



Perceptually aware image inpainting

Ding Ding ^{a,*}, Sundares Ram ^b, Jeffrey J. Rodriguez ^a



^a The Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721, USA

^b School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14850, USA

ARTICLE INFO

Article history:

Received 20 March 2017

Revised 12 April 2018

Accepted 21 May 2018

Available online 25 May 2018

Keywords:

Exemplar-based image inpainting

Object removal

PAMSE

Texture synthesis

Geometric structure propagation

ABSTRACT

Image inpainting is a process of reconstructing missing regions, or removing unwanted objects automatically by propagating intensity and texture information from surrounding parts of the image in a visually plausible manner. We propose a new exemplar-based image inpainting algorithm, which uses the recently developed metric called the perceptual-fidelity aware mean squared error (PAMSE). The PAMSE is a Gaussian-smoothed mean squared error (MSE) and approximates a weighted sum of the gradient of MSE, the Laplacian of MSE, and MSE itself. We show that, compared to MSE, PAMSE is a promising perceptual fidelity metric for application to image inpainting and leads to better performance in propagating texture and geometric structure simultaneously.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Image inpainting was initially introduced to restore damaged photos and films or to remove unwanted text or objects from images [1–13]. In the majority of cases, the damaged regions are narrow and long, and the missing regions to be removed are within images that are smooth and do not have much texture. The key challenge when inpainting narrow or non-textural regions is to find which direction to propagate the local intensity smoothly from areas surrounding the missing regions. Therefore, image inpainting methods based on partial differential equations (PDEs) have been used to restore those types of images [1–11,14,15]. However, when the missing region is large or contains lots of texture, simply propagating the intensity values into that region will result in a blurred or unnatural appearance unless suitable texture matching is applied. As a result, image inpainting involves two main approaches. One is to propagate the surrounding intensity values smoothly into the missing region. The other is to select pixel data from elsewhere in the image or generate suitable texture to fill the missing regions.

In [16,17], Criminisi et al. proposed the exemplar-based image inpainting algorithm to address the two objectives of image inpainting. Criminisi's image inpainting has two steps. First step applies a priority function to determine the next target patch to be inpainted. The second step minimizes the mean squared error (MSE) to find the best matched patch from elsewhere in the image to fill the target patch. This technique is able to reconstruct most

of the texture within a missing region. Compared to PDE-based inpainting, the exemplar-based inpainting method is not just limited to propagating nearby intensity values, but also searches for similar patches within the whole image. In recent years in image inpainting, the focus has shifted to developing exemplar-based methods [11,18–38]. Some exemplar-based inpainting methods make use of techniques used in texture synthesis to copy and fill missing regions in an image [16,17,29–32,39,40]. Such methods work well in general when the missing region consists of primarily one texture pattern that needs to be recovered. However, when images have geometric structures such as line segments and curves in addition to texture or when the missing region is surrounded by various textural regions, the exemplar-based inpainting does not perform well – i.e., does not connect straight lines well or does not separate different textural regions in a visually plausible way [41–43].

Recently, sparse representation has been used toward solving the inverse problems including image inpainting problem [44–56]. Sparsity-based image inpainting methods represent an image as a sparse combination of an over-complete set of transforms (e.g., wavelet, DCT, contourlet, etc.) and infer the missing regions within the image by adaptively updating their sparse representation. Elad et al. [50] proposed an image inpainting algorithm that separates the image into cartoon and texture layers, and sparsely represented these two layers using two incoherent over-complete transforms. Facciolo et al. [44] proposed an iterative nonlocal variational formulation for inpainting a sparsely sampled image, which exploits the principle of self-similarity of the image. This method uses the inpainted pixels from the previous iteration to update the missing region. Xu and Sun [49] proposed an image inpainting algorithm making use of patch-based sparse representations to inpaint

* Corresponding author.

E-mail addresses: dding@email.arizona.edu (D. Ding), sr2255@cornell.edu (S. Ram), jjrodrig@email.arizona.edu (J.J. Rodriguez).

missing regions within the image. These inpainting methods are suitable for filling the missing regions with composite textures and structures, especially when inpainting missing blocks within the image. However, similar to PDE-based methods, sparse inpainting methods may fail to recover structures and may introduce very smooth results when the missing region is large [49].

Other inpainting algorithms in [41,42,57,58] combine both the PDE-based inpainting and exemplar-based inpainting to recover the textural and structural information within an image. These hybrid algorithms [41,42,57] first decompose the input image into two different images, one containing the high frequency component (edges and details) and the other containing the low frequency component (smooth intensity). Next, the algorithms inpaint the high frequency image using an exemplar-based method and inpaint the low frequency image using a PDE-based method. Then the two inpainted images are combined together to obtain the final inpainted result. However, the improvement of the performance of such hybrid algorithms is small compared to the exemplar-based methods, and the hybrid algorithms have a high computational complexity due to the decomposition and multiple inpainting processes.

In this paper, we present a new exemplar-based and perceptually aware image inpainting algorithm. The objective of the proposed algorithm is to recover a large missing region surrounded by multiple textural and structural regions. Like other exemplar-based image inpainting algorithms, our algorithm also consists of two steps: (1) Determine the next target patch to be inpainted, and (2) select the best patches from elsewhere in the image to fill the missing region. We use a filling order similar to that defined by Criminisi et al. [17]. For the selection of patches, we apply a recently developed metric, called the perceptual-fidelity aware mean squared error (PAMSE) [59] to find the best matched patches. PAMSE is a Gaussian-smoothed MSE, so it is computationally efficient in comparison to the structural similarity (SSIM) metric [54,60], thereby motivating its usage for image inpainting. The PAMSE approximates a weighted sum of the gradient of MSE, the Laplacian of MSE, and MSE itself. Thus, it is capable of extracting a patch by properly weighting the geometric structural and textural information within the patch. The main contributions of this paper are as follows:

- To the best of our knowledge, the proposed algorithm is the first application of PAMSE toward image inpainting. (The PAMSE metric was originally developed by Xue et al. [59] for image quality assessment.)
- The proposed iterative algorithm exploits information not only from the known region but also from the patches inpainted during prior iterations.
- Unlike other exemplar-based inpainting algorithms, the proposed algorithm chooses the next-to-be-inpainted target patch to be centered on the contour surrounding the outer border of the missing pixels.

We compare the proposed algorithm with other current inpainting algorithms with respect to both qualitative analysis and observer studies. Results show that the proposed algorithm outperforms other existing algorithms.

2. Perceptual-fidelity aware mean squared error

In recent years, it is widely accepted that MSE is not an effective metric to indicate the perceptual fidelity of natural images [59]. To measure the quality of images, many perceptual fidelity indices have been created according to human visual system models [59–61]. However, most of the full-reference perceptual fidelity indices, which make reference to the original data, are nonlinear and

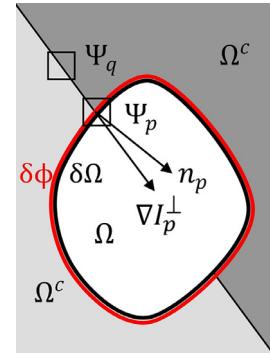


Fig. 1. Original image I with the missing region Ω , and the source region Ω^c . $\delta\Omega$ is the outer border of Ω , $\delta\Omega \in \Omega^c$. $\delta\Phi$ is the single pixel wide contour surrounding the outer border $\delta\Omega$, $\delta\Phi \in \Omega^c$. Ψ_p is a patch centered at the pixel $p \in \delta\Phi$, and Ψ_q is a patch from image source region centered at pixel $q \in \Omega^c \setminus (\delta\Phi \cup \delta\Omega)$. n_p is a unit vector orthogonal to $\delta\Phi$ at the center pixel p , and ∇I_p^\perp is the isophote vector at p .

difficult to apply to some image processing applications [59]. In order to achieve lower computation cost and high perceptual fidelity awareness, Xue et al. [59] proposed the structural mean squared error (SMSE) based on MSE, which is given as

$$\text{SMSE}(u, v) = \frac{1}{M} (\|u - v\|^2 + \alpha_g \|S_g(u - v)\|^2 + \alpha_l \|S_l(u - v)\|^2) \quad (1)$$

where $u, v \in \mathbb{R}^M$ are signals of the same size M . $S_g(\cdot)$ is a gradient operator; e.g., a filter $f = [1, -1]$ can be applied horizontally and vertically to estimate the derivatives needed to compute the gradient. $S_l(\cdot)$ is the Laplacian operator; e.g., a filter $f = [0, 1, 0; 1, -4, 1; 0, 1, 0]$ can be applied to an image to estimate the Laplacian. The values α_g and α_l are constants.

The SMSE is a MSE-like l_2 -norm distance metric. It consists of three different l_2 -norm structural error terms: the MSE itself, the MSE of the gradient, and the MSE of the Laplacian. The SMSE can extract more geometric structural information as compared to MSE, while retaining the low computation of MSE.

By letting $\alpha_g = -2\sigma^2$ and $\alpha_l = \sigma^4$, the SMSE can be turned into a Gaussian-smoothed MSE [59,62]. Xue refers to the Gaussian-smoothed MSE as the PAMSE, defined as

$$\begin{aligned} \text{SMSE}(u, v) &= \frac{1}{M} (\|u - v\|^2 - 2\sigma^2 \|S_g(u - v)\|^2 \\ &\quad + \sigma^4 \|S_l(u - v)\|^2) \approx \frac{1}{M} \|h \otimes (u - v)\|_2^2 \end{aligned} \quad (2)$$

$$\text{PAMSE}(u, v) = \frac{1}{M} \|h \otimes (u - v)\|_2^2 \quad (3)$$

where σ is the standard deviation of the Gaussian filter h , and \otimes is the convolution operator.

3. Method

The objectives of our proposed exemplar-based and perceptually aware image inpainting algorithm are to recover a missing region consisting of textural and structural components and achieve an inpainting result that has high perceptual image quality.

Consider an image I with a missing region denoted as Ω , and define the source region to be the remainder of the image denoted as Ω^c (see Fig. 1). The outer border of Ω is denoted as $\delta\Omega \in \Omega^c$. We define $\delta\Phi \in \Omega^c$ as a single-pixel-wide contour surrounding the outer border $\delta\Omega$ in the source region, and require that all the target patches to be inpainted have the center pixel located on $\delta\Phi$. For the image I , we refer to the intensity at pixel $k \in \Omega \cup \Omega^c$

as $I(x_k, y_k)$, where (x_k, y_k) are the pixel coordinates. Assume that the image is made up of overlapped square patches of size $m \times m$. A single patch with a center pixel p is denoted as Ψ_p . We propagate the intensity along the isophotes, which are the contours of equal intensity values. The main steps of the proposed algorithm are: (i) Inpainting order selection, (ii) finding a candidate patch from the source region, and (iii) filling the missing region.

3.1. Inpainting order selection

The first step of the proposed algorithm is to select the next target patch to be inpainted. Let N be the map indicating whether the pixel belongs to the source region or not.

$$N(x_k, y_k) = \begin{cases} 1, & \text{if } k \in \Omega^c \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

We apply a priority function to select the target patch having the highest priority. The priority function is equal to the product of two terms, a confidence term and a data term. The confidence term $C(\Psi_p)$ for a patch is the ratio of the known pixels to the total pixels in that patch, given by

$$C(\Psi_p) = \frac{|\Psi_p \cap \Omega^c|}{|\Psi_p|} \quad (5)$$

A patch with fewer missing pixels has a higher confidence value. The data term $D(\Psi_p)$ computes the dot product of the isophote vector and the vector normal to the center pixel p on the contour $\delta\Phi$, given by

$$D(\Psi_p) = \frac{|\nabla I_p^\perp \cdot n_p|}{I_{\max}} \quad (6)$$

where ∇I_p^\perp is the isophote vector orthogonal to the gradient ∇I_p at center pixel p , n_p is the unit vector orthogonal to the contour $\delta\Phi$, and I_{\max} is 255 for 8 bits per pixel in our case. The data term $D(\Psi_p)$ is a function of the strength of the isophote projected onto the space of the normal vector. This term favors the reconstruction of local linear structures and textures that intersect the contour $\delta\Phi$ at the pixel p orthogonally. A patch along the direction of the isophote has a higher data-term value. Here, ∇I_p^\perp and n_p are computed as follows:

$$\nabla I_p^\perp = \frac{1}{2I_{\max}} \left\{ [I(x_p, y_p - 1) - I(x_p, y_p + 1)]\mathbf{i} + [I(x_p + 1, y_p) - I(x_p - 1, y_p)]\mathbf{j} \right\} \quad (7)$$

$$n_p^0 = [N(x_p + 1, y_p) - N(x_p - 1, y_p)]\mathbf{i} + [N(x_p, y_p + 1) - N(x_p, y_p - 1)]\mathbf{j} \quad (8)$$

$$n_p = \frac{n_p^0}{\|n_p^0\|_2} \quad (9)$$

where x_p and y_p are the pixel coordinates at pixel p , $I(x_p, y_p)$ is the intensity value at pixel p , $I(x_k, y_k)$ is unknown when pixel $k \in \Omega$, and \mathbf{i} and \mathbf{j} are the standard unit vectors in the x and y direction, respectively. $\|\cdot\|_2$ is the l_2 norm. Thus, the priority function is defined as

$$P(\Psi_p) = C(\Psi_p)D(\Psi_p), \quad p \in \delta\Phi \quad (10)$$

If the image is color instead of grayscale, then Eq. (7) is computed for each color component in the RGB space [19,20,63], and then these computed values are averaged to obtain the mean ∇I_p^\perp value.

When choosing the target patch, we require the center pixel p of the target patch to belong to the contour $\delta\Phi$, which is different from other exemplar-based algorithms where the center pixel p belongs to $\delta\Omega$. We incorporate this change when computing the isophote at pixel p using (7), because choosing $p \in \delta\Omega$ will lead to computation of wrong isophote values as this could involve computing the gradient with unknown pixels from Ω (see Fig. 2).¹ We observe from Fig. 2(c) that when the center pixel p of the target patch lies on $\delta\Omega$ (the outer border of the missing region), the inpainting results have artifacts due to the wrong propagation direction, whereas when p lies on $\delta\Phi$ both texture and structure are recovered well making the result look more visually plausible (see Fig. 2(d)).

3.2. Finding the candidate patch from image source region

We first select a target patch $\Psi_{\hat{p}}$ to be inpainted, by maximizing the priority function from Section 3.1. We then compute the PAMSE between the known pixels from the chosen target patch and the corresponding pixels from every patch Ψ_q from the image source region Ω^c , which is calculated as

$$\text{PAMSE}(\Psi_{\hat{p}}, \Psi_q) = \frac{1}{|\Psi_{\hat{p}} \cap \Omega^c|} \|G_\sigma \otimes (\Psi_{\hat{p}} - \Psi_q)\|_2^2 \quad (11)$$

where σ is the standard deviation from a Gaussian filter G_σ , \otimes is the convolution operator, and $|\Psi_{\hat{p}} \cap \Omega^c|$ is the number of pixels within the target patch that are from the source region Ω^c . When computing the PAMSE using Eq. (11), the unknown pixels within the target patch $\Psi_{\hat{p}}$ and their corresponding pixels within the source region patch Ψ_q are both set to zeros. If the image is color instead of grayscale, then Eq. (11) is computed for each color component in the RGB space [19,20,63], and then these computed values are added up to obtain the total $\text{PAMSE}(\Psi_{\hat{p}}, \Psi_q)$ value. The patch from the image source region that has the least PAMSE value is then chosen as the candidate patch $\Psi_{\hat{q}}$ to the target patch.

Since PAMSE is a Gaussian-smoothed MSE, the parameter of the filter should be chosen carefully. The center of the Gaussian filter is aligned with the center pixel of the patch. We varied the standard deviation σ of the Gaussian filter, and found that $\sigma \in [0.3, 0.5]$ leads to good inpainting results. These σ values make sense because when $\sigma > 0.5$, the MSE between the target patch and a source region patch will be over-smoothed by the Gaussian filter, which will result in more matching errors. When $\sigma < 0.3$, PAMSE will act almost the same as the MSE.

3.3. Fill in the missing pixels of the target patch

The PAMSE helps us to find the candidate patch $\Psi_{\hat{q}}$ for the target patch $\Psi_{\hat{p}}$. In the last step, we copy the intensity values of those pixels of the candidate patch that correspond to pixels in the unknown part of the target patch. Then we update the isophote vector for each pixel that has just been filled in by the isophote of the corresponding pixel within the candidate patch. We also update the confidence value for each missing pixel within the target patch by the confidence value of the pixel \hat{p} , and move the target patch from the missing region to be inpainted into the source region. Our proposed method is summarized in Algorithm 1.

¹ The image in Fig. 2 row 1 is from <http://www.escience.cn/people/dengliangjian/Data.html>. The image in Fig. 2 row 2 is from the USC-SIPI image database. Available: <http://sipi.usc.edu/database/database.php?volume=misc>.

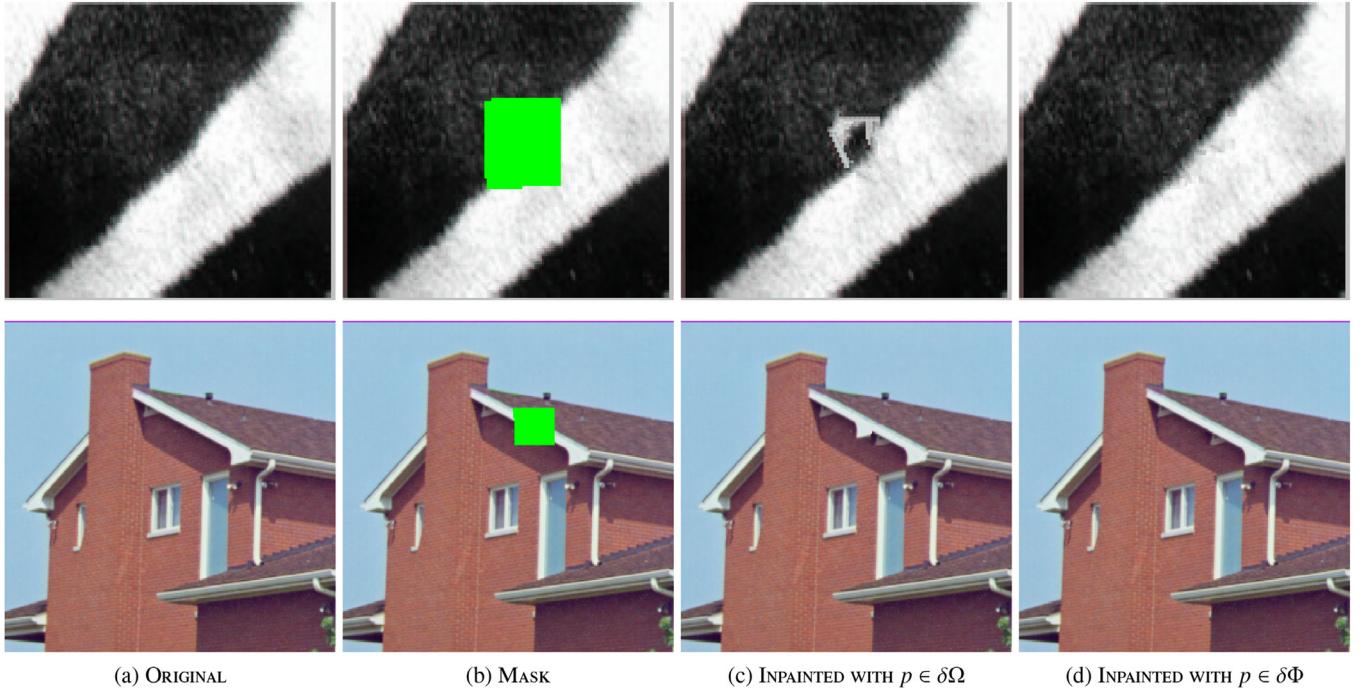


Fig. 2. Effect of choosing the center pixel p of the target patch. Row 1: A simple example image where two textures (white and black) surround the missing region (in green color). Row 2: A more complex image having several textures and a structure surrounding the missing region (in green color).

Algorithm 1: Proposed inpainting algorithm.

input : An image with an unknown region Ω
output: Estimated image with the unknown region filled in

- 1 Compute C at all pixel locations;
- 2 **while** there are missing pixels in Ω **do**
- 3 Compute N and the isophote vector at all pixel locations;
- 4 Find a target patch $\Psi_{\hat{p}}$ by maximizing $P(\Psi_p)$, $p \in \delta\Phi$;
 $\hat{p} = \operatorname{argmax}_p\{P(\Psi_p)\};$
- 5 Find the candidate patch $\Psi_{\hat{q}} \in \Omega^c$ by minimizing
 $\text{PAMSE}(\Psi_{\hat{p}}, \Psi_q)$; $\hat{q} = \operatorname{argmin}_q\{\text{PAMSE}(\Psi_{\hat{p}}, \Psi_q)\};$
- 6 **for** $l \leftarrow 1$ **to** m^2 **do**
- 7 **if** $\Psi_{\hat{p}}(l) = 0$ **then**
- 8 $\Psi_{\hat{p}}(l) = \Psi_{\hat{q}}(l);$
- 9 $\nabla I_{\hat{p}}^\perp(l) = \nabla I_{\hat{q}}^\perp(l);$
- 10 $C(\Psi_{\hat{p}}(l)) = C(\hat{p});$
- 11 Add $\Psi_{\hat{p}}$ to the source region.

4. Results

4.1. PAMSE vs. MSE

In this section, we present some results showing that PAMSE metric outperforms MSE metric when matching a target patch with distinguishing structural information. Fig. 3(a) shows a target patch marked by a red square that contains a horizontal line structure² (see Fig. 3(c) and (f) showing the enlarged target patch). From Fig. 3(e) and (d), we observe that the candidate patch chosen by

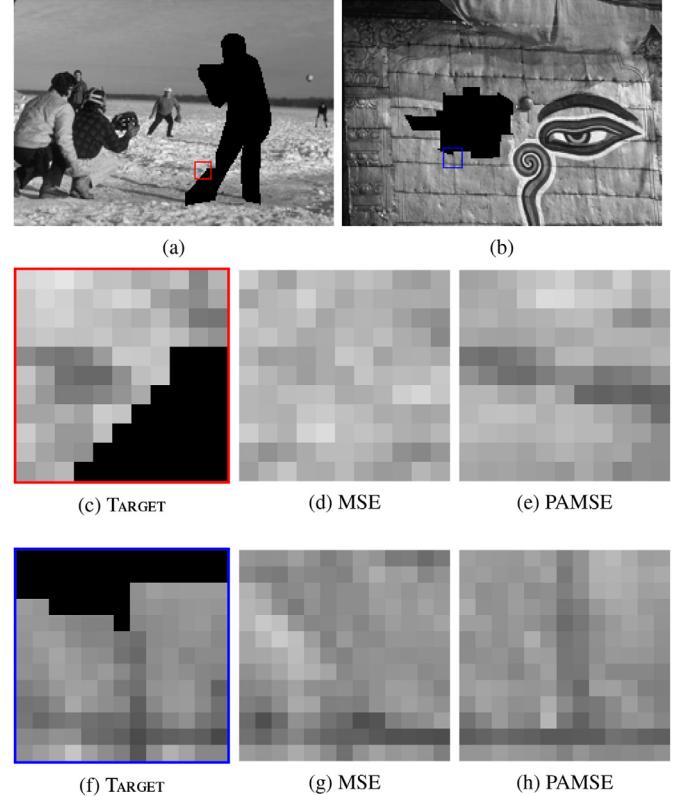


Fig. 3. Comparison of candidate patches chosen by MSE and PAMSE for target patches in two representative images (a), (b) from the Berkeley image dataset.

the PAMSE metric has a similar line structure to that of the target patch, whereas the candidate patch chosen by MSE metric contains no significant structure. Fig. 3(b) shows a target patch marked by a blue square that contains two line structures, horizontal and

² Images in Fig. 3 are from Berkeley segmentation dataset. Available: <https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bbsd/BSDS300/html/dataset/images.html>.

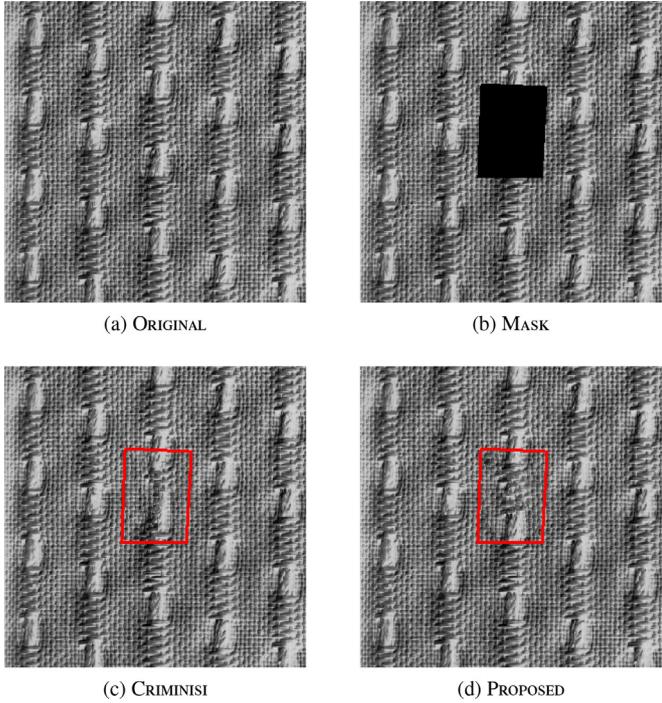


Fig. 4. An example of texture and structure recovery.

vertical. Again, from Fig. 3(h) and (g), we observe that the candidate patch chosen by PAMSE is very similar to the target patch containing the structure information, whereas the candidate patch chosen by MSE fails to preserve the vertical line segment.

In Fig. 4 we show a comparison of the proposed algorithm which uses the PAMSE metric for matching and Criminisi's algorithm using MSE metric for matching towards the task of recovering both structure and texture.³ From Fig. 4(d) we observe that the proposed algorithm recovers the texture while preserving the vertical structure in the missing region, whereas Criminisi's algorithm in Fig. 4(c) fails to recover the correct texture. Fig. 4 also shows an example of restoring an image with both texture and structure.

4.2. Comparison with existing methods

4.2.1. Qualitative analysis

In this section, we show four images in Fig. 5 as examples where an object is to be removed from the image.⁴ For each of these images, we applied the proposed algorithm, Criminisi's exemplar-based inpainting algorithm [17], image recovery via hybrid sparse representations by Li [64], structure completion using planar structure guidance by Huang et al. [65], and annihilating filter-based low-rank Hankel matrix approach (ALOHA) proposed by Jin and Ye [63]. The parameters of all the algorithms are optimized for each image in order to get the best inpainting results.

Object removal applications requiring image inpainting rely on two main aspects of the algorithms: image naturalness (affected by visual artifacts), and the computational speed. In this section, we discuss the image inpainting algorithms with respect to these aspects. Fig. 5 shows image inpainting results of the different algorithms on a variety of images chosen from the Berkeley dataset. The object to be removed in these images is shown in Fig. 5 row 2

³ Fig. 4 is from Brodatz textures dataset. Available: <http://www.ux.uis.no/~tranden/brodatz.html>.

⁴ Images in Fig. 5(b)–(d) are from the Berkeley segmentation dataset. The size of images in Fig. 5(a) is 100 × 150, the size of images in Fig. 5(b) and (c) is 161 × 241, and the size of images in Fig. 5(d) is 241 × 161.

Table 1

Mean and standard deviation run time in HH:MM:SS for 30 Images.

0.8Algorithm	Run time (Std. dev.)
Criminisi	00:02:25 (00:01:52)
Li	00:25:05 (00:12:23)
ALOHA	02:46:27 (01:39:33)
Proposed	00:02:58 (00:02:08)

using a green mask. From Fig. 5 row 3, we observe that Criminisi's image inpainting algorithm can give rise to incorrect texture information in the missing region, resulting in visual artifacts, e.g. dark gaps in the region of water (see Fig. 5(b) row 3), and wrong texture in the wooden fence (see Fig. 5(d) row 3). From Fig. 5 rows 4 and 5, we observe that Li's algorithm and the ALOHA algorithm both lead to very smooth inpainting results, making the images look very blurred. For example, the inpainted region from Fig. 5(b) rows 4 and 5 looks blurred and does not match the surrounding region's texture. The inpainted regions in Fig. 5(c) rows 4 and 5 look unnatural. Huang's algorithm produces a very smooth inpainted region as shown in Fig. 5 row 6, and the structure within the image is not reconstructed (see the left side of the roadway in Fig. 5(a) row 6, and the boundary between the white wall and the ground in Fig. 5(c) row 6). Also, Huang's algorithm produces some wrong texture within the inpainted region (see the gray texture within the area of the river in Fig. 5(b) row 6, and the partial vertical wooden fence within the inpainted region in Fig. 5(d) row 6). In comparison, the results show that our proposed algorithm is able to reconstruct the local texture (see the water region in Fig. 5(b) row 7 and the wall of the house in Fig. 5(c) row 7) and structural details (see Fig. 5(a) and (d) row 7) well without sacrificing the naturalness of the images.

We also compared the run time of the proposed algorithm with that of Criminisi's algorithm, Li's algorithm, and the ALOHA algorithm on a dataset of 30 images taken from the Berkeley image dataset. All the algorithms were coded in MATLAB and were run on an Intel i5-2400 (3.10 GHz) Windows PC, with 8 GB of RAM. The mean and standard deviation of the run time of the compared algorithms are shown in Table 1. Table 1 shows that the proposed algorithm runs 33 s slower than Criminisi's algorithm, 22 min and 7 s faster than Li's algorithm, and 163 min and 29 s faster than the ALOHA algorithm. Huang's method takes about 2–3 s to inpaint the missing region in an image. We do not include the run time of Huang's method in Table 1 as the obtained source code of their method is not completely in MATLAB.

4.2.2. Observer studies

When we use an image inpainting algorithm for object removal, we do not have information about the underlying scene and hence cannot use full-reference image quality assessment metrics, such as peak signal-to-noise ratio and SSIM, for quantitative evaluation. So we evaluated the results of the inpainting algorithms using human observer studies [66,67]. We test the five image inpainting algorithms compared in this paper with 20 images randomly chosen from the Berkeley dataset, where each image has an object to be removed. We presented the results of the five algorithms to ten human evaluators in random order, without identifying the algorithms, and asked them to score the inpainted images between 1 and 5, with 1 being the best inpainting result and 5 being the worst. Table 2 lists all the possible combinations of the scores for each image's five inpainted results from best to worst (when two or more inpainted results are regarded as equally good or equally bad, they share the same averaged score). The evaluators are told to score the five inpainted images using one of the combinations in Table 2, so that we maintain the summation of the five scores

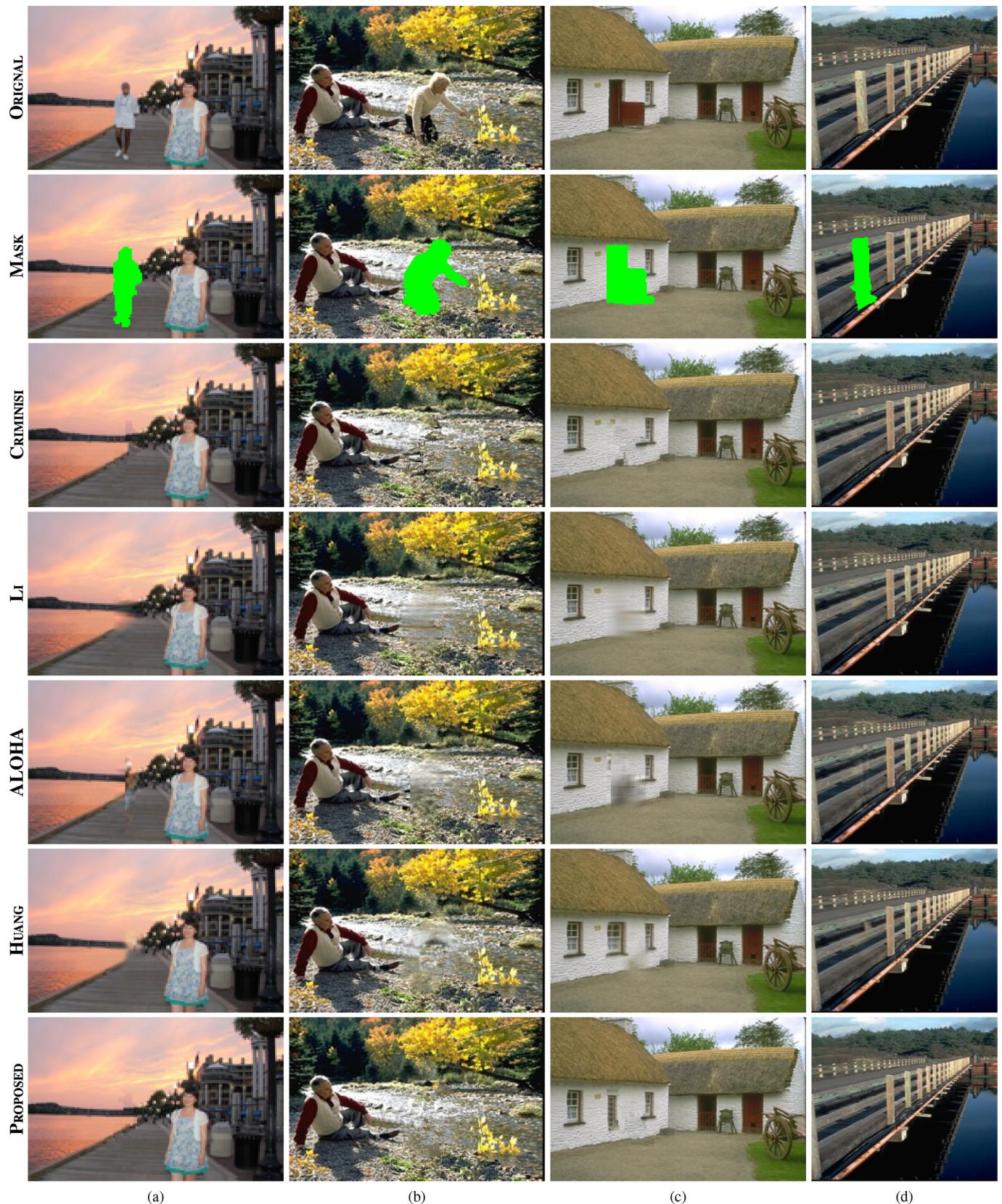


Fig. 5. Image inpainting results of the compared algorithms in the application of object removal. Row 1: Original image. Row 2: Mask. Row 3: Inpainting results using Criminisi's method. Row 4: Li's algorithm. Row 5: ALOHA method. Row 6: Huang's method. Row 7: Proposed algorithm.

Table 2

Possible combinations of scores for each image's five inpainted results from best to worst.

0.8Comb.	Best	Good	Fine	Bad	Worst
1	1	2	3	4	5
2	1.5	1.5	3	4	5
3	1	2.5	2.5	4	5
4	1	2	3.5	3.5	5
5	1	2	3	4.5	4.5
6	1.5	1.5	3.5	3.5	5
7	1.5	1.5	3	4.5	4.5
8	1	2.5	2.5	4.5	4.5
9	2	2	2	4	5
10	2	2	2	4.5	4.5
11	1	3	3	3	5
12	1	2	4	4	4
13	1.5	1.5	4	4	4
14	2.5	2.5	2.5	2.5	5
15	1	3.5	3.5	3.5	3.5
16	3	3	3	3	3

Table 3

Mean score of 20 Images for each algorithm (with grand mean and std. dev.).

0.8Eval.	Criminisi	Li	ALOHA	Huang	Proposed
1	2.15	3.40	4.65	2.85	1.95
2	3.00	3.15	4.50	2.45	1.90
3	2.70	3.35	4.55	2.65	1.75
4	2.50	3.15	4.60	2.70	2.05
5	2.20	3.25	4.60	2.95	2.00
6	2.40	3.35	4.55	2.85	1.85
7	2.20	3.45	4.55	2.80	2.00
8	2.35	3.50	4.60	2.75	1.80
9	2.60	3.38	4.18	3.05	1.80
10	1.93	3.55	4.75	3.00	1.78
μ_{total}	2.40	3.35	4.55	2.81	1.89
σ_{total}	0.30	0.13	0.14	0.17	0.10

equal to 15. As a result, we obtain five scores from each human evaluator, one corresponding to each inpainting algorithm. **Table 3** shows each evaluator's mean scores of the 20 images for each algorithm, along with the grand mean and standard deviation for each algorithm.

Table 3 shows that the proposed algorithm has the lowest mean score for all the images from ten evaluators among the five algorithms.

Further, in order to show that the score of the proposed algorithm will be statistically significantly different from the score of the other algorithms, we apply the Mann–Whitney U test [68] for the reason that it 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 [68]. We define a sample as the collection of the mean scores from all the observers for one inpainting algorithm. In total, we have five samples, one corresponding to each inpainting algorithm. We apply the Mann–Whitney U test between the proposed algorithm and each of the other four image inpainting algorithms. The null and research hypotheses of the tests are described as follows:

H_0 : The two populations of scores have equal median

H_1 : The two populations of scores do not have equal median

In **Table 4**, we show an example of the Mann–Whitney U test between the proposed algorithm and Criminisi's inpainting algorithm. We order the two samples of the ten mean scores of the proposed algorithm and Criminisi's algorithm in **Table 3** from smallest to largest, and assign the ranks from 1–20 (equal scores share the same averaged rank). In **Table 4**, R_1 and R_2 are the sums of the ranks for Criminisi's algorithm and the proposed algorithm,

Table 4

The Mann–Whitney U test between the proposed algorithm and Criminisi's algorithm.

0.8Criminisi	Rank	Proposed	Rank
1.93	7	1.75	1
2.15	12	1.78	2
2.20	13.5	1.80	3.5
2.20	13.5	1.80	3.5
2.35	15	1.85	5
2.40	16	1.90	6
2.50	17	1.95	8
2.60	18	2.00	9.5
2.70	19	2.00	9.5
3.00	20	2.05	11
$R_1=151$		$R_2=59$	
$U_1=4$		$U_2=96$	

respectively. The test statistic U is computed by

$$U = \min(U_1, U_2) \quad (12)$$

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 $n_1 = n_2 = 10$ are the sample sizes. Using **Table 4** we obtain the test statistic value of $U = 4$ using Eq. (12). The appropriate critical value can be found in the Mann–Whitney U test table, and it is 16 based on the sample sizes ($n_1 = 10$ and $n_2 = 10$) and the two-sided level of significance $\alpha = 0.01$. So we reject the null hypothesis because $U = 4 < 16$ at $\alpha = 0.01$, which means the median score of the proposed algorithm is significantly different from the median score of the Criminisi's algorithm. Similarly, we applied the Mann–Whitney U test between the proposed algorithm and the ALOHA algorithm, and between the proposed algorithm and Li's algorithm, and in both cases we are able to reject the null hypothesis. We thus conclude that the median score of proposed algorithm is statistically significantly different from the median score of the other four algorithms in **Table 3**.

4.2.3. Inpainting a specific group of images

When comparing the five algorithms for inpainting images taken randomly from the Berkeley dataset, we observe that the proposed algorithm performs well for inpainting a specific group of images, which have different textural regions meeting at the border of the missing region to be inpainted. In this section, we present three images in **Fig. 6** as examples.⁵ The image in **Fig. 6(a)** row 2 has two different types of textural regions (the window panes and the roof), which meet at the border of the missing region (denoted by a green region), and in **Fig. 6(b)** and (c) row 2, there are different types of textural regions (see the rocks in **Fig. 6(b)** row 2 and the green grass, tree region behind, and roadway in **Fig. 6(c)** row 2) that meet at the border of the missing region. From the inpainted images shown in **Fig. 6**, we can again observe that Li's algorithm and the ALOHA algorithm produce very smooth inpainted regions as shown in **Fig. 6(b)** and (c), rows 4 and 5. Also, the results presented in **Fig. 6(a)** rows 4 and 5 have ghosting-like artifacts, thereby making them look unreal. Huang's algorithm produces smooth inpainted regions shown in **Fig. 6(a)** and (b) row 6. Also, the inpainted region shown in **Fig. 6(c)** row 6 does not look natural. Criminisi's algorithm is able to inpaint the texture of the missing region; however, some structural information appearing at the boundary between different textural regions

⁵ The size of images in **Fig. 6(a)** and (c) is 241×161 , and the size of images in **Fig. 6(b)** is 161×241 .

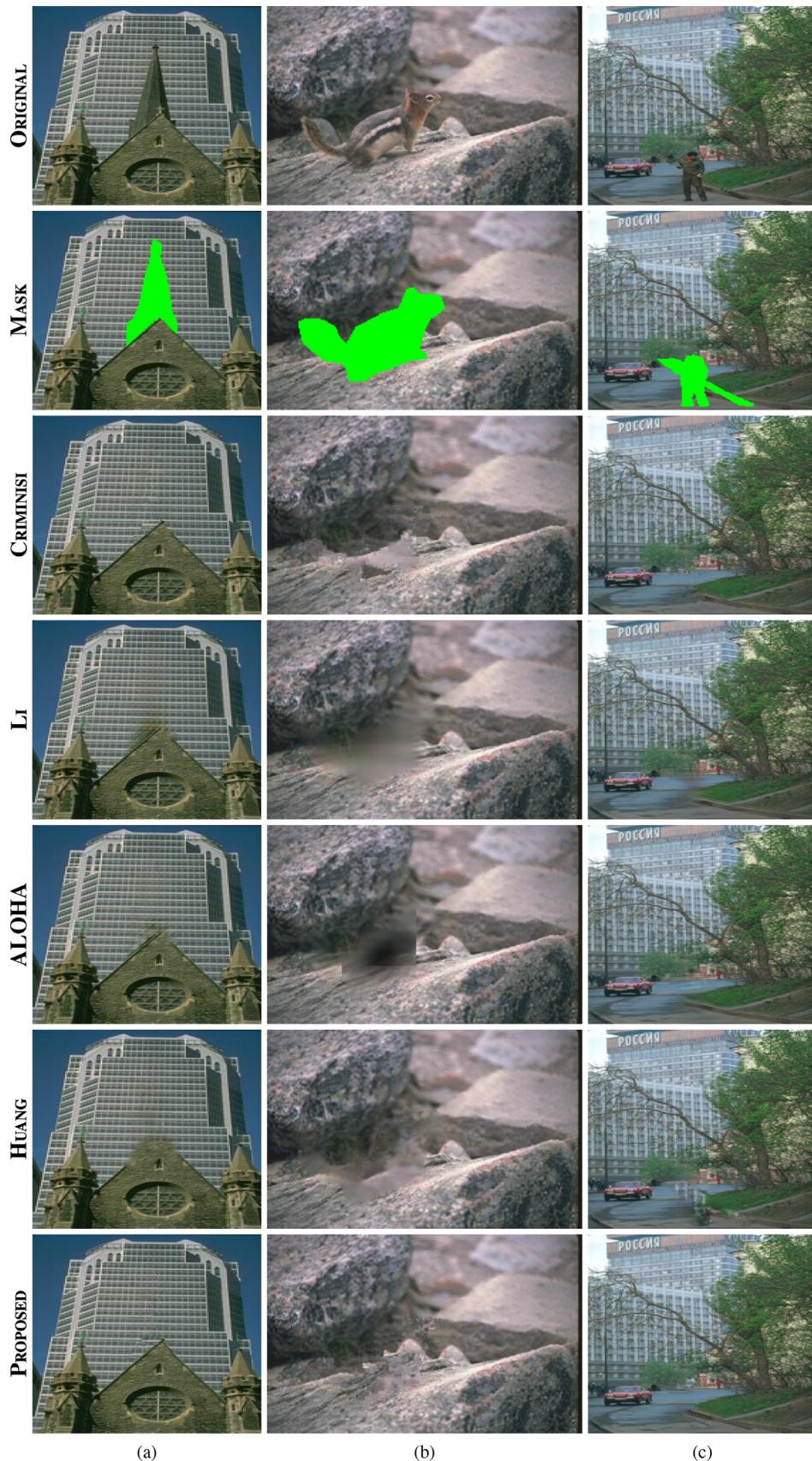


Fig. 6. Image inpainting results of the compared algorithms in the application of object removal in a specific category of images where different textural regions meet at the border of the object to be removed. Row 1: Original image. Row 2: Mask. Row 3: Inpainting results using Criminisi's method. Row 4: Li's algorithm. Row 5: ALOHA method. Row 6: Huang's method. Row 7: Proposed algorithm.

is not recovered (see the right edge of the roof in Fig. 6(a) row 3 and the pavement as shown in Fig. 6(c) row 3). Also, in Fig. 6(b) row 3, Criminisi's algorithm generates a dark gray triangular region within the missing region, which does not look natural. The proposed algorithm performs well in inpainting both the textural and structural information of the missing region as shown in Fig. 6(a)–(c) row 7.

5. Conclusion

In this paper, we have proposed a new exemplar-based perceptually aware image inpainting algorithm. Our algorithm comprises three steps: (1) Determining the next target patch to be inpainted by maximizing a priority function, (2) finding a suitable candidate patch for the target patch, using the PAMSE metric, and (3) filling the missing pixels within the target patch using the chosen candidate patch. The proposed algorithm exploits the matching capability of the PAMSE metric to find a more suitable candidate patch for the target patch, thereby obtaining very plausible inpainting results for a diverse collection of images.

We performed experiments to compare the proposed algorithm with four current algorithms on a dataset of 20 images. From Fig. 5, we observe that the proposed algorithm qualitatively outperforms the other compared algorithms, leading to more visually plausible inpainted images as it recovers both geometric structures (straight lines) and textures well. Further, we conducted observer studies to compare the quantitative performance of the compared algorithms. The results in Table 3 show that the proposed algorithm consistently ranks better than the other algorithms. We also compared the run times of the compared algorithms. From Table 1, we observe that the proposed algorithm is faster than Li's algorithm and the ALOHA algorithm. However, the proposed algorithm runs slower than Criminisi's algorithm because of the Gaussian convolution in the computation of the PAMSE metric.

References

- [1] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Proc. ACM Conf. Comp. Graphics, 2000, pp. 417–424.
- [2] M. Bertalmio, A.N. Bertozzi, G. Sapiro, Navier-strokes, fluid dynamics, and image and video inpainting, in: Proc. IEEE Intl. Conf. Comp. Vis. and Pattern Recognit., 2001, pp. I355–I362.
- [3] T. Chan, J. Shen, Mathematical models for local nontexture inpaintings, SIAM J. Appl. Math. 62 (3) (2006) 1019–1043.
- [4] T. Chan, J. Shen, Non-texture inpainting by curvature-driven diffusions, J. Vis. Commun. Image Represent. 4 (12) (2001) 436–449.
- [5] T.F. Chan, S.H. Kang, Kang, J. Shen, Euler's elastica and curvature based inpaintings, SIAM J. Appl. Math. 63 (2) (2002) 564–592.
- [6] H. Grossauer, O. Scherzer, Using the complex Ginzburg-Landau equation for digital inpainting in 2D and 3D, in: Proc. 4th Intl. Conf. Scale Space Methods in Comp. Vis., 2003, pp. 225–236.
- [7] Z. Xu, X. Lian, L. Feng, Image in-painting algorithm based on partial differential equation, ISECS Intl. Colloq. Comput. Commun. Control Manag. 1 (2008) 120–124.
- [8] P. Li, S. Li, Z. Yao, Z. Zhang, Two anisotropic fourth-order partial differential equations for image inpainting, IET Image Process. 7 (3) (2013) 260–269.
- [9] N. Gopinath, A. K, J.A. Shankar, J.J. Nair, Complex diffusion based image inpainting, in: Proc. 1st. Intl. Conf. Next Gen. Comput. Tech., 2015, pp. 976–980.
- [10] F. Li, L. Pi, T. Zeng, Explicit coherence enhancing filter with spatial adaptive elliptical kernel, IEEE Signal Process. Lett. 19 (9) (2012) 555–558.
- [11] C. Guillemot, O. Le Meur, Image inpainting: overview and recent advances, IEEE Signal Process. Mag. 31 (1) (2014) 127–144.
- [12] S. Ram, Sparse representations and nonlinear image processing for inverse imaging solutions, Dept. Elect. Comput. Engg., Univ. Arizona, Tucson, AZ, 2017 Ph.D. dissertation.
- [13] D. Ding, Image inpainting based on exemplars and sparse representation, Dept. Elect. Comput. Engg., Univ. Arizona, Tucson, AZ, 2017 Ph.D. dissertation.
- [14] W. Casaca, M. Boaventura, M.P.D. Almeida, L.G. Nonato, Combining anisotropic diffusion, transport equation and texture synthesis for inpainting textural images, Pattern Recogn. Lett. 36 (2014) 36–45.
- [15] P. Peter, S. Hoffmann, F. Nedwed, L. Hoeltgen, J. Weickert, Evaluating the true potential of diffusion-based inpainting in a compression context, Signal Process.: Image Commun. 46 (2016) 40–53.
- [16] A. Criminisi, P. Perez, K. Toyama, Object removal by exemplar-based inpainting, in: Proc. IEEE Intl. Conf. Comp. Vis. and Pattern Recognit., 2003, pp. 1–8.
- [17] A. Criminisi, P. Perez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Trans. Image Process. 13 (9) (2004) 1200–1212. github.com/ikuwow/inpainting_criminisi2004.
- [18] J. Wu, Q. Ruan, Object removal by cross isophotes exemplar-based image inpainting, in: Proc. IEEE Intl. Conf. Pattern Recognit., 2006, pp. 810–813.
- [19] T.T. Dang, C. Larabi, A. Beghdadi, Multi-resolution patch and window-based priority for digital image inpainting problem, in: 3rd Intl. Conf. on Image Process. Theory, Tools and Appl., 2012, pp. 280–284.
- [20] B. Wohlberg, Inpainting by joint optimization of linear combinations of exemplars, IEEE Signal Process. Lett. 18 (1) (2011) 75–78.
- [21] A. Wong, J. Orchard, A nonlocal-means approach to exemplar-based inpainting, in: Proc. IEEE Intl. Conf. Image Processing, 2008, pp. 2600–2603.
- [22] Z. Lu, H. Huang, L. Li, D. Cheng, A novel exemplar-based image completion scheme with adaptive TV constraint, in: Proc. 4th Intl. Conf. Genetic and Evolutionary Computing, 2010, pp. 94–97.
- [23] P. Arias, G. Facciolo, V. Caselles, G. Sapiro, A variational framework for exemplar-based image inpainting, Intl. J. Comp. Vis. 93 (3) (2011) 319–347.
- [24] C. Xiang, Y. Cao, P. Duan, L. Shi, An improved exemplar-based image inpainting algorithm, in: Proc. 9th Intl. Conf. Comp. Sci. Edu., 2014, pp. 770–775.
- [25] L. Cai, T. Kim, Context-driven hybrid image inpainting, IET Image Process. 9 (10) (2015) 866–873.
- [26] R. Rodriguez-Sanchez, J.A. Garcia, J. Fdez-Valdivia, Image inpainting with nonsubsampled contoured transform, Pattern Recogn. Lett. 34 (13) (2013) 1508–1518.
- [27] V. Kumar, J. Mukherjee, S.K.D. Mandal, Image inpainting through metric labeling via guided patch mixing, IEEE Trans. Image Process. 25 (11) (2016) 5212–5226.
- [28] T. Ruzic, A. Pizurica, Context-aware patch-based image inpainting using Markov random field modeling, IEEE Trans. Image Process. 24 (1) (2015) 444–456.
- [29] Z. Li, H. He, H. Tai, Z. Yin, F. Chen, Color-direction patch-sparsity-based image inpainting using multidirection features, IEEE Trans. Image Process. 24 (3) (2015) 1138–1152.
- [30] O. Le Meur, J. Gautier, C. Guillemot, Exemplar-based inpainting based on local geometry, in: Proc. 18th IEEE Intl. Conf. on Image Process., 2011, pp. 3401–3404.
- [31] O. Le Meur, J. Gautier, C. Guillemot, Super-resolution-based inpainting, in: Proc. 12th Eur. Conf. Comput. Vis., 2012, pp. 554–567.
- [32] O. Le Meur, J. Gautier, C. Guillemot, Hierarchical super-resolution-based inpainting, IEEE Trans. Image Process. 22 (10) (2013) 3779–3790.
- [33] Y. Liu, V. Caselles, Exemplar-based image inpainting using multiscale graph cuts, IEEE Trans. Image Process. 22 (5) (2013) 1699–1711.
- [34] J. Lee, D. Lee, R. Park, Robust exemplar-based inpainting algorithm using region segmentation, IEEE Trans. Consumer Electron. 58 (2) (2012) 553–561.
- [35] L. Deng, T. Huang, X. Zhao, Exemplar-based image inpainting using a modified priority definition, PLoS ONE 10(10) (2015). doi:10.1371/journal.pone.0141199.
- [36] D. Ding, S. Ram, J.J. Rodriguez, Image inpainting using nonlocal texture matching and nonlinear filtering, IEEE Trans. Image Process., under revision.
- [37] S. Al-Takrouri, A.V. Savkin, A model validation approach to texture recognition and inpainting, Pattern Recogn. 43 (6) (2010) 2054–2067.
- [38] H. Wang, L. Jiang, R. Liang, X. Li, Exemplar-based image inpainting using structure consistent patch matching, Neurocomputing 269 (C) (2017) 90–96.
- [39] N. Komodakis, G. Tziritas, Image completion using efficient belief propagation via priority scheduling and dynamic pruning, IEEE Trans. Image Process. 16 (11) (2007) 2649–2661.
- [40] A.A. Efros, T.K. Leung, Texture synthesis by non-parametric sampling, in: Proc. Seventh IEEE Intl. Conf. on Comp. Vis., 1999, pp. 1033–1038.
- [41] A. Bugean, M. Bertalmio, Combining texture synthesis and diffusion for image inpainting, in: Proc. Intl. Conf. Comp. Vis. Theory and Appl., 2009, pp. 26–33.
- [42] H. Grossauer, A combined PDE and texture synthesis approach to inpainting, in: Proc. Eru. Conf. Comp. Vis., 2002, 2004, pp. 214–224.
- [43] P. Buyssens, M. Daisy, D. Tschumperle, O. Lezoray, Exemplar-based inpainting: technical review and new heuristics for better geometric reconstructions, IEEE Trans. Image Process. 24 (6) (2015) 1809–1824.
- [44] G. Facciolo, P. Arias, V. Caselles, G. Sapiro, Exemplar-Based Interpolation of Sparsely Sampled Images, in: D. Cremers, Y. Boykov, A. Blake, F. Schmidt (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition, 5681, Springer, Berlin, 2009, pp. 331–344.
- [45] Z. Belhachmi, D. Bucur, B. Burgeth, J. Weickert, How to choose interpolation data in images, SIAM J. Appl. Math. 70 (1) (2009) 333–352.
- [46] Y. Chen, R. Ranftl, T. Pock, A bi-level view of inpainting-based image compression, in: Proc. 19th Comp. Vis. Winter Workshop, 2014, pp. 19–26.
- [47] L. Hoeltgen, S. Setzer, J. Weickert, An Optimal Control Approach to Find Sparse Data for Laplace Interpolation, in: A. Heyden, F. Kahl, C. Olsson, M. Oskarsson, X.C. Tai (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition, 8081, Springer, Berlin, 2013, pp. 151–164.
- [48] P. Peter, J. Weickert, Compressing images with diffusion- and exemplar-based inpainting, in: J.F. Aujol, M. Nikolova, N. Papadakis (Eds.), Scale-Space and Variational Methods in Computer Vision, 9087, Springer, Berlin, 2015, pp. 154–165.
- [49] Z. Xu, J. Sun, Image inpainting by patch propagation using patch sparsity, IEEE Trans. Image Process. 19 (5) (2010) 1153–1165.
- [50] M. Elad, J.L. Starck, P. Querre, D.L. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis, Appl. Comput. Harmon. Anal. 19 (3) (2005) 340–358.
- [51] M.J. Fadili, J.L. Starck, F. Murtagh, Inpainting and zooming using sparse representations, Comput. J. 52 (1) (2009) 64–79.

- [52] O.G. Guleryuz, Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising—part i: theory, *IEEE Trans. Image Process.* 15 (3) (2006) 539–554.
- [53] O.G. Guleryuz, Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising—part ii: adaptive algorithms, *IEEE Trans. Image Process.* 15 (3) (2006) 555–571.
- [54] T. Ogawa, M. Haseyama, Image inpainting based on sparse representations with a perceptual metric, *EURASIP J. Adv. Signal Process.* 2013 (179) (2013) 1–26.
- [55] S. Ram, J.J. Rodriguez, Single image super-resolution using dictionary-based local regression, in: Proc. IEEE Southwest Sym. Image Anal. Interp., 2014, pp. 121–124.
- [56] S. Ram, J.J. Rodriguez, Image super resolution using graph regularized block sparse representation, in: Proc. IEEE Southwest Sym. Anal. Interp., 2016, pp. 69–72.
- [57] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, Simultaneous structure and texture image inpainting, *IEEE Trans. Image Process.* 12 (8) (2003) 882–889.
- [58] S. Li, M. Zhao, Image inpainting with salient structure completion and texture propagation, *Pattern Recogn. Lett.* 32 (9) (2011) 1256–1266.
- [59] W. Xue, X. Mou, L. Zhang, X. Feng, Perceptual fidelity aware mean squared error, in: Proc. IEEE Int'l Conf. Comp. Vis., 2013, pp. 705–712.
- [60] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.
- [61] A. Mittal, A.K. Moorthy, A.C. Bovik, No-reference image quality assessment in the spatial domain, *IEEE Trans. Image Process.* 21 (12) (2012) 4695–4708.
- [62] M. Lindenbaum, M. Fischer, A. Bruckstein, On Gabor's contribution to image enhancement, *Pattern Recogn.* 27 (1) (1994) 1–8.
- [63] K.H. Jin, J.C. Ye, Annihilating filter-based low-rank Hankel matrix approach for image inpainting, *IEEE Trans. Image Process.* 24 (11) (2015) 3498–3511. [bispl.weebly.com/aloha-inpainting.html](http://weebly.com/aloha-inpainting.html).
- [64] X. Li, Image recovery via hybrid sparse representations: a deterministic annealing approach, *IEEE J. Sel. Topics Signal Process.* 5 (5) (2011) 953–962. <http://www.csee.wvu.edu/~xini/demo/inpainting.html>.
- [65] J. Huang, S. Kang, N. Ahuja, J. Kopf, Image completion using planar structure guidance, *ACM Trans. Graphics* 33 (4) (2014) 129. github.com/jbhuang0604/StructCompletion.
- [66] J.E. Farrell, Image quality evaluation, in: *Color Imaging: Vision and Technology*, Wiley, New York, 1999, pp. 285–313.
- [67] A.B. Mansoor, A. Anwar, Subjective evaluation of image quality measures for white noise distorted images, in: Proc. 12th Intl. Conference on Adv. Concepts for Intell. Vis. Sys., Lecture notes in Comp. Sci., 6474, 2010, pp. 10–17.
- [68] S. Siegel, N.J. Castellan, *Nonparametric Statistics for the Behavioral Sciences*, 2, New York: McGraw-Hill Book Company, 1988.



Ding Ding received the B.S. in electronic information engineering from 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. Her research interests include signal, image and video processing/analysis, machine learning, image restoration, segmentation and classification, and pattern recognition.



Sundareswaran Ram 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 the Ph.D. degrees in electrical and computer engineering from the University of Arizona, Tucson, AZ, USA, in 2010 and 2017, respectively. He is currently a postdoctoral research fellow in the school of electrical and computer engineering and the school of biomedical engineering at Cornell University, Ithaca, NY, USA. His research interests include signal processing, image and video processing, inverse problems, 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 IEEE and SIAM.



Jeffrey J. Rodriguez received the BS and Ph.D. degrees in electrical engineering from The University of Texas at Austin in 1984 and 1990, respectively, and the SM degree in electrical engineering from the Massachusetts Institute of Technology in 1986. Since 1990 he has been a faculty member in the Dept. of Electrical and Computer Engineering at The University of Arizona in Tucson, where he is director of the Signal and Image Laboratory (Sall). His research area includes signal/image/video processing and analysis, with particular emphasis on automated image analysis. He served as General Chair of the 2016 IEEE Southwest Symp. on Image Analysis and Interpretation, General Chair of the 2007 IEEE Int. Conf. on Image Processing, and has served on the organizing committees for numerous other technical conferences. From 2005 to 2011, he served on the IEEE Signal Processing Society Technical Committee on Image, Video, and Multidimensional Signal Processing. From 2003 to 2008, he was Co-Director of Connection One, a National Science Foundation research center. From 1996 to 2000, he was Associate Editor of the IEEE Trans. on Image Processing. He is a Senior Member of the IEEE.