

Patch size adaptive image inpainting

Huaming Liu¹, Guanming Lu², Xuehui Bi^{1*} and Weilan Wang³

¹ School of Computer and Information Engineering, Fuyang Normal University,
Fuyang 236037, P.R. China

[e-mail : 2017010217@njupt.edu.cn(H.L.), bixuehui888@163.com(X.B.)]

² College of Telecommunications and Information Engineering, Nanjing University of Posts and
Telecommunications, Nanjing 210003, P.R. China

[e-mail : lugm@njupt.edu.cn(G.L.)]

³ Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education, Northwest
Minzu University, Lanzhou 730030, P.R. China

[e-mail : wangweilan@xbmu.edu.cn(W.W.)]

*Corresponding author: Xuehui Bi

*Received January 11, 2021; revised June 6, 2021; accepted September 22, 2021;
published October 31, 2021*

Abstract

Texture synthesis technology has the advantages of repairing texture and structure at the same time. However, during the filling process, the size of the patch is fixed, and the content of the filling is not fully considered. In order to be able to adaptively change the patch size, we used the exemplar-based inpainting technique as the test algorithm, considering the image structure and texture, calculated the image structure patch size and texture patch size, and comprehensively determined the image patch size. This can adaptively change the patch size according to the filling content. In addition, we use multi-layer images to calculate the priority, so that the order of image repair was more stable. The proposed repair algorithm is compared with other image repair algorithms. The experimental results showed that the proposed adaptive image repair algorithm can better repair the texture and structure of the image, which proved the effectiveness of the proposed algorithm.

Keywords: Texture synthesis, exemplar-based techniques, patch adaptive size, object removal.

This research was supported in part by the key projects of natural science research in Anhui colleges and universities under Grant no: KJ2020ZD46, in part by Fuyang Normal University Scientific Research Project under Grant no: 2020KYQD0032, in part by the National Natural Science Foundation of China under Grant no: 61772430, in part by the key fund projects of Young Talents in Fuyang Normal University under Grant no: rcxm202001.

1. Introduction

Image inpainting is to fill in damaged regions or remove targets in an image. It has a wide range of application in reality, such as the restoration of artworks, the removal of objects, the restoration of old photos, and the restoration of old videos, etc. The purpose of image inpainting is to make the filled image satisfy, and the filled content does not need to be the same as the original image. The goal of target removal inpainting is to maintain the visual consistency of the filled image and surrounding information. At present, the image inpainting results are mainly based on human visual judgment, supplemented by objective indicators for auxiliary analysis. In addition, image repair also needs to meet a certain efficiency in order to push the repair algorithm into practical applications; therefore, there are certain requirements for the time spent on image repair.

Image inpainting was proposed by Bertalmio et al. [1] in 2000, and it has attracted extensive attention from many scholars. Later, Bertalmio proposed the BSCB [2] (M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester.) method, which mainly used the same color line diffusion. This method is effective for filling small areas; however, it is easy to produce a blur effect when repairing large areas. Chan et al. [3] proposed a curvature diffusion method to repair simple structure images, which can propagate the structure. However, large-scale repairs still fail to obtain satisfactory repair results. Efros et al. [4] used texture synthesis technique to achieve large texture image, which used the fusion of texture block editing to obtain a natural synthesized texture image.

For texture synthesis, texture patches need to be fused; V. Kwatra et al. [5] used Graph Cut technique to solve and to ensure the natural fusion of the texture boundaries of the image. Bertalmio et al. [6] proposed the decomposition of the image into two components, texture and structure, and then the structural component was repaired using the BSCB [2] method. The texture component adopts the texture synthesis algorithm proposed by A. Efros et al. [4], making full use of the advantages of texture and structural repair method. A. Criminisi et al. [7] proposed an exemplar-based image repair method, which uses texture synthesis to fill preferentially the structural parts of the image. The isophote is used to calculate the priority of the image, and the structural regions are repaired firstly; the search technique is used to obtain similar patches for filling images.

The core of this method is: (1) the priority determines the order of image repair, and priority is given to repairing the structural regions in the image. (2) it has the advantages of texture synthesis. Disadvantages: (1) error filling will lead to more errors. (2) the process of patch filling does not consider fusion between patches, then block effects will be produced easily. In order to speed up the repair of images and introduce the idea of random search, C. Barnes et al. [8] proposed PatchMatch search algorithm, which uses a pyramid decomposition to get the multi-images and starts from the top-layer image. In the process of searching for patches, PatchMatch search quickly for similar patches around the filled area, using less time to find similar patches. Considered image inpainting as an optimization problem, a variational framework is used to minimize the energy constraint function [9]. When combined with the idea of texture synthesis, the best inpainting can be solved.

Considered the search speed of images, PatchMatch is also used to search for patches to reduce the time spent on image repair. Although PatchMatch can speed up the search, in order to avoid falling into local optimality, a random search is added to the algorithm to avoid local optimality. However, PatchMatch uses a fixed patch size. The adaptive patch size is not easy

to implement; therefore, the process of inpainting in the multi-resolution images to avoid the side effects caused by the patch size mismatch.

D. Thanh et al. [10] proposed an adaptive image inpainting using adaptive parameters estimation, parameters can be estimated based on the information of the image, such as discrete gradient. R. Zdunek et al. [11] use hybrid interpolation in tensor representation for inpainting image, it only It can only repair a small area. In order to get similar patch for filling image, S. Yang et al. [12] proposed multi-patch match with adaptive size, this method improves the accuracy of image matching and saves time. R. Zhao et al. [13] proposed a repair method which is based on deep learning, the encoder and generator in the network respectively complete the encoding and decoding tasks based on the residual network.

The content of the image generally contains texture and structure, and even smooth areas sometimes have subtle changes in lighting. In the process of image restoration, the use of patches of different sizes is more conducive to image inpainting when using patch filling. For example, for a smooth area in an image, it is more reasonable to use a larger patch, on the one hand, it can speed up the repair of the image, and on the other hand, it can reduce the filling of the errors. Therefore, it is necessary to select adaptively the size of the patch according to the image content. The image content can be divided into smooth, texture, and structure according to a simple classification.

Large patches can fill in smooth area. As humans are more sensitive to structure, small patches can fill in structural areas; For the texture area, a sample block of intermediate size is used. **Fig. 1** is the image repair result using the Criminisi's algorithm and the patch size of different sizes. **Fig. 1** is the repaired results which contains the original image, marks area to be filled (red area), repaired results are different with patch of size of 5×5 , 7×7 , 9×9 , and 11×11 . Among the repair results with patch size 7×7 , the content of grass is visible in the middle of the road, indicating that there was a wrong filling, and other repair results also include tiny differences.

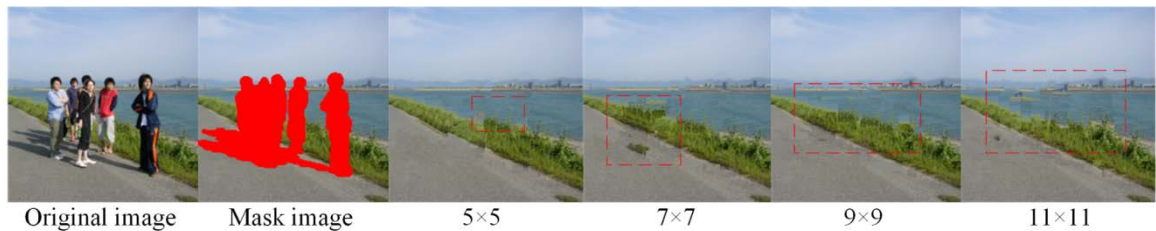


Fig. 1. Repair results using different patch sizes by Criminisi's algorithm [7].

In order to perform repairing by adaptive patch size, the structure and texture patch of the image are predicted, and an integrated patch size is obtained. The data item based on tensor [14] is also an improvement on priority, which is calculated based on the structural tensor ($J = \sum_{c=1}^m \nabla I \nabla I^T$, $m=3$ means color image). According to the eigenvalues, the degree of anisotropy of the local region can be estimated. Therefore, it is feasible to use the structure tensor to analyze the content of the local image and then calculate the patch size. The texture of the image is more complicated, and the texture is divided into random texture and structural texture. The content of the random texture is poor in regularity, which makes it difficult to select the patch size. In order to select an appropriate patch size for the image texture, the size of the texture patch is obtained by statistical analysis of the texture image databases. Through the size of the texture patch and the size of the structure patch, the size of the patch is estimated by weighting calculation. On the one hand, the structure content is taken care of, and on the

other hand, the texture content is also considered.

We introduce related work in section 2. In section 3, we introduce the selection of adaptive patch size. In section 4, we introduce the image repair techniques based on patch. section 5 presents the experimental results and analysis, and section 6 is the conclusions.

2. Related Work

H. Zhou et al. [15] proposed an inpainting method that changes adaptively the patch size, the determination of the patch size is formulated as an optimization problem, mainly to minimize the objective function, because the objective function involves image gradients and saliency maps. The selection principle of the patch size is: for the area with the structure, the size of the patch is small; the area without the structure should be set to a larger patch size. Since the gradient angle histogram is an important tool for measuring the structural variance, in order to distinguish the structure in different directions, a square-size patch is selected on the gradient image to find the gradient angle histogram.

The traditional gradient calculation method are very sensitive to small color changes, and the histogram may be misled by trivial and repeated structures such as grass or trees. In order to improve the gradient of the gradient angle histogram, the saliency map [16] that emphasizes clear and uniform features is used to calculate the saliency value. That is, the saliency value is multiplied by the corresponding gradient vector to obtain an improved gradient field. The greater the difference between the two histograms, the smaller the similarity of the corresponding two patches. Therefore, the patch with the smallest structural change is defined as the patch with the largest gradient angle histogram difference from the adjacent patch of larger size. For the patch $SQ(p, k)$, the gradient angle histogram difference from $SQ(p, k+2)$ is defined as:

$$HD(p, k) = \sum_{\theta=1}^{360} |H(p, k+2, \theta) - H(p, k, \theta)| \quad (1)$$

where $H(p, k, \theta)$ represents the number of pixels with an angle of θ in a gradient image, and p is the center of patch and $k \times k$ is the size of a patch. In the case of differences in the histogram, the optimal patch may be disturbed by some noise. The patch size should also be locally consistent, so that the optimal patch can be selected without being affected by noise. Therefore, the optimal patch size is the solution to the following minimization problem:

$$\min \sum_{p \in S} \left(\frac{1}{HD(p, k(p))} + \lambda \left(k(p) - \frac{\sum_{q \in N(p) \cap S} k(q)}{\|N(p) \cap S\|} \right) \right) \quad (2)$$

where $k(p)$ represents the size of the patch centered on p points, $N(p)$ represents the 8 neighborhoods of p points, and $\|\cdot\|$ represents the number of pixels in the area. By solving the minimization formula, the optimal patch size is obtained.

R. Borole et al. [17] proposed adaptive patch method, searching for similar patches in the neighborhood of damaged regions. The size of similar blocks is traversed from 3×3 to 15×15 , and the patch with the smallest distance is selected to determine the patch size. This method does not need to consider the structure and texture of the image content. It is simple and easy to perform, but it can easily cause time-consuming problems. X. Zhou et al. [18] first fixed the size of the initial patch as 9×9 , and then used the idea of sparsity $S(p)$ to calculate the size of the patch. The sparsity is defined as:

$$S(p) = \sqrt{\left[\sum_{q \in N_s(p)} \omega_{p,q}^2 \right] \cdot C(p)} \quad (3)$$

where $C(p)$ is the confidence level, $N_s(p)$ represents the known pixel in the patch Ψ_p to be filled, and the calculation formula of $\omega_{p,q}$ is:

$$\omega_{p,q} = \frac{1}{Z(p)} \exp\left(-\frac{d(\Psi_p, \Psi_q)}{25}\right) \quad (4)$$

where $Z(p)$ is the normalization constant and $d(\Psi_p, \Psi_q)$ is the color distance between the two patches Ψ_p and Ψ_q . By calculating the size of $S(p)$, the size of the patch can be determined. A larger value of $S(p)$ means that the patch locates in a smooth area, and otherwise it means that the patch locates in structure area. When the patch locates in the structure area, a smaller patch size will be selected for inpainting. When filling in the smooth area, the larger patch size should be selected, which can shorten the processing time and at the same time reduce the errors generated during filling.

J. Cao et al. [19] used structure perception to calculate the structure and texture attributes of the local area of the image, introduced the average correlation factor of the pixel block, and determined the content of the image by the size of the feature value. The calculation is as follows:

$$f_{avg-cor} = \left(\frac{\bar{\lambda}_1 - \bar{\lambda}_2}{\bar{\lambda}_1 + \bar{\lambda}_2} \right)^2 \quad (5)$$

where $\bar{\lambda}_1$ and $\bar{\lambda}_2$ are the average values of the eigenvalues of the structural tensor of the pixels. When the value $f_{avg-cor}$ is larger, it indicates that the image patch is located in the boundary or texture area, and we should select a smaller patch; when the value is smaller, it indicates that the image block is in a smooth area, and it is more reasonable to use a larger patch to repair. Through its range of values, three different patch sizes can be determined.

$$S(p) = \begin{cases} 11 \times 11, & f_{avg-cor} \leq 0.650 \\ 9 \times 9, & 0.65 < f_{avg-cor} < 0.9 \\ 5 \times 5, & f_{avg-cor} \geq 0.9 \end{cases} \quad (6)$$

K. He *et al.* [20] used the SSIM (structural similarity index measure) algorithm to introduce gradient information to achieve the description of local edges and textures. They calculated the structural similarity function between two adjacent blocks using the mean value of the gradient in the eight directions of the image. The smaller the value of the function, the smaller the structural similarity between two adjacent blocks. The patch filled is located in the structure area, and a smaller patch should be selected. If the RSS (region structure similarity) value is large, this means that the similarity between two adjacent blocks is large, and the patch filled is in a smooth area, and the larger block should be selected.

C. Meng et al. [21] adopted the change of gradient in the local block of the image to optimize the selection of patch size, which is similar to the algorithm principle proposed by H. Zhou et al. [15]. Q. Fan [22] used a quadtree to solve the problem of patch size selection, and decomposed the image with a quadtree. The image was decomposed according to the grayscale value difference of the image. If the difference of grayscale value was less than the threshold, the decomposition was stopped; otherwise, the decomposition continued. In order

to determine the size of patch, image quadtree decomposition is needed for each repair. C. Chang [23] used image roughness, entropy mean, moment of inertia mean, etc. to determine the size of the patch. Available size of patch are 9×9 , 13×13 , and 17×17 size.

R. Borole et al. [17, 24] choose the size of the patch from 3×3 to 15×15 . From these different scale patches, the patch with the smallest distance is the filling patch. This method is an enumerated iterative operation, which is computationally intensive and time-consuming. C. Chang [25] proposed adaptive selection based on the filling area. As the filling area decreases, the patch becomes smaller, which is not suitable for images with complex content. A similar processing technique is the adaptive algorithm proposed by Y. He et al. [26]. Unfortunately, only three sizes are available (5×5 , 9×9 , and 11×11). From the block sparse repair algorithm proposed by Z. Xu et al. [27], M. Tang [28] proposed patch sparsity method by which the patch size is determined. Similar to Y. He, only three patch sizes are selected (7×7 , 9×9 , and 11×11). In the algorithm for automatically selecting the patch size as proposed above, the texture features of the image need to be considered firstly, and then the size of the filled area is considered. In this paper, we studied the problem of adaptive selection of patch size and used the local image information to determine the patch size.

3. Patch scale adaptive selection

S. Ye et al. [29] used local gradients and other features to divide image into smooth blocks, texture blocks, and structure (edge) blocks. It is instructive to select the appropriate repair algorithms according to different categories and repair them separately. When Z. Xu et al. [27] repaired the image, they found that the structure sparse values of the patches were different, indicating that the image content was different, which may be in the corner area, structure area, texture area, or smooth area. Structure perception can also predict the content of the image [14, 30], mainly using the structural properties of the local region.

Due to less attention to the texture, the calculation will be interfered to some extent when faced with a complex texture structure. As the texture image contains regular textures (such as walls) and random textures (such as grass and rocks), there is sometimes a certain degree of structure in the image. The repeated texture primitives make the image look more consistent. In order to be able to portray the local information of the image more reasonably and avoid the interference of noise, the image is separated a texture component map and a structural component map. Taking into account the sensitivity of human vision to the structure, the local information of the structural component map is considered for calculating the patch size.

We calculate the structural strength of the patch using the structure-aware item, the interval range of the patch size is given, and then the patch size is calculated according to the energy value of the local area. The boundaries in the image can be expressed through structure-aware items [31], reflecting the local contour information. O. Le Meur et al. [14] used structure items to calculate data items. In this paper, structure-aware items were used to calculate the upper bound of the patch size. The definition of structure-aware items is as follows:

$$J_{\rho} = K_{\rho} * \sum_{c=1}^3 \nabla I_c \nabla I_c^T \quad (7)$$

where ∇I_c represents the gradient value under the c -th channel of the color image, K_{ρ} is the Gaussian convolution kernel, ρ is the standard deviation of the convolution kernel, T is the transpose, J_{ρ} is the positive semidefinite matrix, and J_{ρ} the eigenvalues are calculated as follows:

$$\lambda_{1,2} = \frac{1}{2} \left[J_{11} + J_{22} \pm \sqrt{(J_{11} - J_{22})^2 + 4J_{12}^2} \right] \quad (8)$$

The corresponding feature vector is calculated as follows:

$$v_1 = \begin{bmatrix} 2J_{12} \\ J_{22} - J_{11} + \sqrt{(J_{11} - J_{22})^2 + 4J_{12}^2} \end{bmatrix}, v_1 \perp v_2 \quad (9)$$

The size of the eigenvalues can reflect the boundary strength of the local image. If the feature vector point is in the smooth area, the value of λ_1 and λ_2 are small, and $\lambda_1 \approx \lambda_2$. If the feature vector point is in the boundary area, λ_1 is larger and λ_2 is smaller, at this time $\lambda_1 \gg \lambda_2$. If the feature vector point is in the corner area, and λ_1 and λ_2 are relatively large, $\lambda_1 \approx \lambda_2$. After obtaining λ_1 and λ_2 , the structural strength $S(p)$ of the image can be calculated, which is defined as follows:

$$S(p) = \alpha + (1 - \alpha) \exp \left(-\frac{\eta}{(\lambda_1 - \lambda_2)^2} \right) \quad (10)$$

Among them, p is the center of the patch, $\alpha \in [0, 1]$, $S(p) \in [0, 1]$ and the larger the value of $S(p)$, indicates a stronger structure. This article sets $\alpha=0.3$ for calculating $S(p)$.

To reduce the noise interference, smooth the image, and highlight the structure, the overall variation technique was used to extract the texture image structure [32]. Fig. 2 is a schematic diagram of smoothing, in which (a) is the original image and (b) is the result of the smoothing. The smoothed image has a very clear structure, which can make the calculated value of the structure-aware item in the image more accurate.

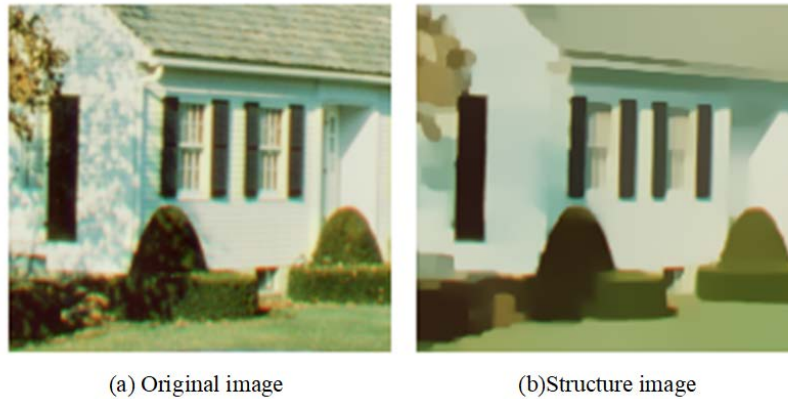


Fig. 2. Image smoothing operation diagram.

In (a), (b), and (c) of Fig. 3, five patches Ψ_1, \dots, Ψ_5 were selected respectively, and the structure-aware items $S(p)$ were calculated for the smooth area, texture area, and structure area in the three images. For a better understanding, $S(p)$, λ_1 , and λ_2 are shown in Table 1. We analyzed the smooth area in the figure. The value of $S(p)$ was 0.3, which was also the minimum value of $S(p)$. As the smooth area had no boundaries, the value of the structure perception item was small, and the value of λ_1 and λ_2 was also small at this time. In the texture area, the structure had a certain intensity; however, the intensity was not very large, so the value of the structure perception item $S(p)$ was larger than the value of the smooth area, $\lambda_1 > \lambda_2$.

In the boundary area, the value of the structure-aware item was large, and the value of λ_1 was much larger than λ_2 , namely $\lambda_1 \gg \lambda_2$. Therefore, it was feasible to calculate the size of the patch through the value of the structure-aware items. If the broken patch was in a smooth area, it is more reasonable to use larger image patch to fill the image; if the broken patch locates in a structural area, we choose a smaller patch to fill. For texture images, if the patch size was larger than the size of the image's texture primitives, images can be effectively repaired. They found that the size of the patch affected the repair quality of the image, and it is important to choose the appropriate patch size.

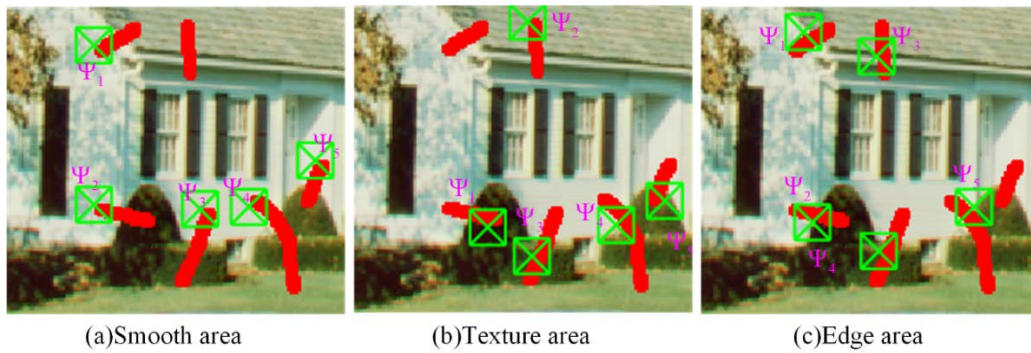


Fig. 3. Structure item value of patch. (a) Smooth area; (b) Texture area; (c) Edge area.

Table 1. Patch structure items and eigenvalues.

λ_1, λ_2	Ψ_1	Ψ_2	Ψ_3	Ψ_4	Ψ_5
S(p)					
Fig. 3	S(p)=0.3	S(p)=0.3	S(p)=0.3	S(p)=0.3	S(p)=0.3
(a)Smooth area	$\lambda_1 = 0.617$	$\lambda_1 = 4.019$	$\lambda_1 = 0.559$	$\lambda_1 = 0.0001$	$\lambda_1 = 0.576$
	$\lambda_2 = 0.005$	$\lambda_2 = 0.462$	$\lambda_2 = 0.104$	$\lambda_2 = 0.00004$	$\lambda_2 = 0.053$
Fig. 3	S(p)=0.3	S(p)=0.3	S(p)=0.354	S(p)=0.403	S(p)=0.616
(b)Texture area	$\lambda_1 = 0.004$	$\lambda_1 = 0.2$	$\lambda_1 = 117.547$	$\lambda_1 = 186.583$	$\lambda_1 = 380.719$
	$\lambda_2 = 0.001$	$\lambda_2 = 0.006$	$\lambda_2 = 0.353$	$\lambda_2 = 30.15$	$\lambda_2 = 4.172$
Fig. 3	S(p)=0.592	S(p)=0.990	S(p)=0.963	S(p)=0.894	S(p)=0.967
(c)Structure area	$\lambda_1 = 347.352$	$\lambda_1 = 21854.993$	$\lambda_1 = 5609.12$	$\lambda_1 = 1912.865$	$\lambda_1 = 6310.747$
	$\lambda_2 = 4.041$	$\lambda_2 = 861.517$	$\lambda_2 = 48.152$	$\lambda_2 = 93.732$	$\lambda_2 = 57.526$

The Brodatz library is an international standard texture image set. In this paper, 112 images were tested using different patch sizes. The patch size ranged from $[5 \times 5]$ to $[33 \times 33]$, and the simulation experiment was performed using A. Criminisi's algorithm[7]. The filling area was 460×10 . We calculated the peak signal to noise ratio (PSNR) for testing results. By repairing 112 images and selecting different patch sizes, the cumulative PSNR value is shown in Fig. 4. The figure shows that the curve has two peaks. The larger peak is suitable for regular texture image repair, so the first peak interval is more reliable. Through analysis, we found that the patch size definition ranges from $[7 \ 7]$ to $[21 \ 21]$. The gray level co-occurrence matrix can reflect the texture characteristics of the image, and find the values of energy; entropy; moment of inertia; the related four aspects in the directions of 0° , 45° , 90° ,

and 135° ; and the mean and standard deviation as the final 24-dimensional texture feature. Choose a block size from $[5\ 5]$ to $[21\ 21]$ for each image, the average of three consecutive patch size PSNR values was counted, and the maximum value was selected to obtain the best patch interval. Then we selected the stable patch size from the best patch interval. If we only refer to the accumulated PSNR value, its maximum value was often not stable.

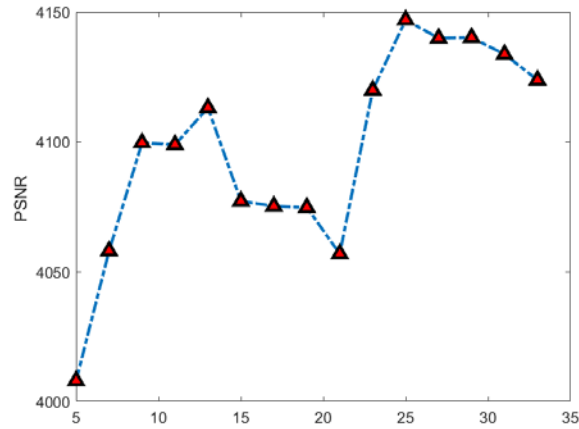


Fig. 4. Cumulative PSNR graph of repair results with different patch sizes.

After determining the patch size of each image, we train network with samples. In each image, we randomly selected $[21 \times 21]$ patches for feature extraction, extracted 100 patches as training samples, and obtained 112×100 data. The number of nodes in the middle layer was 20, and a single node was output. The network structure is shown in **Fig. 5**.

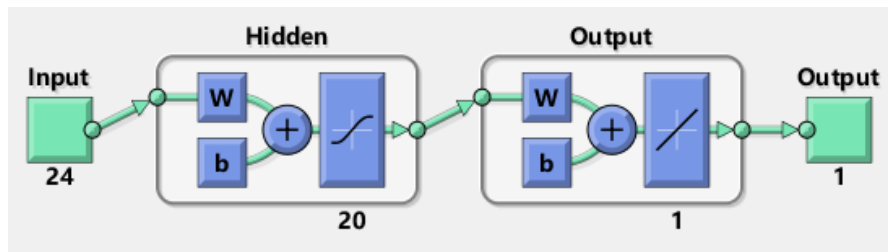


Fig. 5. Neural network structure.

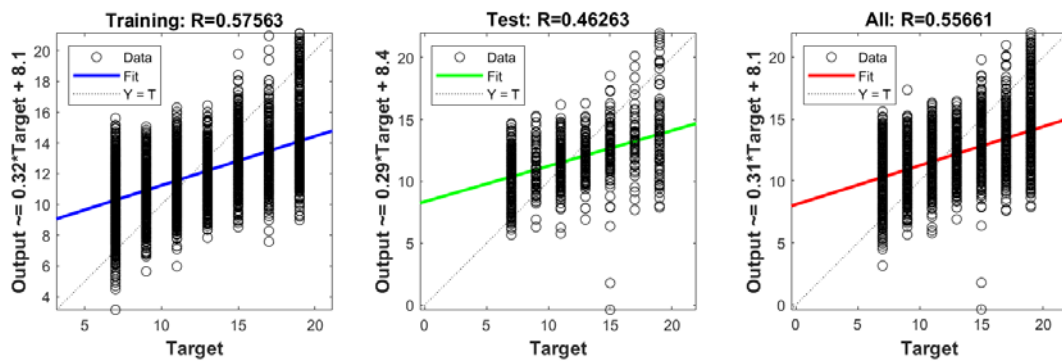


Fig. 6. Neural network structure accuracy.

We used the trained network function to predict the patch size. The accuracy of training is shown in Fig. 6, and the overall prediction accuracy is 55.7%. The size of the patch is more generally based on the combined prediction of the value of the structure-aware item and the value of the texture area. The patch size is defined as follows:

$$PatchSize = \beta Structure_size + (1 - \beta) Texture_size \quad (11)$$

Among them, $Structure_size$ and $Texture_size$ are defined as follows:

$$Structure_size = \begin{cases} 7, PSZ < 7 \\ 17, PSZ > 21 \\ PSZ, else other \end{cases} \quad (12)$$

Among them,

$$PSZ = -127.3716 \times S(p)^4 + 349.1306 \times S(p)^3 - 349.9226 \times S(p)^2 + 135.7657 \times S(p) - 0.6111$$

$$Texture_size = \begin{cases} 7, Neural_Network(\Psi_p) < 7 \\ 21, Neural_Network(\Psi_p) > 21 \\ Neural_Network(\Psi_p), else other \end{cases} \quad (13)$$

Among them, $Neural_Network(\Psi_p)$ is predicted using a trained neural network, and the input value is 24 feature values of the extracted patches. β is the balance factor. Through experimental analysis, $\beta = 0.45$ is the default value. The patch size scale adaptive selection algorithm proposed in this paper combined the information of texture and structure. We integrated these two aspects of information to obtain the size of the patch. Its implementation algorithm is described in Algorithm 1.

Algorithm 1: Adaptive selection of patch size.

Input: Image to be repaired I

Output: **PatchSize**

Image smoothing operation I_s [28];

Calculate the eigenvalues λ_1, λ_2 of each pixel in image I_s ;

Get the pixel position p_i, \dots, p_n on the filling front;

Take p_i of the pixels on the front of the fill and calculate $S(p_i)$;

Calculate **Structure_size**;

Calculate **Texture_size** using patch of size

Structure_size;

Calculate **PatchSize**;

Return **PatchSize** ;

4. Image inpainting technique based on patches

The most representative texture synthesis is the image inpainting method based on patches as proposed by Criminisi et al. [7, 33]. The main idea of this repair method is to first select the filling patch with the highest priority in the filling frontier, the center of the patch is on the filling frontier, the patch size is selected as 9×9 or other sizes, and then the known pixels of patch are used to search for similar patches. Then, the searched similar patches are used to fill the unknown area, after updating the filling front, to find the filling patch with the highest priority on the updated filling front, and the above process is repeated until all the broken regions are filled. Fig. 7 shows a schematic diagram of the repair process based on patches. To facilitate the introduction of image repair ideas, assume that the image is

represented as, the filled area of the image is represented as Ω , and the known area of the image is represented as S , where $I = S \cup \Omega$, the fill front is represented as $\partial\Omega$, There is a point p on the filling front, and the square image patch centered on p is represented as Ψ_p . The flow chart of image inpainting process based on patch is shown in Fig. 7.

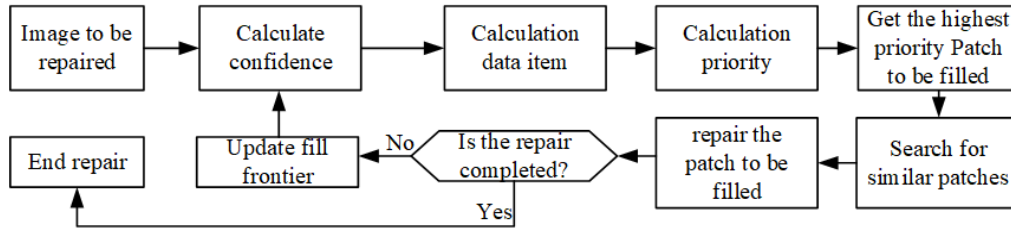


Fig. 7. Image repair flow chart based on patch.

4.1 Sequence of image inpainting

The repair technique based on the patch uses priority to determine the order of image filling. The calculation formula of priority is as follows:

$$P(p) = C(p)D(p) \quad (14)$$

where p represents the position of the pixel, Ψ_p (patch) represents a rectangular block centered on p , and $C(p)$ and $D(p)$ are the confidence and data items, respectively. The meaning expressed by the confidence item $C(p)$ is the proportion of known pixels in the rectangular patch. $D(p)$ denotes the structural information of the local image. The calculation formula for the confidence of image patch Ψ_p is:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap S} C^0(q)}{|\Psi_p|} \quad (15)$$

$$C^0(q) = \begin{cases} 1, \forall q \in S \\ 0, \forall q \in \Omega \end{cases} \quad (16)$$

where $C^0(q)$ is the initial value of confidence, $|\Psi_p|$ is the area of patch Ψ_p , and so $C(p) \in [0, 1]$. The larger the value of $C(p)$ is, the more pixels are known in the patch; that is, the more information is available in the patch, the less information is filled in, and the patch should be repaired first. The calculation formula of $D(p)$ is as follows:

$$D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha} \quad (17)$$

where ∇ represents the gradient operator, \perp represents the vertical operation, n_p is the unit vector perpendicular to the filling front $\partial\Omega$ at the point p , α is the normalization factor, and the grayscale image $\alpha = 255$. Fig. 8 shows the various parameter representations of the data item calculation, ∇I_p^\perp is the size and direction of the isophote, and the calculated value of $P(p)$ comes from the product of confidence and data items. Assuming that the confidence items of the two to-be-filled patches $\Psi_{p'}$ and $\Psi_{p''}$ are equal, then the greater the value of the isometric line ∇I_p^\perp is, the greater the value of the priority $P(p)$, which should be filled first.

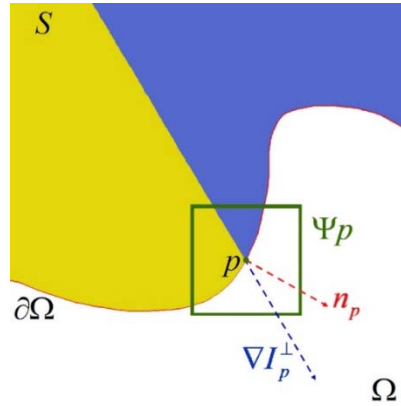


Fig. 8. Schematic diagram of data item parameters.

The intensity of the isophote is calculated by the gradient in the image, and the area with a larger gradient corresponds to the boundary in the image. In addition, human vision is more sensitive to the structure in the image, and the failure of structure filling directly affects the repair effect of the image. Therefore, the priority task of filling the damaged regions where the structure is located is to emphasize the visual connectivity of the structure and improve the repair quality.

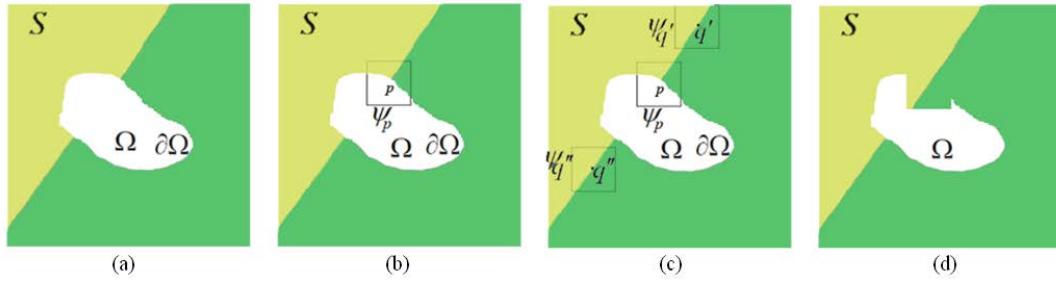


Fig. 9. Schematic diagram of image repair results based on patch.

4.2 Search for similar patch

Assuming that the patch Ψ_p has the highest priority, it should be filled firstly. The strategy of filling is to search for similar patches through the known pixel of patch. The measurement of similar patches is a very critical issue as it is the basis for finding similar patches. The formula for calculating the similarity between Ψ_p and Ψ_q is as follows:

$$d_{ssd}(\Psi_p, \Psi_q) = \sqrt{\sum_{c=1}^3 \sum_{t \in S} (\Psi_p^c(t) - \Psi_q^c(t))^2} \quad (18)$$

Among them, c represents the image channel. If the filled image is a grayscale image, then $c = 1$. If the filled image is a color image, then $c = 3$. The $d_{ssd}(\Psi_p, \Psi_q)$ distance only calculates the distance difference between the two corresponding positions, and the broken region does not participate in the calculation. The most similar patch $\Psi_{\hat{q}}$ is the minimum value of all matching patches, and the calculation formula is as follows:

$$\Psi_{\hat{q}} = \arg \min_{\Psi_q \in S} d_{ssd}(\Psi_p, \Psi_q) \quad (19)$$

The most similar patch ($\Psi_{\hat{q}}$) can be found using formula (19), and $\Psi_{\hat{q}}$ is used to fill the missing region. The filling result is shown in **Fig. 9** (d). If the broken region of patch is filled,

the filling front needs to be updated, and the patch with the highest priority is found on the updated filling front. The calculation of priority requires confidence items and data items.

For the convenience of calculation, on the updated filling front, the confidence level is updated to:

$$C(p') = C(p), \forall p' \in \Psi_p \cap \Omega \quad (20)$$

It can be seen from Equation (15) that the confidence of the filled image will decrease. This is because the confidence of the known pixel points is 1 in the stage of initialization using formula (15), and the confidence of the filling pixel points is 0. When calculating the confidence, the confidence of patch is less than 1. Therefore, in the filled region, the update of confidence is less than 1, which leads to a smaller value of the confidence calculation.

It can be seen from Equation (20) that the confidence of the filled image will decrease. It is known from formula (15) that the confidence degree of the known pixel points is 1 by reason of initialization, and the confidence degree of the pixel points in the broken region is 0. It is known from formula (20) that the confidence value of patch is less than 1. As a result, the confidence value decays during the filling process.

4.3 Priority improvement

In the priority calculation proposed by Criminisi, when calculating the data item ($D(p)$), the gradient information in the image is used. As the gradient is easily interfered by noise and weak texture structure, the $D(p)$ value may be inaccurate, it leads to a big difference of $P(p)$, which causes a change in filling order and repair quality. Multi-resolution decomposition is used for image inpainting [34-39], noise interference can be reduced and the quality of the image inpainting can be improved. Inspired by multi-resolution image repair, the data items are calculated by multi-resolution images to avoid noise interference. In this paper, for the convenience of calculation, the image is decomposed into three layers to obtain three images with different scales. Assuming that p is a point on the boundary, the original image is the first layer image, the position of p point in layer l is p^l , $l=1$ (indicating the first layer image), and the data item $\hat{D}(p)$ is defined as follows:

$$\hat{D}(p) = \frac{\sum_{l=1}^3 D(p^l)}{L} \quad (21)$$

where $L=3$ means to obtain the average value of the data items of each layer. $D(p^l)$ is defined as follows:

$$D(p^l) = \frac{|\nabla I_{p^l}^\perp \cdot n_{p^l}|}{\alpha} \quad (22)$$

Through the improvement of the data items, the average calculation of the three-layer image, which is anti-interference, is carried out. The specific implementation algorithm is described in Algorithm 2.

Algorithm 2: Improvement of data items under multi-resolution features.

Input: The image I to be repaired, fill the front position p_i , fill the mark $mask$

Output: The image I to be repaired, fill the front position p_i , fill the mark $mask$

Obtain the multi-resolution image of the image and get I' , I'' , I''' ;

Calculate the gradient images G' , G'' , G''' of each resolution image;

Calculate the filling front of each resolution image;
 Select the point p_i^1 on the filling front to get the filling positions, p_i^2, p_i^3 at other scales;
 Calculate $n_{p_i^1}, n_{p_i^2}, n_{p_i^3}$ of each resolution image;
 Calculate $D(p_i^1), D(p_i^2), D(p_i^3)$;
 Calculate $\hat{D}(p)$;
 return $\hat{D}(p)$;

5. Experimental results and analysis

5.1 Equations

The result of image inpainting is evaluated subjectively and objectively. Subjective evaluation uses human visual judgment to directly analyze the quality of results. PSNR and SSIM (structural similarity index measure) were used for objective evaluation. PSNR and SSIM are defined as follows:

$$PSNR = 10 \times \log \left(\frac{(2^n - 1)^2}{MSE} \right) \quad (23)$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|A(i, j) - B(i, j)\|^2 \quad (24)$$

where A and B represent two images, m and n are the width and height of the image. The higher the PSNR, the more similar the two images are.

$$SSIM(s, t) = \frac{(2\mu_s\mu_t + C_1)(2\sigma_{st} + C_2)}{(\mu_s^2 + \mu_t^2 + C_1)(\sigma_s^2 + \sigma_t^2 + C_2)} \quad (25)$$

where C_1 and C_2 are constant terms, μ_s and μ_t are the local mean values of image s and t , σ_s and σ_t are the standard deviations of image s and image t , and σ_{st} is the cross covariance of image s and image t . The larger the SSIM value, the more similar the structure of the two images. Similar to PSNR, there is FSIM, whose formula is defined as follows:

$$FSIM(X) = \sum_{x \in \Omega} S_L(X) * PC_m(X) / \sum_{x \in \Omega} PC_m(X) \quad (26)$$

For the definition of $S_L(X)$ and $PC_m(X)$, please see reference[40].

5.2 Parameter setting

The adaptive technique based on the patch scale in this paper involves some parameter settings. When calculating structure-aware items, we set $\alpha=0.3$ and $\eta=300$. When calculated the data items, the image was decomposed into three layers. In order to get a better β value, we repaired 10 images to calculate the average values of PSNR and SSIM, and, to obtain the graph as shown in Fig. 10, we synthesized the two index values and used $\beta=0.75$ as the default when calculating the patch size value. Although only 10 images are used for testing, each image contains multiple repaired areas, which is equivalent to testing 50 images. These images are sufficient for obtaining the value of β .

5.3 Experimental results

In order to evaluate the proposed algorithm in this paper more comprehensively, we filter image types and fill areas. We selected the structure image and texture image to test the algorithm, and verified the performance of the algorithm in repairing the structure and texture image.

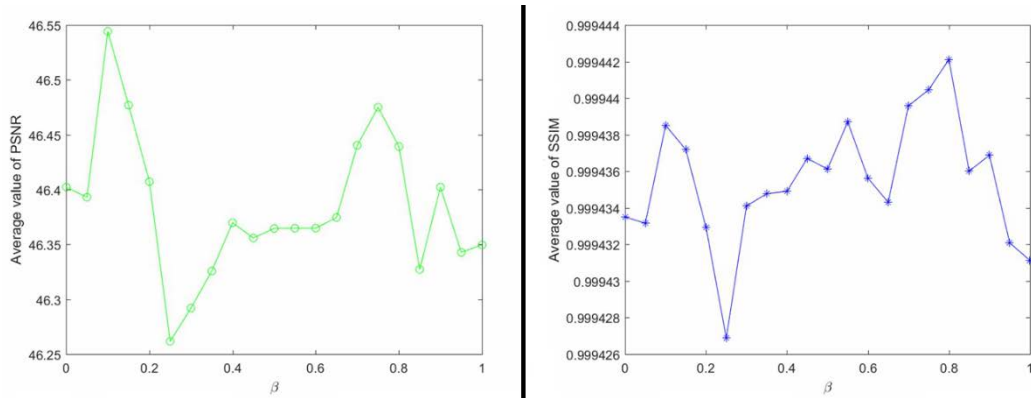


Fig. 10. PSNR and SSIM average curves under different β values. (a) PSNR average at different β values (b) SSIM average at different β values.

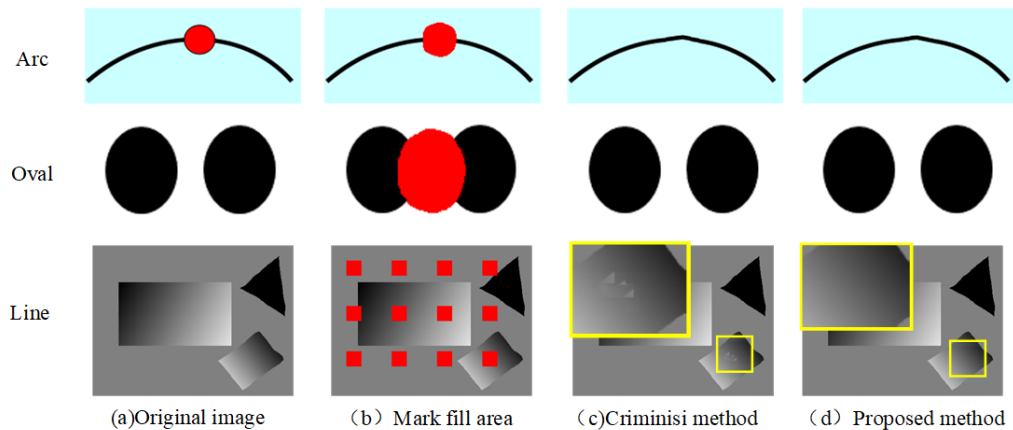


Fig. 11. Comparison with Criminisi's method. (a) Original image; (b) Mark the fill area; (c) Criminisi's method; (d) our method.

Fig. 11 shows several common structural images, including straight-line, arc-shaped, and light changes. Comparing the first row and the second row of images in **Fig. 11** from the human visual judgment, the results of the method proposed and Criminisi's method were relatively satisfactory. The third row of images in **Fig. 11** belong to the images with changes in illumination. As can be seen from the enlarged image, the patch adaptive algorithm proposed in this chapter can better repair the image, while the patch technique proposed by Criminisi had certain repair blemishes. **Table 2** gives the qualitative indicators of the two methods. From the PSNR, SSIM, and FSIM value analysis, the algorithm proposed in this chapter is superior to the patch repair method proposed by Criminisi in structural filling.

Table 2. Comparison with Criminisi patch technique PSNR, SSIM, FSIM.

Image	Subimage	Criminisi's method			Proposed method		
		PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Fig. 11	Arc image	16.8	0.94855	0.9063	16.80	0.94859	0.9065
	Oval image	42.91	0.99941	0.99869	42.90	0.99941	0.99868
	Line image	48.54	0.99827	0.99755	55.27	0.99978	0.99934

Table 3. Comparison with Criminisi patch technique PSNR, SSIM, FSIM.

Image	subimage	Criminisi's method			Proposed method		
		PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Fig. 12	Baboon	33.49	0.98782	0.99159	33.74	0.98843	0.99200
	Fruits	36.0	0.99084	0.98832	37.25	0.99353	0.99153
	Zebra	28.88	0.96335	0.95681	30.73	0.96962	0.97881

For further analysis of the patch adaptive repair method proposed in this chapter to repair the texture image, **Fig. 12** shows four images, and the damaged area gradually increases. **Table 3** shows the comparison values of the two repair results on the PSNR, SSIM, and FSIM indicators. When repairing baboon, fruits, tablecloth and zebra images, from the values of PSNR, SSIM, and FSIM, the proposed algorithm was higher than Criminisi's method. In addition, there were obvious defects in both the tablecloth and zebra images. The adaptive algorithm proposed can maintain the consistency of the overall structure and obtain satisfactory repair results.

In addition to comparing with the Criminisi method, the algorithm proposed is compared with other repair algorithms. The library and BSDS500 proposed in [41] were used.

Table 4. Comparison of PSNR, SSIM and FSIM values with other repair technologies.

Method	Image	PSNR	SSIM	FSIM
Telea et al.[42]	Fig. 13	22.80	0.92216	0.94043
	Fig. 14	32.89	0.98339	0.99321
	Fig. 13	20.76	0.912784	0.968270
Bugea et al.[43]	Fig. 14	31.34	0.980999	0.994110
	Fig. 15	34.88	0.993643	0.995393
	Fig. 13	20.48	0.912272	0.960520
Herling et al.[44]	Fig. 14	30.81	0.981287	0.993172
	Fig. 15	34.96	0.993786	0.995613
	Fig. 13	23.05	0.922129	0.933937
Getreuer[45]	Fig. 14	33.34	0.982982	0.991001
	Fig. 15	33.92	0.993069	0.991684
	Fig. 13	22.93	0.920377	0.956399
Xu & Sun [27]	Fig. 14	33.03	0.981718	0.993532
	Fig. 15	39.79	0.996691	0.998104
	Fig. 15	35.96	0.994500	0.995200
Bertalmio et al.[2]	Fig. 13	20.27	0.91451	0.94699
	Fig. 14	31.22	0.98203	0.99314
	Fig. 15	40.30	0.99825	0.99825
Nazeri et al.[52]	Fig. 13	21.66	0.92315	0.96435
	Fig. 14	32.59	0.98226	0.99535
	Fig. 15	39.33	0.99693	0.99809
Proposed method	Fig. 13	20.48	0.91183	0.96517
	Fig. 14	30.04	0.9812	0.9921
	Fig. 15	38.26	0.9967	0.9975

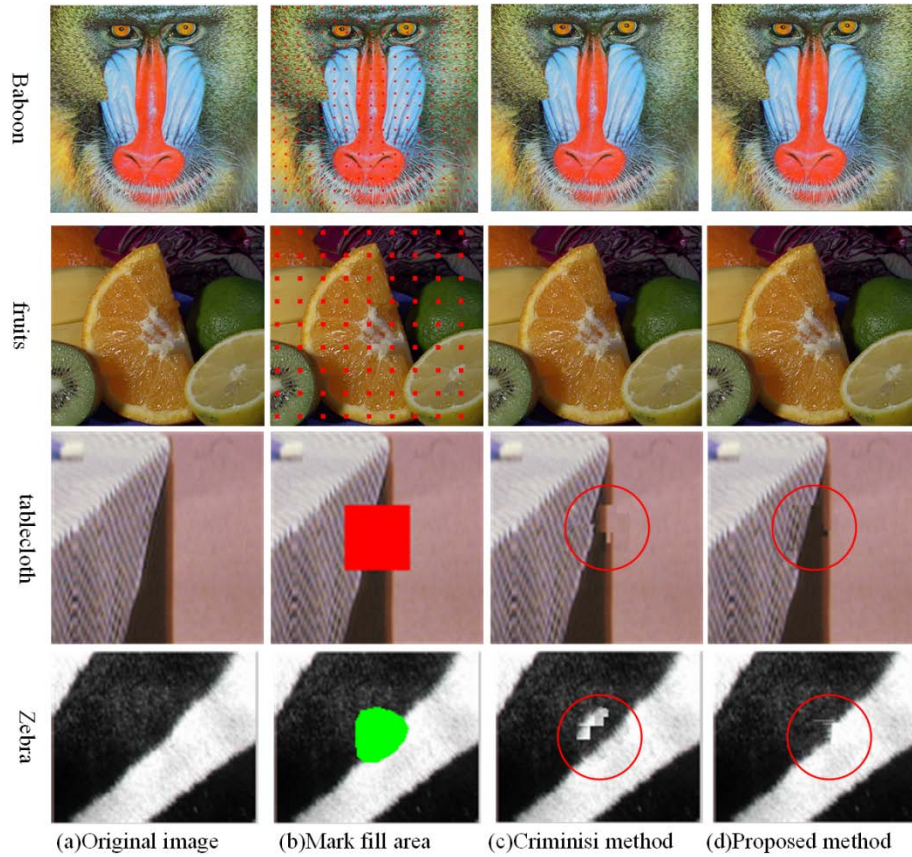


Fig. 12. Comparison with Criminisi's method. (a) Original image; (b) Mark the fill area; (c) Criminisi's method [7] (d) our method.

Fig. 13 is the grassland, which contains random texture information. **Fig. 13** contains origin images, marked filling image, and inpainted results with Telea et al. [42], Bugeau et al. [43], Herling et al. [44], Getreuer et al. [45], Xu & Sun [27], Yu et al. [51], Nazeri et al. [52], and the proposed method. It can be seen from the **Fig. 13** that Bugeau et al. [43], Herring et al. [44], and the proposed methods had good repair effects and left essentially no trace of the repair. Other repair methods had fuzzy effects and poor repair effects. This also shows that the methods of Telea et al. [42] and Getreuer [45] were suitable for repairing small areas and that blurring effects will occur when repairing large areas. The method of Xu & Sun [27] used patch sparsity and used multiple patches, which is not suitable for the repair of random textures. The method of Yu et al. [51] and Nazeri et al. [52] cannot repair image well by deep neural network, when there is no suitable sample image training network.

Fig. 14 is a structured image, which contains random texture information. **Fig. 14** contains origin images, marked filling image, and inpainted results with Telea et al. [42], Bugeau et al. [43], Herling et al. [44], Getreuer et al. [45], Xu & Sun [27], Yu et al. [51], Nazeri et al. [52], and the proposed method. It can be seen from the figure that the repair methods proposed by Herling et al. [44], Yu [51] and our proposed method had good repair effects, and no traces of repairs can be seen. The other repair methods had blurred effects and poor repair effects.

Fig. 15 shows the repair the architectural image, the image contains both structural and texture information. **Fig. 15** contains origin images, marked filling image, and inpainted

results with Telea et al. [42], Bugeau et al. [43], Herling et al. [44], Getreuer et al. [45], Xu & Sun [27], Yu et al. [51], Nazeri et al. [52], and the proposed method. We can see that the proposed method, Yu et al. [51] method and Nazeri et al. [52] method, had less ambiguity. The other repair methods had a certain degree of blur effect, and the repair effect was general. From the data in Table 4, our proposed patch adaptive method could repair both structure and texture of image.

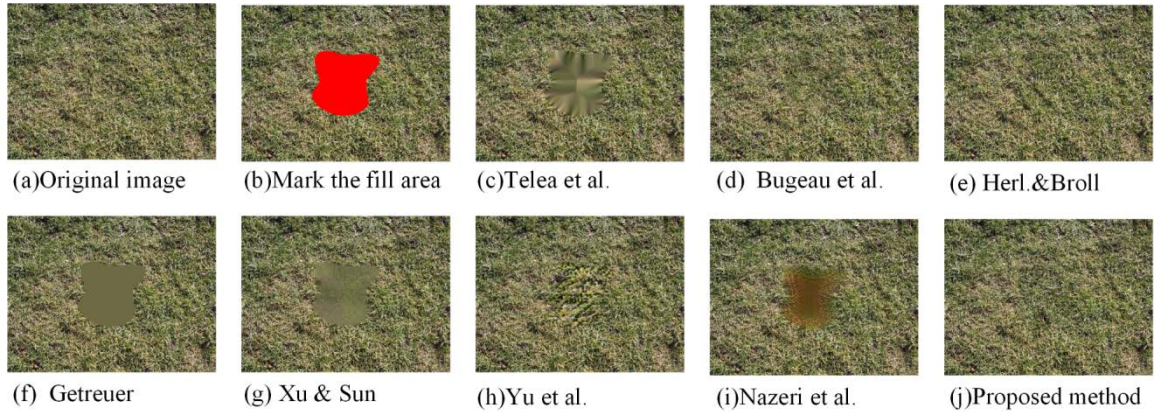


Fig. 13. Comparison of grass image repair results.

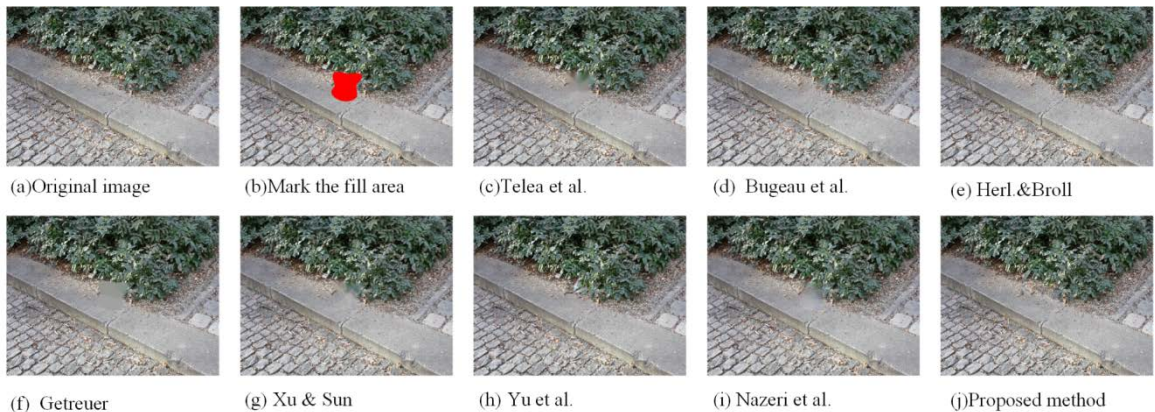


Fig. 14. Comparison of texture image repair results.

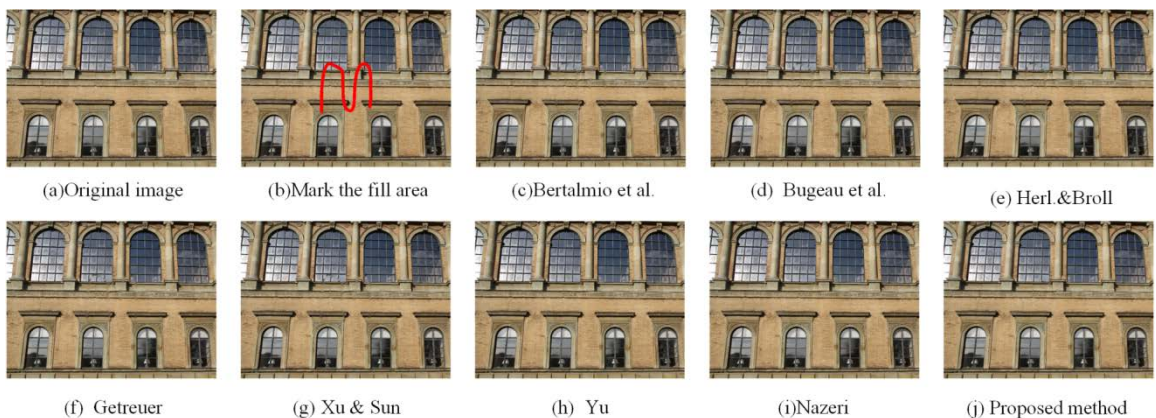


Fig. 15. Comparison of architecture image repair results.

In addition to repairing the image, a comparative experiment was performed to remove the target in the image. **Fig. 16** contains origin images, marked filling image, and inpainted results with Criminisi et al. [7], Barnes et al. [8], Darabi et al. [46], Fedorov et al. [47], Telea et al. [42], Yu et al. [51], Nazeri et al. [52], and the proposed method, respectively. Among them, Telea et al. [42] was suitable for repairing a small area, so the repair results exists blurring results as shown in **Fig. 16(g)**. The inpainting result of Barnes et al. [8], and Fedorov et al. [47] method are a bit unnatural in the water as shown in **Fig. 16(d),(f)**. Yu's and Nazeri's repair method caused some yellow areas in the water. The other method was to use the technique of texture synthesis, and the repair result was satisfactory.

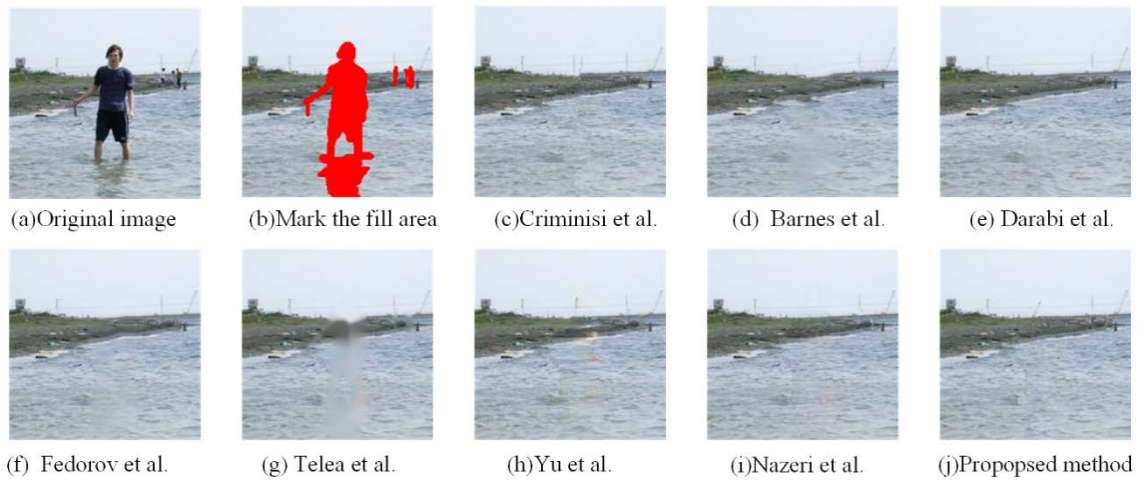


Fig. 16. The inpainting results of water area.

Fig. 17 is a more complex image, which contains texture and structural information. **Fig. 17** contains origin images, marked filling image, and inpainted results with Criminisi et al. [7], Barnes et al. [8], Darabi et al. [46], Fedorov et al. [47], Telea et al. [42], Yu et al. [51], Nazeri et al. [52], and the proposed method, respectively. Among them, Telea et al. [42] was suitable for repairing a small area, so the repair result comes up blurring results. From the point of view of the repair results, the linear structure inconsistencies or fractures happened when using Criminisi et al. [7], Barnes et al. [8], Yu et al. [51], and Fedorov et al. [47] method. As can be seen from **Fig. 17**, it could repair the structure and texture at the same time, and the effect was quite good using Darabi et al. [46], Nazeri et al. [52] and our method.

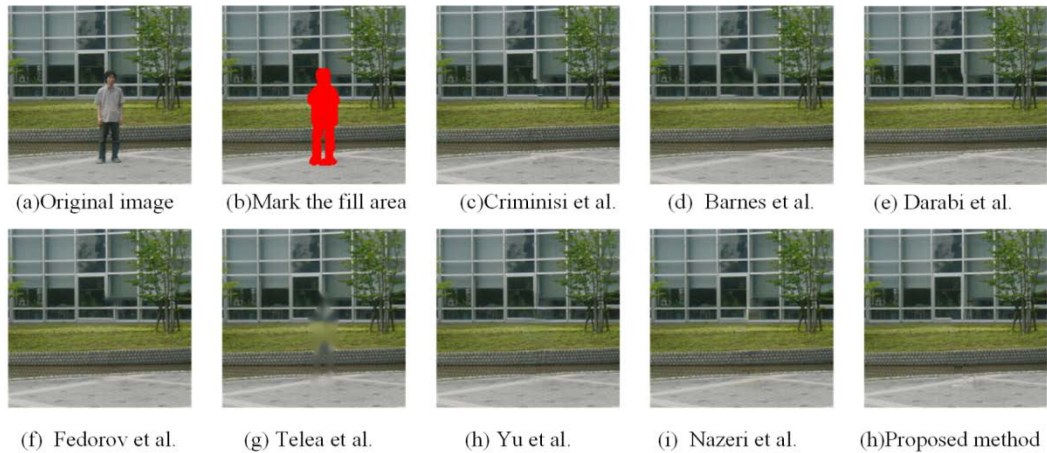


Fig. 17. Inpainting result of complex image.

Fig. 18 is also a more complex image, containing texture and structure information. Among them, **Fig. 18** contains origin images, marked filling image, and inpainted results with Criminisi et al. [7], Barneset al. [8], Darabi et al. [46], Fedorov et al. [47], Telea et al. [42], Yu et al. [51], Nazeri et al. [52], and the proposed method, respectively. As can be seen from the **Fig. 18**, the result of the repair using the Criminisi method is not good. The inpainting method of Telea et al. [42] cannot fill large area as shown in **Fig. 18(g)**. When repairing a large area, the result was blurred. The method of Barnes et al. [8] has a line misalignment in the **Fig. 18(d)**. The inpainted result by Darabi et al. [46], Yu et al. [51] and Nazeri et al. [52] also appeared blurry as shown in **Fig. 18(e), (h), (i)**, but the repair result was acceptable. A clear line misalignment problem occurred by Fedorov et al. [47] method. The method proposed can repair the structure and texture at the same time, and the repair results were satisfactory.

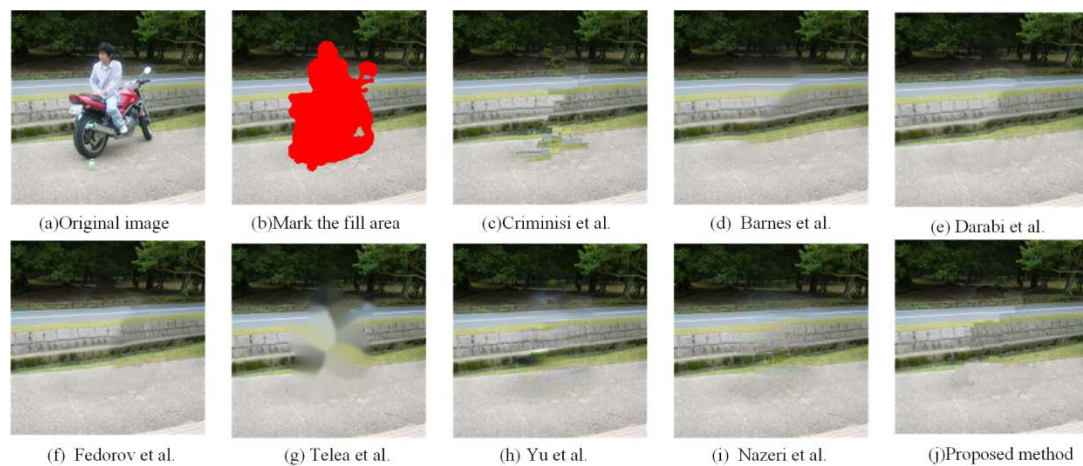


Fig. 18. Target removal.

In view of the time complexity of the repair algorithm, the running time of repairing the images in **Fig. 16**, **Fig. 17**, and **Fig. 18** are compared with other methods in the experiment. The hardware platform used was an Intel core i7 CPU @ 2.0 GHz (MATLAB 9.2.0 with 64 bits). Combining **Table 5** and the comparative experiments above, the Telea method used fast matching technique, and the repair speed was fast; however, the repair result was poor. The Criminisi method did not optimize the patch. It consumed less time, and the repair effect was general. Barnes used the PatchMatch technique with a probability distribution search, which has a short time consumption and is prone to blur. Darabi and Fedorov method use layered restoration of the image, which is time-consuming. Yu and Nazeri are based on the deep neural network method to repair the image.

Table 5. Comparison of repair time (s).

Image	Size	Area	Criminisi [7]	Barnes [8]	Darabi [42]	Fedorov [43]	Telea [38]	Yu [51]	Nazeri [52]	Our method
Fig. 16	200 × 200	5977	2.22	6.22	118.9	274.51	0.024	519.23	331.99	134.78
Fig. 17	200 × 200	2611	1.0	3.76	115.5	110.45	0.011	93.68	344.3	30.96
Fig. 18	200 × 200	7907	2.59	7.55	119.2	198.02	0.031	134.34	406.27	120.75

When using the Darabi method to repair the image, the damaged area did not affect the time consumed, and the repair result was acceptable. The Fedorov's method adopted multi-layer repair and non-local matching technique, which took the longest time. The effect was good when repairing structural images; however, it was easy to produce blur when repairing texture images. Yu's method and Nazeri's method are belong to deep neural network for image inpainting, the repair time is related to the structure of the network. The repair method proposed in the paper had a certain relationship between the repair time and the damaged area. The larger the damaged area, the longer it takes, but the repair effect was better. Whether it was a structural image, a texture image, or an image that contains both structure and texture, the results were relatively satisfactory. In addition, in the process of patch filling, blockiness was not eliminated; for example, Poisson fusion [48–50] eliminated blockiness. Based on the above analysis, the method proposed in this paper has better repair results to a certain extent.

5.4 Time complexity analysis

We assume that the height and width of image are H and W , the patch size is $m \times n$, the filling area is S , The layer number of image decomposition is 3. The area of the filling regions is an important factor affecting the image repair time. In addition, there are also factors that affect the image inpainting time: the size of the image, the size of the sample block, the time required to calculate the size of the sample block, the time to compute the priority, the time to search for the sample block. For the convenience of analysis, the area filled each time is half of the sample block size, so the the number of filling is : $a = \frac{S}{\frac{m*n}{2}} = \frac{2S}{mn}$.

The following is an analysis of other issues. There are two aspects to the time required to calculate the sample block size, (1) the time required to calculate the size of the structure sample block; (2) the time required to calculate the size of the texture sample block. In the first case, the time complexity is $O(HW)$ which is related to the size of the image. In the second case, the time complexity is $O(mn)$, which is about calculating the size of texture structure block. The main time-consuming is feature extraction, which is related to the size of sample block.

The calculation of priority is divided into two parts, (1) calculating the confidence; (2) calculating the data term. The time complexity for calculating confidence is $O(HW)$, which need to scan the whole image. Calculating the data term requires compute the gradient value, the time complexity is $O(HW)$. 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 looking for sample patches, the whole image need to be traversed. The time complexity of the match is $O(HW)$, at each match, it takes a certain amount of time to calculate the color term difference. In the process of each matching, we consider the color image with three channels, the time complexity of the calculating color difference is $O(3mn/2)$. So the time complexity of searching for a sample patch is $O(3HWmn/2) \approx O(HWmn)$.

When the sample patch is found, we can fill the image, the time spent is negligible. According to the number of filling steps α , the time complexity is $O(\alpha(HW + HWmn)) = O(2S(HW + HWmn)/mn) \approx O(SHW(1 + \frac{1}{mn}))$. 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 data in **Table 5**, the size of image is 200*200, the filled area of **Fig. 17** is the smallest, it takes the least time to repair. Comparing with **Fig. 16** and **Fig. 18**, the filling area of **Fig. 18** is larger than the filling area of **Fig. 16**. However, **Fig. 16** need more time to repair. The reason is that the texture of **Fig. 16** is stronger, and smaller sample blocks need to be selected for filling, so it takes longer.

6. Conclusion

This paper analyzed the structure-aware items based on the information around the image filling patches, and the structure-aware items were used to predict the size of the structure patch according to the texture of the image, by extracting the gray level co-occurrence matrix, extracting 24-dimensional features, and predicting the size of the texture patch by training the neural network. Considering the structure patch size and texture patch size comprehensively, the optimal patch size was obtained by weighting. After determining the patch size, we then determined the image confidence and data items. A pyramid-decomposed three-layer image was used to comprehensively calculate the data items. The data items thus had certain anti-interference properties. In addition, in order to reduce the impact of the noise and weak textures in the process of calculating the structural perception, the image was first smoothed, and the structural area was reserved as much as possible to highlight the importance of the structural filling priority. Through repair comparison experiments, we found that the patch adaptive repair method proposed was more effective than the Telea et al. [42], Bugeau et al. [43], Herling et al. [44], Getreuer [45], Xu & Sun [27], Criminisi et al. [7], Barnes et al. [8], Darabi et al. [46], Fedorov et al. [47], Yu et al. [51] and Nazeri et al. [52] methods in comparative experiments. The proposed patch adaptive algorithm can not only repair the structure area and texture area, but also repair the area containing the structure and texture at the same time.

References

- [1] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, "Image inpainting," in *Proc. of ACM SIGGRAPH Conf. Computer Graph.*, New Orleans, LA, USA, pp.417–424, 2000. [Article\(CrossRef Link\)](#)
- [2] M. Bertalmio, A. Bertozzi, G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proc. of IEEE Comput Soc. Conf. Comput. Vision Pattern Recognit*, Kauai, HI, USA, pp.355–362, 2001. [Article \(CrossRef Link\)](#)
- [3] T. F. Chan, and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *J. Vis. Commun. Image R.*, vol. 12, no. 4, pp. 436-449, 2001. [Article \(CrossRef Link\)](#)
- [4] A. Efros, W. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. of ACM SIGGRAPH Conf. Computer Graph.*, Los Angeles, CA, USA, pp.341-346, 2001. [Article \(CrossRef Link\)](#)
- [5] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 277-286, 2003. [Article \(CrossRef Link\)](#)

- [6] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," in *Proc. of IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit*, vol. 12, no. 8, pp. 882-889, Aug. 2003. [Article \(CrossRef Link\)](#)
- [7] 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, 2004. [Article \(CrossRef Link\)](#)
- [8] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 1-11, 2009. [Article \(CrossRef Link\)](#)
- [9] P. Arias, G. Facciolo, V. Caselles G. Sapiro, "A Variational Framework for Exemplar-Based Image Inpainting," *Int. J. Comput. Vision*, vol. 93, no. 3, pp. 319-347, Jul. 2011. [Article \(CrossRef Link\)](#)
- [10] D. Thanh, V. Prasath, S. Dvoenko, L. M. Hieu, "An adaptive image inpainting method based on euler's elastica with adaptive parameters estimation and the discrete gradient method," *Signal Process.*, vol. 178, pp.1-19, 2021. [Article \(CrossRef Link\)](#)
- [11] Zdunek R, Sadowski T, "Image Completion with Hybrid Interpolation in Tensor Representation," *Appl. Sci.*, vol. 10, no. 3, pp. 1-16, 2020. [Article \(CrossRef Link\)](#)
- [12] S. Yang, H. Liang, Y. Wang, H.Y. Cai, X.D. Chen, "Image Inpainting Based on Multi-Patch Match with Adaptive Size," *Appl. Sci.*, vol. 10, no. 14, pp. 1-17, 2020. [Article \(CrossRef Link\)](#)
- [13] R. Zhao, L. Zhao, "Image Inpainting Research Based on Deep Learning," *International Journal of Advanced Network, Monitoring and Controls*, vol. 5, no. 2, pp. 23-30, 2020. [Article \(CrossRef Link\)](#)
- [14] O. Le Meur, J. Gautier, and C. Guillemot, "Exampler-based inpainting based on local geometry," in *Proc. of Int. Conf. Image Process.*, Hongkong, China, pp. 3401-3404, 2011. [Article \(CrossRef Link\)](#)
- [15] H. Zhou, and J. Zheng, "Adaptive patch size determination for patch-based image completion," in *Proc. of Int. Conf. Image Process.*, Hong Kong, China, pp. 421-424, 2010. [Article \(CrossRef Link\)](#)
- [16] N. J. Butko, L. Zhang, G. W. Cottrell, and J. R. Movellan, "Visual saliency model for robot cameras," in *Proc. of IEEE Int. Conf. Rob. Autom.*, Pasadena, CA, USA, pp. 2398-2403, 2008. [Article \(CrossRef Link\)](#)
- [17] R. P. Borole, and S. V. Bonde, "Image Restoration and Object Removal Using Prioritized Adaptive Patch-Based Inpainting in a Wavelet Domain," in *Proc. of J. Inf. Process. Syst.*, vol. 13, no. 5, pp.1183-1202, 2017. [Article \(CrossRef Link\)](#)
- [18] X. Zhou, X. Yan, "Adaptive image inpainting based on structural correlation," *Computer science*, vol.47, no.4, pp.131-135, 2020.
- [19] J. Cao, Y. Li, H. Cui Q. Zhang, "An Adaptive Sample Block and Local Search Algorithm for Inpainting of Ancient Temple Murals," *Journal of Computer-Aided Design & Computer Graphics*, vol.31, no.11, pp.2030-2037, 2019.
- [20] K. He, J. Niu, C. Shen, W. Lu, "Image Inpainting Algorithm with Adaptive Patch Using SSIM," *Journal of Tianjin University (Science and Technique)*, vol.51, no.07, pp.763-767, 2018.
- [21] C. Meng, K. He, Q. Jiao, "Image completion method with adaptive patch size," *Journal of Image and Graphics*, vol.17, no.3, pp.337-341, 2012.
- [22] Q. Fan, "Research on Exemplar-based Image Inpainting Algorithm," *M.S. thesis, University of Electronic Science and Technique of China*, Hefei, Anhui, China, 2018.
- [23] C. Chang, "Research on image inpainting algorithm based on texture," *M.S. Thesis, Fuzhou University*, Fuzhou, Fujian, China, 2016.(in Chinese)
- [24] R. P. Borole, and S. V. Bonde, "Image Restoration using Prioritized Exemplar Inpainting with Automatic Patch Optimization," *J. Inst. Eng. Ser. B*, vol. 98, no. 3, pp. 311-319, 2017. [Article\(CrossRef Link\)](#)
- [25] C. Chang, "Research on image inpainting algorithm based on texture," *M.S. Thesis, Fuzhou University*, Fuzhou, Fujian, China, 2016.

- [26] Y. He, "Research on Color Image Inpainting Based on Gradient Change and Structure Tensor," *M.S. Thesis, Hangzhou Dianzi University*, Hangzhou, Zhejiang, China, 2018.
- [27] Z. Xu, J. Sun, "Image Inpainting by Patch Propagation Using Patch Sparsity," *IEEE Trans. Image Process.*, vol.19, no.5, pp.1153–1165, 2010. [Article\(CrossRef Link\)](#)
- [28] M. Thang, "Image Inpainting Algorithm Based on Patch Structure Sparsity," *M.S. thesis, Huazhong University of Science & Technique*, Wuhan, Hubei, China, 2016.
- [29] S. Ye, X. Lin, G. Wang, "Feature based adaptive error concealment for image transmission over wireless channel," *Computer engineering and applications*, vol.19, pp.61-64, 2002.
- [30] X. Lin, X. Yang, "Effective exemplar-based image inpainting using low-rank matrix completion," in *Proc. of IEEE Int. Conf. Aware. Sci. Technol.*, Qinhuaogdao, China, pp.37-42, 2015. [Article\(CrossRef Link\)](#)
- [31] D. Z. Silvano, "A note on the gradient of a multi-image," *Comput. Vision Graphics Image Process.*, vol.33, no.1, pp.116–125, 1986. [Article\(CrossRef Link\)](#)
- [32] X. Li, Y. Qiong, X. Yang, J. Jia. "Structure extraction from texture via relative total variation," *ACM Tran. Graphics*, vol.31, no.6, pp.1-10, 2012. [Article\(CrossRef Link\)](#)
- [33] A. Criminisi, P. Perez, K. Toyama, "Object removal by exemplar-based inpainting," in *Proc. of IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit*, Madison, WI, USA, pp.1-8, 2003. [Article\(CrossRef Link\)](#)
- [34] L. Meur, C. Olivier Guillemot, "Super-Resolution-Based Inpainting," *Lect. Notes Comput. Sci.*, Florence, Italy, pp.554-567, 2012. [Article\(CrossRef Link\)](#)
- [35] F. Tony Chan, K. Michael, C. Andy Yau, M. Andy, Yip, "Superresolution image reconstruction using fast inpainting algorithms," *Appl. and Comput. Harmonic Anal.*, vol.23, no.1, pp.3–24, 2007. [Article\(CrossRef Link\)](#)
- [36] C. Ledig, T. Lucas, H. Ferenc, C. Jose, C. Andrew, A. Alejandro, A. Andrew, T. Alykhan, T. Johannes, W. Zehan, S. Wenzhe, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *Proc. of IEEE Conf. Comput. Vis. Pattern Recognit*, Honolulu, HI, USA, pp.105-114, 2017. [Article\(CrossRef Link\)](#)
- [37] C. Yang, X. Lu, Z. Lin, S. Eli, W. Oliver, L. Hao, "High-Resolution Image Inpainting Using Multi-scale Neural Patch Synthesis," in *Proc. of IEEE Conf. Comput. Vis. Pattern Recognit*, Honolulu, HI, USA, pp.4076-4084, 2017. [Article\(CrossRef Link\)](#)
- [38] L. Olivier, E. Mounira, G. Christine, "Hierarchical super-resolution-based inpainting," *IEEE Trans. Image Process.*, vol.22, no.10, pp.3779–3790, 2013. [Article\(CrossRef Link\)](#)
- [39] H. Liu, X. Bi, G. Lu, W. Wang, "Exemplar-Based Image Inpainting With Multi-Resolution Information and the Graph Cut Technique," *IEEE Access*, vol.7, pp.101641–101657, 2019. [Article\(CrossRef Link\)](#)
- [40] L. Zhang, L. Zhang, X. Mou, D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE trans. Image Process.*, vol.20, no.8, pp.2378–2386, 2011. [Article\(CrossRef Link\)](#)
- [41] P. Tiefenbacher, V. Bogischef, D. Merget, G. Rigoll, "Subjective and objective evaluation of image inpainting quality," in *Proc. of Int. Conf. Image Process.*, Quebec, QC, Canada, pp.447–451, 2015. [Article\(CrossRef Link\)](#)
- [42] A. Telea, "An image inpainting technique based on the fast marching method," *J. graphics tools*, vol.9, no.1, pp.25–36, 2004. [Article\(CrossRef Link\)](#)
- [43] A. Bugeau, M. Bertalmio, V. Caselles, G. Sapiro, "A comprehensive framework for image inpainting," *IEEE Trans. Image Process.*, vol.19, no.10, pp.2634–2645, 2010. [Article\(CrossRef Link\)](#)
- [44] J. Herling, W. Broll, "Pixmix: A real-time approach to high-quality diminished reality," in *Proc. of IEEE Int. Symp. Mix. Augmented Real., Sci. Technol. Pap.*, Atlanta, GA, USA, pp.141–150, 2012. [Article\(CrossRef Link\)](#)
- [45] P. Getreuer, "Total variation inpainting using split Bregman," *Image Processing on Line*, vol.2, pp.147–157, 2012. [Article\(CrossRef Link\)](#)

- [46] S. Darabi, E. Shechtman, C. Barnes, B. G. Dan, P. Sen, “Image melding: combining inconsistent images using patch-based synthesis,” *ACM Trans. Graphics*, vol.31, no.4, pp.1–10, 2012. [Article\(CrossRef Link\)](#)
- [47] V. Fedorov, G. Facciolo, P. Arias, “Variational framework for non-local inpainting,” *Image Processing on Line*, vol.5, pp.362–386, 2015. [Article\(CrossRef Link\)](#)
- [48] J. Matías, G. Facciolo, E. Meinhardt-Llopis, “Poisson Image Editing,” *Image Processing on Line*, vol.6, pp.300–325, 2016. [Article\(CrossRef Link\)](#)
- [49] D. Lee, S. Yoo, M. Choi, J. Ra, J. Kim, “Block Poisson Method and its application to large scale image editing,” in *Proc. of Int. Conf. Image Process.*, Orlando, FL, USA, pp.2121–2124, 2012. [Article\(CrossRef Link\)](#)
- [50] P. Pérez, M. Gangnet, A. Blake, “Poisson image editing,” *ACM Trans. Graphics*, vol.22, no.3, pp.313–318, 2003. [Article\(CrossRef Link\)](#)
- [51] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T. Huang, “Free-form image inpainting with gated convolution,” in *Proc. of IEEE Int. Conf. Comput. Vision*, Seoul, Korea, pp.4471–4480, 2019. [Article\(CrossRef Link\)](#)
- [52] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, M. Ebrahimi, “Edgeconnect: Generative image inpainting with adversarial edge learning,” *arXiv preprint arXiv:1901.00212*, 2019.



Huaming Liu received the B.S. degree in computer science and technology from Shangqiu Normal University, Shangqiu, China, in 2005, and the M.S. degrees in computer software and theory from Northwest Minzu University, Lanzhou, China. He received the Ph.D. degree in Signal and information processing from Nanjing University of Posts and Telecommunications, in 2020. He was a Visiting Scholar with the University of Science and Technology of China, Hefei, China, in 2015. He is also 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 Wuwei, 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 (Member, IEEE) received the B.S. degree in mathematics from Northwest Normal University, Lanzhou, China, in 1983. She was a Visiting Scholar with 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, IN, USA. She is currently a Professor and a Doctoral Supervisor with the Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University, Lanzhou. She has published more than 70 papers in major journals and international conferences in image processing field. Her research interests include image processing, pattern recognition, Tibetan information processing, and computer vision.