

# Image Inpainting Using Nonlocal Texture Matching and Nonlinear Filtering

Ding Ding<sup>1</sup>, Sundaresh Ram<sup>2</sup>, *Member, IEEE*, and Jeffrey J. Rodríguez<sup>1</sup>, *Senior Member, IEEE*

**Abstract**—Nonlocal texture similarity and local intensity smoothness are both essential for solving most image inpainting problems. In this paper, we propose a novel image inpainting algorithm that is capable of reproducing the underlying textural details using a nonlocal texture measure and also smoothing pixel intensity seamlessly in order to achieve natural-looking inpainted images. For matching texture, we propose a Gaussian-weighted nonlocal texture similarity measure to obtain multiple candidate patches for each target patch. To compute the pixel intensity, we apply the  $\alpha$ -trimmed mean filter to the candidate patches to inpaint the target patch pixel-by-pixel. The proposed algorithm is compared with four current image inpainting algorithms under different scenarios, including object removal, texture synthesis, and error concealment. Experimental results show that the proposed algorithm outperforms the existing algorithms when inpainting large missing regions in images with texture and geometric structures.

**Index Terms**—Exemplar-based image inpainting, nonlocal texture matching, texture synthesis,  $\alpha$ -trimmed mean filter, object removal.

## I. INTRODUCTION

WHEN solving image inpainting problems, we often consider two crucial factors in order to obtain good results: consistency of intensity and consistency of texture. Consistent intensity is needed when connecting lines or recovering edges having similar intensity, whereas consistent texture is needed when reconstructing large missing regions having texture patterns. Many image inpainting algorithms can be classified into two categories: diffusion-based inpainting and patch-based inpainting. Diffusion-based inpainting performs well in achieving consistent intensity, and patch-based inpainting is effective in reconstructing textured regions. Some diffusion-based inpainting methods [1]–[3] inpaint images by propagating higher-order derivatives of local pixel intensity continuously along the lines of equal intensity values (isophotes), toward the interior of the missing region. Other diffusion-based image inpainting methods [4]–[10] reconstruct images

by minimizing a variational inpainting function governed by a high-order partial differential equations (PDE) or systems of PDEs. Such methods are referred to as total variational (TV) inpainting. These methods ensure local intensity smoothness and generally perform better than patch-based inpainting when recovering geometric structures, but they can produce noticeable blurring when inpainting a large missing region having significant texture [11], [12].

Patch-based inpainting [13]–[32] is capable of recovering large missing regions by searching the image source region (the complement of the missing region within an image) and copying a suitable texture into the missing region, patch by patch. For example, Criminisi's exemplar-based inpainting method [13] first maximizes a priority function defined by the direction of the isophote to find the next target patch to be filled. Then, it selects the patch with the least mean-squared-error (MSE) distance to the known pixels of the target patch, and fills in the unknown pixels of the target patch with the corresponding pixels from the selected patch. With the development of image quality assessment indices, corresponding inpainting methods have been developed, such as structural-similarity image inpainting [33] and perceptually aware image inpainting [34]. Patch-based inpainting methods can generate disconnected lines or broken edges due to patch-wise copying. Also, when the missing region is surrounded by complex background (e.g., multiple textures meeting at the border of the missing region), patch-based methods tend to generate unsuitable textural patches [11], [12].

Recently, there have been efforts to combine diffusion-based and patch-based inpainting methods [11], [12], [34], [35]. One approach [11], [12], [35] first decomposes an image into two images. One image captures the high frequency component of the image, and the other captures the low-frequency component of the image. Then the method inpaints the low-frequency image with a diffusion-based method and the high-frequency image with a patch-based method. After that, the inpainted images are combined together to obtain the final inpainted image. However, the performance improvement provided by such methods is small compared to patch-based inpainting. Moreover, the decomposition approach to inpainting is not computationally efficient.

With the advent of the nonlocal means (NLM) filter, it has been applied to image inpainting [36]–[38]. Wond and Orchard [36] evaluate the MSE to choose multiple candidate patches from the image source region and use a NLM filter to aggregate all the candidate patches to fill the target patch.

Manuscript received August 17, 2017; revised February 7, 2018 and October 24, 2018; accepted October 27, 2018. Date of publication November 12, 2018; date of current version December 3, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jean-Francois Aujol. (*Corresponding author: Sundaresh Ram.*)

D. Ding and J. J. Rodríguez are with the Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721 USA (e-mail: dding@email.arizona.edu; jjrodrig@email.arizona.edu).

S. Ram is with the Department of Radiology, Center for Molecular Imaging, and the Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI 48105 USA (e-mail: sundarar@umich.edu).

Digital Object Identifier 10.1109/TIP.2018.2880681

Arias *et al.* [37] formulate image inpainting as an optimization problem and minimize an energy function containing the nonlocal mean of the similarity between patches from the source region and the target patch. Buades *et al.* [38] proposed a denoising algorithm that can also be applied to texture reconstruction; the algorithm estimates each pixel's value using a NLM filter.

With recent advances in machine learning, convolutional neural networks (CNN) have been explored to address the image inpainting problem [32], [39]–[46]. Yang *et al.* [39] propose a hybrid optimization approach, where they combine a trained encoder-decoder structured CNN and a local neural patch synthesizer learned from large scale datasets to inpaint the missing regions within the input image. However, well trained and generalized CNN models can still overfit when presented with random images. Also, the inpainted results are very sensitive to the choice of hyper-parameters of the CNN [40]. In another approach, Ulyanov *et al.* [40] apply an untrained deep CNN with random weight initialization and use it to maximize the likelihood given an image to be inpainted. Ulyanov *et al.* state that the specific structure of the untrained deep CNN is more than enough to learn the image statistics to help complete the missing regions accordingly.

### Contributions

In this paper, we propose a new image inpainting algorithm that uses a novel nonlocal texture similarity measure to select several candidate patches for a given target patch, which are then fused together using the  $\alpha$ -trimmed mean filter to obtain the inpainted results. The objective of the proposed algorithm is to recover a large missing region surrounded by multiple textural and structural regions. The proposed algorithm is capable of reproducing the underlying textures and structures of the large missing regions within an image simultaneously and achieving natural-looking inpainted results. There are several contributions of the proposed algorithm.

First, this work presents an exemplar-based inpainting method using a new nonlocal texture similarity (NLTS) measure to match the target and candidate patches in the image under the assumption that there are many local repetitive textures present within the image. Next, when choosing the next target patch to be inpainted, the proposed algorithm selects a patch centered on the contour surrounding the outer border of the missing region, unlike other exemplar-based inpainting algorithms. Also, the proposed algorithm chooses multiple good candidate patches and combines them using the  $\alpha$ -trimmed mean filter to inpaint the target patch in comparison to other exemplar-based inpainting algorithms which choose only a single candidate patch. Furthermore, in our proposed iterative framework, the information from the already inpainted patches can be used towards inpainting other target patches as the inpainted pixels are added to the source region, whereas other inpainting algorithms use only the original pixel data.

The proposed algorithm is compared to several existing image inpainting algorithms for different applications including object removal, texture synthesis, and error concealment. Both quantitative and qualitative results show that the proposed method outperforms the other inpainting methods.

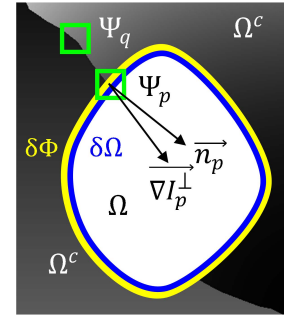


Fig. 1. Original image  $I$  with the missing region  $\Omega$ , and the source region  $\Omega^c$ .  $\delta\Omega \subset \Omega^c$  is the outer border of  $\Omega$ .  $\delta\Phi$  is a single-pixel-wide contour surrounding the outer border  $\delta\Omega$ .  $\Psi_p$  is a patch centered at the pixel  $p \in \delta\Phi$ , and  $\Psi_q$  is a patch centered at pixel  $q$  such that  $\Psi_q \cap \Omega = \emptyset$ .  $\vec{n}_p$  is a unit vector orthogonal to  $\delta\Phi$  at the center pixel  $p$ , and  $\vec{\nabla} I_p^\perp$  is the isophote vector at  $p$  (along the direction of equal intensity line that is the boundary between the two gray regions, and orthogonal to the gradient vector).

## II. METHOD

We propose an image inpainting algorithm that uses non-local texture similarity matching and pixel-wise intensity smoothing. The image inpainting problem involves an input 2D image,

$$I : \mathcal{D} \rightarrow \mathbb{R}, \mathcal{D} \subset \mathbb{Z}^2$$

$$\mathcal{D} = \{(x_k, y_k) : k = 1, 2, \dots, |\mathcal{D}|\}$$

that has a missing region denoted as  $\Omega \subset \mathcal{D}$ . The image source region, denoted as  $\Omega^c$ , is the complement of the missing region as shown in Fig. 1. The outer border of the missing region  $\Omega$  is denoted as  $\delta\Omega \subset \Omega^c$ . We denote  $\delta\Phi$  as a single-pixel-wide contour surrounding the outer border  $\delta\Omega$  in the source region. The intensity of a pixel  $k$  within the image  $I$  is denoted as  $I(x_k, y_k)$ , where  $(x_k, y_k)$  are the pixel coordinates at pixel  $k$ . We use  $M : \mathcal{D} \rightarrow \mathbb{R}$  as a mask to indicate whether a pixel  $k$  belongs to the source region or the missing region, given by

$$M(x_k, y_k) = \begin{cases} 1, & \text{if } k \in \Omega^c \\ 0, & \text{if } k \in \Omega. \end{cases} \quad (1)$$

Assume the image  $I$  is made up of overlapped patches of size  $m \times m$ . A patch centered at pixel  $p$  is defined as a vector using columnwise ordering, denoted as  $\Psi_p$ , whose elements  $k$  are the ordered pair  $(x_k, y_k)$ . The intensities of  $\Psi_p$  are denoted as  $I_p = [I(\Psi_p(l)); l = \{1, 2, \dots, m^2\}]$ , where  $l$  is the columnwise linear index location within the patch. The target patch to be inpainted during each iteration is centered on the single-pixel-wide contour of the outer border of the missing region (see Fig. 1), and the candidate patch is selected from the source region. We propose an iterative inpainting method to inpaint the missing region, patch by patch. There are three main steps of the algorithm, which are described in detail in the next three subsections.

### A. Choosing the Target Patch

The proposed inpainting algorithm is iterative, where, during each iteration, we first apply a priority function to choose the next target patch. The target patch has a known part from

the source region and an unknown part from the missing region. The center of the target patch lies on the contour of the outer border ( $\delta\Phi$ ) of the missing region (see also Section IV-C). The priority function contains two terms, a confidence term and a data term. The confidence term  $C(\Psi_p)$  is the ratio of known pixels within the patch. The data term  $D(\Psi_p)$  computes the dot product of the isophote vector  $\overline{\nabla}I_p^\perp$  (along the direction of equal intensity lines as shown in Fig. 1) and the normal vector  $\vec{n}_p$  at the center pixel  $p$ . The priority function is defined as

$$P(\Psi_p) = C(\Psi_p)D(\Psi_p), \quad p \in \delta\Phi \quad (2)$$

where

$$C(\Psi_p) = \frac{|\Psi_p \cap \Omega^c|}{|\Psi_p|}, \quad D(\Psi_p) = \frac{|\overline{\nabla}I_p^\perp \cdot \vec{n}_p|}{I_{\max}} \quad (3)$$

where  $\overline{\nabla}I_p^\perp$  is the isophote vector orthogonal to the gradient  $\overline{\nabla}I_p$  (computed using the central difference operator) at center pixel  $p$ ,  $I_{\max}$  is the maximum possible gray-level value, which is 255 in our case, and  $\vec{n}_p$  is the unit vector orthogonal to the outer border  $\delta\Omega$ . Here,  $\overline{\nabla}I_p^\perp$  and  $\vec{n}_p$  are computed at center pixel  $p$  as follows:

$$\overline{\nabla}I_p^\perp = \frac{1}{2I_{\max}} \left\{ [I(x_p, y_p - 1) - I(x_p, y_p + 1)] \mathbf{i} + [I(x_p + 1, y_p) - I(x_p - 1, y_p)] \mathbf{j} \right\} \quad (4)$$

$$\vec{n}_p^0 = [M(x_p + 1, y_p) - M(x_p - 1, y_p)] \mathbf{i} + [M(x_p, y_p + 1) - M(x_p, y_p - 1)] \mathbf{j} \quad (5)$$

$$\vec{n}_p = \frac{\vec{n}_p^0}{\|\vec{n}_p^0\|} \quad (6)$$

where  $\mathbf{i}$  and  $\mathbf{j}$  are the unit vectors in the x and y direction, respectively, and  $\vec{n}_p = 0$  when  $\|\vec{n}_p^0\| = 0$ . In each iteration, we want to maximize the priority function to select the patch  $\Psi_{\hat{p}}$ , having the highest priority value, to serve as the target patch.

### B. Nonlocal Texture Similarity Matching

After choosing the target patch, we use a new nonlocal texture similarity (NLTS) measure for selecting the candidate patches from the source region patches  $\Psi_q$  centered at pixel  $q$  such that  $\Psi_q \cap \Omega = \emptyset$ . The proposed similarity measure satisfies the following objectives. First, the measure is able to compute the textural similarity between a chosen candidate patch from the source region and the target patch. Second, the measure is computed using weighted pixel values, where the weights are given by a Gaussian kernel, whose center is adaptively chosen for different target patches with varying spatial distributions of known and unknown pixels. The NLTS measure is given by

$$\text{NLTS}(I_{\hat{p}}, I_q) = \exp \left\{ \frac{-\|(I_{\hat{p}} - I_q)^{\circ 2} \circ G_{\hat{p}}\|_1}{h^2} \right\} \quad (7)$$

where  $\circ 2$  is the element-wise power operator,  $\circ$  is the element-wise product operator, and  $h$  is a user-selected parameter [47].

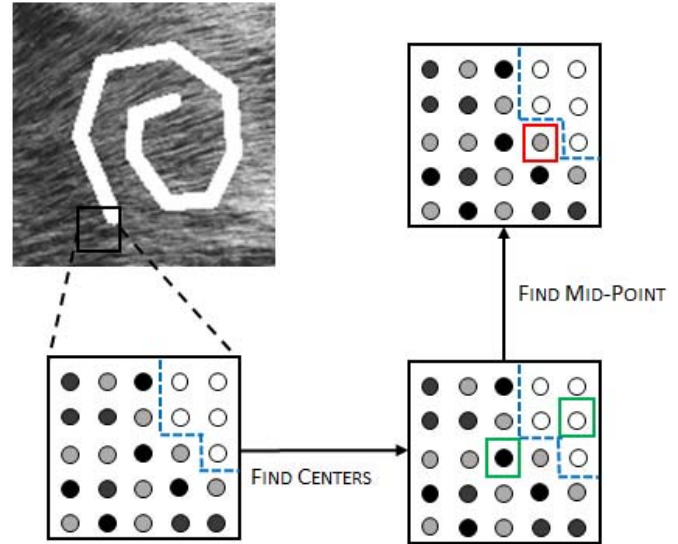


Fig. 2. The flowchart of the process to find the center of the Gaussian kernel for a target patch. The colored dots (black and gray) represent the known pixels with different intensity values within the target patch, and the white dots represent the unknown pixels within the target patch.

We choose a constant  $h = 34$  empirically (the value of  $h$  should be proportionately larger if the maximum gray level is larger than 255).  $G_{\hat{p}} = [\mathbf{G}(\Psi_{\hat{p}}(l)); l = \{1, 2, \dots, m^2\}]$  (columnwise vector) is the Gaussian kernel over the target patch  $\Psi_{\hat{p}}$ . The Gaussian kernel  $\mathbf{G}$  is defined globally, and the Gaussian weight at a pixel  $k$  is computed as

$$\mathbf{G}(x_k, y_k) = \exp \left\{ -\frac{(x_k - x_c)^2 + (y_k - y_c)^2}{2\sigma^2} \right\} \quad (8)$$

where  $(x_c, y_c)$  are the pixel coordinates of the center of the Gaussian filter. Also,  $\sigma$  is the standard deviation of the Gaussian kernel, and how to choose its value is discussed in Section III-A.

The NLTS measure defined in (7) is a measure of the intensity similarity between a patch from the source region and the target patch; a larger value means a larger similarity between the two patches. The Gaussian weight  $\mathbf{G}(x_k, y_k)$  at pixel  $k$  is related to its coordinate distance to the center of the kernel  $(x_c, y_c)$ , and the center adapts depending on the distribution of the known and unknown pixels within the target patch (See Fig. 2). We set the center of the Gaussian kernel  $(x_c, y_c)$  as the midpoint of the geometric center  $(x_1, y_1)$  of the known part and the geometric center  $(x_0, y_0)$  of the unknown part of the target patch, calculated by

$$(x_c, y_c) = \left( \frac{x_1 + x_0}{2}, \frac{y_1 + y_0}{2} \right) \quad (9)$$

where

$$(x_1, y_1) = \frac{1}{|\Psi_{\hat{p}} \cap \Omega^c|} \sum_{k \in \Psi_{\hat{p}} \cap \Omega^c} (x_k, y_k)$$

$$(x_0, y_0) = \frac{1}{|\Psi_{\hat{p}} \cap \Omega|} \sum_{k \in \Psi_{\hat{p}} \cap \Omega} (x_k, y_k)$$

where  $|\Psi_{\hat{p}} \cap \Omega^c|$  is the number of known pixels, and  $|\Psi_{\hat{p}} \cap \Omega|$  is the number of unknown pixels. When calculating the

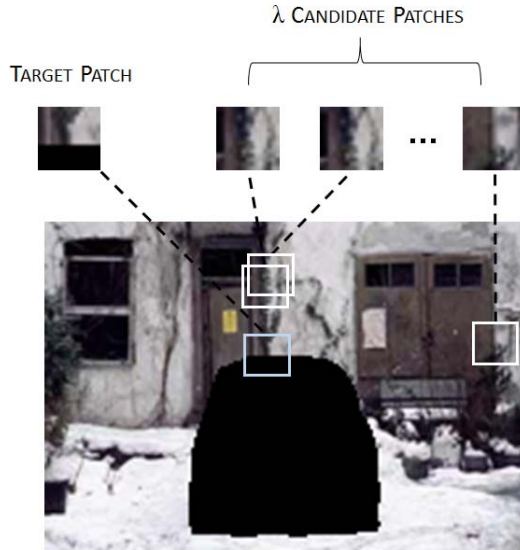


Fig. 3. A set of patches consisting of  $\lambda$  candidate patches and the target patch. For each missing pixel of the target patch (denoted by blue square), we fill it by applying the  $\alpha$ -trimmed mean filter to the sequence of pixels at the same pixel location within the  $\lambda$  candidate patches.

coordinates of the centers, we round the value to the nearest integer.

Using the NLTS measure defined in (7), we select the  $\lambda$  best matched candidate patches with the  $\lambda$  highest NLTS values from the image source region  $\Omega^c$ . In our experiments, we choose  $\lambda = 5$  candidate patches from the source region for each target patch, as we found that this value minimizes inpainting error. The value of  $\lambda$  is further discussed in Section IV-E.

### C. Inpainting Using the $\alpha$ -Trimmed Mean Filter

After choosing the candidate patches, we apply the  $\alpha$ -trimmed mean filter [48], [49] to the  $\lambda$  candidate patches using the procedure described below (see Fig. 3). For each missing pixel from the target patch at the index location  $l$ , we group the corresponding pixels at the same index location within the candidate patches into a set  $\mathcal{S} = \{I_{\hat{q}_i}(l); i = \{1, 2, \dots, \lambda\}\}$ , and order the intensities in the set  $\mathcal{S}$  in ascending order to obtain  $\mathcal{S}_o = \{X_j; j = \{1, 2, \dots, \lambda\}\}$ , satisfying

$$X_1 \leq X_2 \leq \dots \leq X_\lambda. \quad (10)$$

Then, we apply the  $\alpha$ -trimmed mean filter to the set  $\mathcal{S}_o$  to obtain the intensity of that missing pixel, given by

$$I_{\hat{p}}(l) = \text{mean}_\alpha(\mathcal{S}_o) = \frac{1}{\lambda - 2\alpha\lambda} \sum_{j=\alpha\lambda+1}^{\lambda-\alpha\lambda} X_j \quad (11)$$

For a given ratio  $\alpha$ , the  $\alpha$ -trimmed mean of the  $\lambda$  elements in  $\mathcal{S}_o$  is obtained by ignoring the  $\alpha\lambda$  smallest elements and the  $\alpha\lambda$  largest elements, and then computing the mean of the remaining elements. In our experiment, we use  $\alpha = 0.2$ .

The  $\lambda$  candidate patches are chosen based on the similarity between the known part of the target patch and the corresponding part of the candidate patches; thus, the other

### Algorithm 1 Proposed Inpainting Algorithm

**input** : An image with an unknown region  $\Omega$

**output**: Estimated image with the unknown region filled in

---

```

1 while there are missing pixels in  $\Omega$  do
2   Compute  $C(\Psi_p)$  and  $D(\Psi_p)$  using (3)  $\forall p \in \delta\Phi$ ;
3   Find the next target patch  $\Psi_{\hat{p}}$ , where
      $\hat{p} = \underset{p \in \delta\Phi}{\text{argmax}} P(\Psi_p)$ ;
4   Find  $\lambda$  candidate patches  $\Psi_{\hat{q}_i} \in \Omega^c$  having the  $\lambda$ 
     largest NLTS values using  $\text{NLTS}(I_{\hat{p}}, I_{\hat{q}_i})$ ;
5   For each  $l \in \{1, 2, \dots, m^2\}$ , inpaint
      $I_{\hat{p}}(l) = \text{mean}_\alpha(\mathcal{S}_o)$ , if  $\Psi_{\hat{p}}(l) \in \Omega$ ;
6   Add the inpainted pixels of the target patch  $\Psi_{\hat{p}}$  to
     the source region  $\Omega^c$ , and remove them from the
     missing region  $\Omega$ ;
```

---

part of each candidate patch (corresponding to the unknown portion of the target patch) could be very different from that of other candidate patches. For this reason, we choose the  $\alpha$ -trimmed mean instead of a full sample mean or a fully truncated mean (median). When computing the intensity of a missing pixel using the corresponding pixels from the candidate patches, the  $\alpha$ -trimmed mean filter is less sensitive to outliers (extremely large or small intensity values) than the full sample mean. Also, the  $\alpha$ -trimmed mean filter makes use of more than one pixel, which provides a better estimate than the fully truncated mean (discussed in Section IV-B). An outline of the proposed algorithm is presented in Algorithm 1.

The proposed inpainting algorithm can be applied to inpaint small or large missing regions in grayscale or color images. When inpainting color images, the input image is defined as  $I : \mathcal{D} \rightarrow \mathbb{R}^3$ , and the isophote vector at each pixel is computed as the average of the isophote vectors of the three color channels [30] and is given by

$$\overrightarrow{\nabla I_p}^\perp = \frac{1}{3} \sum_{t \in \{R, G, B\}} \overrightarrow{\nabla I_{p,t}}^\perp$$

where  $t$  represents the red (R), green (G), or blue (B) channel, and  $\overrightarrow{\nabla I_{p,t}}^\perp$  is the isophote vector at center pixel  $p$  in channel  $t$ . The NLTS measure for inpainting color images is given by

$$\text{NLTS}(I_{\hat{p}}, I_{\hat{q}}) = \exp \left\{ - \sum_{t \in \{R, G, B\}} \frac{\|(I_{\hat{p},t} - I_{\hat{q},t})^{\circ 2} \circ G_{\hat{p}}\|_1}{h^2} \right\} \quad (12)$$

where  $I_{\hat{p},t}$  and  $I_{\hat{q},t}$  are the intensity values of the patch  $\Psi_p$  and  $\Psi_q$ , respectively, in channel  $t$ . When inpainting a missing pixel at the index location  $l$ , we fill in each channel of the missing pixel separately by computing the  $\alpha$ -trimmed mean of the set  $\mathcal{S}_{o,t} = \{X_{j,t}; j = \{1, 2, \dots, \lambda\}\}$ , which consists of the intensities of the corresponding pixels within the candidate patches, in channel  $t$ .

$$I_{\hat{p},t}(l) = \text{mean}_\alpha(\mathcal{S}_{o,t}) = \frac{1}{\lambda - 2\alpha\lambda} \sum_{j=\alpha\lambda+1}^{\lambda-\alpha\lambda} X_{j,t} \quad (13)$$

### III. EXPERIMENTS AND RESULTS

In this section we evaluate the proposed algorithm along with current inpainting algorithms for three different image inpainting applications: *object removal*, *texture synthesis*, and *error concealment*. In the applications of object removal and texture synthesis, the missing region to be inpainted may have an irregular shape, while in the application of error concealment, the missing region is usually a square or a rectangular block.

We compare our proposed image inpainting algorithm with some of the current image inpainting algorithms including 1) image recovery via hybrid sparse representations by Li [50], 2) adaptive algorithms for image restoration (AAIR) by Wohlberg [51], 3) the annihilating filter-based low-rank Hankel matrix (ALOHA) approach proposed by Jin and Ye [52], 4) image completion using planar structure guidance by Huang *et al.* [28], 5) exemplar-based image inpainting using a modified priority definition by Deng *et al.* [30], and 6) exemplar-based inpainting with new heuristics for better geometric reconstructions by Buysens *et al.* [53].

#### A. Parameter Setting

There are five variable parameters in the proposed algorithm: the patch size  $m$ , the parameter  $h$  in the NLTS measure, the number of candidate patches  $\lambda$ , the value of  $\alpha$  in the  $\alpha$ -trimmed mean filter, and the standard deviation of the Gaussian kernel  $\sigma$ . In all of our experiments, the values of  $h$ ,  $\lambda$ , and  $\alpha$  are fixed. The patch size depends on the size of the image texture pattern. A small patch size relative to the size of the texture pattern may fail to reconstruct the texture of the image. We use a patch size in the range of [3, 17] for our proposed algorithm. The value of  $\sigma$  depends on the patch size ( $\sigma \propto m$ ) and the busyness [54] of the image texture. If  $\sigma$  is very small, the pixels on the edge of a patch would have a Gaussian weight close to zero, so that when computing the similarity between patches, a large patch size may act the same as a smaller patch size. Moreover, when propagating a narrow structure (e.g., a single straight line), the value of  $\sigma$  should be smaller than that used for inpainting a textural image, because pixels near the center of the Gaussian kernel,  $(x_c, y_c)$ , may be sufficient to match the structure. In our experiments, we tried the value of  $\sigma$  in the range of [0.5, 4], and we found that it gives good results in the range of [1.5, 2.5]. The parameters used in all the algorithms compared in the paper are shown in Table I. For Huang's, and Buysens' method we use the default parameter values as given in [28] and [53], respectively.

#### B. Object Removal

In this section, we compare our algorithm with several existing inpainting algorithms for the application of object removal. The images used for testing all the algorithms are

TABLE I  
PARAMETERS USED IN EACH ALGORITHM

Algorithm	Parameters
Li	$T \in [30, 45]$ , $W = 31$ $p_{\max} = 30$ , $\text{step} = 4$ $\text{iter}_{\text{outer}} \in [3, 10]$ , $\text{iter}_{\text{max}} = 50$
AAIR	Block Size: optimized in the range of [7, 31] for every image Others: default as described in [49]
Buades	$n = 128$ , $T = 0.06$ , $d_{\max} = 15$ $N_{\text{dims}} = 20$ , $d_{\text{patchwise}} = 1$ Others: default <sup>1</sup>
ALOHA	$\mu = 10$ , $\text{iter}_{\mu} = 100$ $N_{\text{img}} = 81$ , $N_{\text{fir}} = 25$
Deng	Patch Size: optimized in the range of [3, 17] for every image Neighbor Searching Size: $w = 80$
Proposed	Patch Size: optimized in the range of [5, 17] for every image $\sigma \in [1.5, 2.5]$ $\lambda = 5$ $h = 34$ (chosen empirically)

from the Berkeley segmentation dataset and the PASCAL object recognition database.<sup>2</sup>

1) *Qualitative Analysis*: In Fig. 4, we show some inpainted images from different inpainting algorithms for the application of object removal. Fig. 4 row 1 shows the input images to each algorithm where the object being removed is denoted by a mask (green color). From Fig. 4 rows 2 and 3, we observe that Li's method and the AAIR method produce smooth inpainted regions, and some texture information within the missing region is lost. For example, the inpainted snow covered ground region in Figs. 4(d,e) rows 2 and 3 is very smooth, and the inpainted regions do not match the texture of the surrounding region. The ALOHA algorithm has many dark patches within the missing region that do not look visually plausible, as shown in Figs. 4(c-f) row 4. We also observe that Huang's method, Deng's method, and Buysens' method can produce some unsuitable texture — for example, the unsuitable texture of the wall in Fig. 4(c) rows 5-7 and the dark texture within the snow on the ground in Fig. 4(d) rows 5-7. Not only is the proposed algorithm able to recover the structure of the image (the vertical edge of the wall in Fig. 4(b) row 8, and the door in Fig. 4(c) row 8), but it also generates suitable texture for the missing region (the grass area in Fig. 4(a) row 8, the wall and snow ground in Fig. 4(d) row 8, the pavement in Fig. 4(e) row 8, and the fence in Fig. 4(f) row 8).

2) *Observer Studies*: In most cases when removing an object from the image, there is no prior information of the true pixel values of the underlying scene within the region of the object to be removed. In order to analyze the image quality of the inpainted images from the compared algorithms, we used human observer studies [55], [56]. First, we randomly selected 40 test images from the Berkeley segmentation

<sup>1</sup>See also <http://www.mathworks.com/matlabcentral/fileexchange/13619-toolbox-non-local-means>.

<sup>2</sup><http://host.robots.ox.ac.uk/pascal/VOC/databases.html>.  
<https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images.html>.

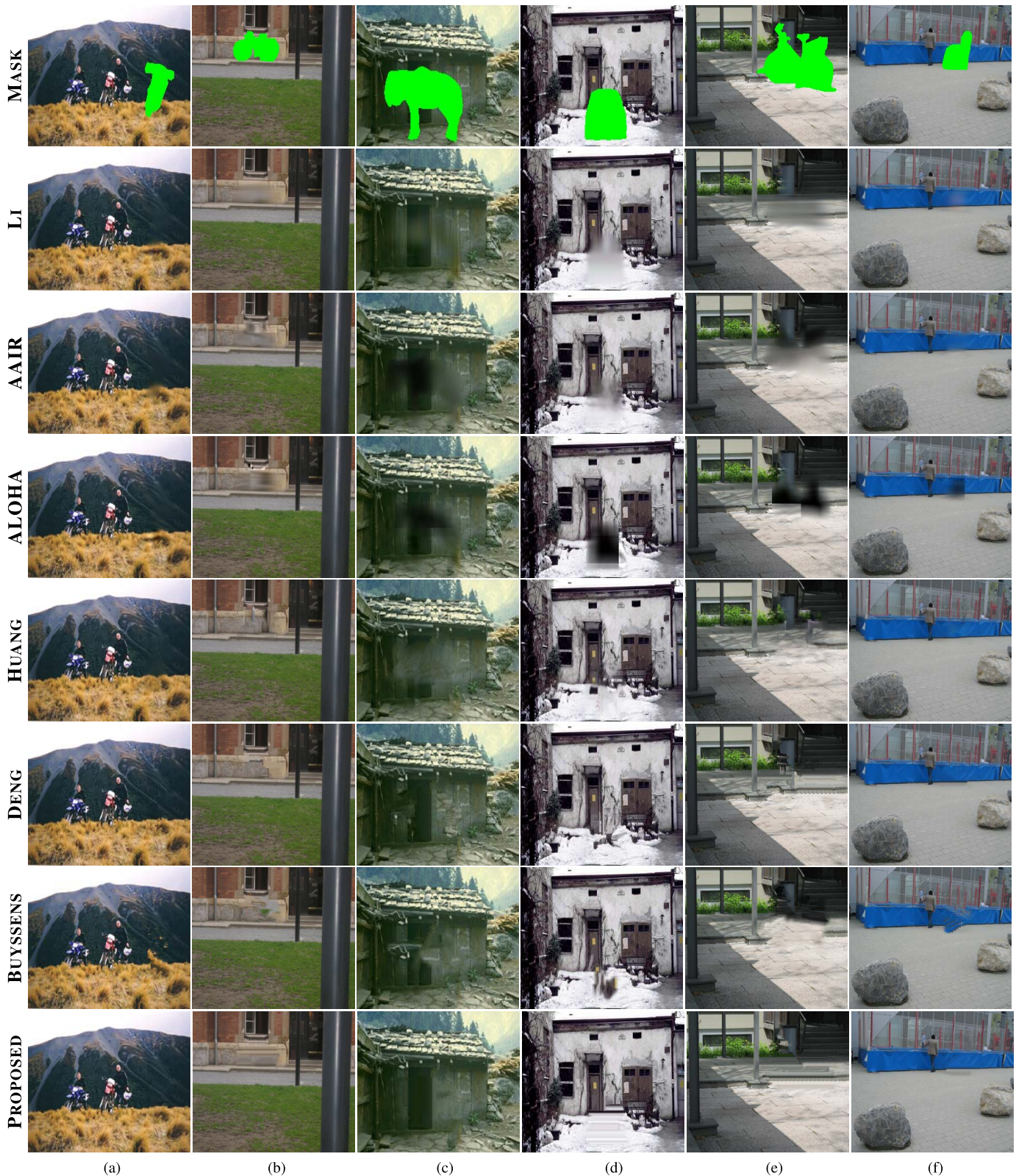


Fig. 4. Comparison of object removal using different methods. Row 1: Input images containing a removed object denoted by a mask (green color). Inpainting results — Row 2: Li's Method, Row 3: AAIR Method, Row 4: ALOHA Method, Row 5: Huang's method, Row 6: Deng's Method, Row 7: Buysens' method, and Row 8: Proposed Method. In each column, we state the percentage of missing pixels to be inpainted. (a) 3.90% missing. (b) 4.25% missing. (c) 14.47% missing. (d) 8.02% missing. (e) 12.65% missing. (f) 3.22% missing.

dataset and the PASCAL object recognition dataset, and we randomly removed one outstanding object (average number of removed pixels: 4639, average percentage of removed

pixels to the image: 8.00%) from each test image. Second, we inpainted all the 40 test images using the proposed algorithm, Li's algorithm, AAIR, ALOHA, Huang's algorithm,

TABLE II  
MEAN SCORE OF 40 IMAGES FOR EACH ALGORITHM  
FROM EACH EVALUATOR

	Li	AAIR	ALOHA	Huang	Deng	Buysens	Prop.
1	4.14	5.15	5.48	3.80	<b>2.21</b>	4.83	2.40
2	4.19	5.54	5.74	4.00	2.18	4.53	<b>1.84</b>
3	4.40	5.21	5.71	3.84	2.25	4.51	<b>2.08</b>
4	4.23	5.45	5.58	3.78	2.30	4.46	<b>2.21</b>
5	4.51	5.53	5.61	3.78	2.19	4.44	<b>1.95</b>
6	4.40	5.63	5.83	3.81	2.09	4.26	<b>1.99</b>
7	4.39	5.79	5.76	3.71	<b>1.99</b>	4.31	2.05
8	4.34	5.56	5.69	3.81	<b>2.09</b>	4.28	2.24
9	4.79	5.53	5.71	3.76	2.23	3.83	<b>2.16</b>
10	4.49	5.71	5.74	4.01	<b>1.99</b>	3.95	2.11
11	4.71	5.61	5.55	4.00	2.09	4.23	<b>1.81</b>
12	4.73	5.63	5.71	3.79	2.06	4.08	<b>2.01</b>
13	4.74	5.61	5.86	3.80	2.15	3.85	<b>1.99</b>
14	4.40	5.63	5.83	3.81	2.09	4.26	<b>1.99</b>
15	4.58	5.54	5.76	3.86	<b>2.00</b>	4.21	2.05
16	4.63	5.80	5.71	3.89	2.10	3.79	<b>2.09</b>
17	4.61	5.54	5.81	3.99	<b>2.00</b>	4.01	2.04
18	4.40	5.21	5.71	3.84	2.25	4.51	<b>2.08</b>
19	4.79	5.51	5.61	3.71	2.31	3.91	<b>2.15</b>
20	4.19	5.54	5.74	4.00	2.18	4.53	<b>1.84</b>
$\mu_{total}$	4.48	5.54	5.71	3.85	2.14	4.24	<b>2.05</b>
$\sigma_{total}$	0.20	0.17	0.10	0.10	0.10	0.28	0.14

Deng's algorithm, and Buysens' algorithm. Third, we presented the seven inpainted results of each test image to twenty human evaluators in a random order, without identifying the algorithms, and asked them to score the seven inpainted results between 1 and 7, with 1 being the best inpainted result and 7 being the worst inpainted result. When two or more inpainted results are regarded as equally good or equally bad, those inpainted results are given the same averaged score. The total score of the seven inpainted results from an evaluator remains the same as 28 ( $= 1 + 2 + 3 + 4 + 5 + 6 + 7$ ), so that the test images are equally weighted.

After the human evaluation, from each evaluator we obtained  $7 \times 40$  (number of compared algorithms  $\times$  number of test images) scores, then we averaged the group of 40 scores corresponding to each algorithm, resulting in 7 mean scores. Table II shows the mean scores of each algorithm from each evaluator and the total mean score ( $\mu_{total}$ ) of each compared algorithm and standard deviation ( $\sigma_{total}$ ) of those mean scores. In Table II, the proposed algorithm has the lowest total mean score ( $\mu_{total}$ ) among those of Li's algorithm, the AAIR method, Huang's method, the ALOHA method, Deng's method, and Buysens' method.

Further, in order to analyze if the mean scores of the proposed algorithm are statistically significant different from the mean scores of the other algorithms, we apply the Mann-Whitney U test [57] to the mean scores in Table II for the reason that the Mann-Whitney U test does not require the assumption of a normal distribution. The Mann-Whitney U test is a nonparametric test for testing the equality of medians in two independent populations [57]. Before the Mann-Whitney U test, we form seven samples of mean scores (size of  $7 \times 20$ , columns in Table II) where each sample is corresponding to one algorithm. We compute the Mann-Whitney U test between the proposed algorithm and each of the other six algorithms.

TABLE III  
THE MANN-WHITNEY U TEST BETWEEN THE PROPOSED  
ALGORITHM AND DENG'S ALGORITHM

Deng	Rank	Proposed	Rank
1.99	7	1.81	1
1.99	7	1.84	2.5
2.00	10.5	1.84	2.5
2.00	10.5	1.95	4
2.06	16	1.99	7
2.09	21	1.99	7
2.09	21	1.99	7
2.09	21	2.01	12
2.09	21	2.04	13
2.10	24	2.05	14.5
2.15	26.5	2.05	14.5
2.18	29.5	2.08	17.5
2.18	29.5	2.08	17.5
2.19	31	2.09	21
2.21	32.5	2.11	25
2.23	34	2.15	26.5
2.25	36.5	2.16	28
2.25	36.5	2.21	32.5
2.30	38	2.24	35
2.31	39	2.40	40
	$R_1=492$		$R_2=328$
	$U_1=118$		$U_2=282$

The null and alternate hypotheses of the Mann-Whitney U test are described as follows:

$H_0$  : The two populations of scores have equal median.

$H_1$  : The two populations do not have equal median.

In Table III, we show the Mann-Whitney U test between the proposed algorithm and Deng's inpainting algorithm. We order the mean scores of the proposed algorithm and Deng's algorithm from smallest to largest, and assign ranks from 1 to 40 (equal scores share the same averaged rank).  $R_1$  and  $R_2$  are the sums of the ranks for Deng's algorithm and the proposed algorithm, respectively, in Table III. The Mann-Whitney U test statistic is computed as

$$U = \min(U_1, U_2) \quad (14)$$

where

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1$$

$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2$$

where  $n_1 = n_2 = 20$  are the sample sizes. From (14), we obtain a test statistic of  $U = 118$  between the proposed algorithm and Deng's algorithm. The appropriate critical value of the Mann-Whitney U test can be found from the Mann-Whitney U test table, which is 127 based on both the sample sizes ( $n_1 = 20$  and  $n_2 = 20$ ) and a two-sided level of significance of  $\alpha = 0.05$ . Since we have  $U = 118 < 127$ , we reject the null hypothesis, thereby concluding that the mean scores of the proposed algorithm and the mean scores of Deng's algorithm do not have the same median.

Similarly, we apply the Mann-Whitney U test between the proposed algorithm and each of the other five algorithms, and

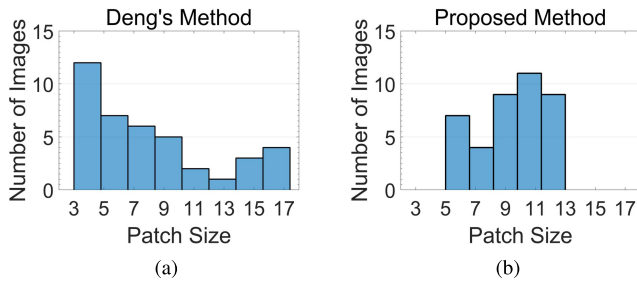


Fig. 5. The histograms of the 40 inpainted images with respect to the optimized inpainting patch-size in the range of [3, 17]. (a) The patch-size histogram when using Deng's method. (b) The patch-size histogram when using the proposed algorithm.

TABLE IV  
RUN TIME MEAN AND STANDARD DEVIATION  
FOR 40 IMAGES (HH:MM:SS)

Algorithm	Run Time (Std. Dev.)
Li	01:44:42 (01:09:55)
AAIR	00:35:22 (00:19:26)
ALOHA	19:02:57 (12:40:49)
Deng	00:03:46 (00:04:40)
Proposed	00:02:54 (00:03:06)

we find that all the test statistic values are zeros. Therefore, we reject the null hypothesis because  $U = 0 < 127$  for all five Mann-Whitney U statistics, concluding that the mean scores of the proposed algorithm and those of each of the other five algorithms have different medians.

We analyzed the common parameter, patch size, of the proposed algorithm and Deng's algorithm. Fig. 5 shows a histogram of the patch size for both algorithms when inpainting the 40 test images. From Fig. 5, we can observe that the occurrence of the patch size for Deng's algorithm distributes broadly between 3 and 17, whereas the occurrence of the patch size for the proposed algorithm distributes more narrowly. Thus, the proposed algorithm is less sensitive to the patch size than Deng's algorithm.

We also computed the mean and standard deviation of the run time of each algorithm in MATLAB for the 40 test images (on an Intel i5-2400 3.10GHz 8.00GB RAM platform). As shown in Table IV, the proposed algorithm has the smallest mean and standard deviation of the run time among all the algorithms.

We do not include the run time of Huang's method and Buysens' method in Table IV as the obtained source code of their methods are not completely in MATLAB. The proposed algorithm has a smaller run time than Li's method, because Li's method is a deterministic annealing method, requiring around 50 inner loops and multiple outer loops to obtain a convergent result. Also, the proposed algorithm has a smaller run time than the AAIR method, because the AAIR method is based on a sparse decomposition model that is solved via the iteratively reweighted least squares method [51], which takes more time. The proposed algorithm has a smaller run time than Deng's method, because the optimal patch size of Deng's method is often smaller than the proposed algorithm, which requires a longer time to fill-in the missing region.

TABLE V

IMAGE QUALITY ASSESSMENT OF INPAINTED IMAGES IN FIG. III-B.2

Algorithm	Avg. PSNR [dB]	Avg. SSIM
Li	29.66	0.9667
AAIR	28.86	0.9606
Buades	27.09	0.9423
ALOHA	23.04	0.9390
Huang	27.27	0.9416
Deng	26.88	0.9488
Buysens	25.85	0.9265
Proposed	29.19	0.9621

The ALOHA algorithm has the longest run time among all the algorithms compared, as it involves solving the block Hankel matrix, which is time-consuming.

### C. Inpainting Textured Images

The proposed image inpainting algorithm is able to reconstruct different types of texture due to the NLTS measure, in contrast to other image inpainting methods. We compared the proposed image inpainting algorithm with the six image inpainting algorithms in Section III-B.1, as well as the nonlocal-means algorithm by Buades [38], coded by Peyre<sup>1</sup>. We tested all eight algorithms on the images (size of  $128 \times 128$ ) in Fig. 6 from the texture image database.<sup>3</sup> Table I shows the parameters used for each algorithm.

From Fig. 6, we observe that Li's algorithm, the AAIR method, and Buades' method reconstruct some structure of the missing region; however, the reconstructed texture does not match the texture of the surrounding region (see Figs. 6(b-d) rows 1-4). When the missing region becomes large, the ALOHA method leaves the center part of the missing region dark, as shown in Fig. 6(e) rows 1-4. Huang's method, Deng's algorithm and Buysens' algorithm reconstruct the texture and structure of the missing region; however, the inpainted region seems to have visual artifacts (see Figs. 6(f-h) rows 1-4). The proposed algorithm has better performance in inpainting both the texture and structure, resulting in natural-looking inpainted images, as shown in Fig. 6(i) rows 1-4.

In order to analyze the image quality of the inpainted images, we computed two image quality assessment indices: peak signal-to-noise ratio (PSNR) in dB, and the structural similarity metric (SSIM) [33], [58]. PSNR and SSIM are calculated by referring to the original image without missing regions. The quantitative results are shown in Table V with the best values highlighted. From Table V, Li's algorithm has the highest average PSNR and SSIM values, and the proposed algorithm has the second highest values. However, the inpainted images from the proposed algorithm look visually better than the inpainted images from Li's algorithm. Table VI lists the mean and standard deviation of the run time of each algorithm in MATLAB for Figs. 6(b-i) (on an Intel i5-2400 3.10GHz 8.00GB RAM platform).

<sup>3</sup>The original images of Fig. 6(a) are from the MIT vision texture database at <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>.



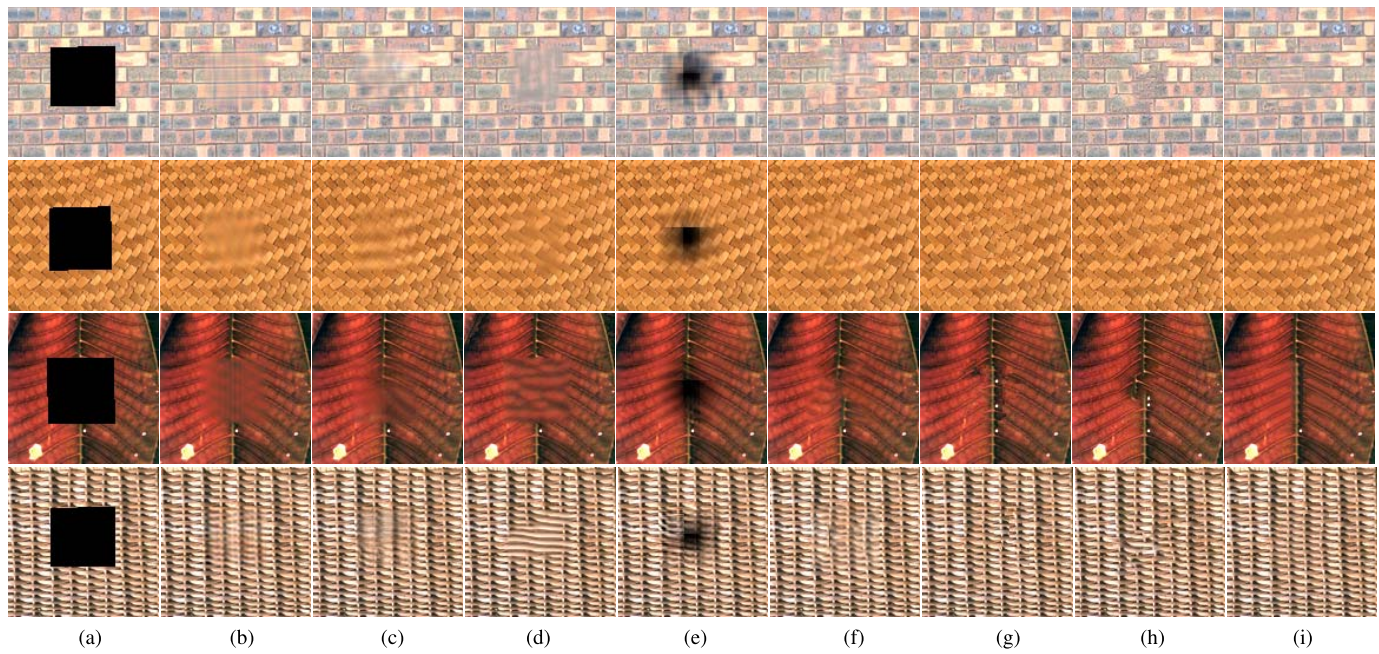


Fig. 6. Comparison of inpainting textured images using different inpainting methods. (a) Input images with a missing region to be restored (Mask). (b) The inpainting results of Li's Method. (c) AAIR Method. (d) Buades' Method. (e) ALOHA Method. (f) Huang's Method. (g) Deng's Method. (h) Buysens' Method. (i) Proposed Method.

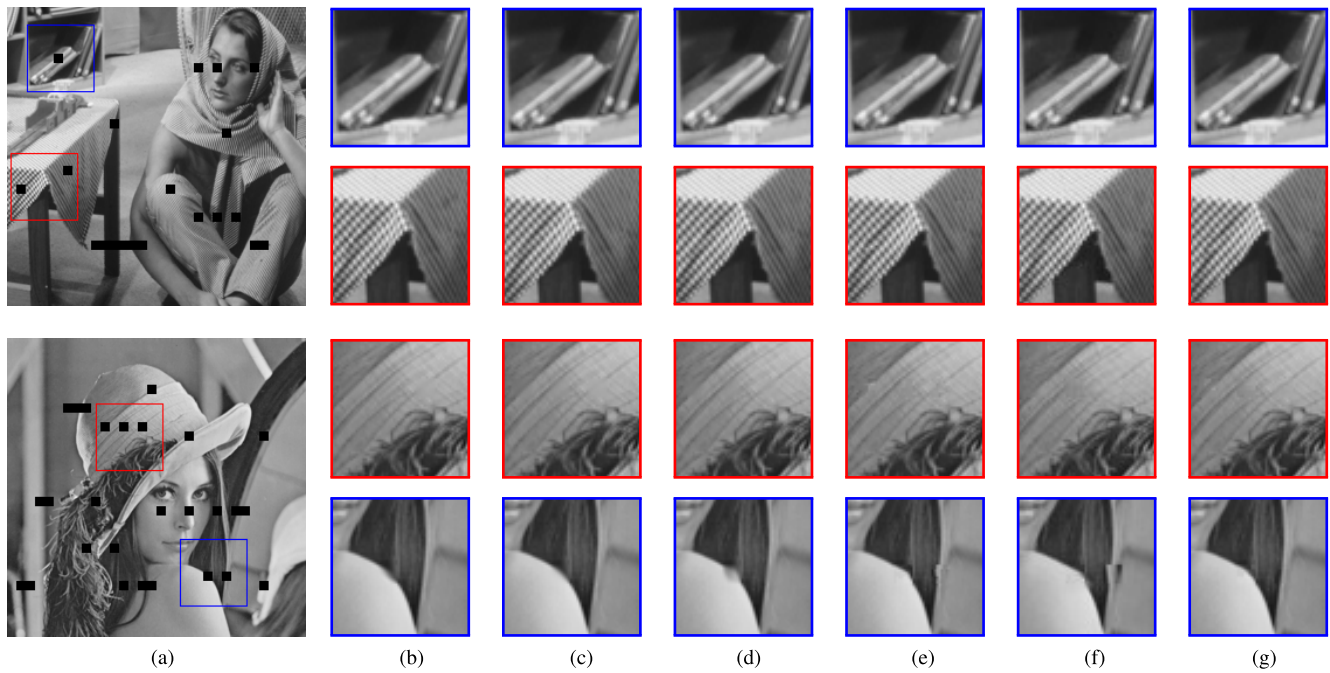


Fig. 7. Comparison of error concealment using inpainting for the different methods. (a) Images obtained at the receiver end after being transmitted with many  $8 \times 8$  blocks missing. (b) The inpainting results of Li's Method. (c) AAIR Method. (d) ALOHA Method. (e) Deng's Method. (f) Buysens' Method. (g) Proposed Method

Deng's algorithm has the lowest mean run time, and the proposed algorithm has the second lowest mean run time.

*D. Error Concealment*

In this section, we apply the proposed image inpainting algorithm for the application of recovering missing data from

digital images after wireless transmission. If the transmitted images are divided into blocks of size  $8 \times 8$  when transmitted (e.g., JPEG), the received images may lose an entire block or consecutive blocks due to noise [59]. In Fig. 7, two received images are shown in column 1. Columns 2 to 7 show the inpainted results of different inpainting algorithms. In Figs. 7(b-d) we observe that Li's algorithm, the AAIR

TABLE VI  
RUN TIME MEAN AND STANDARD DEVIATION  
FOR FIG. III-B.2 (HH:MM:SS)

Algorithm	Run Time (Std. Dev.)
Li	00:24:51 (00:00:47)
AAIR	00:07:26 (00:02:12)
Buades	00:02:14 (00:00:05)
ALOHA	07:35:28 (00:40:36)
Deng	00:00:14 (00:00:13)
Proposed	00:00:24 (00:00:24)

TABLE VII  
RUN TIME FOR FIG. 8 (HH:MM:SS)

	MS-NPS	DIP	Proposed
Training	12:30:00	None	None
Mask 1 (384 × 400)	00:05:24	00:39:13	00:01:39
Mask 2 (300 × 400)	00:04:42	00:50:20	00:02:02

algorithm, and the ALOHA algorithm are able to recover the straight-line structure (see row 1, edge of the book), but they do not perform well in recovering curved structure (see rows 3 and 4). Deng’s algorithm and Buysens’ method, on the other hand, produce some artifacts when recovering structure details as shown in Figs. 7(e-f) rows 3 and 4. In comparison, our proposed algorithm is able to recover texture and geometric structure within the missing blocks as shown in Fig. 7(g) (rows 1-4).

#### E. Comparison With Deep Learning Methods

We also compared the proposed algorithm with two deep learning inpainting methods—Multi-Scale Neural Patch Synthesis (MS-NPS) by Yang *et al.* [39] and Deep Image Prior (DIP) by Ulyanov *et al.* [40]—and tested the three algorithms on the images in Fig. 8 for the application of object removal. From Fig. 8, the inpainted regions from Yang’s method and Ulyanov’s method are more smooth than the inpainted region of the proposed algorithm; thus, some texture information is lost (see Figs. 8(c-d) row 2). The inpainted regions from Yang’s method and Ulyanov’s method can also have some edge defects (see Figs. 8(c-d) row 3), whereas the inpainted results of the proposed method look more natural and visually plausible as shown in Fig. 8(e) rows 2 and 3.

Table VII lists the run time of each algorithm on an Intel Core i7 PC (2.80GHz 16GB RAM) with an NVIDIA GPU (GeForce GTX950M 8GB RAM). For the MS-NPS method, we trained the model on the ImageNet [60] database consisting of 1 260 000 images, in the same manner as described in [39]. From Table VII, the proposed algorithm has the lowest run time compared to the two deep learning inpainting methods and does not require any external database for training.

## IV. DISCUSSION

### A. Nonlocal Texture Similarity Matching and the $\alpha$ -Trimmed Mean Filter

In the proposed algorithm, we apply the NLTS measure to select candidate patches and the  $\alpha$ -trimmed mean filter to

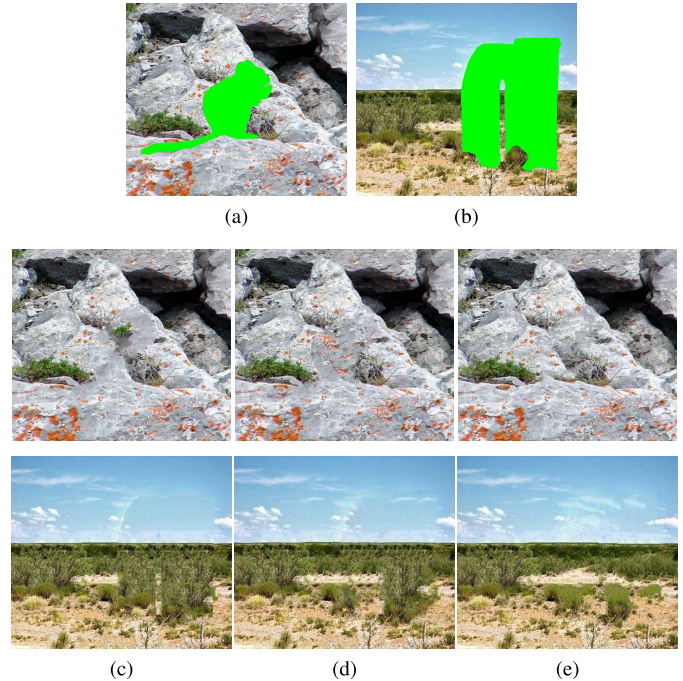


Fig. 8. Comparison of object removal using two deep learning methods and the proposed method. Row 1: Input images containing a removed object denoted by (a) Mask 1 and (b) Mask 2 shown in green color. Inpainting results — (c): MS-NPS, (d): DIP, and (e): Proposed Method.

fill each missing pixel using the chosen candidate patches. The proposed algorithm outperforms other image inpainting algorithms when inpainting large missing regions having both texture and geometric structure. How do the nonlocal measure and the  $\alpha$ -trimmed mean filter individually affect the inpainting result? Are both components essential in the proposed inpainting algorithm? In this section, we consider two variants of our algorithm: Algorithm 2 with only the NLTS measure and Algorithm 3 with only the  $\alpha$ -trimmed mean filter. In Algorithm 2, we apply the NLTS measure to select only one candidate patch from the source region. Then we fill the unknown part of the target patch by copying the corresponding part of the candidate patch. In Algorithm 3, we use the MSE to select five best candidate patches from the source region. Then, we fill the target patch pixel by pixel by applying the  $\alpha$ -trimmed mean filter as described in Section II-C. In Fig. 9, we compare the proposed algorithm with its two variants (Algorithm 2 and Algorithm 3).

In Fig. 9, we reconstruct the texture of the images from Fig. 6. When using Algorithm 2, the inpainted region looks more random and not consistent with the textural pattern of the remaining image as shown in Fig. 9(b) rows 1-4. The inpainted region using Algorithm 3 is overly smooth and blurry, as shown in Fig. 9(c) rows 1-2 and 4, and generates some artifacts as shown in Fig. 9(c) row 3. The proposed algorithm has a better performance in inpainting the missing region with a more plausible-looking result (see Fig. 9(a) rows 1-4). The quality assessment values of the inpainted images in Fig. 9 are listed in Table VIII with the best values highlighted. In Table VIII, the inpainted images using the

**Algorithm 2** Using Only the NLTS Measure

**input** : An image with an unknown region  $\Omega$   
**output**: Estimated image with the unknown region filled in

- 1 **while** there are missing pixels in  $\Omega$  **do**
- 2     Compute  $C(\Psi_p)$  and  $D(\Psi_p)$  using (3)  $\forall p \in \delta\Phi$ ;
- 3     Find the next target patch  $\Psi_{\hat{p}}$ , where  
 $\hat{p} = \underset{p \in \delta\Phi}{\operatorname{argmax}} P(\Psi_p)$ ;
- 4     Find the best candidate patch  $\Psi_{\hat{q}} \in \Omega^c$ , by maximizing NLTS( $I_{\hat{p}}, I_q$ );
- 5     For each  $l \in \{1, 2, \dots, m^2\}$ , inpaint  
 $I_{\hat{p}}(l) = I_{\hat{q}}(l)$ , if  $\Psi_{\hat{p}}(l) \in \Omega$ ;
- 6     Add the pixels of the target patch  $\Psi_{\hat{p}}$  to the source region  $\Omega^c$ , and remove them from the missing region  $\Omega$ ;

**Algorithm 3** Using Only the  $\alpha$ -Trimmed Mean Filter

**input** : An image with an unknown region  $\Omega$   
**output**: Estimated image with the unknown region filled in

- 1 **while** there are missing pixels in  $\Omega$  **do**
- 2     Compute  $C(\Psi_p)$  and  $D(\Psi_p)$  using (3)  $\forall p \in \delta\Phi$ ;
- 3     Find the next target patch  $\Psi_{\hat{p}}$ , where  
 $\hat{p} = \underset{p \in \delta\Phi}{\operatorname{argmax}} P(\Psi_p)$ ;
- 4     Find the  $\lambda$  candidate patches  $\Psi_{\hat{q}_i} \in \Omega^c$  having the  $\lambda$  smallest MSE values using MSE( $I_{\hat{p}}, I_q$ );
- 5     For each  $l \in \{1, 2, \dots, m^2\}$ , inpaint  
 $I_{\hat{p}}(l) = \operatorname{mean}_{\alpha}(S_o)$ , if  $\Psi_{\hat{p}}(l) \in \Omega$ ;
- 6     Add the pixels of the target patch  $\Psi_{\hat{p}}$  to the source region  $\Omega^c$ , and remove them from the missing region  $\Omega$ ;

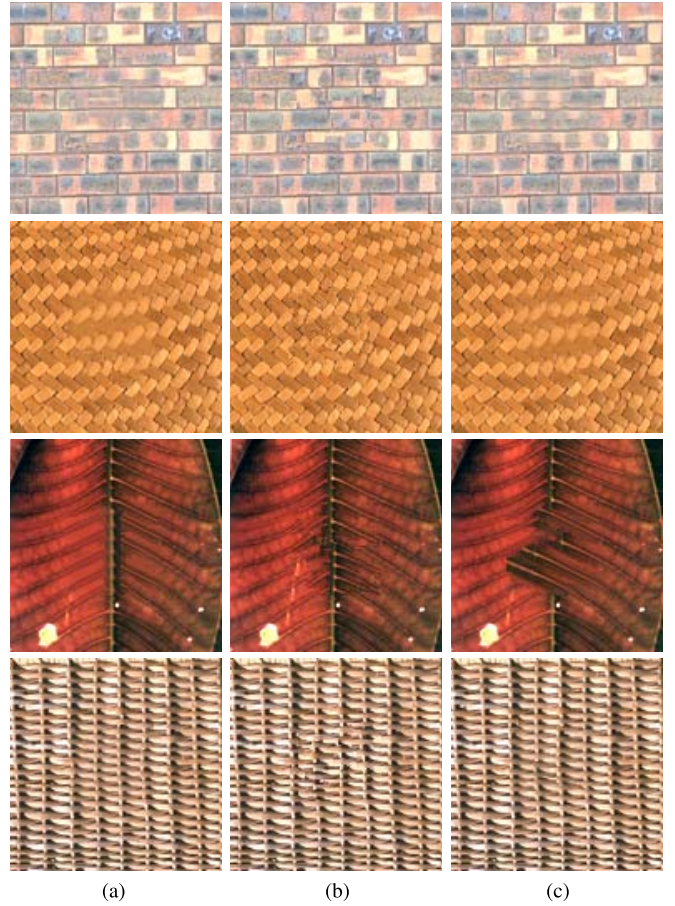


Fig. 9. Example comparing Algorithms 1-3. (a) Proposed. (b) Algorithm 2. (c) Algorithm 3.

TABLE VIII  
IMAGE QUALITY ASSESSMENT OF INPAINTED IMAGES IN FIG. 9

Algorithm	Avg. PSNR [dB]	Avg. SSIM
Proposed	<b>29.19</b>	<b>0.9621</b>
Alg. 2	26.57	0.9406
Alg. 3	28.18	0.9552

proposed algorithm have the highest average PSNR and SSIM values for Fig. 9.

### B. $\alpha$ -Trimmed Mean Filter vs. Other Order-Statistic Filters

In the proposed algorithm, when inpainting a missing pixel within the target patch, we apply the  $\alpha$ -trimmed mean filter to the pixels within the  $\lambda$  candidate patches at the index location corresponding to that of the missing pixel to obtain the intensity value of that missing pixel. The  $\alpha$ -trimmed mean filter is less sensitive to outliers (extremely large or small values) than a full sample mean filter as it discards the lowest and highest  $\alpha\lambda$  values and computes the mean based on the remaining values. Also, the  $\alpha$ -trimmed mean filter uses more

information than a fully truncated mean (median filter) because it uses more than one pixel's intensity value to compute the intensity of the missing pixel. In Fig. 10, we compare the inpainted results using different order-statistic filters [48], the full sample mean filter, the median filter, and the  $\alpha$ -trimmed mean filter.<sup>4</sup> From Fig. 10, the inpainted images using the mean filter and the median filter have some artifacts as shown in Figs. 10(c) and (d), respectively, whereas the artifacts do not appear in the inpainted images using the  $\alpha$ -trimmed mean filter (see Fig. 10(e)).

### C. Location of the Center Pixel $p$ of the Target Patch

Many exemplar-based inpainting methods choose the location of the center pixel  $p$  on the outer border of the missing region, i.e.  $p \in \delta\Omega$ . This leads to using the pixels from the missing region  $\Omega$  (either under the assumption that they have zero pixel intensities or using the pixel intensities from the original image [13], [30]) for the computation of the isophote vector  $\vec{\nabla} I_p^\perp$  in (4), thereby resulting in a different inpainting order, causing unsatisfactory inpainting results<sup>5</sup> (see Fig. 11).

<sup>4</sup>The images in Fig. 10(a) row 1, Fig. 11(a) row 1, and Fig. 12(a) are from <http://www.esience.cn/people/dengliangjian/Data.html>. The image in Fig. 10(a) row 2 is from the PASCAL object recognition database.

<sup>5</sup>The image in Fig. 11(a) row 2 is from the Berkeley segmentation dataset.

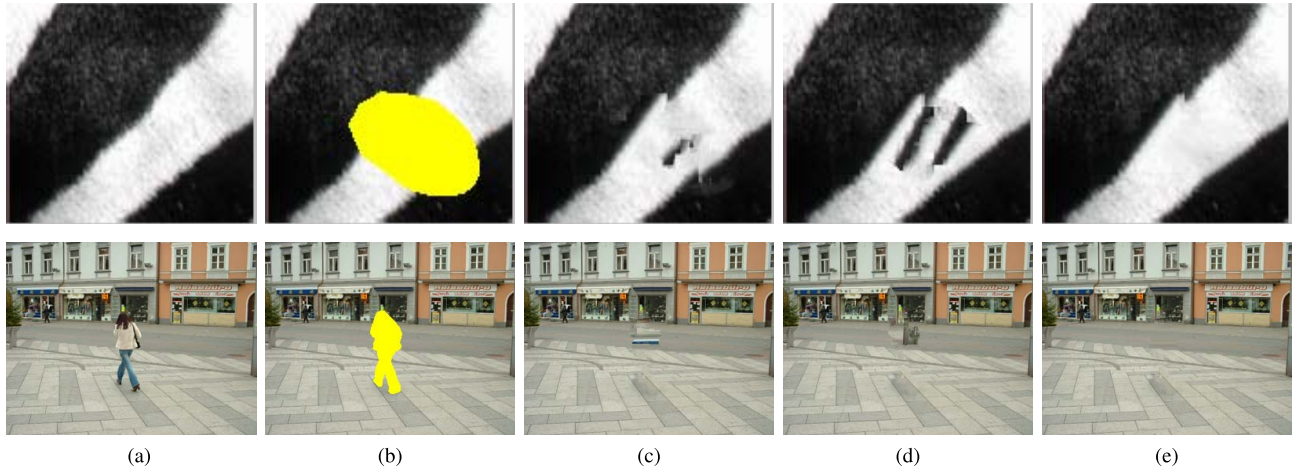


Fig. 10. Effect of various order-statistic filters. (a) Original image (b) Image with a missing region (in yellow mask). Inpainted image using the proposed algorithm with (c) a mean filter, (d) a median filter, (e) the  $\alpha$ -trimmed mean filter.

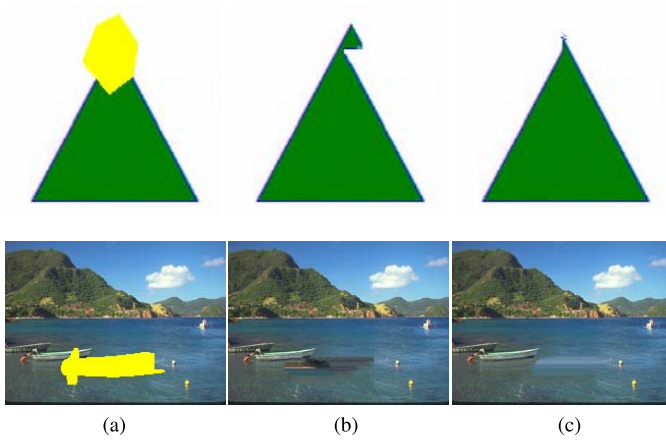


Fig. 11. Effect of choosing the location of center pixel  $p$  of the target patch. (a) Image with a missing region (in yellow mask). (b) Inpainted image using the proposed algorithm with  $p \in \delta\Omega$ . (c) Inpainted image using the proposed algorithm with  $p \in \delta\Phi$ .

In order to overcome this issue, we select the location of the center pixel  $p$  on the contour surrounding the outer border of the missing region, i.e.  $p \in \delta\Phi$ . For example in Fig. 11(b), we see that the inpainted region has artifacts due to the wrong inpainting order, whereas the inpainted images in Fig. 11(c) are more visually plausible.

#### D. Updating the Source Region

In the proposed algorithm, we add the inpainted target patch into the source region at the end of each iteration, thereby increasing the source region with inpainted patches that are often different from other known patches within the source region  $\Omega^c$ . Fig. 12 shows two examples of inpainting the image with or without updating the source region.<sup>4</sup> From Figs. 12(b) and (c), we observe that the inpainted images using our algorithm with updating the source region are better than the inpainted images using our algorithm without updating the source region (see the wrong dark texture in the center of the inpainted region as shown in Fig. 12(b), which does not appear

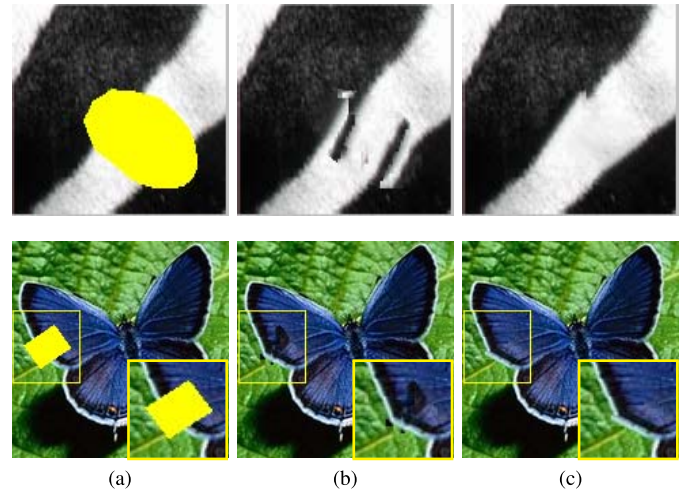


Fig. 12. Effect of updating the source region by including the inpainted patches. (a) Image with a missing region (in yellow mask). (b) Inpainted image using the proposed algorithm with updating the source region during each iteration. (c) Inpainted image using the proposed algorithm without updating the source region.

in Fig. 12(c)). After incorporating the inpainted target patch into the source region, the next target patch can be inpainted by a nearby inpainted patch rather than a patch far away from the source region resulting in a more consistent inpainted image.

#### E. Number of Candidate Patches ( $\lambda$ )

In order to determine how many candidate patches should be chosen for a given target patch in the proposed algorithm, we compute the average PSNR and SSIM values of the inpainted images while varying the number of candidate patches  $\lambda$ . We vary  $\lambda \in [1, 10]$ , as the matching error between the candidate patches and the target patch gets very large for  $\lambda > 10$ . The average PSNR and SSIM values for varying values of  $\lambda$  inpainted using the proposed method on images in Fig. 6(a) are plotted in Fig. 13. From the plot, we observe that the average PSNR and SSIM values of the

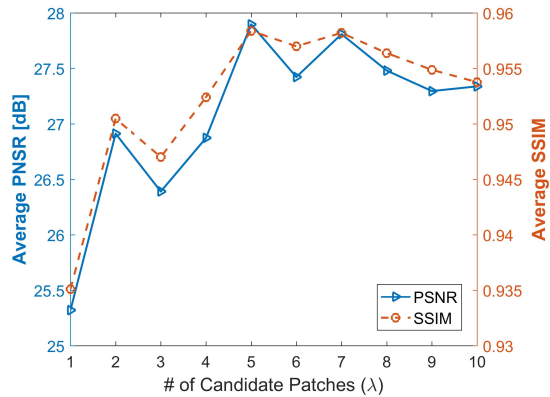


Fig. 13. Varying the number of candidate patches  $\lambda$ .

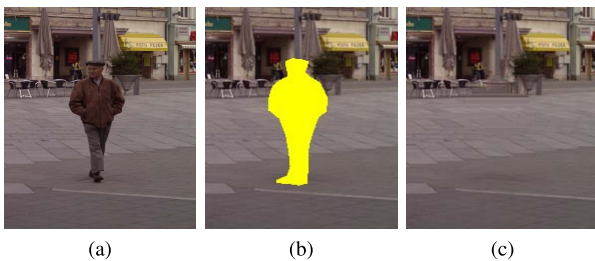


Fig. 14. Example of an inpainted image with texture defects. Some defects appear when there is no patch from the source region to describe the interface of the vertical structure and the horizontal texture that meet in the missing region underneath the man's head. (a) Original. (b) Mask. (c) Proposed.

inpainted images increase by an amount of 2.6 dB and 0.023, respectively, when  $\lambda$  is chosen between 1 and 5, and the values decrease as  $\lambda$  increases from 6 to 10. Thus, it is good to choose about 5 candidate patches (i.e.,  $\lambda \approx 5$ ) for inpainting each target patch using the proposed algorithm.

### F. Limitations

The proposed algorithm can produce some defects when there is no information in the image source region to describe the interface where different texture and structure would meet in the missing region, as shown in Fig. 14. In some cases when the structure is strong, since the priority function is more biased toward strong structure than texture, it sometimes causes problems in term of repeated patches as shown in Fig. 4(e) row 8.

## V. CONCLUSION

In this paper, we have developed an efficient and robust exemplar-based image inpainting method that uses a new nonlocal texture similarity measure to search for several candidate exemplars for each target patch, which are then fused together using the  $\alpha$ -trimmed mean filter to fill in each pixel within the target patch. We evaluate the proposed algorithm against some of the current inpainting algorithms including Li's algorithm, the AAIR algorithm, Buades' algorithm, the ALOHA algorithm, Huang's method, Deng's method, and Buysens' method. For the application of object removal, our

experimental results demonstrate that the proposed algorithm performs better than the other inpainting algorithms w.r.t. both the qualitative analysis and the observer studies. For the application of inpainting texture images, the proposed algorithm outperforms the other inpainting algorithms in terms of the qualitative appearance, even though the quantitative metrics do not always agree. For the application of error concealment, our experimental results show that the proposed algorithm is capable of recovering different textures and structures within the missing blocks of the received image. In addition, our proposed algorithm is also very fast in inpainting images as compared to Li's algorithm, the AAIR algorithm, Buades' algorithm, the ALOHA algorithm, and Deng's method.

## REFERENCES

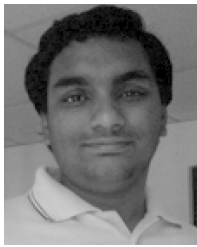
- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM Conf. Comput. Graph.*, 2000, pp. 417–424.
- [2] M. Bertalmio, A. N. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2001, pp. 1355–1362.
- [3] F. Li, L. Pi, and T. Zeng, "Explicit coherence enhancing filter with spatial adaptive elliptical kernel," *IEEE Signal Process. Lett.*, vol. 19, no. 9, pp. 555–558, Sep. 2012.
- [4] T. Chan and J. Shen, "Mathematical models for local nontexture inpaintings," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, Jul. 2006.
- [5] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *J. Vis. Commun. Image Represent.*, vol. 4, no. 12, pp. 436–449, Dec. 2001.
- [6] C. Ballester, 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, no. 8, pp. 1200–1211, Aug. 2001.
- [7] J. Shen, S. H. Kang, and T. F. Chan, "Euler's elastica and curvature-based inpainting," *SIAM J. Appl. Math.*, vol. 63, no. 2, pp. 564–592, Jul. 2002.
- [8] H. Grossauer and O. Scherzer, "Using the complex Ginzburg-Landau equation for digital inpainting in 2D and 3D," in *Proc. 4th Int. Conf. Scale Space Methods Comput. Vis.*, 2003, pp. 225–236.
- [9] J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numer. Algorithms*, vol. 53, no. 1, pp. 67–92, 2010.
- [10] P. Li, S.-J. Li, Z.-A. Yao, and Z.-J. Zhang, "Two anisotropic fourth-order partial differential equations for image inpainting," *IET Image Process.*, vol. 7, no. 3, pp. 260–269, Jun. 2013.
- [11] H. Grossauer, "A combined PDE and texture synthesis approach to inpainting," in *Proc. Eur. Conf. Comput. Vis.*, vol. 3022, T. Pajdla and J. Matas, Eds. Berlin, Germany: Springer, May 2004, pp. 214–224.
- [12] A. Bugeau and M. Bertalmio, "Combining texture synthesis and diffusion for image inpainting," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2009, pp. 26–33.
- [13] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.
- [14] J. Wu and Q. Ruan, "Object removal by cross isophotes exemplar-based inpainting," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Jun. 2006, pp. 810–813.
- [15] L. Cai and T. Kim, "Context-driven hybrid image inpainting," *IET Image Process.*, vol. 9, no. 10, pp. 866–873, Oct. 2015.
- [16] O. Le Meur, J. Gautier, and C. Guillemot, "Super-resolution-based inpainting," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 554–567.
- [17] O. Le Meur, M. Ebdelli, and C. Guillemot, "Hierarchical super-resolution-based inpainting," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3779–3790, Oct. 2013.
- [18] J. Wang, K. Lu, D. Pan, N. He, and B.-K. Bao, "Robust object removal with an exemplar-based image inpainting approach," *Neurocomputing*, vol. 123, pp. 150–155, Jan. 2014.
- [19] V. Kumar, J. Mukherjee, and S. K. Das Mandal, "Image inpainting through metric labeling via guided patch mixing," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5212–5226, Nov. 2016.
- [20] T. Ružić and A. Pižurica, "Context-aware patch-based image inpainting using Markov random field modeling," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 444–456, Nov. 2015.

- [21] Y. Liu and V. Caselles, "Exemplar-based image inpainting using multiscale graph cuts," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1699–1711, May 2013.
- [22] C. Guillemot and O. L. Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 127–144, Jan. 2014.
- [23] O. Le Meur, J. Gautier, and C. Guillemot, "Exemplar-based inpainting based on local geometry," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 3401–3404.
- [24] Z. Li, H. He, H. M. Tai, Z. Yin, and F. Chen, "Color-direction patch-sparsity-based image inpainting using multidirection features," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 1138–1152, Mar. 2015.
- [25] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.
- [26] J. H. Lee, I. Choi, and M. H. Kim, "Laplacian patch-based image synthesis," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2727–2735.
- [27] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen, "Image melding: Combining inconsistent images using patch-based synthesis," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 82:1–82:10, Aug. 2012.
- [28] J. Huang, S. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 129:1–129:10, Jul. 2014. [Online]. Available: <https://github.com/jbhuang0604/StructCompletion>
- [29] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2423–2435, Dec. 2014.
- [30] L.-J. Deng, T.-Z. Huang, and X.-L. Zhao, "Exemplar-based image inpainting using a modified priority definition," *PLoS ONE*, vol. 10, no. 10, pp. 1–18, Oct. 2015. [Online]. Available: <http://www.escience.cn/people/dengliangjian/codes.html>
- [31] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.
- [32] S. Ram, "Sparse representations and nonlinear image processing for inverse imaging solutions," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Arizona, Tucson, AZ, USA, Aug. 2017.
- [33] T. Ogawa and M. Haseyama, "Image inpainting based on sparse representations with a perceptual metric," *EURASIP J. Adv. Signal Process.*, vol. 2013, no. 179, pp. 1–26, Dec. 2013.
- [34] D. Ding, S. Ram, and J. J. Rodríguez, "Perceptually aware image inpainting," *Pattern Recognit.*, vol. 83, pp. 174–184, Nov. 2018.
- [35] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.
- [36] A. Wong and J. Orchard, "A nonlocal-means approach to exemplar-based inpainting," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 2600–2603.
- [37] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, "A variational framework for exemplar-based image inpainting," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 319–347, 2011.
- [38] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.
- [39] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6721–6729. [Online]. Available: <http://vgl.ict.usc.edu/Research/NeuralPatchSynthesis/>
- [40] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9446–9454. [Online]. Available: [https://dmitryulyanov.github.io/deep\\_image\\_prior](https://dmitryulyanov.github.io/deep_image_prior)
- [41] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5505–5514.
- [42] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, (2018). "Free-form image inpainting with gated convolution." [Online]. Available: <https://arxiv.org/abs/1806.03589>
- [43] X. Zhu, Y. Qian, X. Zhao, B. Sun, and Y. Sun, "A deep learning approach to patch-based image inpainting forensics," *Signal Process., Image Commun.*, vol. 67, pp. 90–99, Sep. 2018. [Online]. Available: [https://dmitryulyanov.github.io/deep\\_image\\_prior](https://dmitryulyanov.github.io/deep_image_prior)
- [44] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 341–349.
- [45] S. Ram and J. J. Rodríguez, "Single image super-resolution using dictionary-based local regression," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation*, Apr. 2014, pp. 121–124.
- [46] S. Ram and J. J. Rodríguez, "Image super-resolution using graph regularized block sparse representation," in *Proc. IEEE Southwest Symp. Anal. Interpretation*, Apr. 2016, pp. 69–72.
- [47] V. Duval, J. F. Aujol, and Y. Gousseau, "On the parameter choice for the non-local means," *SIAM J. Imag. Sci.*, vol. 3, pp. 1–37, Mar. 2010. [Online]. Available: <http://hal.archives-ouvertes.fr/docs/00/46/88/56/PDF/nlmeans2.pdf>
- [48] I. Pitas and A. N. Venetsanopoulos, "Order statistics in digital image processing," *Proc. IEEE*, vol. 80, no. 12, pp. 1893–1921, Dec. 1992.
- [49] J. Bednar and T. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 1, pp. 145–153, Feb. 1984.
- [50] X. Li, "Image recovery via hybrid sparse representations: A deterministic annealing approach," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 953–962, Sep. 2011. [Online]. Available: <http://www.csee.wvu.edu/~xinli/demo/inpainting.html>
- [51] B. Wohlberg, "Inpainting by joint optimization of linear combinations of exemplars," *IEEE Signal Process. Lett.*, vol. 18, no. 1, pp. 75–78, Jan. 2011. [Online]. Available: <http://math.lanl.gov/~brendt/Software/LibAAIR>
- [52] K. H. Jin and J. C. Ye, "Annihilating filter-based low-rank Hankel matrix approach for image inpainting," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3498–3511, Nov. 2015. [Online]. Available: <http://bispl.weebly.com/aloha-inpainting.html>
- [53] P. Buysens, M. Daisy, D. Tschumperlé, and O. Lézoray, "Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1809–1824, Jun. 2015. [Online]. Available: <http://gmic.eu/>
- [54] N. Abbadieni, "Computational perceptual features for texture representation and retrieval," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 236–246, Jan. 2011.
- [55] J. E. Farrell, "Image quality evaluation," in *Colour Imaging: Vision and Technology*, L. W. MacDonald and M. R. Luo, Eds. New York, NY, USA: Wiley, 1999, pp. 285–313.
- [56] A. B. Mansoor and A. Anwar, "Subjective evaluation of image quality measures for white noise distorted images," in *Proc. 12th Int. Conf. Adv. Concepts Intell. Vis. Syst.*, vol. 6474, 2010, pp. 10–17.
- [57] S. Siegel and N. J. Castellani, Jr., *Nonparametric Statistics for The Behavioral Sciences*, 2nd ed. New York, NY, USA: McGraw-Hill, 1988.
- [58] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [59] S. D. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 296–303, Mar. 2003.
- [60] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.



**Ding Ding** received the B.S. degree in electronic information engineering from the Hefei University of Technology, Hefei, China, in 2010, and the Ph.D. degree in electrical and computer engineering from The University of Arizona, Tucson, AZ, USA in 2017.

Dr. Ding is currently an AI engineer at Midea Emerging Technology Center, San Jose, CA, USA. Her research interests include signal, image, and video processing/analysis, machine learning, image restoration, segmentation and classification, and pattern recognition.



**Sundaresh Ram** (S'04–M'14) received the B.E. degree in electrical and electronics engineering from the College of Engineering, Anna University, Chennai, India, in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Arizona, Tucson, AZ, USA, in 2010 and 2017, respectively. He was a Post-Doctoral Research Fellow with the School of Electrical and Computer Engineering and the School of Biomedical Engineering, Cornell University, Ithaca, NY, USA, from 2017 to 2018.

Dr. Ram is currently a Post-Doctoral Research Fellow with the Department of Radiology, Center for Molecular Imaging, and the Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI, USA. His research interests include signal processing, image and video processing/analysis, machine learning, computational imaging, and compressive sensing and applications of these methods in the biological sciences, conventional and medical imaging, and data science. He is a member of SIAM and SPIE.



**Jeffrey J. Rodríguez** (M'90–SM'02) received the B.S. and Ph.D. degrees from The University of Texas at Austin in 1984 and 1990, respectively, and the M.S. degree from the Massachusetts Institute of Technology in 1986, all in electrical engineering.

Since 1990, he has been a Faculty Member with the Department of Electrical and Computer Engineering, The University of Arizona, Tucson, where he is currently the Director of the Signal and Image Laboratory. His research area includes signal/image/video processing and analysis, with a particular emphasis on automated image analysis. From 2003 to 2008, he was the Co-Director of Connection One, a National Science Foundation research center. From 2005 to 2011, he served on the IEEE Signal Processing Society Technical Committee on Image, Video, and Multidimensional Signal Processing. He has served on the organizing committees for numerous other technical conferences.

Dr. Rodríguez served as the General Chair for the 2016 IEEE Southwest Symposium on Image Analysis and Interpretation and the 2007 IEEE International Conference on Image Processing. From 1996 to 2000, he was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING.