

## A MODIFIED PATCH PROPAGATION-BASED IMAGE INPAINTING USING PATCH SPARSITY\*

S. HESABI, \*\* AND N. MAHDAVI-AMIRI

Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, I. R. of Iran  
Email: hesabi@alum.sharif.edu

**Abstract**– We present a modified exemplar-based inpainting method in the framework of patch sparsity. In the exemplar-based algorithms, the unknown blocks of target region are inpainted by the most similar blocks extracted from the source region, using the available information. Defining a priority term to decide the filling order of missing pixels ensures the connectivity of the object boundaries. In the exemplar-based patch sparsity approach, a sparse representation of missing pixels is considered to define a new priority term and the unknown pixels of the fill-front patch is inpainted by a sparse combination of the most similar patches. Here, we modify this representation of the priority term and take a measure to compute the similarities between fill-front and candidate patches. Also, a new definition is proposed for updating the confidence term to illustrate the amount of the reliable information surrounding pixels. Comparative reconstructed test images show the effectiveness of our proposed approach in providing high quality inpainted images.

**Keywords**– Image inpainting, texture synthesis, patch sparsity

### 1. INTRODUCTION

Restoring damaged regions of an image and removing undesired objects are termed as image inpainting. The basic idea is to fill in the lost or broken parts of an image using the surrounding information in such a way that the final restored result appears to be natural to a not familiar observer.

The applications of image inpainting techniques include removal of scratches in old photographs, repairing damaged regions of an image, removal of undesired objects, and restoration of missing blocks of transmitted images.

The user is to identify the missing or damaged areas objectively, since these areas cannot be easily classified. These specified regions are called inpainting domain or target regions and the undamaged parts, whose information is used to repair the target region, are called source regions.

Recently, several image inpainting approaches have been developed. They are roughly categorized into two main types: PDE (Partial Differential Equation)-based methods, and texture synthesis approaches. PDE-based methods use partial differential equations, which propagate edge information along isophote (i.e., a line with all the points having the same gray value) directions with diffusion techniques. Bertalmio et al. [1] were the first to introduce an image inpainting method. Their PDE-based scheme propagated boundary information of the inpainting domain along the isophote directions. Inspired by the work in [1], two other PDE-based algorithms [2, 3] were proposed by Chan and Shen. The Total Variational (TV) model [2] uses an Euler-Lagrange equation coupled with an anisotropic diffusion to preserve the direction of isophotes. This method does not restore a single object well when its disconnected remaining elements are separated far apart within the target region. The Curvature Driven Diffusion (CDD) model [3] considers geometric information by defining the strength of isophotes. This extended version of the TV approach can inpaint larger damaged regions.

---

\*Received by the editors December 8, 2012; Accepted December 29, 2013.

\*\*Corresponding author

A simple inpainting algorithm based on prior models was presented by Roth and Black [4]. They modified the diffusion technique of denoising approaches to learn image statistics from natural images and then applied it to the target region. Noori et al. [22] proposed a method based on bilateral filters which was originally used for image denoising. They iteratively convolved the damaged image with a space variant kernel to restore the thin regions.

In almost all PDE-based algorithms, blurring artifacts may be produced when the missing regions are large and textured. So, these methods perform well on images with pure structures or thin target regions.

To fill in large regions with pure textures, the second class of approaches, texture synthesis techniques were proposed. The common idea in these methods is to duplicate the information on the source region into the target region [5-10]; hence, the texture information is preserved.

Texture synthesis approaches are classified into pixel-based sampling [5-7] and patch-based sampling [8-10] according to the sample texture size. Since the filling process in pixel-based scheme is performed pixel by pixel, the algorithm is very slow. However, the speed of patch-based sampling was greatly improved even though filling in the target region is by blocks of pixels, but discontinuous flaws between neighboring patches still remains.

As natural images usually contain both structure and texture components, Bertalmio et al. [11] decomposed the input image into its structure and texture components and then restored them separately. The final outcome was the sum of two reconstructed components. This method is not appropriate for repairing large and thick damaged regions, because the PDE-based approach being used to construct the structure component often admits blurring artifacts. Criminisi et al. [12] presented an exemplar-based inpainting technique to propagate the known patches (i.e., exemplars) into the missing ones by ordering the synthesizing process. They can remove large objects from the image according to the defined patch priority values assigned to the pixel. Constructing information on both the structure and the texture characteristics for large and thick damaged regions is a special feature of this method.

Other exemplar-based schemes were also proposed [13-15]. Compared with the PDE-based approaches, the exemplar-based inpainting algorithms have produced plausible results; however, they appear to fail on repairing other types of structures, such as curves, and are often faced with some artifacts in the output image.

Some approaches based on image sparse representation were also introduced for the inpainting problem [16-19]. In these methods, an image is presented by a sparse combination of an overcomplete set of transformations (e.g., wavelet, contourlet, DCT, etc.), and then the missing pixels are inferred by adaptively updating the sparse representation. In [16], Elad et al. proposed an approach to separate the image into cartoon (structure) and texture components, and then represented the sparse combination of the two obtained components by two incoherent over-complete transformations. Although this approach can effectively fill in the regions with structure and texture, it may fail to repair the structure or might produce smoothing defects similar to the PDE-based approaches. Fadili et al. [17] proposed a sparse representation-based iterative algorithm for image inpainting. They used the Expectation Maximization (EM) framework to consider recovering the missing samples based on representations. The proposed approach in [18] is considered a simple exemplar-based model via global optimization. Hence, some problems associated with progressive fill-ins were avoided. Xu and Sun [19] presented an exemplar-based inpainting method using a patch sparsity representation. They introduced the idea of sparse representation under the assumption that the missing patch could be represented by sparse linear combinations of candidate patches. Then, a constrained optimization model was proposed for the patch inpainting. Nonetheless, at times the edges in the filled regions are not connected properly.

Here, we intend to improve the patch sparsity image inpainting scheme based on the patch propagation scheme proposed in [19].

The remainder of our work is organized as follows. In Section 2, we explain the patch sparsity-based image inpainting. We present our proposed method in Section 3. Performance of our technique is investigated in Section 4. Finally, Section 5 gives the concluding remarks.

## 2. PATCH SPARSITY-BASED IMAGE INPAINTING

We use the common notations being used in the inpainting literature. The target region, i.e., the region to be filled in, is shown by  $\Omega$ , and its boundary is denoted by  $\delta\Omega$ . The source region,  $\Phi$ , which remains fixed throughout the algorithm, supplies samples to fill in the missing regions (refer to Fig. 1).

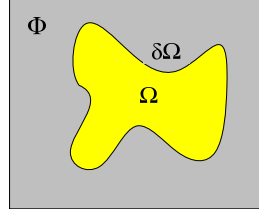


Fig. 1. Notation diagram: The original image to be filled with target region  $\Omega$ , its boundary  $\delta\Omega$  and the source region  $\Phi$

The conventional exemplar-based method [12] is as follows.

**Algorithm 1:** A conventional exemplar-based method.

**Step 1:** For each point  $p$  on the boundary  $\delta\Omega$ , construct a patch  $\psi_p$ , with  $p$  in the center of the patch.

**Step 2:** Compute the patch priority  $P(p)$ :  $P(p)$  is defined as the product of two terms: a confidence term  $C(p)$ , and a data term  $D(p)$  (see Fig. 2):

$$P(p) = C(p) \cdot D(p), \quad (1)$$

$$C(p) = \frac{\sum_{q \in \psi_p \cap \bar{\Omega}} C(q)}{|\psi_p|}, \quad D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha}, \quad (2)$$

where,  $|\psi_p|$  is the area of  $\psi_p$ ,  $\alpha$  is a normalization factor (e.g.,  $\alpha = 255$  for a typical grey-level image),  $n_p$  is a unit vector orthogonal to the boundary at the point  $p$ , and  $\nabla I_p^\perp$  is an isophote vector.  $D(p)$  lets linear structures to be synthesized first, and thus propagates securely into the target region,  $C(p)$  illustrates the amount of the reliable information surrounding the pixel  $p$  and is initialized to be  $C(p) = 0, \forall p \in \Omega$ , and  $C(p) = 1, \forall p \notin \Omega$ .

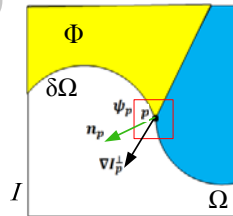


Fig. 2. Given the patch  $\psi_p$  for image  $I$ ,  $n_p$  is the normal to the boundary at  $p$ , and  $\nabla I_p^\perp$  is the isophote vector

**Step 3:** Find the patch  $\psi_{\hat{p}}$  with the highest priority being filled in with the information extracted from the source region  $\Phi$  (Fig. 3a).

**Step 4:** Make a global search on the whole image to find a patch  $\psi_q$  having the most similarity with  $\psi_{\hat{p}}$ . Formally,

$$\psi_{\hat{q}} = \arg \min_{\psi_q \in \Phi} d(\psi_{\hat{p}}, \psi_q) \quad (3)$$

where the distance  $d$  between two generic patches is simply defined as the sum of squared differences (SSDs) of the already known pixels in the two patches (Fig. 3b).

**Step 5:** Copy the value of each pixel to be filled in,  $p' | p' \in \psi_{\hat{p}} \cap \Omega$ , using its corresponding position inside  $\psi_{\hat{q}}$  (Fig. 3c).

**Step 6:** Update the confidence term  $C(p)$  in the area encircled by  $\psi_{\hat{p}}$  as follows:

$$C(q) = C(\hat{p}), \quad \forall q \in \psi_{\hat{p}} \cap \Omega \quad (4)$$

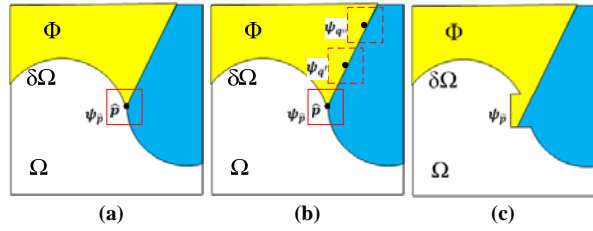


Fig. 3. *Algorithm 1*: (a) The patch  $\psi_{\hat{p}}$  with the highest priority is found to be filled in, (b) The most similarity candidate patches with  $\psi_{\hat{p}}$  are determined, e.g.  $\psi_q$  and  $\psi_{q'}$ , (c) The best matching patch in the candidate set copied into the position occupied by  $\psi_{\hat{p}}$ , and thus partial filling of  $\Omega$  is achieved

Before discussing our improvement of the patch sparsity inpainting method proposed in [19], we first explain the general steps of the algorithm.

The method investigates the sparsity of image patches and measures the confidence of the patch located at the structure region by the sparseness of its nonzero similarities to the neighbouring patches. The patch with larger structure sparsity is assigned a higher priority for further inpainting. The algorithm can be presented in six steps. Steps 1, 3, 5, and 6 are quite similar to the conventional exemplar-based method (*Algorithm 1*). The priority term in Step 2 is computed in a different way. A sparse linear combination of weighted candidate patches is also used to fill in the missing patch instead of using a single best match in Step 4. Therefore, the steps 2 and 4 of *Algorithm 1* are modified as follows:

**Step 2:** A new definition for patch priority, namely *structure sparsity*, is proposed. For any selected patch, a collection of neighbouring patches with the most similarities is also distributed in the same structure or texture. Therefore, the confidence of structure for a patch is measured by the sparseness of its nonzero similarities to the neighbouring patches. The patch with more sparsely distributed nonzero similarities is laid on the fill-front due to the high structure sparseness.

For the patch  $\psi_p$ , located at the fill-front  $\delta\Omega$ , a neighbourhood window  $N(p)$ , with the center  $p$ , is set (refer to Fig. 4a). The sparseness of similarities for the patch is measured by

$$\rho(p) = \sqrt{\left[ \sum_{p_j \in N_s(p)} w_{p,p_j}^2 \right] \frac{|N_s(p)|}{|N(p)|}} \quad (5)$$

where the patch  $\psi_{p_j}$  is located in the known region centered at  $p_j$ , and  $w_{p,p_j}$  refers to the similarity between  $\psi_p$  and  $\psi_{p_j}$  (Fig. 4b), as defined by:

$$w_{p,p_j} = \frac{1}{Z(p)} \exp\left(-\frac{d(\psi_p, \psi_{p_j})}{\sigma^2}\right) \quad (6)$$

with  $d$  measuring the mean squared distance of the already known pixels in the two patches,  $Z(p)$  being a normalization constant so that  $\sum_{p_j \in N_s(p)} w_{p,p_j} = \mathbf{1}$ , and  $\sigma$  being set to 5. Finally,  $N_s(p)$  is defined to be

$$N_s(p) = \{p_j: p_j \in N(p) \text{ and } \psi_{p_j} \subset \bar{\Omega}\} \quad (7)$$

The patch priority (or structure sparsity) term is defined as the product of the transformed structure sparsity term and the patch confidence term that is,

$$P(p) = T_{[\zeta, 1]}(\rho(p)) \cdot C(p) \quad (8)$$

where  $T_{[\zeta, 1]}$  is a linear transformation taking  $\rho(p)$  into the interval  $[\zeta, 1]$ . This transformation scales the structure sparsity variations to be comparable with  $C(p)$ .

**Step 4:** Contrary to the traditional exemplar-based method [19], in which the fill-front patch  $\psi_{\hat{p}}$  is filled by the best match  $\psi_{\hat{q}}$ , in the patch sparsity inpainting method,  $\psi_{\hat{p}}$  is inpainted by the sparse combinations of multiple exemplars in the framework of sparse representation. Indeed, from the source region, the top  $N$  most similar patches are selected as the set of candidates

$\{\psi_q\}_{q=1}^N$ . Therefore, the unknown pixels in patch  $\psi_{\hat{p}}$  is approximated by linear combinations of the  $\{\psi_q\}_{q=1}^N$ , that is,

$$\psi_{\hat{p}} = \sum_{q=1}^N \alpha_q \psi_q \quad (9)$$

where the coefficient vector  $\vec{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$  is obtained by solving a constrained optimization problem in the framework of a sparse representation (Fig. 4c).

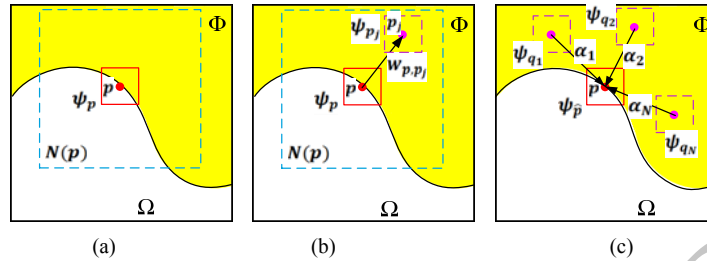


Fig. 4. The patch sparsity inpainting method proposed in [19]: (a) The neighbourhood window  $N(p)$ , with the center  $p$ , is set for the patch  $\psi_p$  located at the fill-front  $\delta\Omega$ , (b) To measure the sparseness of similarities for  $\psi_p$  the similarity between  $\psi_p$  and  $\psi_{p_j}$ ,  $w_{p,p_j}$ , is computed, (c) The unknown pixels in patch  $\psi_{\hat{p}}$ , which has the highest priority, are filled by linear combinations of the top  $N$  most similar patches  $\{\psi_q\}_{q=1}^N$  weighted by coefficients  $\{\alpha_q\}_{q=1}^N$  obtained by solving an optimization problem

This optimization problem minimizes the  $\ell_0$  norm of  $\vec{\alpha}$ , i.e., the number of nonzero elements in the vector  $\vec{\alpha}$ , with the linearity assumption of the combination.

### 3. THE PROPOSED METHOD

Our proposed algorithm is given next. We modify steps 2, 4 and 6 to attain better results.

*Algorithm 2: A modified patch propagation-based image inpainting using patch sparsity.*

**Step 1:** For each point  $p$  on the boundary  $\delta\Omega$ , a patch  $\psi_p$  is constructed with  $p$  as the center of the patch (as in [12] and [19]).

**Step 2:** To compute the patch's priority  $P(p)$ , a stable definition is used as follows:

$$P(p) = \alpha T_{[\zeta, 1]}(\rho(p)) + \beta T'_{[\gamma, 1]}(C(p)) \quad (10)$$

where the terms  $T_{[\zeta, 1]}(\rho(p))$  and  $C(p)$  are the same as the ones defined in [19], and  $\alpha$  and  $\beta$  are the component weights with  $0 \leq \alpha, \beta \leq 1$  and  $\alpha + \beta = 1$ . As illustrated in [20], the confidence value rapidly drops to zero as the filling process goes on. When the dropping effect occurs, error continually propagates to the central part of the reconstructed image, causing noticeable visual artifacts. Therefore, a regularizing transformation is used to control the decreasing rate of the confidence term. We propose using a linear transformation  $T'$  to take  $C(p)$  into the interval  $[\gamma, 1]$ . Also, the priority term is changed to an additive form instead of a multiplicative form (because the numerical multiplication is effectively sensitive to extreme values, while the additive form has been shown to be more robust with respect to its input and hence more stable).

**Step 3:** Find the patch  $\psi_{\hat{p}}$  with the highest priority to be filled in with the information extracted from the source region  $\Phi$  (as in [12] and [19]).

**Step 4:** Perform a global search on the whole image to find the  $N$  most similar patches. Formally,

$$\psi_{\hat{q}} = \arg \min_{\psi_q \in \Phi} \{d(\psi_{\hat{p}}, \psi_q) + d_{grad}(\psi_{\hat{p}}, \psi_q) + d_{div}(\psi_{\hat{p}}, \psi_q)\} \quad (11)$$

where the distance  $\mathbf{d}$  between two generic patches is simply defined as the sum of squared differences (SSDs) of the already known pixels in the two patches.

As a patch may contain texture details, we also include the gradient and divergence differences ( $\mathbf{d}_{grad}$  and  $\mathbf{d}_{div}$ , respectively) between the already known pixels in the two patches to attain more accuracy. Hence, these added terms cause the preservation of texture properties.

**Step 5:** Once the  $N$  most similar patches are found, we use a linear combination of patches  $\{\psi_q\}_{q=1}^N$  to finally obtain the prediction of the missing pixels in  $\psi_{\hat{p}}$  (as in [19]).

**Step 6:** Let  $\psi_{\hat{q}} = \sum_{q=1}^N \alpha_q \psi_q$  (12)

Denote the unknown pixels in the patch  $\psi_{\hat{p}}$ , by a matrix  $\mathbf{P}$ , filled by the corresponding pixels in  $\psi_{\hat{q}}$ :

$$\mathbf{P}\psi_{\hat{p}} = \mathbf{P}\psi_{\hat{q}}. \quad (13)$$

As illustrated in [19], the coefficients  $\vec{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$  are obtained by solving the following constrained optimization problem:

$$\begin{aligned} & \min \|\vec{\alpha}\|_0 \\ & \text{s. t. } \|\bar{\mathbf{P}}\psi_{\hat{q}} - \bar{\mathbf{P}}\psi_{\hat{p}}\|^2 < \varepsilon, \\ & \beta \left\| \mathbf{P}\psi_{\hat{q}} - \mathbf{P} \sum_{p_j \in N_s(p)} \mathbf{w}_{p,p_j} \psi_{p_j} \right\|^2 < \varepsilon, \\ & \sum_i^N \alpha_i = 1. \end{aligned} \quad (14)$$

The first and the second constraints concern the local patch consistency. The first constraint constrains the estimated patch  $\psi_{\hat{q}}$  approximated by the target patch  $\psi_{\hat{p}}$  over the already known pixels, and the second one forms the consistency between the newly filled pixels and the neighboring patches in appearance. It measures the similarity between the estimated patch and the weighted mean of the neighboring patches over the missing pixels. The last constraint imposes a normalization summation on the coefficients vector  $\vec{\alpha}$ . This constraint is used to achieve invariancy while reconstructing the target patch from its neighboring candidate patches. The parameter  $\varepsilon$  is to control the error tolerance,  $\beta$  balances the strength of the two first constraints, which is set to 0.25 in our implementation (as in [19]), and  $\mathbf{w}_{p,p_j}$  is defined to be as given in (6).

The local patch consistency constraint can be rewritten in a compact form:

$$\|\mathbf{D}\psi_{\hat{q}} - \psi_T\|^2 < \varepsilon, \quad (15)$$

where,

$$\mathbf{D} = \begin{bmatrix} \bar{\mathbf{P}} \\ \sqrt{\beta}\mathbf{P} \end{bmatrix}, \quad \psi_T = \begin{bmatrix} \bar{\mathbf{P}}\psi_{\hat{p}} \\ \sqrt{\beta}\mathbf{P} \sum_{p_j \in N_s(p)} \mathbf{w}_{p,p_j} \psi_{p_j} \end{bmatrix} \quad (16)$$

So, the optimization problem can be formulated as

$$\begin{aligned} & \min \|\vec{\alpha}\|_0 \\ & \text{s. t. } \|\mathbf{D}\psi_{\hat{q}} - \psi_T\|^2 < \varepsilon \\ & \sum_i^N \alpha_i = 1 \end{aligned} \quad (17)$$

which can be solved in the same way as in Locally Linear Embedding (LLE) for data reduction [21]. In our case, the Gram matrix is  $\mathbf{G} = (\boldsymbol{\psi}_T \mathbf{1}^T - \mathbf{X})^T (\boldsymbol{\psi}_T \mathbf{1}^T - \mathbf{X})$ , where  $\mathbf{X}$  is a matrix with columns  $\{\mathbf{D}\boldsymbol{\psi}_{q_1}, \dots, \mathbf{D}\boldsymbol{\psi}_{q_N}\}$ , and  $\mathbf{1}$  is a column vector of ones. Then, we have a closed form solution as

$$\vec{\alpha} = \frac{\mathbf{G}^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{G}^{-1} \mathbf{1}} \quad (18)$$

**Step 7:** Update the confidence term  $\mathbf{C}(\mathbf{p})$  in the area encircled by  $\boldsymbol{\psi}_{\hat{\mathbf{p}}}$  as follows:

$$C(q) = \frac{|\{q|q \in \boldsymbol{\psi}_{\hat{\mathbf{p}}} \cap \bar{\Omega}\}|}{|\boldsymbol{\psi}_{\hat{\mathbf{p}}}|} \cdot C(\hat{\mathbf{p}}), \quad \forall q \in \boldsymbol{\psi}_{\hat{\mathbf{p}}} \cap \Omega \quad (19)$$

where the numerator in the first term is considered to be the number of known pixels in the patch  $\boldsymbol{\psi}_{\hat{\mathbf{p}}}$ .

By this definition, we differentiate between the centered pixel,  $\hat{\mathbf{p}}$ , and the pixels recently filled in,  $\mathbf{p}' | \mathbf{p}' \in \boldsymbol{\psi}_{\hat{\mathbf{p}}} \cap \Omega$ ; the new filled in pixels do not have information as reliable as the centered pixel. Hence, we can reduce the rate of error diffusion as the filling proceeds.

Next, we present our experimental results.

#### 4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

We have implemented our proposed method and compared the results with those obtained by a PDE-based technique [4], conventional approach [12], and the patch sparsity method [19] on several different images.

To set up the same computing environment, we have implemented *Algorithm 2*, the conventional approach in [12], and the patch sparsity method in [19] using Matlab 7.10.0 instructions and executed the programs on a laptop with CPU specifications: Intel® Core™ i5, 2.4 GHz, and 4GB of RAM. The results for the method in [4] were obtained using the code provided by the first author of [4].

As proposed in [12], we applied the conventional method with the patch size (i.e., size of  $\boldsymbol{\psi}$ ) being equal to  $9 \times 9$  pixels. The parameters used in the patch sparsity approach are the same as those given in [19].

In our proposed algorithm, the size of the patch window  $\boldsymbol{\psi}$  was set to  $7 \times 7$  pixels. For most cases in Fig. 1, the size of the neighbourhood window,  $\mathbf{N}(\mathbf{p})$ , was set to  $11 \times 11$  and the value of  $\alpha$  and  $\beta$  were set to 0.5 and 0.5, respectively. Also, the number of candidate patches,  $\mathbf{N}$ , was fixed to be 21.

Figure 5 shows the constancy definition for the patch priority  $\mathbf{P}(\mathbf{p})$ . As the numerical multiplication is effectively sensitive to extreme values, the additive form is more robust and more stable than the multiplicative form. So, the obtained result for this new definition is better than the result obtained by the conventional one.

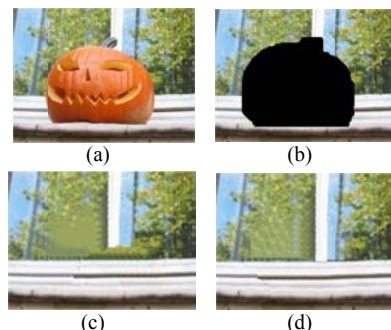


Fig. 5. The effect of new definition for the patch priority: (a) Original image, (b) Degraded image, (c) Inpainted image with the multiplicative form of  $\mathbf{P}(\mathbf{p})$ , and (d) Inpainted image with the additive form of  $\mathbf{P}(\mathbf{p})$

Figure 6 shows some results obtained by our method in comparison with the proposed methods in [12] and [19] for the application of object removal.

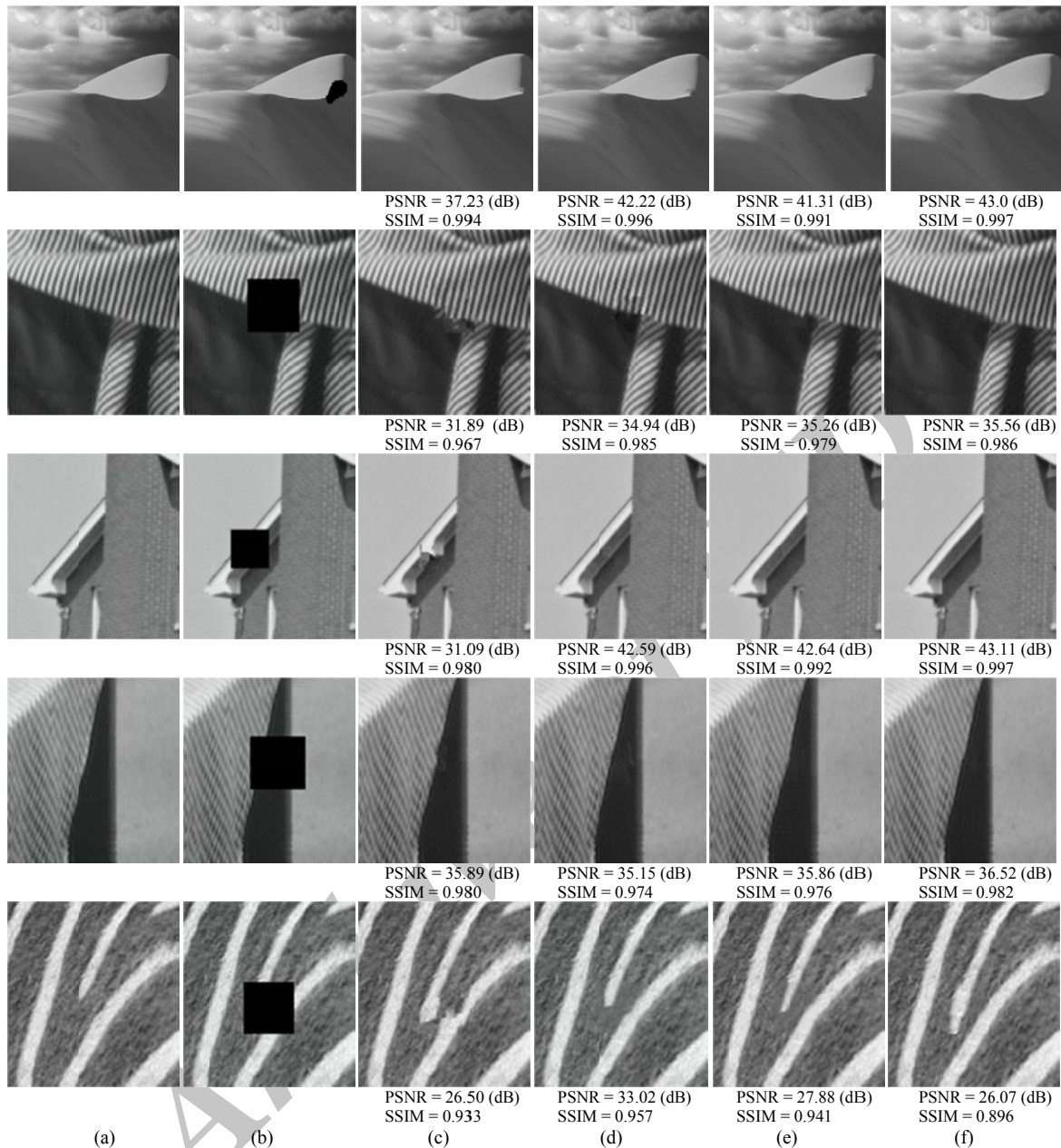


Fig. 6. Results obtained by *Algorithm 2* in comparison with these obtained by the methods in [12] and [19]: (a) Original image, (b) Degraded image, (c) Inpainted image from [12], (d) Inpainted image from [19], (e) Inpainted image by *Algorithm 2* using (3) as similarity metric, and (f) Inpainted image by *Algorithm 2* using the proposed similarity metric, (11)

The first and second columns are the original and degraded images, respectively, the third column illustrates the results obtained from the conventional algorithm, the fourth column shows the results of the patch sparsity method, and the last two columns demonstrate the results obtained by our proposed algorithm using the conventional and the newly proposed similarity metric, respectively.

As observed in Fig. 6, the results of our proposed algorithm are superior over the two other ones, even using the conventional similarity metric. Visually it is found that the presented results in column (e) are more acceptable than those illustrated in columns (c) and (d). Also, robustness of the newly proposed similarity metric is affirmed by comparing the fifth and sixth columns. We notice that patches chosen with just the conventional similarity metric, i.e., (3), contain artifacts that are not well suited for the



neighborhood, whereas our metric, i.e., (11), considering more properties between the patches, selects more visually pleasant patches and produces better results.

The modification (19) for updating the confidence term after filling in the unknown pixels reduces the error propagation, since the rate of reliable information is controlled by the difference of the centered pixel and the newly filled ones. The effect of this new definition is shown in Fig. 7. The first and second rows show original and degraded images, and the third and fourth rows show the images obtained by our proposed method using the conventional definition (4) and the newly proposed one (19) for the confidence term, respectively.

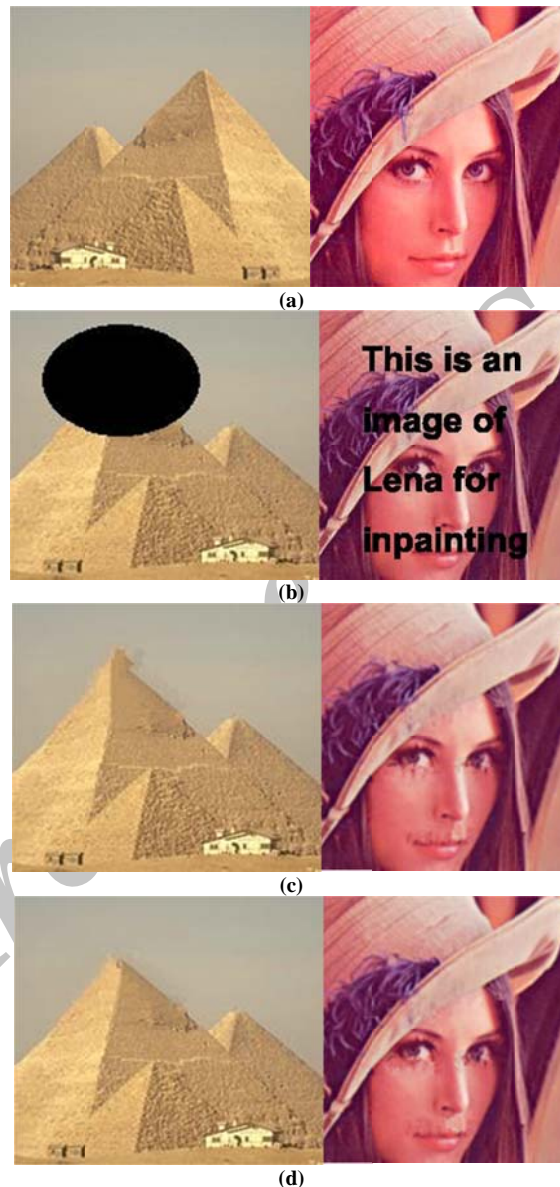


Fig. 7. The effect of the new definition for confidence term,  $C(p)$ : (a) Original images, (b) Degraded images, (c) Inpainted images with the conventional definition for  $C(p)$ , and (d) Inpainted images with the new definition for  $C(p)$

As observed in Fig. 7, the result obtained by the new confidence term is more plausible.

Figure 8 presents more examples for the application of text removal, scratch restoration and object removal. The first and second rows show the original and degraded images, and the third to the sixth rows respectively show the images obtained by a PDE-based technique [4], the conventional algorithm [12], the patch sparsity method [19] and our proposed algorithm.

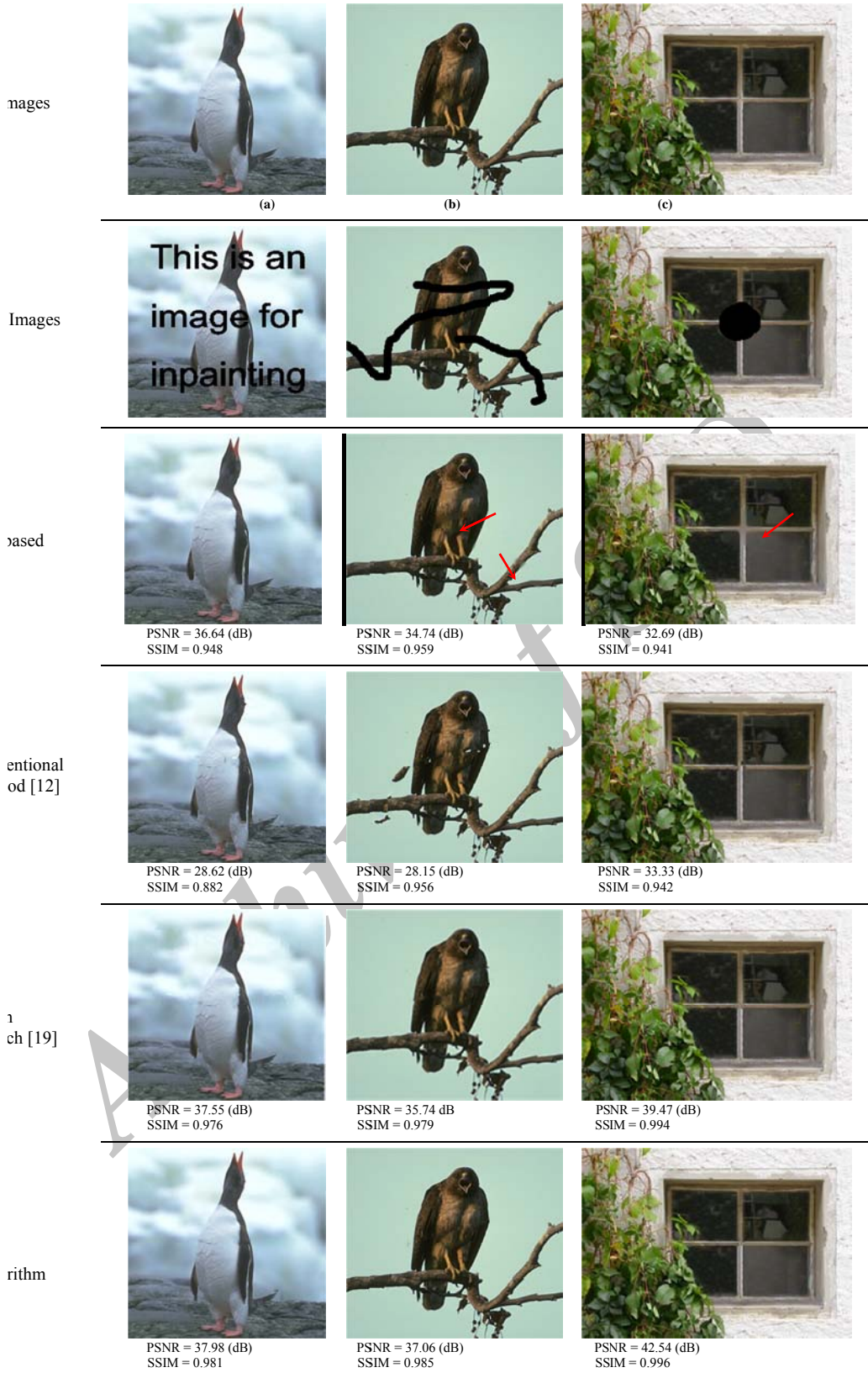


Fig. 8. Results obtained by *Algorithm 2* in comparison with the ones obtained by the methods in [4], [12] and [19]: First and second rows are the original and degraded images, respectively. The third row shows the results of PDE-based technique [4], the results of conventional inpainting method [12] are presented in the fourth row, the results obtained by [19] are illustrated in the fifth row, and the last row demonstrates the results of our proposed algorithm

Roth and Black [4] used the prior models to restore the image; the models have been shown to be useful when images have noise or uncertainty. They proposed a framework for learning image priors. The prior model used in their work was Markov random field (MRF), which assumes an image to be the result of a random process, described by an MRF.

For images in Fig. 8, the size of the patch window  $\psi$  was set to  $7 \times 7$  pixels and the value of  $\alpha$  and  $\beta$  were set to 0.5 and 0.5, respectively. The size of the neighbourhood window,  $N(p)$ , was set to  $11 \times 11$ ,  $19 \times 19$ , and  $51 \times 51$  pixels for images (a), (b) and (c), respectively. Also, the number of candidate patches,  $N$ , was fixed to be 11, 3 and 25, correspondingly.

We have used the parameters of the approach in [4] unchanged for all the examples considered here. This scheme converted color image to YCbCr color model, and the algorithm was independently applied to all 3 channels. We used a FoE prior with 8 filters of  $3 \times 3$  pixels. The algorithm inpainted the missing areas by iteratively propagating information. We set the number of iterations to 200.

As seen in Fig. 8, the PDE-based method [4] performs well on piecewise smooth image, Fig. 8 (a), but fails to reconstruct areas containing texture with fine details and tends to blur the inpainted image, Fig. 8 (b) and (c). The conventional algorithm [12] cannot preserve the edge continuity and the texture consistency, and thus produce unpleasant artifacts. The patch sparsity algorithm [19] produces more pleasant results; however, it fails to reconstruct some edges properly.

In contrast, the results obtained by our algorithm appear to be closer to the original image than the ones obtained by the other three methods. Because of the improvement in the priority term, high importance was given to the structures in a more robust way. Also, the proposed similarity metric encompasses more properties of the patches and selects the most visually pleasant ones. Furthermore, the modification in the confidence term after filling in pixels helps reduce the error propagation.

For a quantitative comparison, we computed the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) values between the original and inpainted images, also observing overall better obtained PSNR and SSIM values by *Algorithm 2*.

However, similar to [19], our algorithm cannot properly recover large missing areas consisting of structures while the known region doesn't contain any structure cues. This is a limitation of our algorithm, which needs to be investigated in a future work.

## 5. CONCLUSION

We presented a modified patch sparsity scheme for inpainting degraded images. Addressing the patch sparsity approach as a robust inpainting method, the suggested modifications lead to an improvement in producing better results. We applied the proposed algorithm to several images and compared the obtained result with those obtained by three other methods. The high visual quality of the results obtained by our approach affirmed the effectiveness of the proposed algorithm.

In future, we will also investigate the effects of varying patch sizes and try to select an optimal size. Moreover, the number of sufficient candidate patches would be explored since it has a direct impact on the computational time of inpainting.

**Acknowledgment:** The authors sincerely thank Stefan Roth for distributing the inpainting code and also Jian Sun for providing the original and degraded images as well as the results obtained by the patch sparsity method. The second author also thanks Sharif University of Technology for its support.

## REFERENCES

1. Bertalmio, M., Sapiro, G., Caselles, V. & Ballester, C. (2000). *Image inpainting, computer graphic (SIGGRAPH)*, pp. 417–424.
2. Chan, T. F. & Shen, J. (2002). Mathematical models for local non-texture inpainting. *SIAM Journal of Applied Mathematics*, Vol. 62, No. 3, pp.1019–1043.
3. Chan, T. F. & Shen, J. (2001). Non-texture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, Vol. 12, No. 4, pp. 436-449.
4. Roth, S. & Black, M. J. (2009). *Fields of experts. International Journal of Computer Vision Computer*, Vol. 82, No. 2, pp. 205-229.
5. Ashikhmin, M. (2001). Synthesizing natural textures. *ACM Symposium on Interactive 3D Graphics*, pp. 217–226.
6. Efros A. A. & Leung, T. K. (1999). Texture synthesis by non-parametric sampling. *IEEE International Conference on Computer Vision*, Vol. 2, pp. 1033–1038.
7. Wei, L. Y. & Levoy, M. (2000). Fast texture synthesis using tree-structured vector quantization. *Proc. ACM Conf. Comp. Graphics*, pp. 479–488.
8. Efros, A. A. & Freeman, W.T. (2001). Image quilting for texture synthesis and transfer. *Proc. ACM Conf. Comp. Graphics (SIGGRAPH)*, Eugene Fiume, pp. 341–346.
9. Freeman, W. T., T. R. Jones and E.C. Pasztor, “Example-based super-resolution”, *IEEE Computer Graphics and Applications*, vol. 22, issue 2, pp. 56–65, 2002.
10. Liang, L., Liu, C., Xu, Y., Guo, B. & Shum, H. Y. (2001). Real-time texture synthesis using patch-based sampling. *ACM Trans. on Graphics*, Vol. 20, No. 3, pp. 127–150.
11. Bertalmio, M., Vese, L., Sapiro, G. & Osher, S. (2003). Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.*, Vol. 12, pp. 882–889.
12. Criminisi, A., Perez, P. & Toyama, K. (2003). Object removal by exemplar based inpainting. *IEEE Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pp. 721–728.
13. Drori, I., Cohen-Or, D. & Yeshurun, H. (2005). Fragment-based image completion. *ACM Trans. Graph.*, Vol. 22, No., pp. 303–312.
14. Hung, J. C., Hwang, C. H., Liao, Y. C., Tang, N. C. & Chen, T. J. (2008). Exemplar-based image inpainting based on structure construction. *Journal of Software*, Vol. 3, No. 8, pp. 57-64.
15. Wong, A. & Orchard, J. (2008). A nonlocal-means approach to exemplar-based inpainting. *Proceeding of IEEE International Conference on Image Processing*, pp. 2600-2603.
16. Elad, M., Starck, J. L., Querre, P. & Donoho, D. L. (2005). Simultaneous cartoon and texture image inpainting using morphological component analysis. *Appl. Comput. Harmon. Anal.*, Vol. 19, pp. 340–358.
17. Fadili, M. J., Starck, J. L. & Murtagh, F. (2009). Inpainting and zooming using sparse representations. *The Comput. J.*, Vol. 52, No. 1, pp. 64–79.
18. Wohlberg, B. (2011). Inpainting by joint optimization of linear combinations of exemplars. *IEEE Signal Processing Letters*, Vol. 18, No. 1, pp. 75–78.
19. Xu, Z. & Sun, J. (2010). Image inpainting by patch propagation using patch sparsity. *IEEE Transactions on Image Processing*, Vol. 19, No. 5, pp. 1153–1165.
20. Cheng, W. H., Hsieh, C. W., Lin, S.K., Wang, C.W. & Wu, J. L. (2005). Robust algorithm for exemplar-based image inpainting. *CGIV*.
21. Roweis, S.T. & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Scienc*, Vol. 290, pp. 2323–2326.
22. Noori, H., Saryazdi, S. & Nezamabadi-pour, H. (2011). A bilateral image inpainting. *Iranian Journal of Science & Technology, Transactions of Electrical Engineering*, Vol. 35, No. E2, pp. 95–108.