IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 9, 2018, accepted October 8, 2018, date of publication October 15, 2018, date of current version November 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2876161

Screen Window Propagating for Image Inpainting

HUAMING LIU[1,2], **XUEHUI BI**[2], **GUANMING LU**[1], **WEILAN WANG**[3], **JINGJIE YAN**[1], **AND ZHENGYAN ZHANG**[4]

[1]College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[2]School of Computer and Information Engineering, Fuyang Normal University, Fuyang 236037, China
[3]Key Laboratory of China's Ethnic Languages and Information Technology, Ministry of Education, Northwest Minzu University, Lanzhou 730000, China
[4]School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China

Corresponding author: Xuehui Bi (bixuehui888@163.com)

This work was supported in part by the key projects of natural science research in Anhui colleges and universities under Grant KJ2018A0345, in part by the key fund projects of Young Talents in Fuyang Normal University under Grant rcxm201706, in part by the National Natural Science Foundation of China under Grant 61162021 and Grant 61772430, in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX18_0901, in part by the Natural Fund Project in Fuyang Normal University under Grant 2017FSKJ17, in part by the National Natural Science Foundation of China under Grant 61501249, in part by the Key Research and Development Program of Jiangsu Province under Grant BE2016775, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20150855, and in part by the Project funded by China Postdoctoral Science Foundation under Grant 2018M632348.

ABSTRACT Image repair often has errors in filling. An exemplar-based technique has the advantage of filling texture and structure simultaneously, but there exist some problems with this, such as not considering the overall situation of inpainting, the unreasonable calculation of priority, and random selection when there are multiple candidate patches. In view of these situations, this paper proposes a method of screen window propagating where the upper layer guides the lower layer inpainting from the top layer by means of multiple resolution decomposition. Screen window propagating aims to constrain the inpainting of the lower layer and maintain the overall profile. Improving the priority calculation using the upper repair result can lead to getting a better repair order. When there is more than one candidate patch, the structural similarity index measure (SSIM) is used to obtain the best candidate patch. We consider four cases of rotation and inversion when looking for the best exemplar. The experiments show that our method can obtain more satisfactory results than other methods based on subjective and objective indicators, such as peak signal-to-noise ratio (PSNR), SSIM, and feature similarity index measurement. Screen window propagating for image inpainting aims toward image restoration. Whole contours are preserved and the lower layer can utilize more upper layer information. Our method considers information from the whole and its parts.

INDEX TERMS Exemplar-based technique, pyramid decomposition, guidance strategy of screen window propagating, exemplar matching.

I. INTRODUCTION

In image processing, we often need to remove text, broken regions, objects, and other elements in the image; this problem can be solved using image inpainting. The researched achievements of inpainting have been applied in image editing software. For example PatchMatch [1]has been applied in Photoshop. Inpainting, which imitates painter repair, was firstly proposed by Bertalmo [2]. To date, more and more image restoration methods are being proposed.

The key to image restoration is to consider global and local consistency, especially with regards to geometric structure and texture features. We first observe for global consistency, such as in the structure outline, and then we examine the local detail, such as texture filling. In addition, post-processing is added to improve the quality of repair. An example is Possion blending [3], [4] to improve the color consistency between the target areas and their surrounding areas because we are sensitive to drastic changes in color in the local regions.

Chan proposed nontextured inpainting [5] for geometric structure inpainting; it mainly realizes the connectivity which does not consider in the total variation (TV) [6], the filling of textures and global structures were not considered. Texture and geometric structural components were divided respectively in [7], as proposed by Bertalmo *et al.* Texture layer is inpainted by texture synthesis technique [8] and structure layer is infilled using diffusion-based method [2], then two components were repaired and then combined. It overcomes the smoothing effect of diffusion repair, but it is difficult to repair larger missing structures. In [2], [5], and [7], the image structures were propagated from outside the target regions.

This method is effective for some structures and repairs a small area. Texture and structure are two significant factors in reconstructing an image. Criminisi *et al.* [9] put forward an exemplar-based technique which could repair texture and structure simultaneously. It is also very effective when repairing large areas. The key factors of this method are calculating priority and the search for a matching patch. However, the value of priority is influenced by high-frequency components, such as noise and complicated textures, because they obtain higher priority than the structural components. In addition, large missing regions can be inpainted well when we adopt this approach. Inspired by this, Barnes *et al.* [1] proposed the PatchMatch algorithm to quickly find matching patches through the contiguous propagation of similar patches.

To improve the quality of repairs, the target regions are repaired layer-by-layer, starting from the top layer, using pyramid decomposition. Other researchers have proposed improved methods for calculating priority or for searching for the best matching patches. For example, such methods for improving calculating priority was proposed in [10]–[14]. Barnes *et al.* [15] improved the search strategy by adding conditions of rotation and scale. All these methods [1], [12], [15] use super-resolution analysis. It is commonly believed that inpainting low-resolution images is easier than inpainting high-resolution ones. An approach outlined in [16] combines an exemplar-based technique with super-resolution. First, a coarse image is inpainted. Second, an enhanced resolution picture is inferred from the coarse inpainted image. This method takes advantage of other algorithms such as exemplar-based techniques [9], single image super resolution techniques [17] and a K-NN (K Nearest Neighbors) techniques [1].

Here are some other methods [18]–[20] similar to [16]. A patch-based method and hierarchical Wavelets decomposition are combined in [18], it makes use of the advantages that wavelets handle structure information and texture information separately. The patch-based method and hierarchical pyramid decomposition are combined in [19], which uses multiscale scheme inpaiting with the help of K-NN searching and energy minimum constraints, however, structural connectivity is not yet complete. Variational framework and exemplar-based techniques are combined in [20], which employ variational framework [21], this method obtains the optimal inpainting by taking the minimum energy on multiple layers, and a K-NN searching for similar patches.

An expectation maximization (EM) algorithm is adopted in [1], [19], [20], and [22] for inpainting, EM can search for similar sample patches to improve the accuracy of filling. Low-rank was utilized in an exemplar-based method for reducing mismatching in [23], which modifies the calculation of priority and synthesize the most similar patch using low-rank optimization, the calculation of structure terms uses the edge map technique rather than isophotes directions. Jurio *et al.* [26] proposed two-step algorithm for inpainting, however, the number of clusters cannot be determined automatically, which can easily lead to failure of inpainting.

Although there has been much progress in the past years on hierarchical inpainting and exemplar-based technique, there are still some problems, such as determining the number of pyramid decompositions, the size of patches, the defining of priority, the searching strategy, enhancing super-resolution image from a coarse image, and etc.

The approach in [24] and [25] was first proposed in a more rigorous minimization framework. They used a multi-scale method to combine better inpainting solutions. The approach in [19] also uses multi-scale strategy, PatchMatch searching [1], and modifying the patch distance. In this paper, we take a similar approach. We use a patch-based technique [9] and pyramid decomposition inpainting from coarse pictures. Although the algorithm is similar to method [19], there are some differences here. Firstly, we improved the computation of data term which was different from [9]. Secondly, we provided a screen window propagating guidance strategy from the coarse pictures to the final super-resolution in hierarchical inpainting. Thirdly, we provided an approach to setting the numbers of levels automatically in hierarchical inpainting. This approach is different from approach [16] by Newson *et al.* Finally, we cut down the candidate patches to improve filling quality.

Algorithm 1 Screen Window Propagating Inpainting Algorithm

```
Input: target image, mask of the target image
Output: inpainted image
Compute the number ($N$) of level with respected to
  target image; /* Using (6) */
/* Down-sampling operation */
Get $\{L^1, L^2, \cdots, L^N\}$, $\{M^1, M^2, \cdots, M^N\}$;
for ($i = N; i \leq 1; i - -$) do
    if ($i == N$) then
        Generate $\hat{L}^i$ image using exemplar-based
          technique by inputing $L^i$ and $M^i$ ;
    else
        Generate $\hat{L}^i$ image using screen window
          propagating technique $L_c^i$ and $M_c^i$;
    end
    Get $\hat{L}_{up}^i$ using up-sampling of zero filling;
    Get $L_c^{i-1}$ and $M_c^{i-1}$ by combining $\hat{L}_{up}^i$ and $L^{i-1}$;
    /* using (5)*/
end
return $\hat{L}^1$;
```

A pseudo-code for our algorithm has been listed in Algorithm 1. In summary, our work addresses an image inpainting algorithm which uses multi-scale inpainting strategy, exemplar-based technique, improved priority calculation, screen window guidance from the top layer to the bottom layer, and automatic computing of the number of levels.

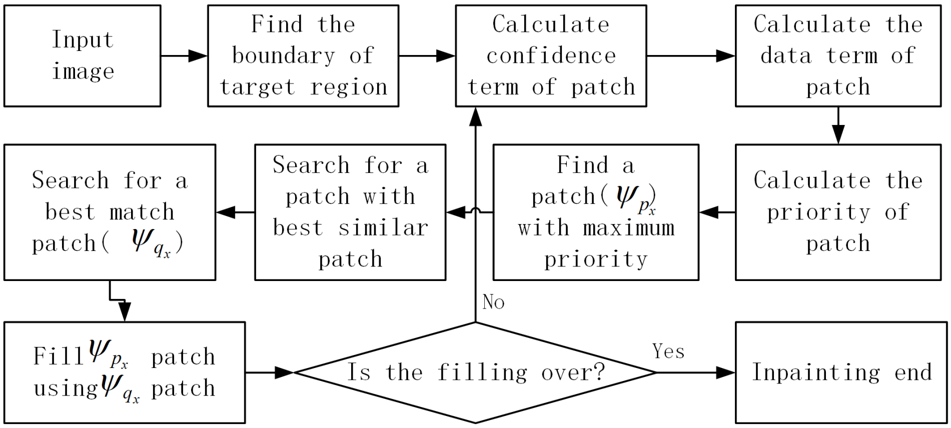

Input image
Find the boundary of target region
Calculate confidence term of patch
Calculate the data term of patch
Calculate the priority of patch
Find a patch(ψ_{p_x}) with maximum priority
Search for a patch with best similar patch
Search for a best match patch(ψ_{q_x})
Fill ψ_{p_x} patch using ψ_{q_x} patch
Is the filling over?
No
Yes
Inpainting end

FIGURE 1. The process of exemplar-based inpainting.

II. NOTATIONS AND ALGORITHM OVERVIEW

A. NOTATIONS

We list the notations to be used in this paper. I is a color image

$$I = \left| \begin{array}{l} \Omega \subset \mathcal{R}^n \to \mathcal{R}^m \\ \quad x \to I(x) \end{array} \right. \tag{1}$$

where x represents a vector which labels the spatial coordinates of a pixel p_x and I is a two-dimensional (2-D) ($n = 2$) color image which contains three components ($m = 3$) defined in the (RGB) color space.

- $I^c : \Omega \to \mathcal{R}$ represents the cth image color channel of I.
- The definition domain Ω includes two parts: $\Omega = S \cup T$. S is the known part of I (source region) and T is the unknown part of I (target region).
- $M \subset \Omega$ is a binary mask corresponding to the target regions with a value of 1 wherever a pixel was dropped; 0 denotes a known pixel.
- ψ_{p_x} is the patch centered on the pixel p_x.
- $\psi_{p_x}^{uk}$ presents the unknown pixels of the patch, whereas $\psi_{p_x}^{k}$ denotes its known pixels.
- L^i denotes ith layer pictures. Here L^1 represents the original image.
- M^i denotes a mask with respect to the ith layer pictures.
- $\hat{L}^i$ is the inpainted image of L^i.
- L_{up}^i is the up-sampling image of L^i, whereas L_{dw}^i denotes its down-sampling image. Up-sampling is the operation which fills zero in the even row and even column. The down-sampling operation is that where even rows and even columns are deleted.
- $\hat{L}_{up}^i$ is the up-sampling image of $\hat{L}^i$, whereas $\hat{L}_{up-nn}^i$ denotes the up-sampling result using the nearest neighbor algorithm.
- L_c^i represents the merger of L^i and L_{up}^{i+1}.
- M_c^i denotes a mask with respect to L_c^i pictures.

B. EXEMPLAR-BASED ALGORITHM

An exemplar-based technique was proposed by Criminisi *et al.* [9]. It has an advantage in that it can simultaneously fill the large missing target regions with structure and texture. It first searches for a patch with maximum priority in the outside edge of the target regions. Then it fills the patch, updates the edges of target regions, looks for the next patch with maximum priority in the new edge, and repeats this process until the target regions are filled. The details of the inpainting process referred to in [9] is shown in Fig.1.

Given a patch ψ_{p_x} centered at the point p_x for $p_x \in \delta T$, the priority patch is defined to be

$$P(p_x) = C(p_x)D(p_x) \tag{2}$$

where $C(p_x)$ is confidence term and $D(p_x)$ is the data term. They are defined as follows

$$C(p_x) = \frac{\sum_{q_x \in \psi_{p_x} \cap (I-T)} C(q_x)}{\left|\psi_{p_x}\right|} \tag{3}$$

$$D(p_x) = \frac{\left|\nabla I_{p_x}^{\perp} \cdot n_{p_x}\right|}{\alpha} \tag{4}$$

where $\left|\psi_{p_x}\right|$ is the area of ψ_{p_x}, α is a normalization factor, n_{p_x} is a unit vector orthogonal to the front δT in the point p_x, and $\perp$ represents the orthogonal operator. Here, the initialized value of $C(p_x)$ is set to $C(p_x) = 1$, $\forall p_x \in S$, and $C(p_x) = 0$, $\forall p_x \in T$.

III. PROPOSED ALGORITHM

In the inpainting process of our algorithm, the images to be filled are decomposed multi-scale pictures. Firstly, the top-layer picture (a low resolution or coarse image) is inpainted using a patch-based technique using rotated and reversal patches searching for the best matching patches. Secondly, the inpainted image of the top layer may guide inpainting of next layer picture. Here we use up-sampling of the screen window for constraining the inpainting of the next layer. The calculation of priority combines the current layer picture with the upper layer picture and uses up-sampling to fill the target regions. The process of filling the target regions is carried out from the outside edge of the filling regions.

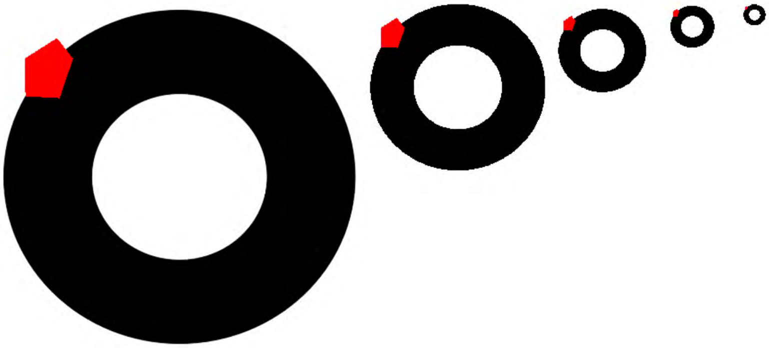


FIGURE 2. The multi-scale image obtained using the down-sample operation.

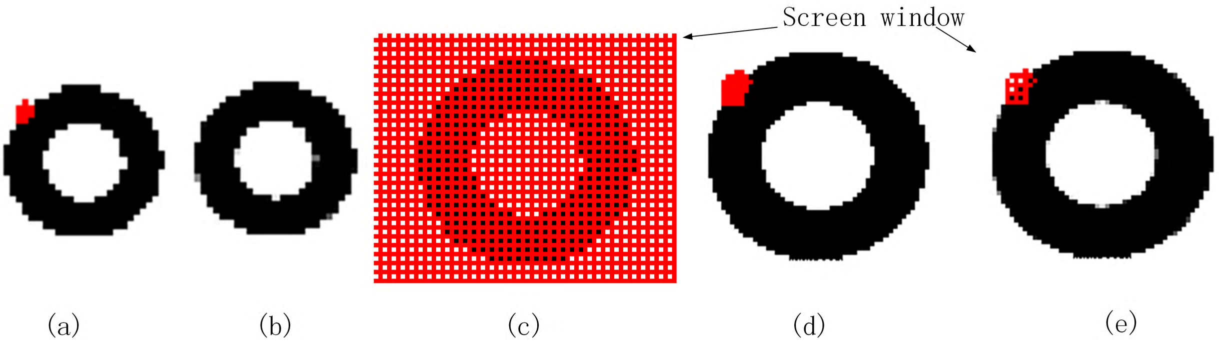


FIGURE 3. The screen window guiding the inpainting of the next layer. (a) L^{i+1} image. (b) $\hat{L}^{i+1}$ image. (c) $\hat{L}_{up}^{i+1}$ image. (d) L^i image. (e) L_c^i image.

A. SCREEN WINDOW PROPAGATING INPAINTING

The image is decomposed into several lower resolution pictures. The decomposition process is different from pyramid decomposing; we use the down-sampling operation. Down-sampling means that the lower layer is obtained by deleting the pixels located in the even row and even column in the upper layer. The process is simple. The top image is the lowest resolution image. Repair starts from the top image. This has two advantages: (1) the area of the target region in a low-resolution picture is smaller than one in a full resolution picture after decomposing multi-scale pictures (many experiments also verified this); and (2) a low-resolution picture contains less noise and includes the main scene structures. In other words, noises could affect the filling order of inpainting. The computing of priority is less influenced by high frequency components. In summary, the start of inpainting is from the top level picture and the result of filling in the top layer picture guides the inpainting of the next lower layer. Fig.2 illustrates the multi-scale images.

Assume that the top level image (L^{i+1}) is repaired ($\hat{L}^{i+1}$). The top level image undergoes up-sampling. The up-sampling operation ($\hat{L}_{up}^{i+1}$ is result of up-sampling) is performed by filling zeros (the zero filling is indicated by red color in Fig.3(c)) in the even rows and even columns. Then, the next layer image L^i and the $\hat{L}_{up}^{i+1}$ image are merged. The result of this combination is represented by L_c^i. The L_c^i is defined as follows

$$L_c^i = \hat{L}_{up}^{i+1} \odot M + L^i \odot (1 - M) \tag{5}$$

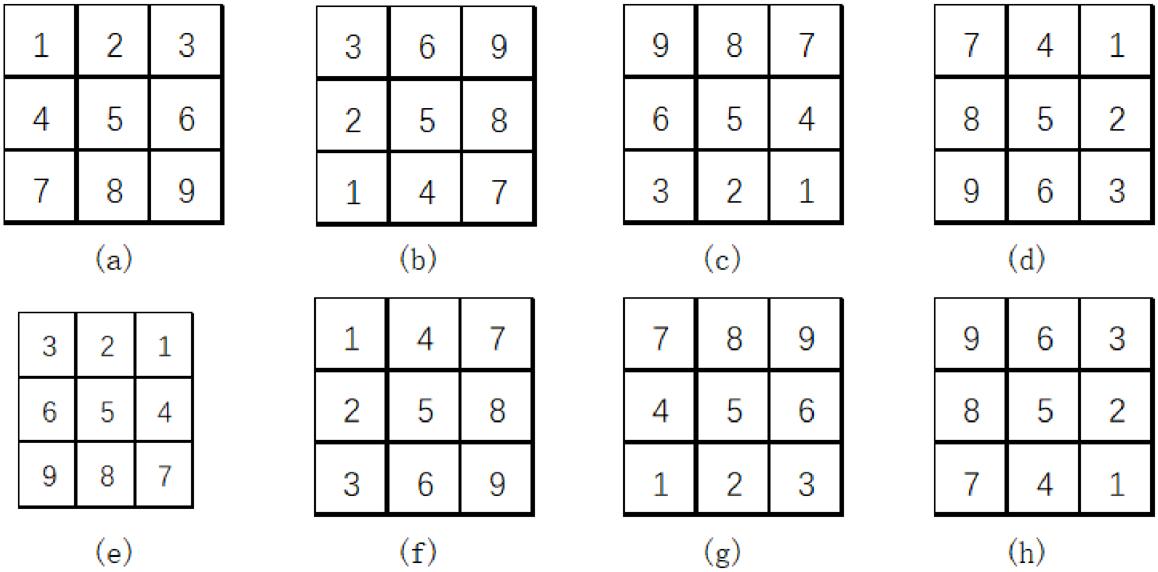
where $\odot$ is the element-wise product operation. The Fig.3(e) shows the combined result of L^i and $\hat{L}_{up}^{i+1}$ image.

The zero filling operation aims to preserve the upper layer information in order to guide lower layer filling. As seen in Fig.3(c), the picture resulting from the filling is similar to the screen window. This is where the screen window propagation repair gets its name. The upper layer picture information propagates in the lower layer. The area of the target regions in the lower layer image is reduced, so filling quality will improve. This is also the main innovation of our paper. The algorithm process is listed as Algorithm 1.

The number of decomposition is bounded by the area of the filling regions and the resolution of image. If the target regions disappear when decomposing the target image, we should stop decomposing the image. In addition, we adopt an exemplar-based technique to fill target regions so that the patch size will also affect the number of the decomposition layer. The formula to compute the number of decomposition is defined as

$$N = \lfloor \log_2[min(I_{width}, I_{height})] \rfloor - \lceil \log_2[min(patch_{width}, patch_{height})] \rceil \tag{6}$$

where I_{width}, I_{height}, $patch_{width}$, and $patch_{height}$ are the width, height of image, width, and height of the patch respectively.



(a)

1	2	3
4	5	6
7	8	9

(b)

3	6	9
2	5	8
1	4	7

(c)

9	8	7
6	5	4
3	2	1

(d)

7	4	1
8	5	2
9	6	3

(e)

3	2	1
6	5	4
9	8	7

(f)

1	4	7
2	5	8
3	6	9

(g)

7	8	9
4	5	6
1	2	3

(h)

9	6	3
8	5	2
7	4	1

FIGURE 4. Multi-direction match. The four rotate cases and the four reversals of a patch of 3 × 3 size: (a) 0° direction patch. (b) 90° direction patch. (c) 180° direction patch. (d) 270° direction patch. (e) horizontally flipped 0° direction patch. (f) vertically flipped 90° direction patch. (g) horizontally flipped 180° direction patch. (h) vertically flipped 270° direction patch.

B. INPAINTING ALGORITHM OF THE TOP LEVEL IMAGE

The top level picture is inpainted first. We employ an exemplar-based technique for filling the target regions because of its advantage in inpainting large missing regions and inpainting texture and structures simultaneously. Here, searching for the best matching patch is different from the process of an exemplar-based technique in [9]. We search similar patches in four directions with respect to 0°, 90°, 180°, and 270° and in four reversal cases with respect to horizontally flipping the 0° direction patch, vertically flipping the 90° direction patch, horizontally flipping the 180° direction patch, and vertically flipping the 270° direction patch. Fig.4 shows these eight cases. In the paper, we call these eight cases of the search. It can search for more proper patches under the conditions of rotate and reversal. The inpainting result from this will be better than the result from an exemplar-based technique [9].

Assume that f is a transformation function and $[f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8]$ denote eight cases with respect to the four rotate and four reversal patches. We look for a minimum distance patch. The distance is computed as follows.

$$D^m_{SSD}(\psi_p, \psi_q) = \sqrt{\sum [\psi_p - f_i(\psi_q)]^2}, i \in [1, 8] \tag{7}$$

where $D^m_{SSD}(q)$ presents the Euclidean distance between ψ_p and $f_i(\psi_q)$. We adopt the square of the difference between two given pixel patches. We should find patch $\psi_{\hat{q}}$, which has the shortest distance to patch ψ_p. The $\psi_{\hat{q}}$ is defined as

$$\psi_{\hat{q}} = \mathrm{argmin} D^m_{SSD}(\psi_p, \psi_q), m \in [1, 8] \tag{8}$$

Next, the unknown pixel in the ψ_p will be filled by the corresponding pixel in the $f_i(\psi_{\hat{q}})$.

C. INPAINTING ALGORITHM OF THE MIDDLE LEVEL IMAGE

After inpainting the top level picture, we employ an up-sampling operation by filling zeros into the even rows and even columns. Fig.3(c) shows the result of the up-sampling operation. Assume that the image (L^{i+1}) is the top level picture and the L^i_c image is the combined result. Thus, we need to inpaint the L^i_c image using the guidance of the upper level picture. By comparing Fig.3(d) with Fig. 3(e), we can see that the area of the target region in Fig.3(e) is reduced and the difficulty of repair is reduced. The inpainting results of Fig.3(e) would be better than the results of Fig.3(d).

In addition, the resolution of the lower layer picture is higher than the resolution of the upper layer picture. This is because of the effect of noise which makes the priority of computing in the lower layer picture more easily affected. Here, we can further utilize the upper layer information for improving the calculation of priority.

$$D(p_x) = \alpha D_1(p_x) + (1 - \alpha) D_2(p_x) \tag{9}$$

$$D_1(p_x) = \frac{\left| (\nabla L^i_c)^{\perp}_{p_x} \cdot n_{p_x} \right|}{\alpha} \tag{10}$$

$$D_2(p_x) = \frac{\left| (\nabla \hat{L}^{i+1}_{up-nn})^{\perp}_{p_x} \cdot n_{p_x} \right|}{\alpha} \tag{11}$$

where α is the scale factor, ∇ is the gradient image, and n_{p_x} is a unit vector orthogonal to the front δT in the point p_x with respect to L^i_c. The calculation of $D(p_x)$ is more noise resistant. In order to get the best value of α, we calculate the cumulative values of PSNR in about 30 images. The PSNR is computed by equation(12).

$$PSNR = 10 \times \log\left(\frac{(2^n-1)^2}{MSE}\right)$$
$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\|A(i,j)-B(i,j)\|^2 \tag{12}$$

where the MSE is the mean square error with respect to image A and the image B; the term $n = 8$ denotes grayscale image.

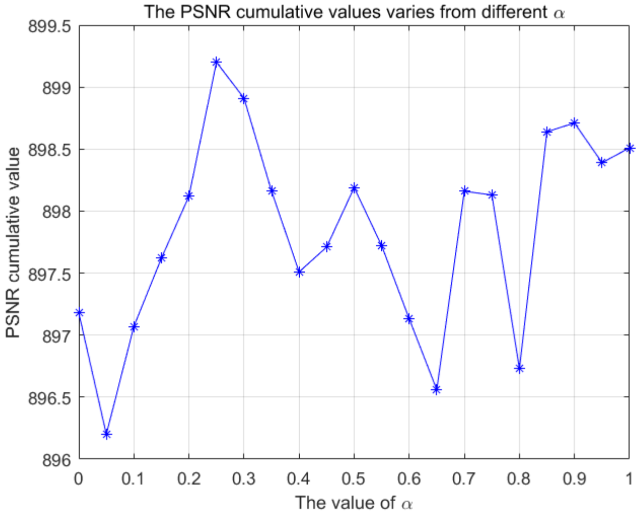


FIGURE 5. **The changes of the cumulative values with α.**

Fig.5 shows the change of the cumulative values with α. When $\alpha = 0.25$, cumulative values of PSNR reach the peak value. Thus, we set $\alpha = 0.25$ in all experiments in this paper. The process for the data term of the middle layer is noted in Algorithm 2.

Algorithm 2 Calculating the Data Term in the Middle Layer

```
Input: the combined image ($L_c^i$), mask of the combined
       image ($M_c^i$), the upper layer picture ( $\hat{L}^{i+1}$), $\alpha$
Output: the data term of the exemplar at the edge of the
        target region
Set $\alpha$ value ($\alpha = 0.25$ is default);
Get the up-sampling (using nearest neighbor algorithm)
    image ($\hat{L}_{\text{up}-nn}^{i+1}$) of $\hat{L}^{i+1}$;
Calculate the points of the edge of the target region of
    the $L_c^i$ by $M_c^i$ information;
Calculate data terms ($D_1$) of the patch at the edge of the
    target region of $L_c^i$ ;
Calculate data terms ($D_2$) of the patch at the edge of the
    target region of $\hat{L}_{up-nn}^{i+1}$;
Calculate the data term ($D$) of the patch at the edge of
    the target region of $L_c^i$; /* Using (9) */
return $\hat{L}^1$;
```

D. CHOOSE THE BEST CANDIDATE PATCH

In the process of inpainting, the best patch is not unique. The reason is that there is more than one minimum distance. If we randomly choose one, it may not be suitable for filling. We must pick the most suitable patch.

Assume the patch ψ_p (Fig.6(a)) is to be filled. There exist several candidate patches ($\psi_1, \psi_2, \cdots, \psi_{\text{m}}$, $m \in [2, 8]$). The filling results of ψ_p are $\hat{\psi}_1, \hat{\psi}_2, \cdots, \hat{\psi}_m$ (Fig.6(c)) by $\psi_1, \psi_2, \cdots, \psi_{\text{m}}$. ψ_p' (Fig.6(b)) is the patch in the picture, which is the up-sampling image made using nearest neighbor interpolation. ψ_p and ψ_p' are located in the same position of the different images. We aim to find $\hat{\psi}_{m'}$, which gets a maximum structural similarity index measurement (SSIM) value between $\hat{\psi}_m$ ($m \in [2, 8]$) and ψ_p'. The SSIM is defined as

$$SSIM(\text{a}, b) = \frac{(2\mu_a\mu_b + C_1)(2\sigma_{ab} + C_2)}{(\mu_a^2 + \mu_b^2 + C_1)(\sigma_a^2 + \sigma_b^2 + C_2)} \tag{13}$$

where $\mu_a, \mu_b, \sigma_a, \sigma_b$, and σ_{ab} are the local means, the standard deviations, and the cross-covariance for images a and b, respectively. C_1 and C_2 are constant values. The larger the SSIM value of the two images, the better the structural consistency of the two images. The best filling patch $\hat{\psi}_{m'}$ should be met (14)

$$\hat{\psi}_{m'} = \arg\max SSIM(\hat{\psi}_m, \psi_p') \tag{14}$$

If $\hat{\psi}_{m'}$ is picked out so that we can find a corresponding patch, $\psi_m, m \in [2, 8]$, we can greatly reduce the number of candidate patches after such processing.

IV. RESULTS

To test our method, we checked its capability to fill structure and texture images as shown Fig.7-Fig.12. (a), (b), (c), (d), (e), (f), (g), and (h) are the original image, target image, inpainting result of our method, inpainting result of Criminisi's method [9], inpainting result of Barnes's method [1], inpainting result of Newson's method [19], inpainting result of Jurio's method [26], and inpainting result of Anupam's method [27], respectively.

Fig.7, Fig.8, Fig.9 belong to the structure image. They are the circle, grid, and arch images, respectively. The results show that our method is better than others [1], [9], [19], [26], [27]. Criminisi's method generates error when inpainting because there are multiple candidate patches, with a patch chosen randomly from multiple candidate patches. Fig.6 shows the error result of this patch inpainting. Thus, Criminisi's [9] method cannot obtain satisfactory results. Newson's methods [19] also works well, and is close to our method, as seen in Fig.7(f), Fig.8(f), and Fig.9(f). Barnes's method [1] result of inpainting is not too bad, as shown in Fig.7(e), Fig.8(e), and Fig.9(e). It is very close to the Newson's method [19], because Newson's method is an improvement method based on the Bar's method, Newson modifies the measure of similar sample blocks by adding gradient term. Anupam's method [27] use assigned resource surrounding target regions, sometimes this can lead

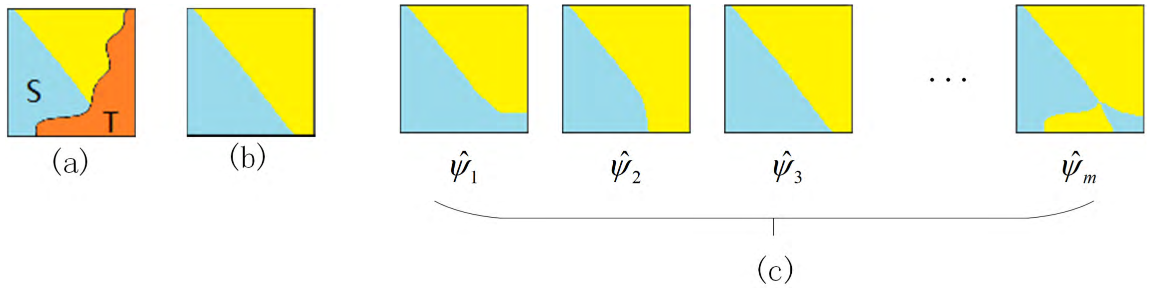


FIGURE 6. Pick out best patch from the candidate patches. (a) The patch ψ_p to be filled. The red region is target region. (b) The ψ'_p patch which is located in the picture using nearest neighbor interpolation up-sampling of the upper layer. (c) The filling results($\hat{\psi}_1, \hat{\psi}_2, \cdots, \hat{\psi}_m$) of ψ_p using $\psi_1, \psi_2, \cdots, \psi_m$.

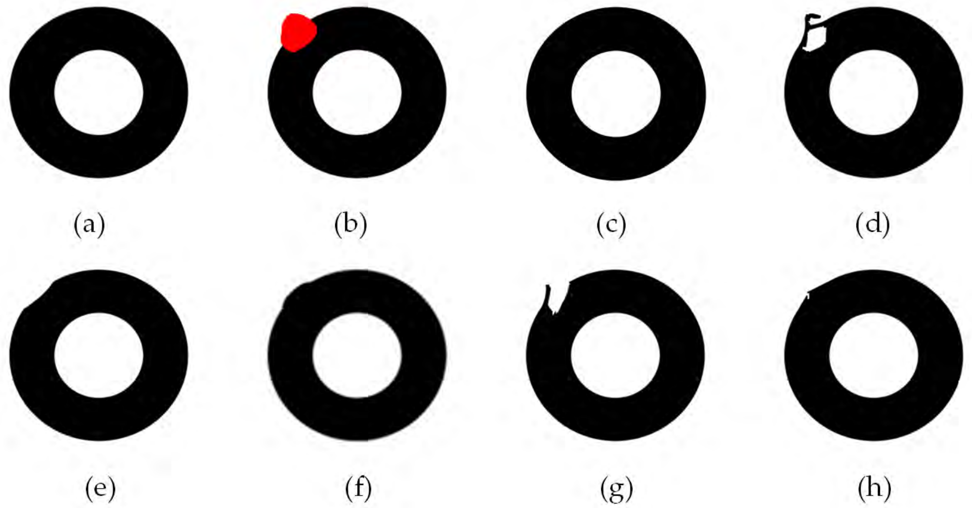


FIGURE 7. Circle structure image inpainting. (a) Origin image. (b) Target image. (c) Our method. (d) Criminisi's method [9]. (e) Barnes's method [1]. (f) Newson's method [19]. (g) Jurio's method [26]. (h) Anupam's method [27].

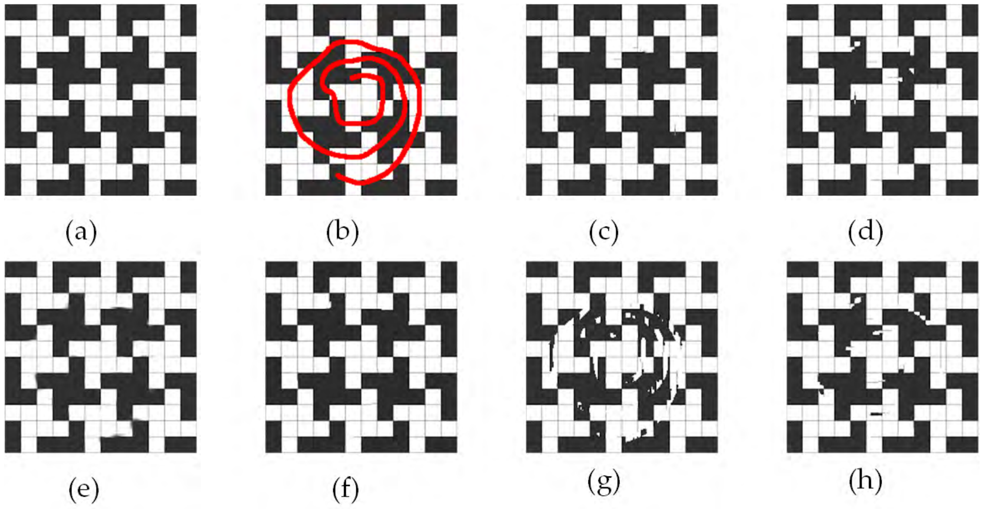


FIGURE 8. Grid structure image inpainting. (a) Origin image. (b) Target image. (c) Our method. (d) Criminisi's method [9]. (e) Barnes's method [1]. (f)Newson's method [19]. (g) Jurio's method [26]. (h) Anupam's method [27].

to failures filling due to insufficient resources, as shown in fig.7(h), Fig.8(h), and Fig.9(h). Jurio's method [26] is the worst method, because the number of clusters is difficult to determine, so it will affect the search region and ultimately affect the repair results, as shown in Fig.7(g), Fig.8(g), and Fig.9(g).

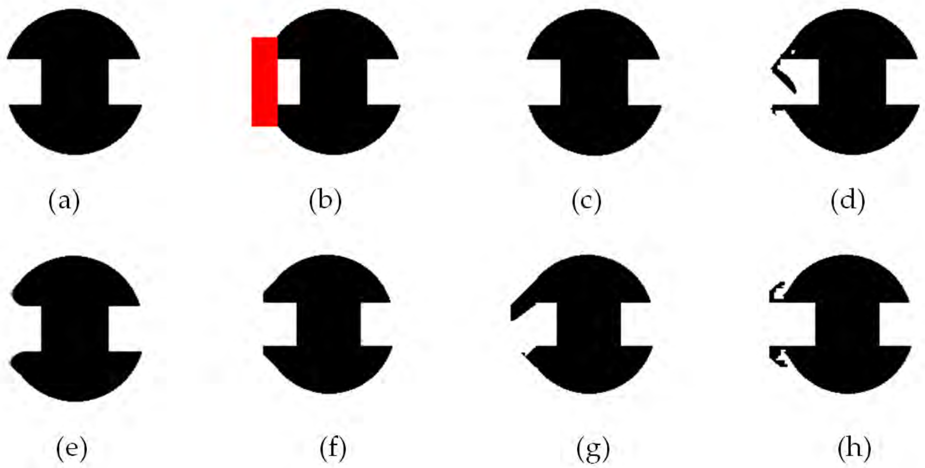


FIGURE 9. Arch structure image inpainting. (a) Origin image. (b) Target image. (c) Our method. (d) Criminisi's method [9]. (e) Barnes's method [1]. (f)Newson's method [19]. (g) Jurio's method [26]. (h) Anupam's method [27].

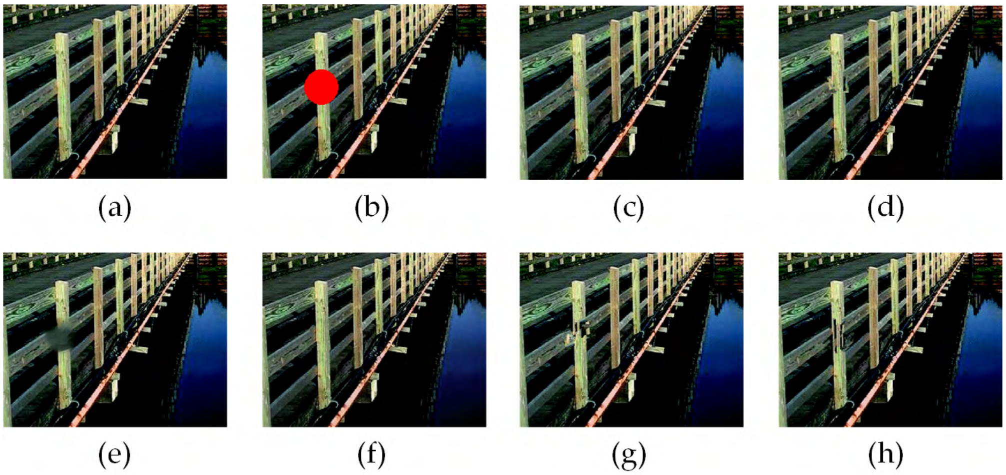


FIGURE 10. Texture and structure image inpainting of fence. (a) Origin image. (b) Target image. (c) Our method. (d) Criminisi's method [9]. (e) Barnes's method [1]. (f)Newson's method [19]. (g) Jurio's method [26]. (h) Anupam's method [27].

Fig.10-Fig.12 shows the texture and structure inpainting. There is some difficulty in repairing these images. The images in Fig.10, Fig.11, and Fig.12 are fence, wall, and window pictures. Barnes's method [1] cannot obtain satisfactory results, lines are fractured such as in the fence and window image as shown Fig.10(e) and Fig.12(e). Criminisi's method [9] has some tiny flaws when repairing fence and wall as shown in Fig.10(d) and Fig.11(d). Newson's method [19] and our method have obtained good results as shown in (c) and (f) in Fig.10-Fig.12. In addition, the filling results of methods Jurio's and Anupam's method are not satisfactory as shown in (g) and (h) in Fig.10-Fig.12.

If visual judgment could not be obtained, we used PSNR, SSIM, and Feature Similarity Index Measurement (FSIM) [28] to help us analyze the results. Table 1, Table 2 and Table 3 give the PSNR, SSIM and FSIM values of our method and other's method. These values measured with our method are, in many cases, higher than the values of other methods. The PSNR and FSIM value of our method are lower than Criminisi's method for filling the window picture as shown in table 1 and table 3, but the result of our method is better than the result of Criminisi's method [9] by vision as shown in Fig.12(c) and Fig.12(d). Otherwise, the SSIM value of our method is equal or greater than other methods.

Fig.13, Fig.14 shows the inpainting of the removed object. Fig.13(c), Fig.14(c) are the result of Criminisi's method; the repair results were of poor quality. Fig.13(b), Fig.14(b) are the pleasing result of our method, which can obtain a similar result with Barnes's [1] method and Newson's [19] method, as shown in Fig.13(d), Fig.14(d), Fig.13(e) and Fig.14(e). In addition, Jurio's method [26] results are not physically plausible as shown in Fig.13(f) and Fig.14(f).

To illustrate the inpainting ability of our method, the regions to be filled having highly textured irregular patterns, such as 'water', 'grass', or 'flower' images as shown in [9], have been filled as shown in Fig.15. The first columns

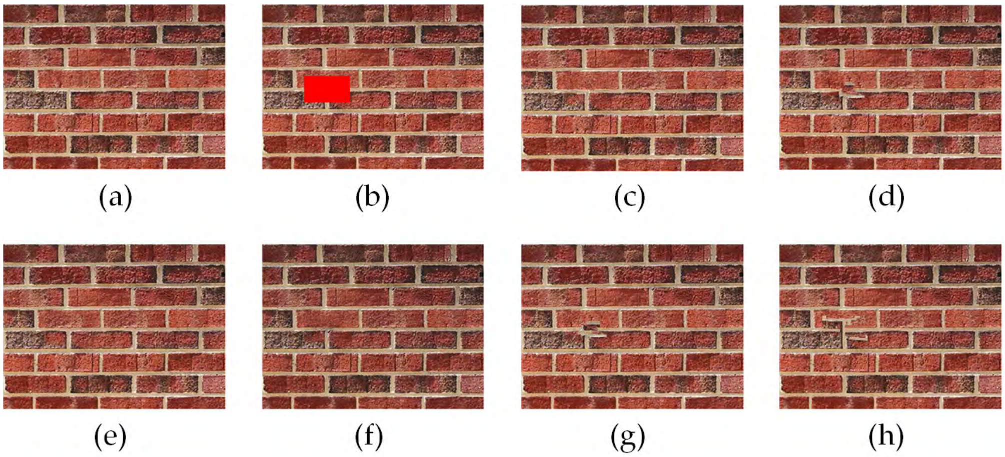


FIGURE 11. Texture and structure image inpainting of wall. (a) Origin image. (b) Target image. (c) Our method. (d) Criminisi's method [9]. (e) Barnes's method [1]. (f)Newson's method [19]. (g) Jurio's method [26]. (h) Anupam's method [27].

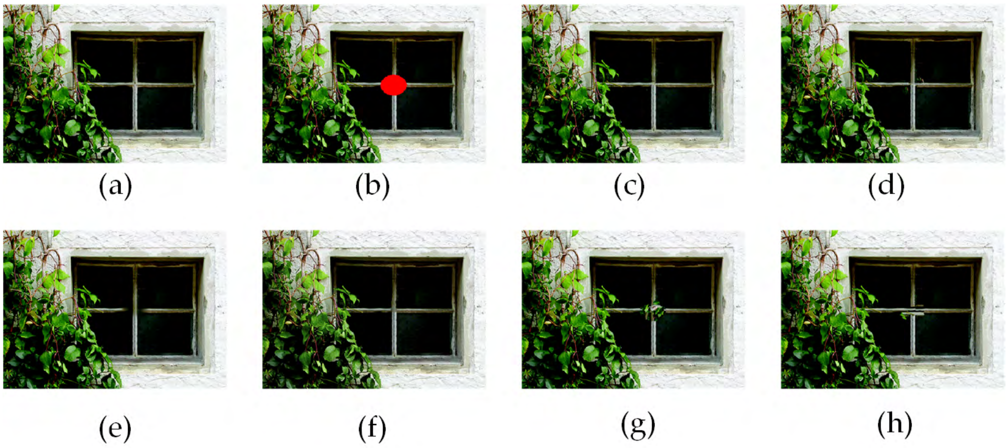


FIGURE 12. Texture and structure image inpainting of window. (a) Origin image. (b) Target image. (c) Our method. (d) Criminisi's method [9]. (e) Barnes's method [1]. (f)Newson's method [19]. (g) Jurio's method [26]. (h) Anupam's method [27].

TABLE 1. A comparison of the PSNR values.

Image	Our method	Criminisi's [9]	Barnes's [1]	Newson's [19]	Anupam's [27]	Jurio's [26]
circle	**43.22**	19.80	26.80	27.07	30.42	21.30
grid	**36.52**	30.97	28.72	30.91	23.04	16.81
arch	**30.69**	19.95	24.64	25.56	19.93	19.29
fence	**31.40**	30.41	26.48	30.22	14.58	27.59
wall	**31.42**	28.92	30.34	26.73	28.44	28.81
window	35.63	**36.35**	32.87	31.33	30.82	29.42

TABLE 2. A comparison of the SSIM values.

Image	Our method	Criminisi's [9]	Barnes's [1]	Newson's [19]	Anupam's [27]	Jurio's [26]
circle	**0.9973**	0.9789	0.9924	0.9924	0.9950	0.9833
grid	**0.9927**	0.9773	0.9771	0.9883	0.9617	0.8768
arch	**0.9895**	0.9682	0.9850	0.9881	0.9715	0.9693
fence	**0.9864**	0.9855	0.9809	0.9746	0.8036	0.9828
wall	**0.9897**	0.9862	0.9877	0.9800	0.9841	0.9857
window	**0.9947**	0.9943	**0.9947**	0.9784	0.9923	0.9895

TABLE 3. A comparison of the FSIM values.

Image	Our method	Criminisi's [9]	Barnes's [1]	Newson's [19]	Anupam's [27]	Jurio's [26]
circle	**0.9994**	0.9829	0.9920	0.9917	0.9949	0.9865
grid	**0.9950**	0.9865	0.9719	0.9926	0.9765	0.9315
arch	**0.9963**	0.98	0.9890	0.9925	0.9806	0.9787
fence	**0.9880**	0.9846	0.9814	0.9866	0.8252	0.9815
wall	**0.9909**	0.9875	0.9892	0.9899	0.9839	0.9861
window	0.9938	**0.9946**	0.993	0.9902	0.9890	0.9874

shows the original images named dog, flower, grass, and sea. The second columns shows the target images which specified the removing regions in red. The third columns in Fig.15 shows the infilled results of our method, the inpainted results are generally pleasing. Criminisi's method failed to inpaint as shown in fourth columns in Fig.15. The repair ability of the Barnes's method also performed well except failed to inpaint flower due to infilling a part of cloth or body into flowers. Newson's method failed to inpaint flower and sea image as shown in Fig.15(f). The Jurio's method removed the dog well in Fig.15(g), but other images failed to infill.

Our method searches for best patch in eight cases, so it will consume more time in the searching patch than Criminisi's method, Table 4 lists a time comparisons of several methods.

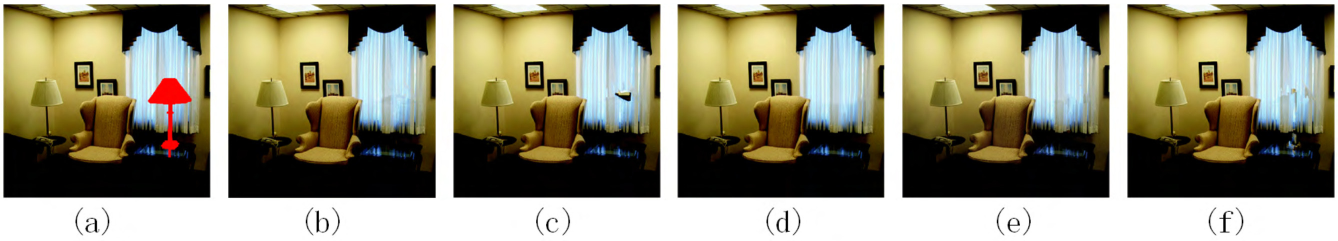


FIGURE 13. Lamp removing. (a) Target image. (b) Our method. (c) Criminisi's method [9]. (d) Barnes's method [1]. (e) Newson's method [19]. (f) Jurio's method [26].

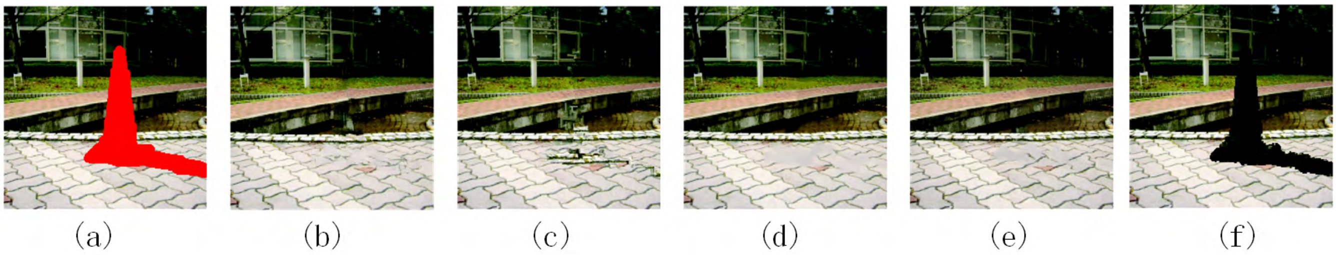


FIGURE 14. People removing. (a) Target image. (b) Our method. (c) Criminisi's method [9]. (d) Barnes's method [1]. (e) Newson's method [19]. (f) Jurio's method [26].

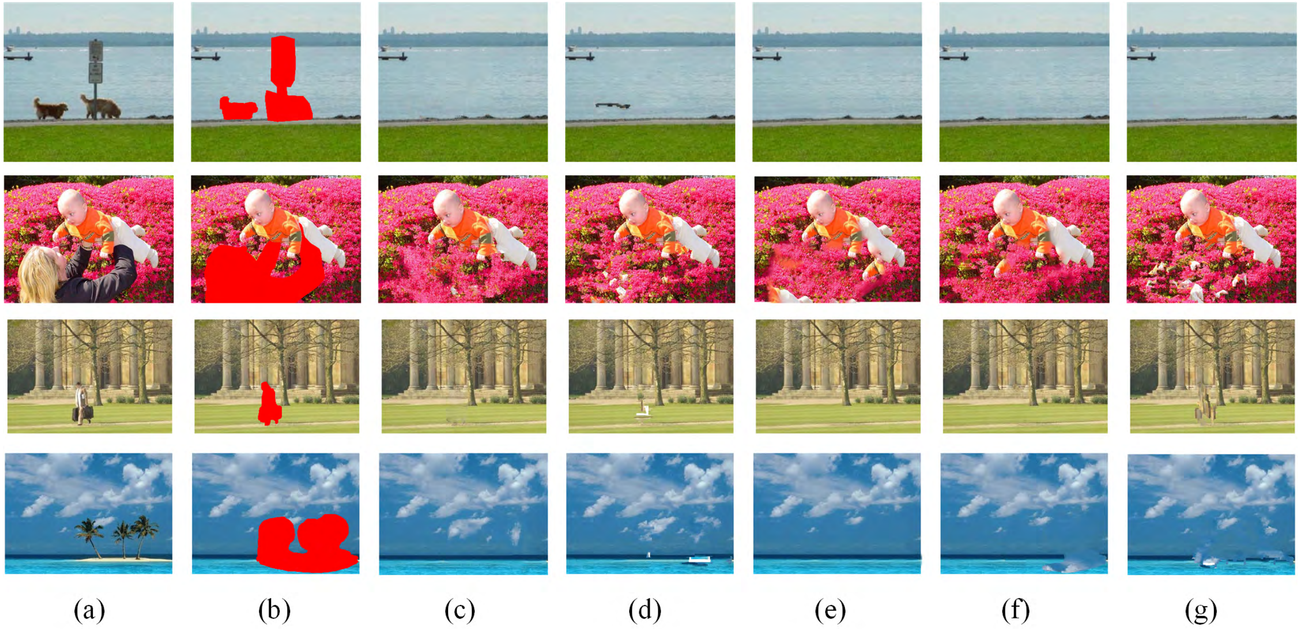


FIGURE 15. Inpainting with highly textured irregular patterns. The first columns shows four original images named as dog, flower, grass, sea. (a) Origin image. (b) Target image.(c) Our method. (d) Criminisi's method [9]. (e) Barnes's method [1]. (f) Newson's method [19]. (g) Jurio's method [26].

From the table 4, our method spends more time than others, because it searches eight cases in the global image. Anupam's method [27] spends lower time than others owing to searching patches within its adjacent regions, this will shorten the time. Our experiments are carried out on an Intel core i7CPU@2.0GHz using Matlab implementation (Matlab 9.2.0 with 64 bit). So our method also requires further improvements to reduce the time when searching for sample patches.

V. DISCUSSION

Our method uses an exemplar-based technique. To avoid choosing the wrong patch, we consider four cases of rotation and four cases of inversion so the most suitable patch can be found and some errors avoided. We applied a multiple resolution inpainting method. The filling region area in the top layer is small. It can produce good repair results, so we can utilize the filled result of the top layer image in the next layer, and this is equivalent to giving the overall profile information.

TABLE 4. The comparison of the computation time of several algorithmes(in seconds).

Image	Size of image	Percentage area to be removed	Our method	Criminisi [9]	Barnes [1]	Newson [19]	Anupam [27]	Jurio [26]
circle	300×249	1.06	30.84	1.56	11.81	10.01	0.70	21.88
grid	200×200	11.19	40.91	9.35	8.32	20.36	1.71	15.80
arch	225×189	5.44	34.29	1.91	14.18	28.31	0.46	12.92
fence	512×400	2.41	195.70	13.17	28.83	34.83	3.87	80.20
wall	296×224	3.24	26.82	1.85	9.88	22.79	0.53	20.19
window	512×352	1.19	77.89	5.68	18.34	29.47	1.48	53.33
lamp	644×604	2.69	777.41	54.91	52.36	78.26	17.95	201.61
road	645×647	11.18	4576.75	285.2	112.17	136.81	90.30	546.52
dog	361×343	11.11	361.45	27.51	20.30	29.02	8.41	61.88
flower	460×346	25.83	1074.78	85.99	369.00	116.34	30.04	171.34
grass	378×266	3.23	55.59	24.66	15.46	20.13	1.42	31.05
sea	315×250	20.55	174.97	20.80	143.13	48.02	12.66	39.85

Screen window propagating inpainting has two advantages. First, it can give more detail when filling the next layer as the candidate patches can be further reduced; second, it fixes some pixels in the next layer, as shown in Fig.3(c). Towards the aim of avoiding high-frequency information interference, such as noises, calculating data items using upper layer repair results makes calculating for the priority more reasonable. In most cases, there is no one best patch, so we pick the best one with the help of SSIM. All candidate patches try to fill in the area. Whichever can maximize the SSIM value is the best exemplar. This method can obtain the most suitable patch. The number of decomposition layers is calculated automatically, avoiding the trouble of manual selection. The user can choose the size of the exemplar for optimal repair.

VI. CONCLUSIONS

In the paper, we propose the inpainting method of screen window propagating. It considers the whole image and its parts. The inpainting starts from top layer. The whole filling is very important; in our experiments, the top layer can obtain satisfactory results. Screen window propagating from the upper layer to the lower layer maintains the whole profile. Our method is better than multiple resolution inpainting [1], [9], [19], [26], [27].

REFERENCES

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, Jul. 2009, Art. no. 24.

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM Special Interest Group Comput. Graph. Interact. Techn.*, Jul. 2000, pp. 417–424.

[3] D. H. Lee, S. B. Yoo, M. Choi, J. B. Ra, and J. Kim, "Block Poisson method and its application to large scale image editing," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 2121–2124.

[4] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *Proc. SIGGRAPH*, Jul. 2003, pp. 313–318.

[5] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *J. Vis. Commun. Image Represent.*, vol. 12, no. 4, pp. 436–449, May 2001.

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

[7] M. Bertalmío, 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.

[8] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 1033–1038.

[9] 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.

[10] M. Daisy, P. Buyssens, D. Tschumperlé, and O. Lézoray, "A smarter exemplar-based inpainting algorithm using local and global heuristics for more geometric coherence," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 4622–4626.

[11] M. Ghorai and B. Chanda, "An image inpainting method using pLSA-based search space estimation," *Mach. Vis. Appl.*, vol. 26, no. 1, pp. 69–87, Jan. 2015.

[12] C. Guillemot, M. Turkan, O. Le Meur, and M. Ebdelli, "Object removal and loss concealment using neighbor embedding methods," *Signal Process., Image Commun.*, vol. 28, no. 10, pp. 1405–1419, Nov. 2013.

[13] V. Kumar, J. Mukhopadhyay, and S. K. D. Mandal, "Modified exemplar-based image inpainting via primal-dual optimization," in *Proc. PReMI*, Warsaw, Poland, Jun. 2015, pp. 116–125.

[14] C. Xiang, Y. Cao, P. Duan, and L. Shi, "An improved exemplar-based image inpainting algorithm," in *Proc. 9th Int. Conf. Comput. Sci. Educ.*, Oct. 2014, pp. 770–775.

[15] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *Proc. 11th Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 29–43.

[16] O. Le Meur and C. Guillemot, "Super-resolution-based inpainting," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 554–567.

[17] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.

[18] S. Padmavathi, B. Priyalakshmi, and K. P. Soman, "Hierarchical digital image inpainting using wavelets," *J. Signal Image Process.*, vol. 3, no. 4, pp. 85–93, Aug. 2012.

[19] A. Newson, A. Almansa, Y. Gousseau, and P. Pérez, "Non-local patch-based image inpainting," *Image Process. On Line*, vol. 7, pp. 373–385, Dec. 2017.

[20] V. Fedorov, G. Facciolo, and P. Arias, "Variational framework for non-local inpainting," *Image Process. On Line*, vol. 5, pp. 362–386, Dec. 2015.

[21] 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, Jul. 2011.

[22] Y.-Q. Wang, "E-PLE: An algorithm for image inpainting," *Image Process. On Line*, vol. 3, pp. 271–285, Dec. 2013.

[23] X. Lin and X. Yang, "Effective exemplar-based image inpainting using low-rank matrix completion," in *Proc. IEEE 7th Int. Conf. Awareness Sci. Technol.*, Sep. 2015, pp. 37–42.

[24] N. Komodakis and G. Tziritas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2649–2661, Nov. 2007.

[25] Y. Pritch, E. Kav-Venaki, and S. Peleg, "Shift-map image editing," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 151–158.

[26] A. Jurio, D. Paternain, M. Pagola, C. Marco-Detchart, and H. Bustince, "Two-step algorithm for image inpainting," in *Proc. Eur. Soc. Fuzzy Logic Technol.*, Sep. 2017, pp. 302–313.

[27] Anupam, P. Goyal, and S. Diwakar, "Fast and enhanced algorithm for exemplar based image inpainting," in *Proc. Pacific-Rim Symp. Image Video Technol.*, 2010, pp. 325–330.

[28] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

HUAMING LIU received the B.S. degree in computer science and technology from Shangqiu Normal University, Shangqiu, China, in 2005, and the M.S. degree in computer software and theory from Northwest Minzu University, Lanzhou, China. He is currently pursuing the Ph.D. degree with the Nanjing University of Posts and Telecommunications, Nanjing, China. He was a Visiting Scholar with the University of Science and Technology of China, Hefei, China, in 2015. He is currently an Associate Professor with the School of Computer and Information Engineering, Fuyang Normal University, Fuyang, China. His research interests include image processing, pattern recognition, and software engineering.

XUEHUI BI was born in Wuhu, Anhui, China, in 1982. She received the B.S. degree in mathematics and applied mathematics from Fuyang Normal University, Fuyang, China, in 2004, and the M.S. degree in applied mathematics from the Hefei University of Technology, Hefei, China, in 2007. Her research interests include image processing and pattern recognition.

GUANMING LU received the B.E. degree in radio engineering and the M.S. degree in communication and electronic systems from the Nanjing University of Posts and Telecommunications (NUPT), Nanjing, China, in 1985 and 1988, respectively, and the Ph.D. degree in communication and information systems from Shanghai Jiao Tong University, Shanghai, China, in 1999. He is currently a Professor with the College of Communication and Information Engineering, NUPT. His current research interests include image processing, affective computing, and machine learning.

WEILAN WANG received the B.S. degree in mathematics from Northwest Normal University, Lanzhou, China, in 1983. She was a Visiting Scholar with the Sun Yat-sen University, Guangzhou, China, in 1987. From 2001 to 2002, she was a Visiting Scholar with Tsinghua University, Beijing, China. From 2006 to 2007, she was a Visiting Scholar with Indiana University Bloomington, Bloomington, USA. She is currently a Professor with the College of Mathematics and Computer Science, Northwest Minzu University, Lanzhou. Her research interests include image processing, pattern recognition, and Tibetan information processing.

JINGJIE YAN received the B.E. degree in electronic science and technology and the M.S. degree in signal and information processing from the China University of Mining and Technology, Beijing, China, in 2006 and 2009, respectively, and the Ph.D. degree in signal and information processing from Southeast University, Nanjing, China, in 2014. Since 2015, he has been with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, as a Lecturer. His current research interests include pattern recognition, affective computing, computer vision, and machine learning.

ZHENGYAN ZHANG received the B.S. degree in electronic information engineering and the M.S. degree in signal and information processing from the Jiangsu University of Science and technology, Zhenjiang, Jiangsu, China, in 2004 and 2007, respectively. He is currently pursuing the Ph.D. degree with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications. His current research interests include pattern recognition, machine learning, and computer vision.

• • •