Exemplar-based Image Inpainting with Multi-resolution Information and the Graph Cut Technique

HUAMING LIU, XUEHUI BI, GUANMING LU, AND WEILAN WANG

1 College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003 China (e-mail: 2017010217@njupt.edu.cn)
2 School of Computer and Information Engineering, Fuyang Normal University, Fuyang 236037 China (e-mail: bixuehui888@163.com)
3 Key Laboratory of China’s Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University, Lanzhou 730000 China (e-mail: wangweilan@xbmu.edu.cn)

Corresponding author: Guanming Lu (e-mail: lugm@njupt.edu.cn)

ABSTRACT Filling holes in an image is achieved in a manner similar to peeling the onion. The order of filling affects the image inpainting results, especially concerning the content of complex images. When high-resolution images are used to extract edge information, they are susceptible to high-frequency information, such as complex textures and noise. Furthermore, edge information is extracted in different resolutions, while the main contour information of the image can be obtained more easily. In this paper, multi-resolution information is used to prioritize which target patches in an image to fill, which helps to elucidate the optimal sequence for image repair. Multi-resolution images provides more information than single-resolution images, and similar patches are computed on multi-resolution images to obtain multiple candidate patches. Similar patch calculations use a variety of information on colors, gradients and boundaries to more accurately search for similar patches. We chose the most reasonable candidate patch by means of the structural similarity index measure (SSIM). When pasting the patch to fill the target region, we used graph cut technology to eliminate blockiness. Compared with the state-of-the-art repair algorithm, the experimental results prove that the proposed repair algorithm can repair the image very well.

INDEX TERMS exemplar-based inpainting technique, priority calculation, patch matching, graph cut, multi-resolution information

I. INTRODUCTION

IMAGE inpainting aims to fill corrupted regions such as scratches, damaged regions, dates, and so on. Image inpainting is also called image completion, image restoration, or image repairing. Restoration with noise, blur, etc. is called blind restoration. Our work is dedicated to non-blind image restoration, which is employed to fill large regions. Bertalmio et al. firstly proposed inpainting in Reference [1]. His approach was a diffusion-based method, in which the filling process involves diffusing information by isophotes from the known regions to the target regions. We call this type of approach the partial differential equation (PDE) [2] or the total variational method [3], [4]. As PDE can be derived from variational principles, these approaches can propagate smooth level lines; however, the texture region cannot be well maintained. Therefore, we consider them structural repair methods. They generally inpaint small regions well, but fail to fill large regions when the image to be repaired is rich in texture.

Exemplar-based techniques involve copying and pasting similar patches in the image using a searching strategy. This technique was initiated by Efros et al. for texture synthesis [5], in which each pixel value of the target region is copied from the sample in the image based on a similar patch. The exemplar-based technique has also been employed for inpainting [6]–[8]. Contrary to the geometry-based method, the exemplar-based technique is non-local, and searches for the matching patch in the whole image. Target images with textures or repetitive patterns can be excellently inpainted using this technique. However, this approach will fail to fill an image geometry correctly if a matching sample cannot be found in the image.

Bertalmio et al. [9] proposed an approach which combines geometry and texture repair. In this method, the target im-
The current error map, which uses data optimization strategies, exemplar-based technique, new pixels are added by shaking Karos et al. [18] proposed optimizing data for inpainting of however, the filling of large texture regions is still blurry. was greedy scheme. In [15], [32], the minimization problem regularization was presented, unfortunately, this algorithm correspondence map in the known part. The total variation where each pixel in the missing regions was assigned a man algorithm. In [17], variational method was proposed, process used total variation and is optimized using split Bregman thesis. But EM-like algorithms call for initialization which schemes was applied for inpainting image or texture synthesis. Anupam et al. [10] proposed an approach which searches for exemplar patches near the missing regions, so searching for samples takes less time. Similar to work [10], Ghorai et al. [11] presented a group-based image inpainting method in a Markov random field (MRF) framework. In this method, known regions are divided into different groups; subsequently searching for candidate patches in the same group and refining the patch scheme makes patch identification more reasonable in addition to reducing the search time.

The above exemplar-based methods are greedy procedures, so each pixel of target region is process only once, it leads to the inpainting result sensitive to the order. Oppositely, some techniques [12]–[15], [32] formulate restoration as an optimization problem. In [13], [14], the image is modeled through MRF, and objective function is minimized by belief propagation. In [7], [16], expectation-maximization (EM) schemes was applied for inpainting image or texture synthesis. But EM-like algorithms call for initialization which may stuck into local minima. In [15], [32], the inpainting process used total variation and is optimized using split Bregman algorithm. In [17], variational method was proposed, where each pixel in the missing regions was assigned a correspondence map in the known part. The total variation regularization was presented, unfortunately, this algorithm was greedy scheme. In [15], [32], the minimization problem of total variation was solved by split Bregman algorithm, however, the filling of large texture region art still blurry. Karos et al. [18] proposed optimizing data for inpainting of exemplar-based technique, new pixels are added by shaking the current error map, which use data optimization strategies for inpainting.

In addition to the above optimization methods, matrix completion has recently become a popular image restoration method. If the rank of an image is low, it means that there is a lot of duplicate information in the image. Candès et al. [19] proposed a low-rank theory to complete the matrix. Viewing image data as a matrix, low-rank theory was applied to recover the missing region of image matrix [20]. They solved the problem using the accelerated proximal gradient line (APGL). They illustrated many experimental images in which the missing regions were small and the rank of the image was low. Liu et al. [21] presented tensor completion for image inpainting, and provided several solutions for different conditions. However, when some images have a higher rank, the final result is not pleasing, as mentioned in [22]; images with irregular or near-random texture patterns cannot be inpainted plausibly since the low-rank assumption is violated.

The exemplar-based technique [6] has an advantage for inpainting large region with missing texture information. However, it also has several drawbacks: (1) the filling order affects the repair results due to the high-frequency components of full-level images; (2) the best sample patch randomly selected is not necessarily the most suitable patch, which is located by Euclidean distance on the color component; and (3) copying and pasting patches may generate the appearance of visual artefacts such as block effect (blockiness). In this paper, we propose a novel inpainting approach which is optimized using multi-resolution information and the graph cut technique. Priority, which ensures a more reasonable order of filling, is computed using multi-resolution image information. The search for the best sample block is carried out more efficiently by adding gradient and boundary constraints. Images are decomposed into multi-resolution images, and they are also viewed as image resources, which enriches the sample resources of a given image.

After finding a suitable sample block, we use the graph cut technique to copy/paste and to eliminate blockiness. Here, we summarize our contribution in this paper. First, we calculate the data terms by means of multi-resolution information. Second, we search for similar patches using color terms, gradient terms and boundary terms, making it easier to identify similarities among patches. Third, multi-resolution images are all considered to be sample resources, which further expands the shortage of single image resources. Fourth, we use the structural similarity index measure (SSIM) to pick out

![The inpainting framework of our work.](image-url)
best similar patch. Fifth, we eliminate blockiness by means of the graph cut technique.

Our inpainting framework is shown in Fig.1.

This paper is organized as follows. Related studies are presented in Section II. Our method is mainly introduced in Section III. The content mainly involves priority calculation, similar block search, and determination of the best patch. The graph cut technique is used to fill target region to eliminate blockiness in Section IV. Experimental results and conclusions are discussed in Sections V and VI, respectively.

II. RELATED WORK

In recent years, exemplar-based approaches have been widely used in image restoration. In response to the shortage of exemplar-based techniques, many research works have provided improved approaches. Multi-resolution image can provide much more information than one. However, Patch filling may generate blockiness effect, the repair results will be affected. In this section, we review the relevant research literature related to our task, including exemplar-based techniques, multi-resolution inpainting and Graph cut technique.

A. EXEMPLAR-BASED TECHNIQUE

The exemplar-based technique have received widespread attention from most scholars. Fig.2 shows the processing of inpainting. Image $I$ denotes the target image, $\Omega$ denotes missing regions or regions to be removed. $S$ ($I = \Omega \cup S$) denotes the known region, also referred to as the sample resources. $\partial \Omega$ denotes the boundary of missing region $\Omega$.

$\psi_p$ is a patch which is centered on boundary $\partial \Omega$. These parameters are shown in Fig.3(a). The priority of patches determines which piece is the first to be filled. Assuming that $\psi_p$ has the highest priority, then similar patches are searched for in the sample regions (S), as shown in Fig. 3(b). Assuming that patch $\psi_{q'}$ and patch $\psi_{q''}$ are similar to $\psi_p$, as shown in Fig. 3(c), then the missing region in $\psi_p$ is filled with the most similar patch. The filling result is shown in Fig. 3(d). The priority calculation is defined as follows:

$$P(p) = C(p)D(p)$$

(1)

where $P(p)$ is the priority of $\psi_p$, whose center is pixel $p$ located on the boundary of the missing regions(target region).

$C(p)$ and $D(p)$ are the confidence term and data term of $\psi_p$, respectively.

$$C(p) = \frac{\sum_{t \in \psi_p \cap S} A(t)}{|\psi_p|}$$

(2)

$$A(t) = \begin{cases} 1, t \in \psi_p \cap S \\ 0, t \in \psi_p \cap \Omega \end{cases}$$

(3)

where $C(p)$ stands for the proportion of known pixel points to the patch $\psi_p$ area. The more information $\psi_p$ contains, the higher the confidence value obtained.

$$D(p) = \frac{\nabla I_p \cdot n_p}{\alpha}$$

(4)

where $\nabla I_p$ is the orthogonal operator in pixel $p$ and $\nabla$ represents the gradient. $n_p$ denotes a unit vector orthogonal to the front $\partial \Omega$ in the point $p$. If the image is a grayscale image, we set $\alpha = 255$.

As the priority determines the sequence of repair, Xu et al. [7] redefined the patch priority based on structure sparsity, allowing the texture and structure information to be better distinguished. Deng et al. [23] separated priority definition according to geometry and texture filling; however, their results still suffered from high-frequency noise, so their proposed fill sequence may not be suitable. People are more sensitive to structural connectivity; therefore, it may be a good solution that filling the target regions is carried out in the structure regions first to avoid error repair. Sun et al. [24] mainly improved the structure propagation, their method encouraged filling the structure region first, following which the remainder would be filled using exemplar-based texture synthesis. However, the structure region needs to be specified manually by the user by extending the necessary curves [24], and this increases the burden on the user and means that the missing region cannot be inpainted automatically. Want et al. [25] redefined confidence and structure-consistent patch matching, but high frequency was also found to affect their priority calculation.

B. MULTI-RESOLUTION INPAINTING

High-frequency components such as noise may affect repair quality by exemplar-based techniques [6]; however, the edge information of low-resolution images is less affected by high-frequency information, as shown in Fig.4. Fig.4(b) is low-resolution image using Gauss convolution kernel. Its edge was obtained using the Canny operator in Fig.4(d). Meanwhile, Fig.4(c) shows the Canny edge at the full-level image of Fig. 4(a). Compared with the Fig. 4(c) image, the Fig. 4(d) image shows less affection when extracting edges. In some previous research works employing the multi-resolution approach, the process of inpainting was carried out from the coarsest level to the finest level. In [7], [8], [26]–[28], the top layer image was inpainted first, then the next layer image was filled, etc. Here, the previous layer was upsampled for initialization, this has become a general method.
of energy optimization using multi-resolution information. The main structures are easily retrieved in the coarse version since the local singularities are less numerous, and thus the inpainting results are less sensitive to noise. In [8], Drori et al. improved the K-NN search method using a rough estimate of the coarse level result. The coarse resolution picture was inpainted, then the result is used as guiding image to search for high-frequency details in the next layer image. However, the drawback of this multi-resolution approach is obvious [26], as error that occurs at a coarse level can spread across all levels of an image. Despite this, in most cases, a fine inpainting effect can be achieved [26]–[28].

C. GRAPH CUT TECHNIQUE
Concerning texture synthesis, Alexei A et al. used minimum error boundary cuts [36] and Vivek K. et al. proposed graph cut texture synthesis, they all used graph theory to solve texture synthesis. Later on, Yunqing L. et al. used the multiscale graph cut technique to inpaint images. In our work, we use the graph cut technique to complete filling work as well as to eliminate blockiness. In section IV, we will detail this technique.

III. PROPOSED APPROACH
In the paper, we mainly use exemplar-based technique, multi-resolution and graph cut technique. Exemplar-based technique can repair texture and structure regions. Our inpainting framework is shown in Fig. 1, including priority defining, patch matching, multi-resolution resources searching, and graph cut technique filling. We introduce those content as follows. Priority definition is introduced in section 3.1. Patch matching strategy is defined in Section 3.2. Searching multi-resolution resources and pick the best patch is described in Section 3.3 and section 3.4, respectively.

A. PRIORITY DEFINITION
Inspired by multi-resolution approaches [26], [27] and confidence calculation [25], we redefined the data terms. This may serve to reduce the effects of high-frequency components, which would make the priority calculation more reasonable.

The data term is computed by:

$$D(p) = \alpha_1 D(p) + \alpha_2 D(p) + \cdots + \alpha_m D(p)$$

where $D(p)$ is the data term of $p$ in the $i$-th scale image. Here, we calculate the priority of $p$ using a multi-resolution image, where $\alpha_i$ is the weight, indicating the degree of contribution. The global outline of the image is provided by the low-resolution image. Therefore, the lower the resolution, the larger the value of $\alpha_i$. In the process of repair, this can provide a better understanding of the global contour. So, $\alpha_i$ is defined as follows:

$$\alpha_i = \frac{\sum_{t=1}^{i} t}{1 + 2 + \cdots + L}, 1 \leq i \leq L$$

FIGURE 3. Exemplar-based inpainting. (a) Target image; (b) Patch $\psi_p$ to be filled, where $p$ has the highest priority; (c) searching for similar patches such as patch $\psi_{q'}$ and patch $\psi_{q''}$; (d) the filling result using the exemplar-based technique.

FIGURE 4. Comparison of edge images at different scales. (a) Camera man; (b) low-resolution image of Camera man using Gauss convolution kernel; (c) the Canny edge of (a); (d) the Canny edge of (b).
where $L$ represents the maximum scale layer of image decomposition computed by:

$$L = \left\lfloor \log_2 \min \{I_{\text{height}_\text{size}}, I_{\text{width}_\text{size}}\} \right\rfloor - \left\lfloor \log_2 \min \{patch_{\text{height}_\text{size}}, patch_{\text{width}_\text{size}}\} \right\rfloor$$

where $I_{\text{height}_\text{size}}$, $I_{\text{width}_\text{size}}$ denote the height and width of the image, respectively, while $patch_{\text{height}_\text{size}}$, $patch_{\text{width}_\text{size}}$ are the height and width of the patch, respectively.

**Algorithm 1:** Priority computation.

**Input:** target image, mask

**Output:** patch (which has the highest value of priority in the boundary of the target region)

1. compute gradient image $G$;
2. compute decomposition layer number; // (using (7))
3. obtain multi-resolution image of target image, $G$, mask;
4. compute data term ($D(\psi_p^i)$) in different resolutions; // (using (4))
5. compute data term ($D(p)$); // (using (5))
6. compute $\alpha_i$; // (using (6))
7. compute $C(p)$; // (using (2))
8. compute $C'(p)$; // (using (8))
9. compute $P(p)$; // (using (9))
10. sort $P(p)$;
11. return $\psi_p'$ (It has the highest priority value);

The priority of patches determines the sequence of filling. Criminisi et al. [6] proposed a priority determination method which is easy susceptible to high-frequency information interference such as noise. Multi-resolution images, especially low-resolution images, have a main outline structure and less interference from high-frequency components, as shown in Fig. 4. Therefore, multi-resolution priority calculation may provide greater stability, we give experiment result to illustrate his advantage in Fig. 5.

The calculation of priority may be inaccurate. When one of $C(p)$ and $D(p)$ is greater than the other, this causes the priority to be dominated by one factor. We can turn $C(p)$ and $D(p)$ into a unified scale as follows:

$$C'(p) = \frac{\max(D(p)) - \min(D(p))}{\max(C(p)) - \min(C(p))}C(p)$$

And the priority calculation is modified as follows,

$$P(p) = C'(p)D(p)$$

We give the calculation of the priority in algorithm 1.

To illustrate the advantage of multi-resolution data term, we use Criminis’s method [6] to demonstrate the advantages of multi-resolution data items, we used the peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and feature similarity index measurement (FSIM) to compare Criminis’s data term and the multi-resolution data term. We obtained PSNR, SSIM, and FSIM values by testing 30 images. as shown in Fig. 5(a)-5(c), and the average values of PSNR, SSIM, and FSIM are shown in Fig. 5(a)-5(c), the PSNR, SSIM and FSIM values of the inpainting results using multi-resolution information to compute priority are greater than those of the single resolution results, it demonstrate that the multi-data term can improve the inpainting effect.

**B. PATCH MATCHING**

Patch matching only uses color values to find the best similar patch. However, sometimes the best similar patch is not a suitable sample patch for filling the target region when the square difference distance (SSD) is used, as demonstrated in detail in [29]. In [25], [28], [30]–[32], the authors redefined the patch matching method and improved the matching of patches to some extent. Inspired by those patch matching approaches, we used color terms, gradient terms and boundary blend terms for patch matching to improve the inpainting effect.

Exemplar-based techniques [6] involve a search for the best patch, but the identified patch may not be the only suitable one. The algorithm may randomly pick out a patch when searching best candidate patches, which leads to filling errors, and the error will affect other patch fillings. In [7], [33], Xu et al. used a combination of multiple candidate patches to create an optimal patch for filling the missing region. This method avoided the random selection of a single patch. Synthetic patches can be generated using sparse representation. Similar combination strategies have also been used in low-rank consolidation [34], [35], achieving low-rank optimization using multiple candidate patches. The above combination strategies demonstrate that the use of multiple candidate patches avoids the drawback of a single, randomly picked out exemplar.

Once the patches to be filled are determined, patch matching can be used to search for similar patches. In [6], Criminisi et al. only used color to search for similar patches. Lee et al. [37] further constrained similar patches by adding gradient terms. In addition, Chung et al. [38] added the boundary matching algorithm. Ghorai et al. [11] provided histogram-based distance between the selected patch and the mean patch of a group. Inspired by those improved schemes, we calculated the patch distance using color terms, gradient terms and boundary terms.

The patch distance is computed as follows:

$$d(\psi_p, \psi_q) = \beta(\alpha d_c(\psi_p, \psi_q) + (1 - \alpha)d_g(\psi_p, \psi_q)) + (1 - \beta)d_e(\psi_p, \psi_q)$$

$$d_c(\psi_p, \psi_q) = \frac{1}{N} \sum_{c=1}^{3} \sum_{i=1}^{N} (\psi_p^c(i) - \psi_q^c(i))^2$$

$$d_g(\psi_p, \psi_q) = \frac{1}{N} \sum_{c=1}^{3} \sum_{i=1}^{N} (\nabla \psi_p^c(i) - \nabla \psi_q^c(i))^2$$

$$d_e(\psi_p, \psi_q) = \frac{1}{N} \sum_{c=1}^{3} \sum_{i=1}^{N} (B_p^c(i) - B_q^c(i))^2$$
where \(d_c(\psi_p, \psi_q), d_g(\psi_p, \psi_q)\) and \(d_e(\psi_p, \psi_q)\) are the color term, gradient term and boundary term, respectively, and \(\alpha, \beta\) are the coefficients. Parameter \(c\) denotes the color channel. \(\psi^c_i\) is the pixel \(i\) color value of the \(c\)-th channel in patch \(\psi_p\). \(\nabla \psi^c\) denotes the gradient of patch \(\psi_p\) in the \(c\)-th channel of the image. \(N\) is the number of known pixels in patch \(\psi_p\). \(\mathbf{E}_p\) denotes the boundary of patch \(\psi_p\) in the \(c\)-th channel of the image. \(E\) is the number of edge pixels.

The gradient value is computed by

\[
\nabla F = \frac{\partial F}{\partial x} + |\frac{\partial F}{\partial y}| \quad (12)
\]

where \(\nabla \) is an absolute value operation, \(\frac{\partial F}{\partial x}\) and \(\frac{\partial F}{\partial y}\) represent gradient values in the \(x\) and \(y\) directions.

To obtain the best valued of \(\alpha\), we tested the accumulative values of the PSNR, SSIM, and FSIM using 30 images. We considered \(d_c(\psi_p, \psi_q)\) and \(d_g(\psi_p, \psi_q)\) for testing the \(\alpha\) value, where the range value of \(\alpha\) is \([0, 1]\).

Fig. 6 shows the accumulative curve of the PSNR, SSIM, and FSIM by testing 30 images. When \(\alpha = 0.95\), the value of the PSNR, SSIM and FSIM reaches the maximum value, so we set \(\alpha = 0.95\) as the default.

We then take \(\alpha d_c(\psi_p, \psi_q) + (1-\alpha) d_g(\psi_p, \psi_q)\) as a whole to test the \(\beta\) value. The calculation of similar patch is conducted as follows:

\[
d(\psi_p, \psi_q) = \beta \alpha d_c(\psi_p, \psi_q) + (1-\alpha) d_g(\psi_p, \psi_q) + (1-\beta) d_e(\psi_p, \psi_q)
\]

We tested 30 images to find the best \(\beta\) value. Fig. 7 shows the cumulative value of PSNR, SSIM, and FSIM. From the graph, it is more appropriate to set \(\beta = 0.1\). Although \(\beta = 0.15\), cumulative value of PSNR achieves its maximum value. SSIM and FSIM can also better satisfy visual consistency when \(\beta = 0.1\). So we set \(\beta = 0.1\) as default.

C. SEARCHING MULTI-RESOLUTION RESOURCES

The known region of an image is regarded as the sample resources [6]; however, these resources are still far from enough. Utilizing multi-resolution resources is the easiest solution, which is different super-resolution-based inpainting [26]. The search for similar patches is shown in Fig.8. We searched for multi-resolution samples to find multiple candidates for similar patches.

In many cases, similar regions of the image are in the vicinity of the filled area, so it is possible to search near the filled area, reducing the range of matching and reducing the overall repair time. We extend 80 pixels outward along the boundary of the filled area to get the sample area. At other resolutions, depending on the sampling mechanism, the range of the search will be reduced accordingly.

D. PICK THE BEST PATCH

We obtained more than one candidate patch by using the multi-resolution resources. Thus, we must determine how to pick out the best candidate patch to be used to fill the target region. The pleasing result of inpainting is judged by human vision. High peak signal-to-noise ratios (PSNRs) cannot meet the standards of human vision, as has been reported in many research works. The structural similarity index measure (SSIM) is more suitable for human visual effects. When searching for candidate patches, we used the known region in the target patch, so the known region can be used to calculate the SSIM value to aid in picking out the best candidate patch. The SSIM is defined in (13).

\[
SSIM(s, t) = \frac{(2\mu_s\mu_t + C_1)(2\sigma_{st} + C_2)}{\mu^2_s + \mu^2_t + C_1(\sigma^2_s + \sigma^2_t + C_2)} \quad (13)
\]

where \(C_1 \) and \(C_2\) are constant values. \(\mu_s\), \(\mu_t\) are the local means of images \(s\) and \(t\), respectively; \(\sigma_s\), \(\sigma_t\) are the standard deviations, respectively; \(\sigma_{st}\) is the cross-covariance of images \(s\) and \(t\). The larger the SSIM value of image \(s\) and \(t\), the better the structural consistency of image \(s\) and \(t\).
In our work, we found the top 5 patches that were most similar to the target patch in every layer image, using only the known part in different multi-resolution images. We then found the patch which obtained the maximum value of the SSIM. Fig. 8 shows the inpainting process. The image in Fig. 8 is from the network https://themillions.com/2010/11/sunday-mornings.html. The main process of our repair work can be described by figure 1.

In order to remove the interference patches, we get average distance using top patches in the full resolution image. Suppose the filling patch is $PT$, we pick out $m$ reference patches which is the most similar patch obtained in the full resolution image, they are denoted $P_1$, $P_2$, $P_3$, ..., $P_m$. So the Euclidean distance can be computed as followed,

$$D = \text{Dis}(PT, PX)$$

we can get distance between $PT$ and top k reference patches and compute average distance $D_{avg}$. If $D > D_{avg}$, the patch($PX$) will be discarded. we use top five patches to compute $D_{avg}$. This means that we remove some patches which are different from the filling patch. This way you can reduce some interference patches.

The target patch can be presented by $target\_patch$, $ref\_num$ represents the number of reference patches used to calculate $D_{avg}$, $patch\_mask$ represents the mask image of the $target\_patch$, 0 is the filling regions and 1 is the known pixel area in $patch\_mask$. $\psi_{i,k}$ indicates that the $k$-th similar patch on the $i$ layer image. $L$ represents the number of layers of image decomposition, the process of selecting the best patch is represented by algorithm 2.

IV. GRAPH CUT FILLING

In the process of the exemplar-based technique, the unknown region is filled by candidate patches. However, the blockiness effect may be generated in the target region. The graph cut technique can obtain seam boundaries in the overlap region. If we fill the target region using candidate patches and consider the known region as the overlap region, the graph cut technique can be used to find the best seam boundaries, and the seam boundaries can eliminate the blockiness effect. The seam boundary is shown in Fig. 9. Our work employs the graph cut technique proposed in [39]. Suppose there are overlaps of three pixels, as shown in Fig. 9. The overlapping pixels are treated as nodes, where the edge is the connection between nodes. In the overlap region, let two adjacent pixel are $s$ and $t$. $A$ and $B$ are old and new patches, respectively. $A(s)$ and $B(s)$ are the pixel colors at the position $s$. The matching quality cost $C$ between $s$ and $t$ that copy from patches $A$ and $B$ respectively is defined in (14).

$$C(s, t, A, B) = \|A(s) - B(s)\| + \|A(t) - B(t)\|$$  (14)

where $\|\cdot\|$ refers to the appropriate norm. We can use the graph theory called min-cut or max-flow in order to easily understand and code the process. The detail of the graph cut can be found in [39].
Huaming Liu et al.: Preparation of Papers for IEEE Access

VOLUME 4, 2019

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

FIGURE 8. Flow chart of the exemplar-based image inpainting with multi-resolution information and graph cut technique. The unknown parts are labeled in red. The priority is calculated using multi-resolution images, and the search for similar patches is also carried out using the multi-resolution images. More than one candidate similar patches is employed to generate the best similar patch by SSIM optimization, which identifies the best patch to fill the missing part.

V. EXPERIMENT

A. THE CHOICE OF PARAMETER

The calculation of priority by (9) does not require the parameters to be set manually. The parameter $\alpha_i$ is obtained automatically. The number of multiscale levels is computed by (7), although the parameter patch size must be set by the user. In our work, we needed to set the patch size, and different patch sizes affect the inpainting results. We tried to obtain the block size automatically, but the result was not satisfactory. When picking the best patch, we selected the first 5 similar sample blocks in every layer, because too many candidate blocks can affect the speed of the repair and may affect the repair results.

B. INPAINTING RESULT

In order to highlight our proposed algorithm, our method compares it with other repair methods. The peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM), and feature similarity index measurement (FSIM) [40] are used to evaluate the quality of repair. At the same time, in order to illustrate the edge-preserving ability of the proposed algorithm, Edge Preservation Ratio(EPR) [53] is used to calculate the edge-preserving index. We get edge using Canny algorithm. We compute EPRs defined in [53]. The larger the EPRa, the better the edge preservation, $EPRa \in [0,1]$. However, the object removing inpainting cannot use these indexes, we use visual analysis of the human eye. Repair time is also an important index. The shorter the repair time, the better. Our experiments were conducted on an Intel core i7CPU@2.0GHz using Matlab implementation (Matlab 9.2.0 with 64 bit/C++). Table1 shows the results of different algorithms.

To test our inpainting method, we selected some images to test our algorithm and compare with others method. Fig. 10 and Fig. 11 are structure images; they do not include texture information. Fig. 10 shows the line structure image, and Fig. 11 shows the arc structure image. The red region in the figures denote the target region. Subgraphs (c), (d), (e), (f), (g), and (h) in Fig. 10 are the inpainting results by Criminisi’s method [6], Barnes’s method [7], Huang’s method [45], Newson’s method [40], Deng’s method [23], and our method, respectively. In Fig. 10, the results of (c), (d), (e), and (g) are not satisfactory, and the results of (f) and (h) are pleasing. The patch size was set to $23 \times 23$ pixel.
Algorithm 2: Selecting best patch

Input: target_image, patch_mask (1 denotes known pixel), ψ^i_q,k (i ∈ [1 ··· L], k ∈ [1 ··· 5])

Output: best_patch

1 ref_num = 5;
2 patch_num = 5;
3 for i = 1 to ref_num do
4     \[ D_i = \text{Dis}(\text{target_image} \odot \text{patch_mask}, \psi^i_q,k \odot \text{patch_mask}); \]
5     end
6 \[ D_{avg} = \frac{1}{\text{ref_num}} \sum_{k=1}^{\text{ref_num}} D_k; \]
7 ssim_max = -1;
8 foreach i = 1 to L do
9     foreach k = 1 to patch_num do
10        D = \text{Dis}(\text{target_patch} \odot \text{patch_mask}, \psi^i_q,k \odot \text{patch_mask});
11        if D < D_{avg} then
12            continue;
13        end
14        ssim_v = \text{ssim}(\text{target_patch} \odot \text{patch_mask}, \psi^i_q,k \odot \text{patch_mask});
15        if ssim_v > ssim_max then
16            ssim_max = ssim_v;
17            best_patch = \psi^i_q,k;
18        end
19     end
20 return best_patch;

size using our method, because the patch size can obtain the image base structure. Table 1 shows the comparison of our results with those of different methods. The values of the PSNR, SSIM, and FSIM using Newson’s method and our method are higher than those in other’s method. The EPRa value of our method is higher than the EPRa value of Newson’s method. This shows that our method can better maintain the boundary.

Subgraphs (c), (d), (e), (f), (g), and (h) in Fig. 11 are the inpainting results using Criminisi’s method [6], Barnes’s method [7], Anupam’s method [10], Newson’s method [40], Deng’s method [23], and our method, respectively. Our’s method and Deng’s method [23] can obtain pleasing result, there are some flaws in the results of other methods. Although Newson’s method [40] obtains better value of PSNR, SSIM, and FSIM than that of our method, but the result of our repair method is better visually. The EPRa value of our method is higher than the EPRa value of Newson’s method. This shows that our method can better maintain the boundary. The patch size is set 15 × 15 pixel size.

The images in Fig. 12, Fig. 14 are from https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/. The images in Fig. 16, Fig. 17 are from http://yokoya.naist.jp/research2/inpainting/. The red regions are the target region in Fig. 12(b)-Fig. 14(b).

Fig. 12 shows an image with a complex structure. The results of Barnes’s method [7], Ding’s method [46] and Huang’s method [45] are unpleasing, while Criminisi’s method [6], Deng’s method [23] and our method can obtain pleasing results. Table 1 shows the comparison of the PSNR, SSIM, and FSIM values. Our’s method and Criminisi’s method [6] obtained the higher values of the PSNR, SSIM, and FSIM, the patch size is set as 9 × 9 in our method. However, the EPRa value of our method is higher than EPRa value of Criminisi’s method [6]. Our approach keeps the boundaries better.

Fig. 13 shows the another texture inpainting. Fig.13(c)-Fig.13(h) are the inpainted result by Criminisi’s method [6], Barnes’s method [7], Newson’s method [40], Ding’s method [46], Huang’s method [45], and our’s method, respectively. The results inpainted by Barnes’s method and Ding’s method are not satisfactory. The results inpainted by Criminisi’s method, Huang’s method and our’s method are pleasing. The patch size is set as 9 × 9. Table1 shows that Criminisi’s method obtained highest value of PSNR, SSIM, FSIM and EPRas.

Fig.14 shows a complex image containing texture and structure components. Antler repair failed using Criminisi’s method [6] and Barnes’s method [7] and Liu’s method [50], as shown in Fig.14(c), Fig. 14(d) and Fig. 14(g), respectively.
The inpainted result as shown in the Fig. 14(e), Fig. 14(f) and Fig. 14(h) using Darabi’s method [48], Damelin’s method [49] and our’s method are pleasing. Darabi’s method [48] obtained the highest value of PSNR, SSIM and FSIM. The patch size is set as $11 \times 11$ in our’s method. Our’s results are very close to the results of Damelin’s method as shown Fig. 14(f) and Fig. 14(h). The EPRa value of our method is higher than other’s methods.

Image inpainting can also be applied to target removal. Fig. 15 shows the object removal result. The results using Criminisi’s method [6] and Newson’s method [40] were unpleasing, as shown in Fig. 15(c) and Fig. 15(f). There are extra balls in the image. Barnes’s method [7] generated a blurring effect, as shown in Fig. 15(d). The repair results using Anupam’s method [10] and Deng’s method [23] in Fig. 15(e) and Fig. 15(g), also show some defects. Our method obtained a pleasing result, as shown in Fig. 15(h).

Fig. 16 is another object removal inpainting. The inpainted result as shown in Fig. 16(c), Fig. 16(d) and Fig. 16(e) are not very well using Ding’s method [46], Liu’s method [50] and Damelin’s method [49] respectively. The result of Ding’s method is filled with error patches, resulting in incorrect repair results. Liu’s method [50] can’t to fill large regions using low tensor technique. Damelin’s method [49] generate
By calculating priority with multi-resolution information, the interference of texture or noise can be shielded. Fig. 5 shows that priority computed by multi-resolution information improves the inpainting effect. If image has not rich samples, our method can find suitable patches to fill target regions through decomposing image. Our method can’t solve the problem of rotating matching patch yet. By adding graph cut technology, the blocking effect can be reduced and sometimes the original information (around the target regions) can be changed. In addition, we need set patch size in out experiment test, generally speaking, patch size should be larger than texture block size. Fig. 19 shows the results using different patch size. When the patch size is $11 \times 11$ pixel size, we can achieved the highest value of PSNR.

At present, the deep learning repair method is booming. However, it requires a large amount of data to train network. In fact, it uses image data distribution to generate similar images for filling target regions. At present, it is also faced with the disadvantages of vague repair results, large data training and fixed image scale. The traditional repair method, in some cases, has certain advantages, such as speed and texture clarity.

**D. TIME COMPLEXITY ANALYSIS**

Assume that the width and height of the image are $w$ and $h$, the area of damage regions is $s$, the patch size is $m \times n$, and the number of image decomposition layers is $L$. In the inpainting process of our method, the factors affecting time mainly include the area of the damaged regions, the size of the patch $m \times n$, the time to calculate the priority, the time to find the sample patch, and the fill time of the graph cut technique. For the convenience of analysis, it is assumed that the filling area of each is half of the patch, and the number of times to fill is: $a = \frac{1}{4} \cdot \frac{2}{m \times n}$. Each fill needs to calculate the priority, search the patch and use the graph cut fill. The time complexity of priority calculating, patch searching, and graph cut filling is analyzed below.

The calculation of priority is divided into two parts, (1) calculating the confidence $C(\psi_p)$; (2) calculating the data term $D(\psi_p)$. It is necessary to traverse the image to calculate confidence, so the time complexity for calculating confidence is $O(hw)$. The calculated data item needs to be decomposed into $L$ different resolution images. The first layer is the original image, and the resolution image size of the $L-th$ layer image is: $\frac{h}{2^{L-1}} \cdot \frac{w}{2^{L-1}}$, calculating the data term requires calculating the gradient value, and considering the multi-resolution image, the time complexity is $O(hw + \ldots + \frac{h}{2^{L-1}} \cdot \frac{w}{2^{L-1}}) \approx O(hw)$. The time complexity of the calculated priority is the sum of the time complexity of calculating the confidence and the time complexity of the calculated data item. $O(2hw) \approx O(hw)$.

When searching for sample patches, you need to traverse the area around the damaged area. The time complexity of the match is less than $O(hw)$, at each match, it takes a certain amount of time to calculate the color term difference, the gradient term difference and the boundary term difference. In the process of each matching, we consider the color image...
the graph cut calculation is a function of the complexity of the number of pixels in the overlapping area. The complexity of searching for a sample patch is a function of the complexity of the boundary difference calculation. When the best sample patch is found, the image is filled using the algorithm, let $a = \frac{1}{2}mn$, $b = \frac{1}{5}mn$, $c = \frac{1}{6}mn$. Because of the search under multi-resolution images, the image size and the size of the sample patch.

When the best sample patch is found, the image is filled with graph cut technique. We use the EK (Edmonds-karp) algorithm, let $d = \frac{1}{2}mn$ be the number of pixels in the area where the sample patch is filled, and $e$ is the number of edges. According to the knowledge of graph theory, the number of pixels in the overlapping area is $\frac{mn}{2}$. Adding a source point and an end point, then the time complexity of the graph cut calculation is $O(\text{de}^2) = O((\frac{mn}{2} + 2)(\frac{mn}{2} + 1)^2) \approx O(m^3n^3)$.

According to the number of filling steps $a$, the time complexity is $O(a(nwmn + nhmn^2)) = O((nwmn + nhmn^2))$, through the time complexity analysis, it can be seen that the program running time is related to the area of the filled regions, the image size and the size of the sample patch.

From the table1, Criminisi’s method [6], Deng’s method [23], Anupanm’s method [10] are all searches on a single image, which takes less time. Anupanm’s method [10], in particular, searches for information around the damage, reducing the search time. In this paper, the same strategy is used to search for similar patches in multi-resolution images, which reduces search time. Both Barnes’s method [7] and Newson’s method [40] adopt multi-resolution restoration, starting from low resolution, layer by layer, until the full resolution layer is completed. In Newson’s method [40], color and gradient constraints are used to search for similar blocks, which takes longer time than Barnes’s method [7]. Huang’s method [45] takes a lot of time to calculate structural characteristics, and it takes a long time. Darabi’s method [48] uses multi-scale and rotation methods to find similar blocks, and conduct poisson fusion processing when considering the consistency of filling, the filling process also takes longer time. Ding’s method [46], which does not use c++ acceleration, takes longer. Liu’s method [50] adopts low tensor completion technique, which requires multiple optimizations and takes a long time. Damelin’s method [49] uses biharmonic functions to inpaint image, which won’t take too long time.

VI. CONCLUSION

In this paper, we proposed an exemplar-based image inpainting with multi-resolution resources and the graph cut technique. To improve the priority computation, we used multi-resolution resources. To further utilize the sample resources, our method searched for similar patches at multiple sources.
resolutions. Our method makes up for the lack of a single reference. When searching sample patches, we added gradients and boundaries to further constrain the candidate patches. Considering the blockiness of the boundary filling, we used the graph cut technique to copy and paste patches. Our method can also fill texture images. However, our method may require the user to choose the patch size, so we will continue to improve the sample patch size problem in future research.

ACKNOWLEDGMENT

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 Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX18_0901, in part by the National Natural Science Foundation of China under Grant 61672430, in part by the Natural Fund Project in Fuyang Normal University under Grant 2017FSKJ17, in part by the Key Research and Development Program of Jiangsu Province under Grant BE2016775.

REFERENCES


HUAMING LIU received the B.S degree in computer science and technology from Shangqiu Normal University, Shangqiu city, China, in 2005, and received the M.S. degrees in computer software and theory from Northwest Minzu University, Lanzhou City, China. He is currently a Ph.D. candidate from 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 City, China. His research interests include image processing, pattern recognition, Software Engineering and so on.

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

HUAMING LIU received the B.S degree in computer science and technology from Shangqiu Normal University, Shangqiu city, China, in 2005, and received the M.S. degrees in computer software and theory from Northwest Minzu University, Lanzhou City, China. He is currently a Ph.D. candidate from 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 City, China. His research interests include image processing, pattern recognition, Software Engineering and so on.

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, USA. She is currently a professor with the college of mathematics and Computer Science, Northwest Minzu University, Lanzhou City, China. Her research interests include image processing, pattern recognition, Tibetan information processing and so on.
FIGURE 12. Complex structure image inpainting. (a) Original image; (b) mask; (c) Criminisi’s method [6]; (d) Barnes’s method [7]; (e) Ding’s method [46]; (f) Huang’s method [45]; (g) Deng’s method [23]; (h) our method.

FIGURE 13. Complex structure and texture image inpainting. (a) Original image; (b) mask; (c) Criminisi’s method [6]; (d) Barnes’s method [7]; (e) Newson’s method [40]; (f) Ding’s method [46]; (g) Huang’s method [45]; (h) our method.
FIGURE 14. Complex structure and texture image inpainting. (a) Original image; (b) mask; (c) Criminisi’s method [6]; (d) Barnes’s method [7]; (e) Darabi’s method [48]; (f) Damelin’s method [49]; (g) Liu’s method [50]; (h) our method.

FIGURE 15. Object removing inpainting. (a) Original image; (b) mask; (c) Criminisi’s method [6]; (d) Barnes’s method [7]; (e) Anupam’s method [10]; (f) Newson’s method [40]; (g) Deng’s method [23]; (h) our method.

FIGURE 16. Object removal inpainting. (a) Original image; (b) mask; (c) Ding’s method [46]; (d) Liu’s method [50]; (e) Damelin’s method [49]; (f) Huang’s method [45]; (g) Darabi’s method [48]; (h) our method.
FIGURE 17. Object removal inpainting. (a) Original image; (b) mask; (c) Criminisi’s method [6]; (d) Darabi’s method [48]; (e) Xu’s method [47]; (f) Komodakis’s method [51]; (g) Ghorai’s method [52]; (h) our method.

FIGURE 18. Object removal inpainting. (a) Original image; (b) mask; (c) Criminisi’s method [6]; (d) Barnes’s method [7]; (e) Newson’s method [40]; (f) our method; (g) Yu’s method [43]; (h) Liu’s method [44].
FIGURE 19. The patch size influence the inpainting result, image size is 100×100 pixel. (a) Original image; (b) mask; (c) 5×5 (PSNR=32.81); (d) 7×7 (PSNR=32.98); (e) 9×9 (PSNR=34.87); (f) 11×11 (PSNR=35.12); (g) 13×13 (PSNR=33.23); (h) 15×15 (PSNR=29.82).