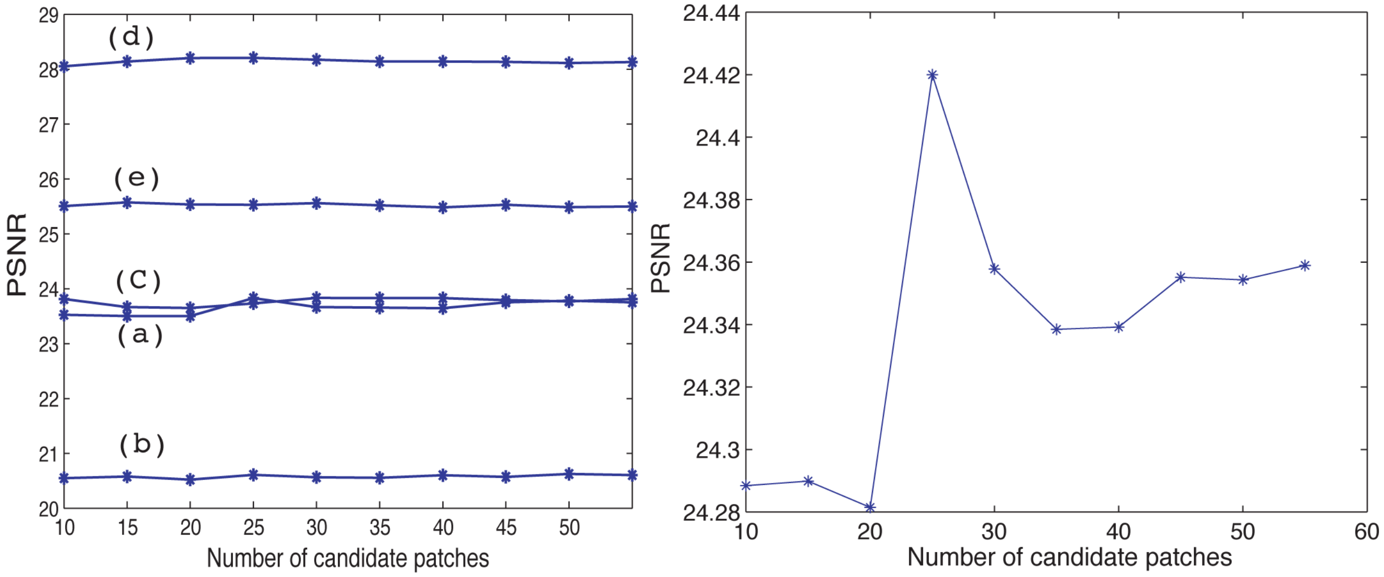


Fig. 13. Effect of the number of candidate patches (N) on inpainting quality. The left figure shows the PSNR values of inpainting results for the five examples in Fig. 8 with different N. The right figure shows the mean PSNR values with different N. The inpainting algorithm is stable to N, and achieves the highest quality when $N = 25$.

on the fill-front in each iteration, and the similarities are also used for the computation of local patch consistency constraint in (7). On the other hand, only a small subset of candidate patches are used for the optimization of sparse representation model, which deceases its computational overhead. It takes 103 s to fill in the missing region with 5310 missing pixels in the first example of Fig. 6 using C++ programming language on Intel 2.0GHz CPU.

V. Conclusion

This paper proposed a novel patch propagation based inpainting algorithm for scratch or text removal, object removal and missing block completion. The major novelty of this work is that two types of patch sparsity were proposed and introduced into the examplar-based inpainting algorithm. This was inspired from the recent progress of the research in the fields of image sparse representation and natural image statistics.

Structure sparsity was designed by measuring the sparseness of the patch similarities in the local neighborhood. The patch with larger structure sparsity, which is generally located at the structure, tends to be selected for further inpainting with higher priority. On the other hand, the patch sparse representation was proposed to synthesize the selected patch by the sparsest linear combination of candidate patches under the local consistency constraint. Experiments and comparisons showed that the proposed examplar-based patch propagation algorithm can better infer the structures and textures of the missing region, and produce sharp inpainting results consistent with the surrounding textures.

In the future, we will further investigate the sparsity of natural images at multiple scales and orientations, and apply it to the image inpainting, super-resolution and texture synthesis. We are also interested in incorporating the human-labeled structures into our framework in order to recover the totally removed structures.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Proc. SIGGRAPH*, 2000, pp. 417–424.

[2] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, “Navier–Strokes, fluid dynamics, and image and video inpainting,” in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2001, pp. 417–424.

[3] T. Chan and J. Shen, “Local inpainting models and tv inpainting,” *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2001.

[4] T. Chan and J. Shen, “Non-texture inpainting by curvature-driven diffusions,” *J. Vis. Commun. Image Represent.*, vol. 4, no. 12, pp. 436–449, 2001.

[5] C. Bertalmio, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, “Filling-in by joint interpolation of vector fields and gray levels,” *IEEE Trans. Image Process.*, vol. 10, pp. 1200–1211, 2001.

[6] A. Levin, A. Zomet, and Y. Weiss, “Learning how to inpaint from global image statistics,” in *Proc. Int. Conf. Comp. Vision*, pp. 305–313.

[7] S. Roth and M. J. Black, “Fields of experts: A framework for learning image priors,” in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2005, pp. 860–867.

[8] S. Roth and M. J. Black, “Steerable random fields,” in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[9] A. Efros and T. Leung, “Texture synthesis by non-parametric sampling,” in *Proc. Int. Conf. Comp. Vision*, 1999, pp. 1033–1038.

[10] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” *IEEE Trans. Image Process.*, vol. 12, pp. 882–889, 2003.

[11] A. Criminisi, P. Perez, and K. Toyama, “Object removal by examplar-based image inpainting,” in *Proc. Int. Conf. Comp. Vision*, 2003, pp. 721–728.

[12] J. Wu and Q. Ruan, “Object removal by cross isophotes examplar-based image inpainting,” in *Proc. Int. Conf. Pattern Recognition*, 2006, pp. 810–813.

[13] A. Wong and J. Orchard, “A nonlocal-means approach to examplar-based inpainting,” presented at the IEEE Int. Conf. Image Processing, 2008.

[14] G. T. N. Komodakis, “Image completion using efficient belief propagation via priority scheduling and dynamic pruning,” *IEEE Trans. Image Process.*, vol. 16, pp. 2649–2661, 2007.

[15] J. Jia and C. K. Tang, “Image repairing: Robust image synthesis by adaptive nd tensor voting,” in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recogition*, 2003, pp. 643–650.

[16] I. Drori, D. Cohen-Or, and H. Yeshurun, “Fragment-based image completion,” *ACM Trans. Graph.*, vol. 22, no. 2003, pp. 303–312, 2005.

[17] M. Elad, J. L. Starck, P. Querre, and D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis," *Appl. Comput. Harmon. Anal.*, vol. 19, pp. 340–358, 2005.

[18] O. G. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions," presented at the IEEE Int. Conf. Image Processing, 2003.

[19] O. G. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructures and iterated denoising-part i: Theory," *IEEE Trans. Image Process.*, vol. 15, pp. 539–554, 2006.

[20] O. G. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructures and iterated denoising-part ii: Adaptive algorithms," *IEEE Trans. Image Process.*, vol. 15, pp. 555–571, 2006.

[21] M. J. Fadili, J. L. Starck, and F. Murtagh, "Inpainting and zooming using sparse representations," *The Comput. J.*, vol. 52, no. 1, pp. 64–79, 2009.

[22] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by examplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, pp. 1200–1212, 2004.

[23] Y. P. Li and D. Huttenlocher, "Sparse long-range random field and its application to image denoising," presented at the European Conf. Computer Vision, 2008.

[24] M. F. Tappen, B. C. Russell, and W. T. Freeman, "Exploiting the sparse derivative prior for super-resolution and image demosaicing," presented at the IEEE Workshop on Statistical and Computational Theories of Vision, 2003.

[25] J. Sun, J. Sun, Z. B. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," presented at the IEEE Computer Society Conf. Computer Vision and Pattern Recogition, 2008.

[26] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, 2006.

[27] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 70:1–70:9, 2007.

[28] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vis. Res.*, vol. 37, no. 33, pp. 3311–3325, 1997.

[29] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, pp. 3736–3745, 2006.

[30] J. Maire, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, pp. 53–69, 2008.

[31] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce, "Discriminative sparse image models for class-specific edge detextion and image interpretation," presented at the IEEE Computer Society Conf. Computer Vision and Pattern Recogition, 2008.

[32] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionary for local image analysis," presented at the European Conf. Computer Vision, 2008.

[33] J. Winn, A. Criminisi, and N. Cristianini, "Object categorization by learned universal visual dictionary," presented at the Int. Conf. Comp. Vision, 2005.

[34] J. C. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," presented at the IEEE Computer Society Conf. Computer Vision and Pattern Recogition, 2008.

[35] G. Peyre, "Non-negative sparse modeling of textures," presented at the Scale Space and Variational Methods in Computer Vision, 2007.

[36] H. Chang, D. Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," presented at the IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 2004.

[37] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[38] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Trans. Signal Process.*, vol. 41, pp. 3397–3415, 1993.

[39] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Non-negative sparse modeling of textures," presented at the Scale Space and Variational Methods in Computer Vision, 2007.

[40] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[41] R. Tibshirani, "Regression shrinkge and selection via the lasso," *J. Roy. Statist. Soc. B.*, vol. 58, no. 1, pp. 267–288, 1996.

[42] R. F. i Ventura, P. Vandergheynst, P. Frossard, and A. Cavallaro, "Color image scalable coding with matching pursuit," *IEEE Acoust. Speech, Signal Process.*, 2004.

[43] R. Gribonval, C. Fvotte, and E. Vincent, "Performance measurement in blind audio source separation," *IEEE Trans. Speech, Audio, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, 2006.

[44] B. Shen, W. Hu, Y. Zhang, and Y. Zhang, "Image inpainting via sparse representation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2009, pp. 697–700.

[45] J. Sun, Y. Lu, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," in *Proc. ACM SIGGRAPH*, 2005, pp. 861–868.

[46] J. L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Trans. Image Process.*, vol. 14, pp. 1570–1582, 2005.

Zongben Xu received the M.S. and Ph.D. degrees in mathematics from Xi'an Jiaotong University, China, in 1981 and 1987, respectively.

In 1988, he was a Postdoctoral Researcher in the Department of Mathematics, University of Strathclyde, U.K. He has been with the Faculty of Science and Institute for Information and System Sciences, Xi'an Jiaotong University, since 1982, where he became Associate Professor in 1987 and Full Professor in 1991, and now serves as Director of the Institute for Information and System Sciences. He now serves as Chief Scientist of one National Basic Research Project of China (973 Project). His current research interests include nonlinear functional analysis, mathematical foundation of information technology, and computational intelligence.

Dr. Xu is owner of the National Natural Science Award of China in 2007 and winner of CSIAM Su Buchin Applied Mathematics Prize in 2008.

Jun Sun received the B.S. degree from the University of Electronic Science and Technology of China in 2003 and the Ph.D. degree in applied mathematics from Xi'an Jiaotong University in 2009.

He worked as a visiting student at Microsoft Research Asia from November 2005 to March 2008, and a postdoctoral research associate at the University of Central Florida from August 2009 to April 2010. He now serves as a faculty at the School of Science of Xi'an Jiaotong University. His current research interests are image denoising, inpainting, segmentation, and super-resolution.