FAST IMAGE INPAINTING ALGORITHM BASED ON PRINCIPAL COMPONENT ANALYSIS

Guanjun XU[1]

In this paper, an improved fast algorithm based on Principal Component Analysis (PCA) is proposed to reduce the computational complexity and avoid wrong structure diffusion of the exemplar-based inpainting algorithms. The proposed algorithm replaces the global sample patch with a neighboring sample patch from a local window to reduce algorithm complexity. The PCA dimension reduction technology is used to select the sample patch corresponding to the centralized region updated to the original image among multiple well-matched neighboring sample patches, so as to avoid the diffusion of the wrong structure as much as possible. The proposed algorithm determines the local window parameters through the simulation experiments. The results show that in the single image restoration, the improved algorithm is superior to the original algorithm in terms of image effect and computational efficiency.

Keywords: image inpainting, exemplar-based inpainting, principal component analysis, local window, neighboring sample patch

1. Introduction

Image inpainting [1-3] aims to reconstruct a damaged image by using the rest of the image information according to certain rules so that the restored image is visually plausible for an observer. The image inpainting technology has been widely used in photo restoration, special film effects, cultural relics protection, and many other fields. Namely, it has always been a hot topic in the field of computer image processing.

The exemplar-based image inpainting algorithm [2] is one of the most-studied image restoration algorithms. This algorithm considers the texture and structural information simultaneously and can achieve a relatively good restoration effect in complex damaged regions. However, this algorithm can easily produce mismatches, which can lead to the diffusion of the wrong image structure, causing local image discordance, and using the global sample patch matching increases the computation burden.

In view of the shortcomings of the exemplar-based algorithm, this paper proposes an improved fast algorithm based on the Principal Component Analysis

[1] Sr. Eng., Information Center, Taizhou Vocational College of Science and Technology, Taizhou, Zhejiang, China, email: xug@tzvcst.edu.cn

(PCA). The proposed algorithm takes account of the continuity of the natural image's local structure and high correlation of pixel information [4,5] and applies exemplar-based inpainting to an appropriate local window. In the matching process, the PCA method is used for dimension reduction of k best-matching patches of the pending pixels, and a high degree of concentration patch is used to fill the missing region in the original image, so as to avoid a mismatch caused by the error structure generation and diffusion. Due to the locality of neighboring sample patches, the fusion degree of the restored image is higher.

2. Exemplar-based image inpainting algorithm

The exemplar-based image inpainting algorithm uses the edge information of the area to be repaired so that the pixel information along the direction of the isophote can spread to the area to be repaired. This algorithm can retain the image structure and restore the image quickly by using a sample patch in the known area.

2.1. Algorithm description

The steps of the exemplar-based image inpainting algorithm are as follows:

(1) Calculate the priority of a patch to be repaired.

In Fig. 1, Φ denotes a known image region, Ω denotes a region to be filled, and $\partial\Omega$ represents the contour of region Ω. In order to repair patch $\Psi_{\mathbf{p}}$ on the boundary of region Φ, whose center is at point $\mathbf{p}$, the priority function is defined as follows:

$$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p}) \tag{1}$$

The priority function $P(\mathbf{p})$ defines the order of image filling, which comprehensively reflects the confidence level and structural strength of a patch to be repaired. The confidence $C(\mathbf{p})$ indicates the information reliability at point $\mathbf{p}$, while $D(\mathbf{p})$ reflects the strength of the image structure, and they are defined as:

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\Psi_{\mathbf{p}}\cap\Phi} C(\mathbf{q})}{|\Psi_{\mathbf{p}}|}, \quad D(\mathbf{p}) = \frac{|\nabla I_{\mathbf{p}}^{\perp}\cdot\mathbf{n}_{\mathbf{p}}|}{\alpha} \tag{2}$$

In (2), $|\Psi_{\mathbf{p}}|$ represents the patch area to be repaired. Initially, if $\mathbf{q}\in\Phi$, then $C(\mathbf{q})=1$, and if $\mathbf{q}\notin\Phi$, then $C(\mathbf{q})=0$; $\nabla I_{\mathbf{p}}^{\perp}$ denotes the isophote at $\mathbf{p}$, which is the vertical direction of the gradient of pixel $\mathbf{p}$, $\mathbf{n}_{\mathbf{p}}$ represents the normal vector of $\partial\Omega$ at $\mathbf{p}$, and the normalization factor is $\alpha = 255$.

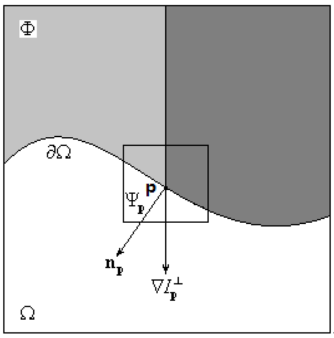


Fig. 1. Patch to be repaired

The algorithm first computes $P(\mathbf{p})$ for all points on $\partial\Omega$ and then determines a patch $\Psi_{\mathbf{p}}$ that contains more known pixel information and has a distinct image structure.

(2) Search for the best-matching sample patch and fill

Find a sample patch $\Psi_{\mathbf{q}}$ in the known area Φ, which is most similar to $\Psi_{\mathbf{p}}$, and which satisfies the following relation: $\Psi_{\mathbf{q}} = \arg\min(d(\Psi_{\mathbf{p}}, \Psi_{\mathbf{q}}))$, where $d(\Psi_{\mathbf{p}}, \Psi_{\mathbf{q}})$ represents the similarity between patches $\Psi_{\mathbf{p}}$ and $\Psi_{\mathbf{q}}$. Generally, the sum of squared differences (SSD) of the pixels value of $\Psi_{\mathbf{q}} \cap \Phi$ and $\Psi_{\mathbf{p}} \cap \Phi$ is calculated, and the best-matching sample patch $\Psi_{\mathbf{q}}$ is copied to the patch $\Psi_{\mathbf{p}}$.

(3) Update the confidence function

Update the confidence of the newly filled pixel in patch $\Psi_{\mathbf{p}}$, and set the confidence value of the new filled pixels to $C(\mathbf{p})$.

2.2. Algorithm analysis

The Step (2) of the exemplar-based image inpainting algorithm requires to search for the best-matching sample patch in the known area of the whole image for each patch to be repaired. Assume the image size is $M \times N$ and the size of the sample patch is 9×9. For these sizes, the computation complexity of one-time matching [2] is approximately $81MN$, and the computation complexity of the improved algorithm [6] is βMN, where $\beta << 81$. Thus, the computation complexity of these algorithms is related to the image size.

The sample patch selection diagram is shown in Fig. 2. In Fig. 2, the most left element in the first row represents the patch to be repaired, where the missing part is marked in grey color, and the remaining elements denote the sample patches for matching. The original algorithm[2] will choose the middle element in the first row as the best-matching patch and restore the patch to be repaired according to it. The SSD matching degree of the other sample patches is also high, but if any of them is selected to be used in the patch repairing process, the final result will be a circle.

The improved algorithm mainly aims to reduce the computational complexity and select a proper sample patch.

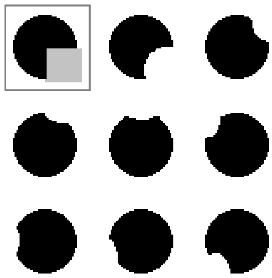


Fig. 2. Sample patch selection diagram

3. Fast image inpainting algorithm based on PCA

The neighboring sample patch [7] represents a sample patch in a local window. The improved algorithm replaces the global sample patch [2] with the neighboring sample patch, thus reducing algorithm complexity. Then, the PCA is used to reduce the pixel information to be filled and select a sample patch corresponding to the higher concentration area, so as to avoid the mismatching of sample patches.

3.1. Local window selection

A patch $\Psi_{\mathbf{p}}$ on $\partial\Omega$ to be repaired is chosen using the priority function $P(\mathbf{p})$. Then, a local window $W_{\mathbf{p}}$ with a size of $m \times n$ is selected such that to satisfy $\Psi_{\mathbf{p}} \subset W_{\mathbf{p}}$; also, $W_{\mathbf{p}}$ should contain as much known image information as possible [8]. The local window $W_{\mathbf{p}}$ is expressed as follows:

$$W_{\mathbf{p}} = \max_{W_{\mathbf{p_i}} \supset \Psi_{\mathbf{p}}} \sum_{\mathbf{q} \in W_{\mathbf{p_i}} \cap \Phi} C(\mathbf{q}) \tag{3}$$

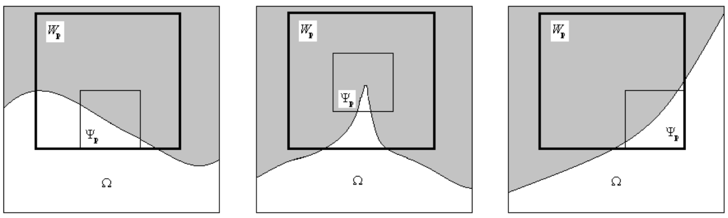
The improved algorithm applies a simplified local window selection method, and its average complexity is similar to that in equation (3). The simplified method classifies the local windows into three types: open type, enclosed type, and corner type.

Open type: A patch to be repaired is located on the middle side of the local window, as shown in Fig. 3(a), so one or two vertices of the repaired patch are located in area Ω. There are four sub-types of the open type windows: up (U type), down (D type), left (L type), and right (R type). Open type selection for a two vertex in a patch is obvious. If only one vertex of the patch to be repaired is located in area Ω, such as the up-left vertex, there are two options, U type and L type. The selection criterion of the method is to calculate the total confidence of the pixels in two windows by (4), and select the window type with greater total confidence.

$$C(W_{\mathbf{p}}) = \sum_{\mathbf{q} \in W_{\mathbf{p}} \cap \Phi} C(\mathbf{q}) \tag{4}$$

Enclosed type (E type): Patch $\Psi_{\mathbf{p}}$ is located in the center of the local window, as shown in Fig. 3(b), and the four vertices of patch $\Psi_{\mathbf{p}}$ are in Φ.

Corner type: Patch $\Psi_{\mathbf{p}}$ is located in the corner of the local window, as shown in Fig. 3(c), and three vertices of patch $\Psi_{\mathbf{p}}$ are located in area Ω. The corner type has four sub-types: up-left (UL type), up-right (UR type), down-left (DL type) and down-right (DR type).



(a) D Open type (b) Enclosed type (c) DR Corner type

Fig. 3. Local window types

Before the local window selection, first the four vertices of patch $\Psi_{\mathbf{p}}$ located in area Ω are determined. The four vertices are recorded as: the up-left vertex (UL), the up-right vertex (UR), the down-left vertex (DL), and the down-right vertex (DR). According to the distribution of the four vertices, the type of local window is determined. If only one vertex is located in area Ω, the total confidence values of the two open window types are obtained by formula (4) and compared. The total confidences of U type, D type, L type, and R type are denoted as $C(U)$, $C(D)$, $C(L)$ and $C(R)$, respectively. The specific window selection process is shown in Fig. 4.

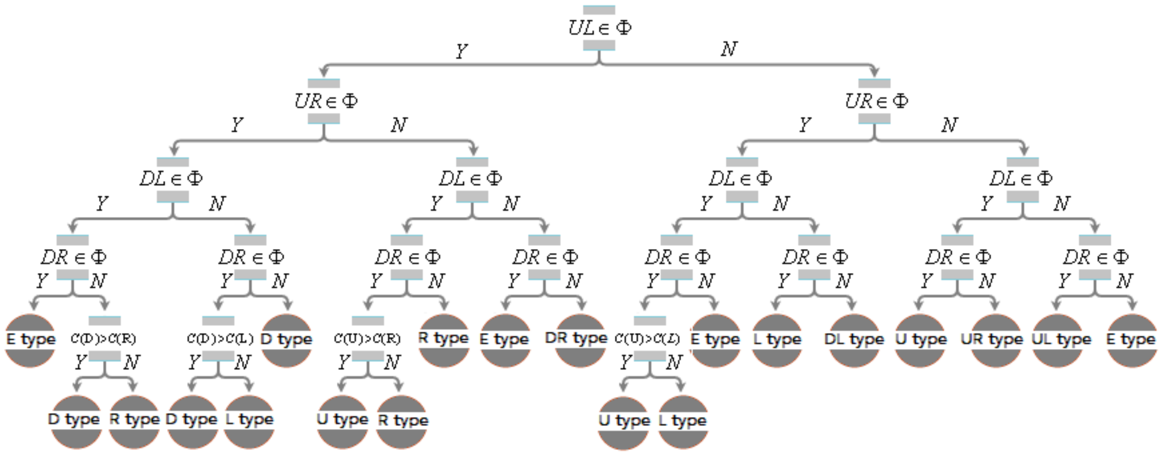


Fig. 4. Local window selection

3.2. PCA-based sample patch matching

In image feature extraction, the dimension of the local image block data is reduced by the PCA, and the isolated points in the low-dimensional space usually correspond to the image features [9,10]. In the proposed algorithm, the best-matching neighboring sample patch is determined by calculating the SSD. Then the sample patch information is mapped to the low-dimensional space points using the PCA; the sample patches that correspond to relatively concentrated points are most likely to be the best-matching patch, while relatively isolated points are likely to be the mismatched patches, as shown in Fig. 5.

Compared to the sample patch matching based on the smallest SSD, the PCA matching neighboring sample patch has three advantages: (1) using the local similarity of an image, the maximum probability of restoring the real image is achieved [11]; (2) the diffusion of a wrong structure can be effectively avoided; (3) it is avoided to make the image structure blurred due to the average pixel value obtained by the weighted algorithm [3,12,13]. The PCA-based sample patch matching algorithm is as follows.

Step 1: Select k best-matching patches, and let the pixel value of the best-matching patches to be filled in a matrix $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \cdots \mathbf{x_k}]$.

Step 2: Calculate $\bar{\mathbf{x}} = \frac{1}{k}\sum_{i=1}^{k}\mathbf{x_i}$, $\hat{\mathbf{X}} = [\mathbf{x_1} - \bar{\mathbf{x}}, \mathbf{x_2} - \bar{\mathbf{x}}, \cdots \mathbf{x_k} - \bar{\mathbf{x}}]$, and its covariance matrix is given by:

$$\mathbf{A} = \frac{1}{k^2}\hat{\mathbf{X}}^T\hat{\mathbf{X}} \qquad (5)$$

Step 3: Obtain the eigenvalue decomposition of matrix $\mathbf{A}$; sort the eigenvalues from large to small, and take the former eigenvectors $\mathbf{U}=[\mathbf{u}_1,\mathbf{u}_2,\cdots\mathbf{u}_\mathbf{d}]$.

Step 4: Calculate $\mathbf{Y}=\mathbf{U}^T\mathbf{X}$ to obtain the points after dimensional reduction.

Step 5: Update the original image with the sample patch corresponding to $median(\mathbf{Y})$.

Fig. 5. The PCA-based sample patch matching

3.3. Improved algorithm working principle

The improved algorithm consists of the PCA-based neighboring sample patch matching (Algorithm 1) and the algorithm for the final search and repair of the prioritized patch. In Algorithm 1, instead of the global sample patch matching [2] the neighboring sample patch matching is performed based on the PCA. The specific algorithms are as follows:

Algorithm 1 (PCA-based neighborhood sample patch matching):

Step 1: Determine the type and position of a local window according to the structure of the patch to be repaired $\Psi_\mathbf{p}$.

Step 2: Search the neighboring sample patches ($\Psi_{\hat{\mathbf{p}}}\in W_\mathbf{p}\bigcap\Phi$) in a local window $W_\mathbf{p}$; get the minimum $d(\Psi_{\hat{\mathbf{p}}},\Psi_\mathbf{p})$ of the neighboring sample patches, and record them as $\{\Psi_{\hat{\mathbf{p}}_i}\},(\mathrm{i}=1,2\cdots k)$.

Step 3: $\mathrm{PCA}\{\Psi_{\hat{\mathbf{p}}_i}\}\rightarrow\{t_i\}$, let $t_m=median\{t_i\}$, and the corresponding neighboring sample patch is $\Psi_{\hat{\mathbf{p}}_m}$.

Step 4: Fill in $\Psi_\mathbf{p}\bigcap\Omega$ with the pixel value of neighboring sample patch $\Psi_{\hat{\mathbf{p}}_m}$.

After patch $\Psi_\mathbf{p}$ is repaired, the confidence values of the new filling pixels are updated to $C(\mathbf{p})$. The overall algorithm is as follows:

Algorithm 2 (Fast image inpainting based on the PCA):

Step 1: Identify the fill front $\partial\Omega$, if $\Omega=\phi$, exit.

Step 2: Compute priorities $P(\mathbf{p})$, $\forall \mathbf{p} \in \partial\Omega$.

Step 3: Find the repair patch $\Psi_{\mathbf{p}}$, where $p = \arg\max_{\mathbf{p}\in\partial\Omega} P(\mathbf{p})$.

Step 4: Apply Algorithm 1 to repair patch $\Psi_{\mathbf{p}}$.

Step 5: Update $C(\mathbf{p})$, $\forall \mathbf{p} \in \Psi_{\mathbf{p}} \cap \Phi$.

Step 6: Update known image area $\Phi = \Phi \cup \Psi_{\mathbf{p}}$, $\Omega = I - \Phi$.

4. Experimental results and analysis

The grayscale images from the standard test image library [14] were used in the experiments; the size of all the images was 512×512. The Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) were used as evaluation indices of image restoration quality. The greater their values were, the better the image restoration effect was achieved.

4.1. Experimental results

The simulation experiments included searching for neighboring sample patches in the local window, sample patch matching, applying the SSD (i.e., finding the minimum SSD among the neighboring sample patches in the local window) and PCA (Algorithm 2) methods, and comparison of the obtained results with the restoration results in [6]. The PSNR and SSIM values obtained in the experiments are shown in Table 1.

Table 1

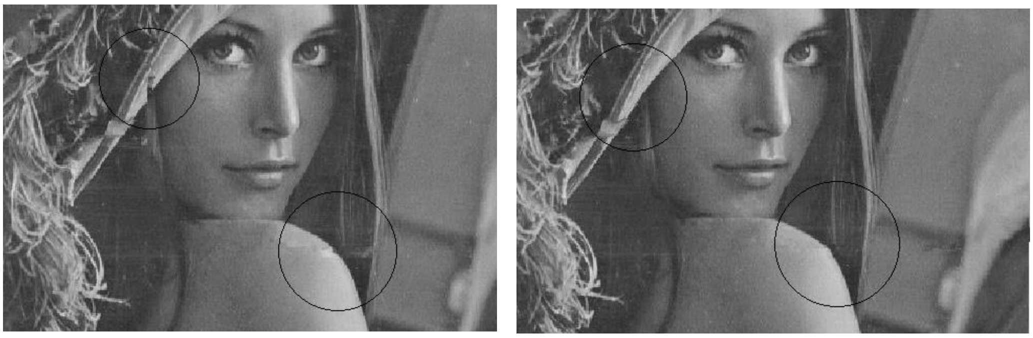
Grayscale images restoration comparison

Image	Sample patch radius R_Ψ	Local window radius R_W	Proposed method				Reference [6]	
			SSD		PCA			
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	4	20	40.15	0.9759	41.51	0.9763	37.76	0.9762
		24	40.28	0.9764	41.41	0.9769		
Bara		20	38.66	0.9869	39.92	0.9871	37.68	0.9866
		24	39.21	0.9872	40.19	0.9876		

The neighboring sample patch represents a subset of the global sample patch, but according to the values presented in Table, the overall image restoration effect achieved by neighboring patch matching was better than that of the global patch matching [2,6]. The PSNR and SSIM values in Table 1 show that the PCA performed better than the SSD; thus, the proposed method effectively avoided mismatching.

The image restoration results of the Lena image are shown in Fig. 6, where it can be seen that by using the proposed method, good continuity brim lines in the shoulder area without an obvious sense of hierarchy are achieved, and the hair region has no obvious repairing marks; thus, the achieved image restoration effect is excellent.

(a) image to be repaired (b) reference[6] (c)proposed method

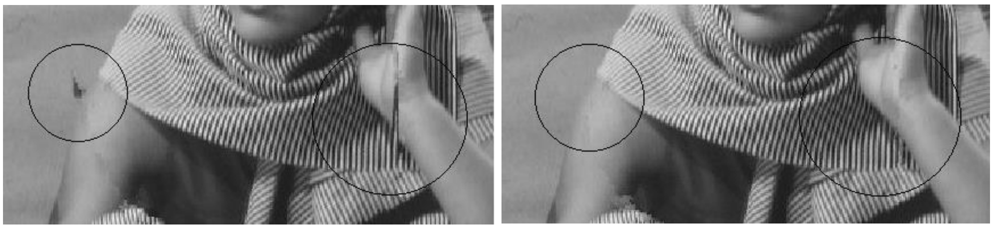


(d)enlarged view of the results of reference[6] (e) enlarged view of the results of the proposed method

Fig. 6. The image restoration results of the Lena image

The image restoration results of the Barb image are presented in Fig. 7, where it can be seen that when the proposed method is used, in the arm and wrist areas, there are no obvious mismatch and wrong structure; the repaired image looks relatively natural.

(a) image to be repaired (b) reference[6] (c) proposed method

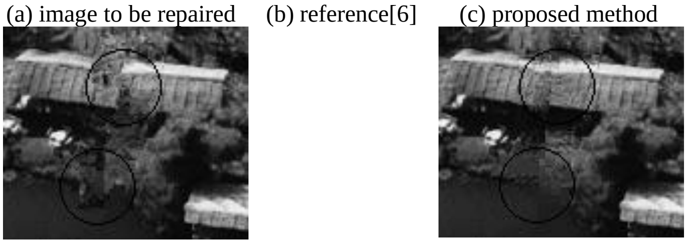


(d) enlarged view of the results of reference[6] (e) enlarged view of the results of the proposed method

Fig. 7. The image restoration results of the Barb image

The proposed algorithm successfully restored the Bungee image, as shown in Fig. 8. As displayed in Fig. 8(e), the roof lines are coherent, there is no fragmentation, and the local fusion degree is high. However, in the center of the repaired area, the results of all the algorithms have obvious segmentation lines.

(a) image to be repaired (b) reference[6] (c) proposed method



(d) enlarged view of the results of reference[6] (e) enlarged view of the results of the proposed method

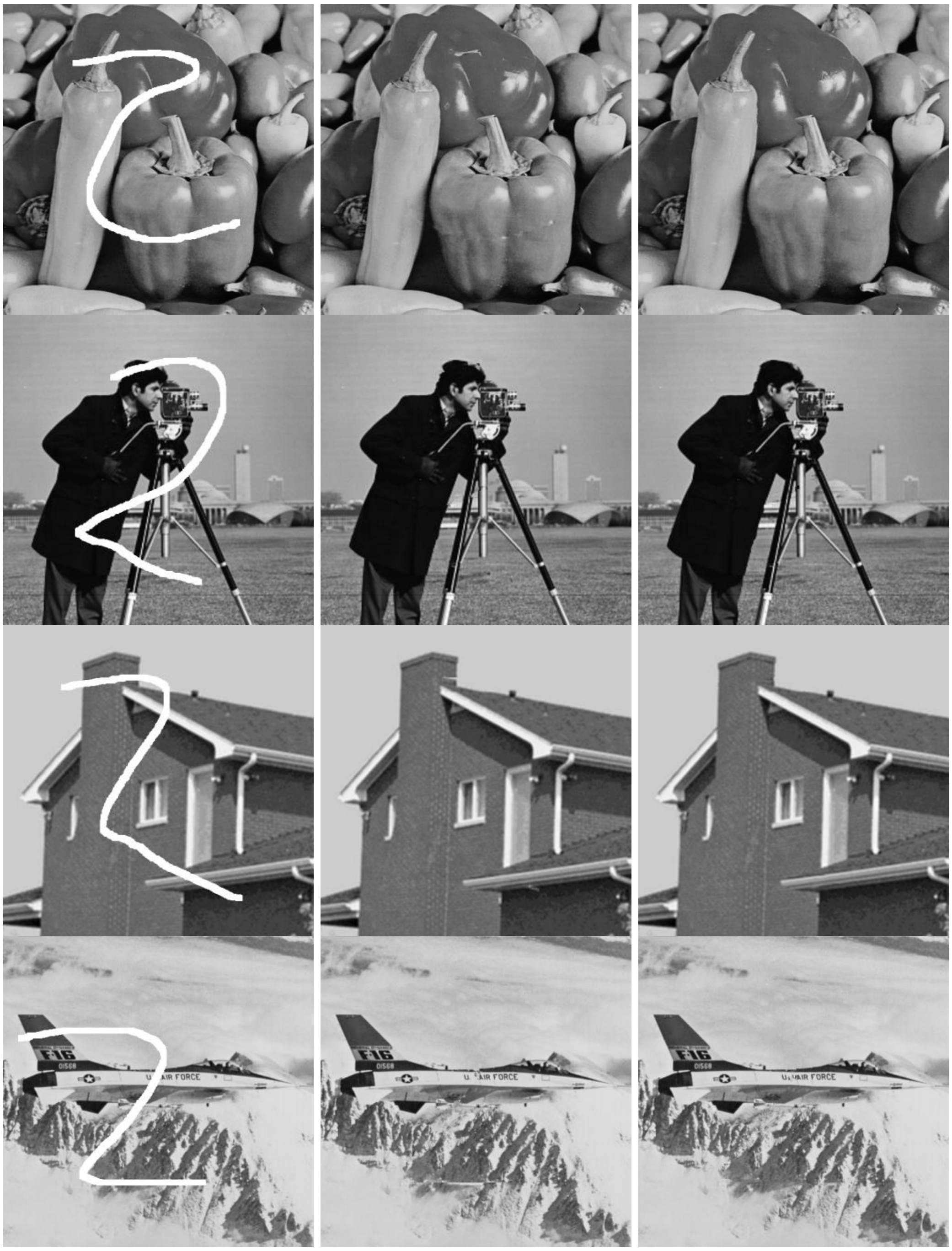
Fig. 8. The image restoration results of the Bungee image

More grayscale images restoration results are shown in Table 2 and Fig. 9.

Table 2

More grayscale image restoration results

Image	Proposed method		Reference[6]	
	PSNR	SSIM	PSNR	SSIM
peppers	41.0336	0.9675	37.8102	0.9665
cameraman	37.1573	0.9703	35.7910	0.9705
house	46.2056	0.9769	42.5529	0.9778
jetplane	34.3152	0.9721	32.6661	0.9707



(a) image to be repaired (b) reference[6] (c) proposed method

Fig. 9. More grayscale images restoration results

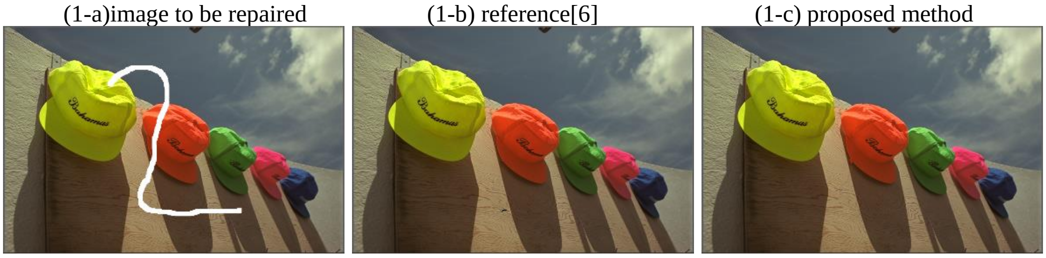
The proposed algorithm was further tested on the color images. Test images were selected from the Kodak database (Kodak24), and the method presented in [6], was used for comparison. The PSNR and SSIM values of both the methods are shown in Table 3, and the restoration results are shown in Fig. 10.

Table 3

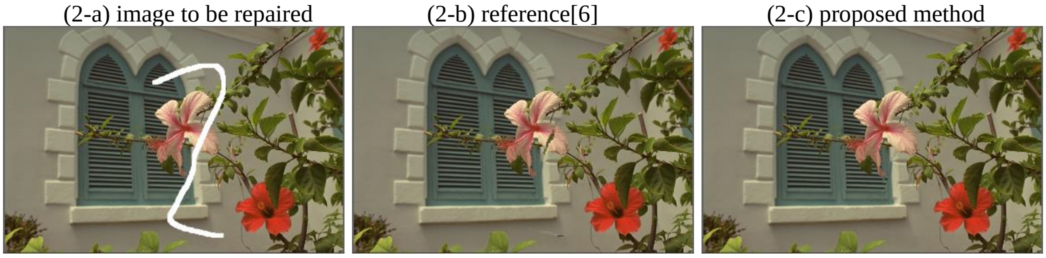
Color image restoration results

Image	Proposed method		Reference [6]	
	PSNR	SSIM	PSNR	SSIM
1-a	43.2919	0.9786	42.5876	0.9780
2-a	44.1169	0.9828	43.4132	0.9839
3-a	39.1037	0.9799	38.7532	0.9807
4-a	40.9043	0.9852	36.9539	0.9850
5-a	39.3524	0.9873	37.6477	0.9873
6-a	37.4780	0.9857	37.0536	0.9857

(1-a)image to be repaired (1-b) reference[6] (1-c) proposed method

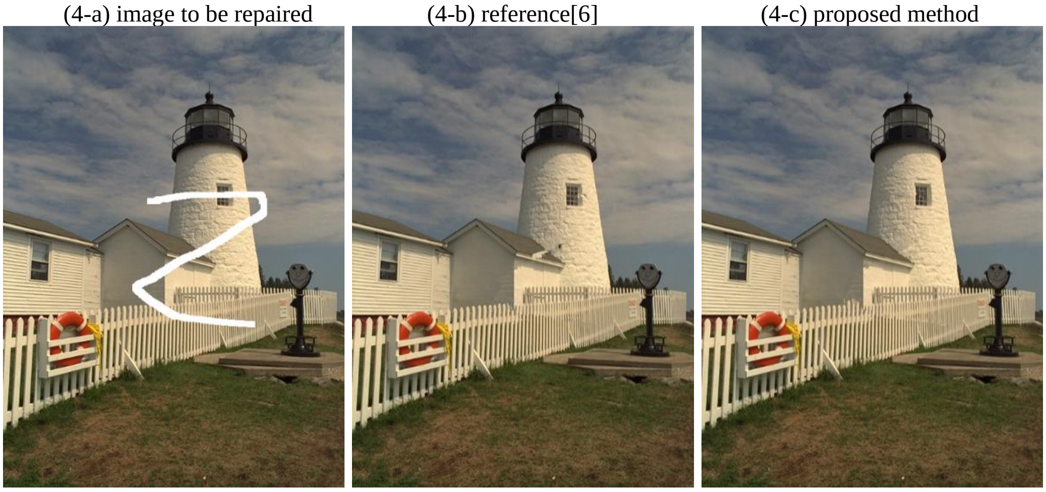


(2-a) image to be repaired (2-b) reference[6] (2-c) proposed method

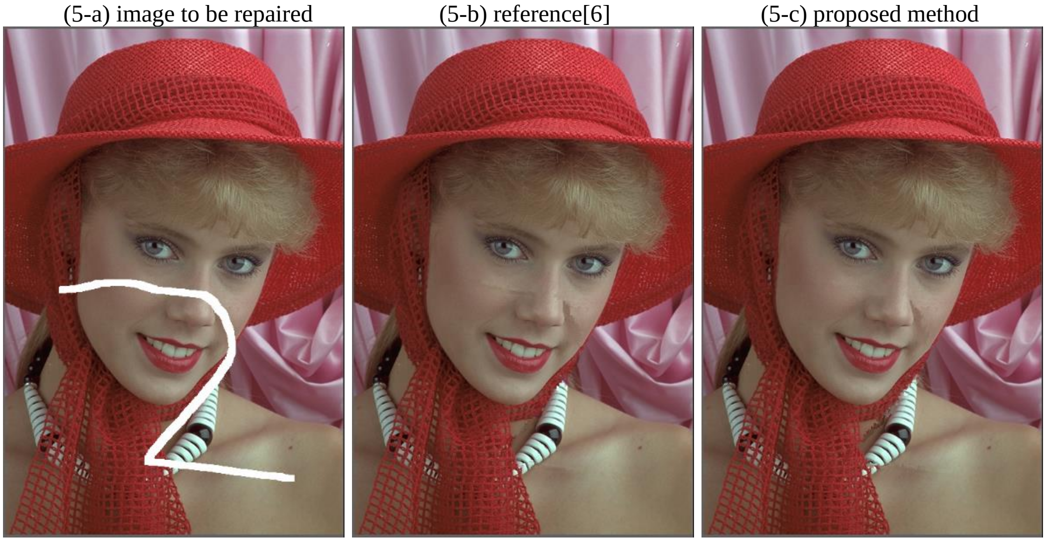


(3-a) image to be repaired (3-b) reference[6] (3-c) proposed method

(4-a) image to be repaired (4-b) reference[6] (4-c) proposed method



(5-a) image to be repaired (5-b) reference[6] (5-c) proposed method



(6-a) image to be repaired (6-b) reference[6] (6-c) proposed method

Fig. 10. The restoration results comparison of color images from Kodak24 database

In conclusion, the PSNR value of the proposed algorithm was significantly better than that of the original algorithm. The proposed algorithm achieved good continuity in image structure and texture, there was no obvious wrong structure diffusion, and the restored images were more natural and harmonious in a visual sense compared to the ones restored by the original algorithm.

4.2. Complexity analysis

4.2.1. Complexity analysis of exemplar-based algorithm

Assume the image size is $M \times N$, and the number of pixels to be repaired is K. If the repair area is rectangular, set the radius of a sample patch to R_Ψ, so the size of the sample patch is $(2R_\Psi + 1) \times (2R_\Psi + 1)$, and the average number of pixels for each repairing is at least $\frac{1}{4} \times (2R_\Psi)^2 = R_\Psi^2$; then, the complexity of the algorithm is given by:

$$\frac{K}{R_\Psi^2}(M - R_\Psi)(N - R_\Psi) \approx \frac{KMN}{R_\Psi^2} \tag{6}$$

4.2.2. Complexity analysis of PCA-based algorithm

Assume the local window size is $m \times n$; then, the complexity of each repair is $\gamma(m - R_\Psi)(n - R_\Psi)$, where γ represents a positive number, and the number of repairs is about $\frac{K}{R_\Psi^2}$; further, the overall complexity of the algorithm is given by:

$$\gamma \frac{K}{R_\Psi^2}(m - R_\Psi)(n - R_\Psi) \approx \frac{\gamma Kmn}{R_\Psi^2} \tag{7}$$

According to (6) and (7), the complexity of the proposed algorithm is related to the local window size $m \times n$ (or R_W), but there is no direct relation to the image size. Compared to the algorithms presented in [2,6], the proposed algorithm has a lower complexity. All simulation experiments were implemented by Matlab2015b on a PC with the 64-bit Win10 operating system, CPU Intel Core Duo E7500 2.93GHZ, and 8G memory. The obtained results are shown in Table 4.

Table 4

Computing time comparison of different methods

Image	Local window radius R_W	Computing time (s)	
		Proposed method	Reference [6]
Lena	20	23.31	44.92
	24	23.76	
Barb	20	23.19	44.32
	24	23.83	

4.3. Parameters analysis

The neighboring sample patch [7] is chosen to repair the image using the local correlation of the natural image [4,5]. The criterion for determining the local window size is that the local widow should provide enough information on the local structure and texture to the patch to be repaired. The size of the patch to be repaired should be the same as the size of the texture [2], and the local window should include as many textures or image structures as possible.

In order to ensure that sufficient local pixel information is provided, the known pixel rate of the repaired image should be greater than 90% [6]. Based on this assumption, the area ratio of the patch to be repaired and the local window should satisfy the following condition: $S(\Psi_{\mathbf{p}}):S(W_{\mathbf{p}})\le 0.05$. Therefore, the relationship between the local window radius R_W and the radius of a patch to be repaired R_Ψ can be expressed as:

$$R_W \ge 5R_\Psi \tag{8}$$

The neighboring sample patch matching reduces the algorithm's computational complexity. Since the neighboring sample patch set represents a subset of the global sample block, the SSD matching degree is not as good as the global one in theory. In order to verify the image restoration effect, this paper conducts an image restoration experiment using the neighboring sample patch matching at different local window radii. The image restoration results are presented in Fig. 11, where it can be seen that when the local window radius is about 5 to 8 times larger than the sample patch radius, the PSNR value of the results is larger, and the restoration effect is better; the experimental result is consistent with (8).

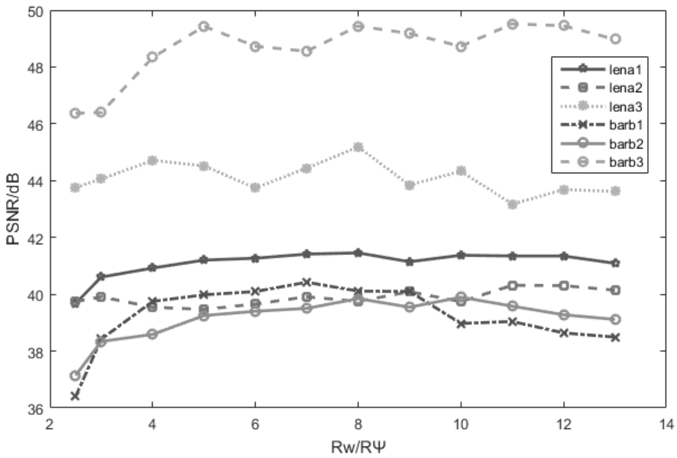


Fig. 11. The PSNR at different R_W / R_Ψ values

However, the overall image restoration effect has not improved with the larger local window radius, so it is reasonable to use the neighboring sample patch matching. In the simulation experiments, the sample patch size was 9×9, and R_Ψ was set to 4. According to (8), $R_W \geq 20$, thus, the local window size was at least 41×41.

In the PCA-based neighboring sample patch matching, the top 11 patches of the best-matched SSD were taken, so $k=11$. The size of the pixel value vector was reduced to one dimension point, and the sample patch corresponding to the median point was chosen. Although the reduction to one dimension was relatively rough, the excellent restoration effect was achieved.

(a) image to be repaired (b) post processed of reference[15] (c) proposed method

Fig. 12. The restoration results comparison with reference [15]

Recently, the machine learning-based method [15,16] proposes to encode the semantic context of an image into a latent feature space by using deep neural networks and then generate semantic-coherent patches by generative models. A simple inpainting region image restoration results comparison with reference [15] is shown in Fig. 12, compared with the machine learning-based method, the proposed method is simpler, and it achieves more in the single natural image restoration.

5. Conclusion

This paper proposes a fast image inpainting algorithm based on the PCA, which can extend the main lines of the known region and achieves excellent restoration effect for linear structures. The proposed algorithm has low calculation complexity, and it can also reduce the generation and diffusion of the wrong image structure caused by the sample patch mismatching.

The simulation results show that the computational complexity of the proposed algorithm is lower, and the visual effect of image restoration is better than those of the algorithm presented in [2,6]. When the texture difference between the two sides of the repaired area is too large, and the repair area is closed, there is a significant area segmentation line in the results of both the original algorithm and the proposed algorithm. Therefore, how to avoid the generation of region segmentation lines and construct more reasonable priority functions, and how to select a suitable sample patch, will be the research direction of our future work.

R E F E R E N C E S

[1]. *M. Bertalmio and G. Sapiro*, “Image Inpainting”, ACM SIGGRAPH’00, 2000.

[2]. *A. Criminisi, P.Perez and K. Toyama*, “RegionFilling and Object Removal by Exemplar-based Image Inpainting”, in IEEE Transactions on Image Processing, **vol. 13**, no. 9, Sept. 2004, pp. 1200-1212.

[3]. *A. Wong and J. Orchard*, “A nonlocal-means approach to exemplar-based inpainting”,in IEEE International Conference on Image Processing, 2008.

[4]. *H. M. Liu, W. L. Wang and H. Xie*, “ThangKa Image Inpainting Using Adjacent Information of Broken Area”, DOAJ, 2008.

[5]. *H. M. Liu, X. H. Bi, Z. F. Ye and W. L. Wang*, “Arc promoting image inpainting using exemplar searching and priority filling”,in Journal of Image and Graphics, **vol. 21**, no. 8, Aug. 2016, pp. 993-1003.

[6]. *V. K.Alilouand F. Yaghmaa*, “Introducing a New Fast Exemplar-based Inpainting Algorithm”, in Proceedings of the 22nd Iranian Conference on Electrical Engineering, Washington D.C.,USA: IEEE Press, May 2014, pp. 874-878.

[7]. *G. J. Xu and C. Y. Yan*, “Study on the Fast Image Restoration Algorithm with Structure Priority”,in Technical Bulletin, **vol.55**, no. 17, Dec. 2017, pp. 98-106.

[8]. *J. Sun and K. He*, “Computing nearest-neighbor fields via Propagation-Assisted KD-Trees”, IEEE Conference on Computer Vision & Pattern Recognition, IEEE Computer Society, 2012.

[9]. *X. J. Liu, J. Yang, H. Liu and H. B. Shen*, “Affine Invariant Features Image Matching Approach Based on Principal Components Analysis”,in Journal of System Simulation, **vol. 20**, no. 4, Feb. 2008, pp. 977-980.

[10]. *H. Y. Wang and S. B. Gao*, “Saliency Detection Based on Fusion of Global and Local Features”, in Journal of Data Acquisition and Processing, **vol. 29**, no. 5, Sept. 2014, pp. 801-808.

[11]. *K. M. He and J. Sun*,“Image Completion Approaches Using the Statistics of Similar Patches”,in IEEE Transaction on Pattern Analysis and Machine Intelligence, **vol. 36**, no. 12, Dec. 2014, pp. 2423-2435.

[12]. *C. J. Xie and J. Q. Tan*, “Improved Method of Exemplar-based Inpainting”, in Journal of System Simulation,**vol. 20**, no. 10, May 2008, pp. 2606-2608.

[13]. *Z. K. Zhao, N. Z. Liu and W. Li*, “Image Inpainting Algorithm Based on Synthetic Sample Block by Gary Entropy”, in Computer Engineering, **vol. 42**, no. 11, Nov. 2016, pp. 267-271.

[14]http://imageprocessingplace.com/downloads_V3/root_downloads/image_databases/standard_test_images.zip

[15] *S. Iizuka E. Simo-Serra and H. Ishikawa,* "Globally and Locally Consistent Image Completion",ACM Transactions on Graphics (Proc. of SIGGRAPH),vol. 36, no. 4, Jul. 2017,pp.1-14.

[16]. *Y. H. Zeng, J. L. Fu, H. Y. Chao and B. N. Guo*, "Learning Pyramid-Context Encoder Network for High-Quality Image Inpainting", in CVPR, 2019.