

Thangka Image Inpainting Algorithm Based on Wavelet Transform and Structural Constraints

Fan Yao*

Abstract

The thangka image inpainting method based on wavelet transform is not ideal for contour curves when the high frequency information is repaired. In order to solve the problem, a new image inpainting algorithm is proposed based on edge structural constraints and wavelet transform coefficients. Firstly, a damaged thangka image is decomposed into low frequency subgraphs and high frequency subgraphs with different resolutions using wavelet transform. Then, the improved fast marching method is used to repair the low frequency subgraphs which represent structural information of the image. At the same time, for the high frequency subgraphs which represent textural information of the image, the extracted and repaired edge contour information is used to constrain structure inpainting in the proposed algorithm. Finally, the texture part is repaired using texture synthesis based on the wavelet coefficient characteristic of each subgraph. In this paper, the improved method is compared with the existing three methods. It is found that the improved method is superior to them in inpainting accuracy, especially in the case of contour curve. The experimental results show that the hierarchical method combined with structural constraints has a good effect on the edge damage of thangka images.

Keywords

Image Inpainting, Structural Constraint, Texture Synthesis, Wavelet Transform

1. Introduction

The image inpainting methods can be divided into structure and texture feature inpainting methods [1-3]. The texture feature inpainting methods can be further classified into decomposition-based [4] and texture-based synthesis [5]. In the image wavelet decomposition model, a specific way is used to decompose the damaged images. The images are decomposed into structural and textural information. The method based on variation and partial differential equation is used to repair the structural information, while the texture synthesis method is used to repair textural information. Then the subgraph reconstruction is completed. In actual simulation experiment, the texture subgraphs include less image energy when repairing textural information. The texture synthesis algorithm can easily cause false matching, and leads to a sharp deterioration in the repair effect, so the visual effect of final reconstructed image is undesirable.

In view of the image decomposition inpainting, many scholars done a lot of improvement research. Chen et al. [6] used TV decomposition model to decompose images into structure and texture subgraphs, and then the two parts were repaired according to their characteristics. The Poisson equation was used

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received May 31, 2019; first revision May 11, 2020; second revision June 11, 2020; accepted June 14, 2020.

Corresponding Author: Fan Yao (fannyao@xzmu.edu.cn)

* College of Information Engineering, Xizang Minzu University, Xian Yang, Shaanxi, China (fannyao@xzmu.edu.cn)

for structure subgraph, and the texture block synthesis was used for texture subgraph, but the inpainting effect was not very ideal for the large scale. Wang et al. [7] presented a new technique for image inpainting. The damaged area was roughly repaired by fast marching inpainting method in spatial domain. Zhang and Dai [8] decomposed image into texture subgraph and structure subgraph based on wavelet decomposition, then reconstructed each one of these functions separately with the CDD algorithm and the improved texture synthesis algorithm, and a good repair effect can be obtained. Deshmukh and Mukherji [9] separated the damaged image into two principal components which encompass image texture and color. The fast texture image completion algorithm based on dependencies was presented by He et al. [10]. Xiao and Zhang [11] proposed a fast inpainting algorithm for texture image by determining the order of filling blocks in wavelet domain, and filled the inpainting blocks by using texture synthesis method. But it could not keep connectivity about the edge information of texture subgraphs.

In order to reduce mismatch and overcome the defect that the texture is superior to the edge in repairing high frequency subgraph, the advantages of image decomposition restoration algorithm and wavelet transform domain coefficient characteristics will be made full use of in this paper. By introducing edge change factor and improving matching block search method, an image inpainting algorithm combining wavelet transform with texture synthesis will be discussed in detail.

The rest of this paper is organized as follows. The proposed method will be described in Sections 2–4. In Section 5, some experimental results will be presented. Finally, the conclusion will be drawn in Section 6.

2. Our Proposed Method

At present, the basic principle of image inpainting method which uses wavelet transform is as below [12]: first, the image is decomposed into structure and texture through wavelet transform; then an appropriate method is used to repair each part respectively according to its information characteristics. Due to the fact that the wavelet coefficient energy is relatively concentrated in low frequency subgraphs, the coefficient change has a relatively large influence on the reconstructed image. The wavelet coefficient energy is scattered in high frequency subgraphs; that is to say the matrix is sparse. Therefore, different inpainting methods are put forward according to various matrix characteristics.

Based on the above analysis, the principle of the image restoration algorithm based on wavelet transform and texture synthesis was discussed in this paper. Wavelet transform was adopted to decompose the damaged image into low frequency subgraphs and high frequency subgraphs with different resolutions. Considering the low frequency component represented the structural part of the image, while the high frequency component reflected the change of the image edge, the low frequency position and high frequency position had certain relationship. The low frequency subgraphs were restored by the improved fast marching algorithm. According to the coefficient characteristics, the high frequency subgraphs were restored by the algorithm of texture synthesis. Finally the subgraphs after image inpainting were reconstructed.

The thangka image inpainting algorithm that combined wavelet transform with structural constraints was shown in Fig. 1. The algorithm implementation steps were as follows:

- (1) J -level wavelet transform of the original thangka image was done (Note the higher the wavelet transform series was, the more concentrated the low frequency subgraphs were. The restoration

errors of low frequency subgraphs would have a great impact on the subsequent restoration at this time. Therefore, the wavelet transform series of image $J=2-3$ were proposed.)

(2) Subgraph inpainting

- For the low frequency subgraphs, the improved FMM was used.
- For the high frequency subgraphs, first the edge contour was extracted, and the contour underwent curve fitting. Then the improved texture synthesis algorithm was used to restore the edge and texture region successively.

(3) The first level wavelet reconstruction was carried out, and then step (2) was performed until the complete restoration image was obtained.

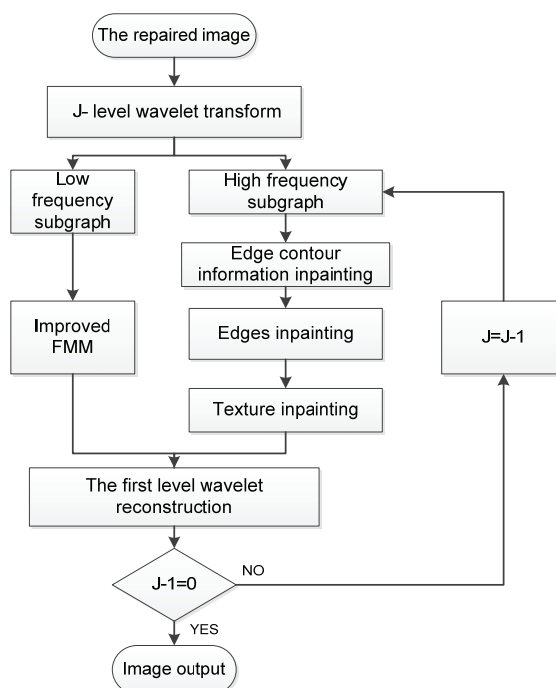


Fig. 1. The flow chart of the proposed inpainting algorithm.

3. Low Frequency Subgraph Inpainting

Telea [13] proposed a fast marching method (FMM) image inpainting algorithm in 2004. Based on the arrival time function $T(x, y)$, the algorithm aims to construct the curve evolution and FMM is used to restore the edge. Although FMM image inpainting algorithm can quickly restore damaged image and has good restoration effect, the weighting function of three factors has certain limitations, and some irrelevant information is introduced to the restoration process. As a result, the quality of restoration is undermined. To overcome the concentration of gradient information in the neighborhood, the main texture direction information was introduced and the inpainting template size was adaptively selected in this paper. The inpainting process was conducted according to the main texture direction in the neighborhood, so as to better maintain texture and edge direction.

In order to solve the arrival time function, the normal velocity β was introduced, with $\beta(x, y)$ as the normal velocity of the point (x, y) . Because the restore process was always done from edge to internal restoration area, $\beta(x, y)$ was always larger than zero [7]. In the process of curve evolution, the curve reached a point at most once. Next, $T(x, y)$ could be obtained through Eikonal equation $|\nabla T|\beta = 1$:

$$\left\{ [\max(D_{\bar{x}}^- T_{ij}, -D_{\bar{x}}^+ T_{ij}, 0)]^2 + [\max(D_{\bar{y}}^- T_{ij}, -D_{\bar{y}}^+ T_{ij}, 0)]^2 \right\}^{1/2} = \frac{1}{\beta_{ij}} \tag{1}$$

In Eq. (1), $\beta_{ij} = 1$, the finite difference scheme of $D_{\bar{x}}^- T_{ij}$ was $D_{\bar{x}}^- T_{ij} = T_{i,j} - T_{i-1,j}$, and the finite difference scheme of $D_{\bar{x}}^+ T_{ij}$ was $D_{\bar{x}}^+ T_{ij} = T_{i+1,j} - T_{i,j}$. Similarly, $D_{\bar{y}}^- T_{ij}$ and $D_{\bar{y}}^+ T_{ij}$ could also be secured. In this way, T of each pixel point to the edge could be calculated by Eq. (1).

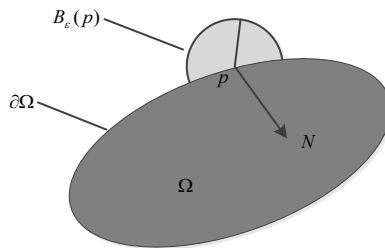


Fig. 2. Improved FMM principle.

To correctly select an order path, a mathematical model should be set up for each pixel. As shown in Fig. 2, the point p was the restored point close to $\partial\Omega$, and $B_\epsilon(p)$ was a small region around p . At this time, every known pixel point in the neighborhood q had an impact on p . Eq. (2) was added into the first-order approximation process.

$$I_q(p) = I(q) + \nabla I(q)(p - q) \tag{2}$$

$I_q(p)$ was the function of all known pixel point q about p in the neighborhood, and a normalized weighting function $\omega(p, q)$ should be located, which was used to measure the effect of each pixel point q on p , so as to estimate the inpainting value of point p :

$$I(p) = \frac{\sum_{q \in B_\epsilon(p)} \omega(p, q) [I(q) + \nabla I(q)(p - q)]}{\sum_{q \in B_\epsilon(p)} \omega(p, q)} \tag{3}$$

First, the gradient value and the direction of all pixels (outside the edge) were calculated to determine the main texture direction in the domain:

$$m(x, y) = \sqrt{[I(x + 1, y) - I(x - 1, y)]^2 + [I(x, y + 1) - I(x, y - 1)]^2} \tag{4}$$

$$\theta(x, y) = a \tan\{[I(x, y + 1) - I(x, y - 1)]/[I(x + 1, y) - I(x - 1, y)]\} \tag{5}$$

The gradient direction of 360° could be divided into 36 directions in Eq. (4); then the accumulative gradient value of each direction could be calculated. The direction of the maximum accumulative value

was taken as the main texture direction N_p in the domain. Given the isochromatic line was perpendicular to the gradient direction, the isochromatic line direction in the domain was N_p^\perp .

The weighted points were selected from the domain, and their directions were consistent with the main texture direction. Starting from point p to the known point q along N_p^\perp , the distance between the two points was $d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$, and the vector was $\overrightarrow{pq} = (x_p - x_q, y_p - y_q)$. When the isochromatic line direction was N_p^\perp , $D(p, q)$ was defined to satisfy certain condition, and then q could be added to the collection of the weighted points:

$$D(p, q) = \left| \frac{\overrightarrow{pq}}{d(p, q)} \cdot N_p^\perp \right| < \lambda \quad (6)$$

In Eq. (10), if $D(p, q)$ was smaller than the threshold λ , q was the weighted point. λ could not be set too small; otherwise there will be no weighted point in the main texture direction. According to the experiment, the threshold λ was generally taken as 0.1, and the window size could be adaptively expanded according to the number of weighted points.

Because the direction of q was determined, in order to avoid double-counting, the weighting function $\omega(p, q)$ was modified as:

$$\omega(p, q) = dst(p, q) \cdot lev(p, q) \quad (7)$$

Finally, the size of pixel value was calculated through Eqs. (3) and (7).

The improved FMM image restoration algorithm was implemented as follows:

- (1) Pre-processing: three matrices should be extracted from the known image, namely the distance matrix T of all known pixels and edge, the known pixel matrix I , and the flag bit matrix f . Next, the points were marked in the image as the following three sets:
 - KNOWN: the known pixels. The corresponding positions T and I were known; f was recorded as 0.
 - INSIDE: the unknown pixels to be restored in the region. The corresponding positions T and I were unknown; f was denoted as 255.
 - BAND: the pixel points on the damaged edge. The corresponding position I was known; T was recorded as 0 along the edge advancement during the restore process; and f was recorded as 1.
- (2) Initialization: in the matrix T , the values of the points marked KNOWN and BAND were 0, namely $T = 0$. The points of INSIDE were marked to be a larger number, namely $T = 10^6$. 0, 1, and 255 were set according to the above set points in matrix f . Finally, the coordinates of the point marked as BAND were placed in the new edge stack.
- (3) Inpainting: the value T of the points in the stack was calculated and put in order of numerical size. The minimum value was selected to be set P_{min} , marked as KNOWN and deleted from the stack. Its four neighborhood points were traversed. When there was an INSIDE point, it was marked as BAND and the pixel value of the point was calculated by Eq. (3). If there was a BAND point, the point was retained in BAND. The value of BAND was updated in the neighborhood using Eq. (1). When the edge stack was not empty, the point was reselected. When the edge stack was empty, namely there was no unknown pixel point, the image was restored completely and the loop ended.

4. High Frequency Subgraph Inpainting

4.1 Edge Contour Information Extraction

The precondition of contour restoration was edge information extraction. The high frequency subgraphs reflected the edge information of image, so the edge contour information could be directly extracted from high frequency subgraphs. Through observation, it was found that the high frequency coefficient with large absolute value represented the contour line of image. Therefore, the threshold value could be set to judge whether the coefficient was an edge information. The high frequency components in different directions represented the marginal changes in the different directions of original thangka image.

The position information of three high frequency subgraphs corresponded to each other on the same layer. Therefore, the whole edge contour information could be combined after the edge extraction of the components in different directions. The binary matrix of edge contour information was denoted as B , whose size was the same as the subgraph in this layer. The points on the edge contour line were denoted as "1", while the points on other areas were denoted as "0". The threshold value σ was set, in order to determine whether each coefficient in the high frequency subgraphs represented edge information. If the absolute value of high frequency coefficient was larger than σ , the value of the corresponding position in the binary image was 1, or otherwise it was 0. A binary image representing edge contour of this layer could be obtained. The value of coordinates (m, n) in the binary image B was defined as:

$$B_j(m, n) = \begin{cases} 1, & |W_j^b| \geq \sigma \\ 0, & |W_j^b| < \sigma \end{cases} \quad b \in (LH_j, HL_j, HH_j) \quad (8)$$

$|W_j^b|$ was the absolute value of coefficients in j layer high frequency subgraph (LH_j, HL_j, HH_j). σ was the threshold, whose size could be determined according to the strength information of actual damaged edges to ensure the extraction of the edges of the damaged area.

4.2 Edge Contour Information Inpainting

After the information extraction of edge contour, the edge contour information was restored. The specific implementation process was as follows: firstly the intersection of the contour line and the boundary to be restored was set as a collection $P = \{p_0, p_1, \dots, p_i, \dots, p_n\} \quad 0 \leq i \leq n$ (as shown in Fig. 3(c)) after the edge information was extracted from the contour of defects (as shown in Fig. 3(b)), and the contour line of p_i was C_i . In order to make these intersections match orderly, two rules should be established:

- (1) The connection between the two points must pass through the area to be restored.
- (2) Any connection line should not intersect.

It was assumed that only if the directions of the two curves to be fitted were nearly opposite, the two curves were selected for matching. To meet the first rule, P was divided into two subsets P_1 and P_2 , as shown in Fig. 3(d). The curve directions of the points were almost identical in the same subset, and the curve directions of the points were almost inverse in different subsets. Based on the above hypothesis, the angles could be calculated between every two curves to be restored. To find out the minimum value of these angles: P was divided into two subsets, and the direction of each point on the curve C was fixed

in collection P . In binary image B , along curve C_i starting from p_i , the point $p_{i,j}$ ($j=1,2,3,4$) marked as "1" was found in its 8-neighborhood; $p_{i,1} = p_i$. If $\vec{p}_{i,j} = (x(p_{i,j}), y(p_{i,j}))$ was the coordinate vector of $p_{i,j}$, the direction of curve C_i was defined as:

$$\vec{S}_{p_i} = 0.5(\vec{p}_{i,2} - \vec{p}_{i,1}) + 0.35(\vec{p}_{i,3} - \vec{p}_{i,1}) + 0.15(\vec{p}_{i,4} - \vec{p}_{i,1}) \tag{9}$$

In Eq. (13), \vec{S}_{p_i} was the direction vector of a point on the curve, and then this vector was normalized:

$$\vec{S}'_{p_i} = \frac{\vec{S}_{p_i}}{|\vec{S}_{p_i}|} \tag{10}$$

The unit direction vectors of all points could be calculated in collection P . Any point p_i was put in subset P_1 , and the vector angle cosine of point p_i and other points p_m was calculated:

$$\cos \theta = \frac{\langle \vec{S}'_{p_i}, \vec{S}'_{p_m} \rangle}{\langle \vec{S}'_{p_i}, \vec{S}'_{p_m} \rangle} \tag{11}$$

The resulting cosine values ranged from large to small, and then p_i was put into subset P_1 .

- (1) When the number of points on both sides was equal, n was an even number. The element number of subset P_1 was equal to $n/2$, the remaining points were put into subset P_2 , and the classification of P was completed.
- (2) When the number of points on both sides was unequal, n was an odd number (as shown in Fig. 3). When the number of subset P_1 was equal to $(n-1)/2$, the arrangement of the points was as follows:

$$p_m \in \begin{cases} P_1, \cos \theta \geq 0 \\ P_2, \cos \theta < 0 \end{cases} \tag{12}$$

Finally, the rest of the points were placed in subset P_2 , and the classification of the collection was completed.

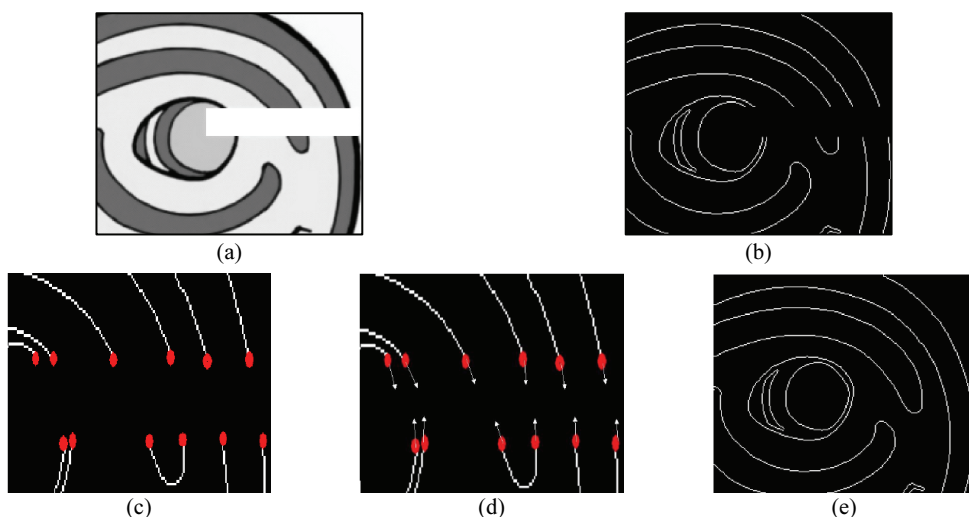


Fig. 3. Restored the edge information: (a) original image, (b) the restored edges information, (c) intersection of the repaired edges, (d) direction vector of each point, and (e) edge information of this algorithm.

In order to meet the second criterion, the shortest Euclidean distance of all points in two subsets was identified, and two nearest points were used in curve fitting of quadratic polynomial. The points corresponding to the fitting curve were marked as “1” in B ; then the two points were removed from subsets P_1 and P_2 . All points were fitted in the similar way, as shown in Fig. 3(e). In the fitting process, the following settings were built for different situations:

- (1) When n was an even number, $n(P_1) \neq n(P_2)$, and every two points were fitted until all points were connected.
- (2) When n was an odd number, $n(P_1) \neq n(P_2)$, and $n(P_1) = n(P_2) + 1$ (other situations were handled in the similar way). As shown in Fig. 4, midpoint p_1 and its nearest point p_2 in subset P_1 were taken, and the shortest distance between point p_5 and p_1 was located in the corresponding subset P_2 . If $\|\vec{p}_{1,2} - \vec{p}_{2,2}\| < \|\vec{p}_{1,1} - \vec{p}_{2,1}\|$, the two points were in opposite directions. Then p_1 and p_5 were connected, and p_2 was retained in subset P_1 . Otherwise p_1 and p_5 , p_2 and p_6 at the same time were connected.

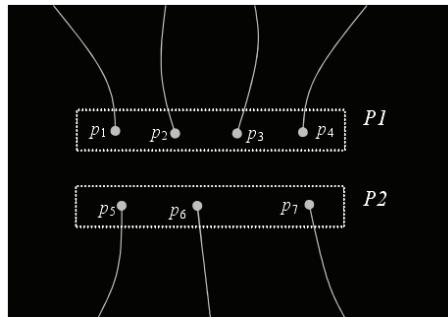


Fig. 4. $n+1$ point to n point.

4.3 Edges Inpainting

When the edge contour information of the wavelet coefficients was restored, it could be used as the restoration constraint condition of this layer subgraph. The edge information of the subgraph enjoyed better smoothness due to the use of defined priority of the wavelet coefficient. The inpainting should be carried out in accordance with the direction of edge, and the restoration effect was better than traditional isophote [14-17]. However, only the improvement in priority was not enough to realize the restoration along the actual edge. In order to further optimize the restoration of edge part, the inpainting order should be optimized on the basis of texture synthesis. The binary image B' of the restored edge contour information was used as a constraint condition. When the restoration in the region was to be repaired and the point in B' was marked as 1, the priority was to synthesize the texture block centered on the point. In this way, the curve matching along edge contour information was restored. The edge information in the high frequency subgraphs was repaired, taking priority over other parts to overcome the fact that the edges were not connected.

4.4 Texture Inpainting

In priority function, the edge change factor $E(p)$ was introduced, and degree $C'(p)$ of confidence was improved. The synthetic process could be carried out according to the strongest direction along the edge,

so as to prevent the texture portion in the area.

The priority factor was redefined as:

$$P(p) = C'(p) \cdot D(p) + E(p) \quad (13)$$

$$D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha} \quad (14)$$

∇I_p^\perp was the isophote of point p . n_p was the unit normal vector of the boundary line of point p . α was the normalization factor. $E(p)$ was the edge change factor, whose size reflected the intensity of edge changes to be repaired in different directions and regions. The absolute value of the high frequency coefficient reflected the intensity of edge in different directions, so the factor of edge change $E(p)$ was defined as:

$$E(p) = \frac{\max\{\sum_{\Psi_p \cap \Phi} \|W_{LH}^j\|, \sum_{\Psi_p \cap \Phi} \|W_{HL}^j\|, \sum_{\Psi_p \cap \Phi} \|W_{HH}^j\|\}}{\alpha} \quad (15)$$

In layer j , $\sum_{\Psi_p \cap \Phi} \|W^j\|$ represented the sum of absolute values in the known high frequency coefficients, and α was the normalization factor reaching 255. Since the different high frequency subgraphs represented the edge properties in different directions, confidence $C'(p)$ was defined as:

$$C'(p) = \frac{\sum_{\Psi_p \cap \Phi} N_b}{|\Psi_p|}, \quad b \in (LH, HL, HH) \quad (16)$$

$\sum_{\Psi_p \cap \Phi} N_b$ was the number of effective pixels in different directions, and $|\Psi_p|$ was the pixel number of the block to be repaired.

In search of a matching block, the prediction difference value was introduced from the very beginning to reduce the defects caused by poor matching in the restoration process. The high layer subgraph was used to constrain the inpainting of sublayer subgraph, and to better constrain the search of the restoration sequence and the best matching block. The inpainting order depended not only on the size of priority, but also on whether the matching block was the optimal. In terms of the restoration order of subgraphs, the low frequency subgraph on the highest layer was repaired first, the high frequency subgraph was repaired next, and the high frequency subgraph of sublayer was repaired last.

5. Experiment and Analysis

In order to verify the effectiveness of the designed method, a contrast experiment was conducted using the traditional criminisi algorithm [18], two improved inpainting algorithm [19,20], and this method respectively. The hardware environment of this experiment was Intel Core i7-8550u, 1.99 GHz and 8 GB internal storage, and the software environment was the MATLAB 2014a under Windows 7 operating system. Haar wavelet was applied in this paper, with wavelet decomposition level $j = 2$.

5.1 Image Inpainting

To illustrate the effectiveness of these algorithms on thangka image inpainting, the thangka images with rich texture features and complex structural information were selected for testing in this paper.

Figs. 5(a), 6(a), and 7(a) demonstrated the original images, and Figs. 5(b), 6(b), and 7(b) displayed the artificially broken images. Figs. 5(c), 6(c), and 7(c) were the experiment results based on the traditional criminisi algorithm. As could be seen from the inpainting results, the surrounding texture of the damaged area diffused into the damaged area, eventually filling the damaged area. A relatively messy texture could be clearly seen. Figs. 5(d), 6(d), and 7(d) were the experiment results based on literature [19], and Figs. 5(e), 6(e), and 7(e) were the experiment results based on literature [21].



Fig. 5. Inpainting results for image_1: (a) original image, (b) broken image, (c) algorithm of [18], (d) algorithm of [19], (e) algorithm of [20], and (f) the proposed algorithm.

As could be seen from the inpainting results, both edge closing degree and smoothness of the latter were obviously superior to those of the criminisi algorithm, and the inpainting effect could also meet human visual needs. To some extent, the phenomenon of structure fault caused by error propagation could be overcome, but the smoothness of the structure was hardly maintained. Figs. 5(f), 6(f), and 7(f) were the inpainting results based on the method designed in this paper. It could be argued that its visual effect

outperformed the above algorithms, and that there was no obvious block matching error in the inpainting results. The restored area was uniformly distributed. The edge was smoother than the above algorithms, and there was not texture extension phenomenon in the strong edge area.



Fig. 6. Inpainting results for image 2: (a) original image, (b) broken image, (c) algorithm of [18], (d) algorithm of [19], (e) algorithm of [20], and (f) the proposed algorithm.

Table 1 showed the peak signal-to-noise ratio (PSNR) value and consumed time of the four algorithms in the experiment. As was shown in Table 1, PSNR of this algorithm increased 1–3 dB compared with other algorithms. However, the consumed time of this algorithm was longer than literatures [19] and [20]. The reason was that the area to be repaired “shrank” after wavelet decomposition, and the increased factor and the improved search way would take a long time when the high frequency subgraphs were repaired. As a result, this algorithm consumed more time.

Table 1. PSNR (dB) and consumed time (s) of four algorithms

	Image size	[18]		[19]		[20]		Proposed	
		Time (s)	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)	PSNR (dB)
Fig. 5	682×570	76.8459	36.4581	75.42897	37.4529	63.3889	37.7219	81.4596	39.1562
Fig. 6	400×350	61.0259	36.4893	59.6489	37.2772	51.6972	37.4293	62.0348	39.1579
Fig. 7	432×306	60.8547	36.1457	59.4586	37.6578	53.2497	37.1689	61.0245	39.0389

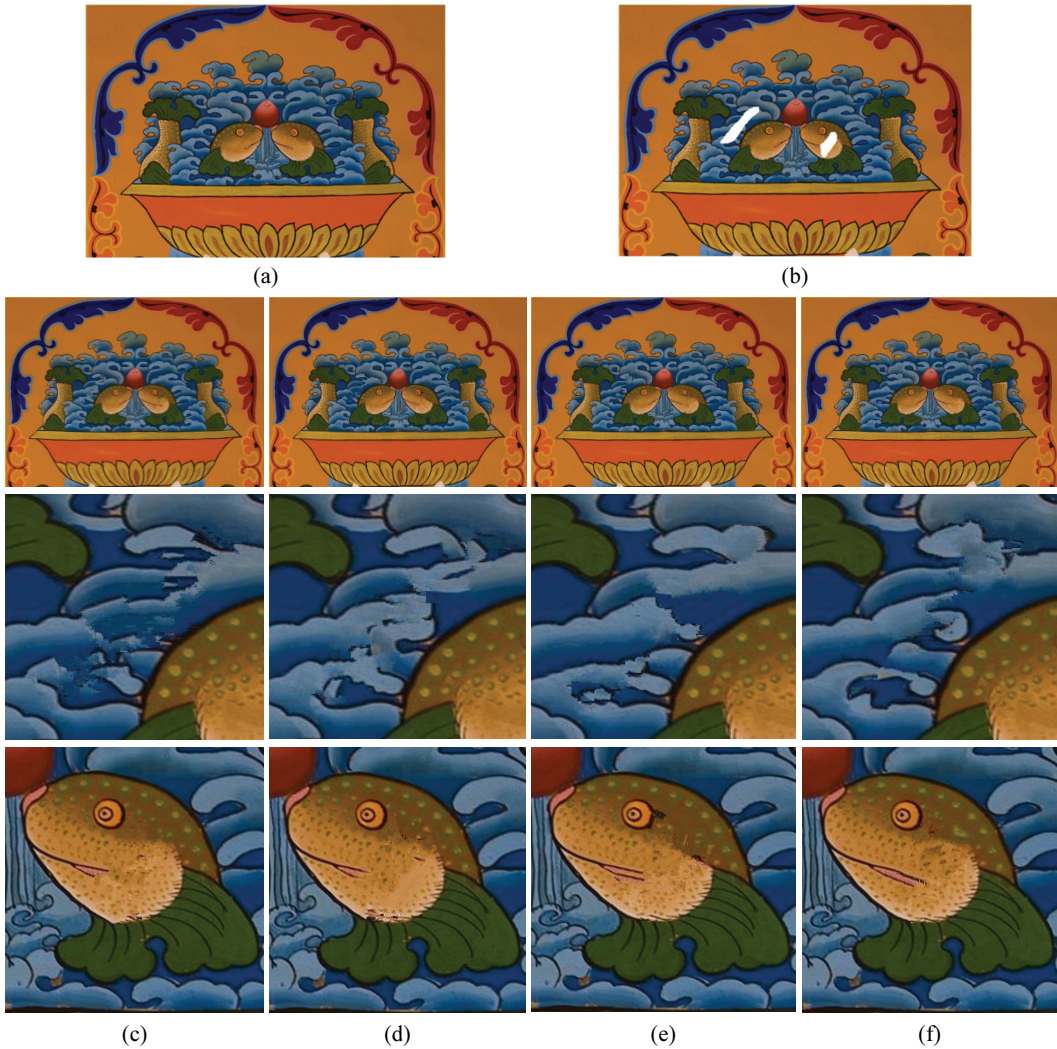


Fig. 7. Inpainting results for image_3: (a) original image, (b) broken image, (c) algorithm of [18], (d) algorithm of [19], (e) algorithm of [20], and (f) the proposed algorithm.

5.2 Effects of Different Base Wavelets on Inpainting

Fig. 8 indicated the restoration effect on the damaged thangka image (Fig. 8(b)) of Haar wavelet (a support length of 1) and biorthogonal Bior4.4 wavelet (a support length of 9) with the wavelet decomposition level at 2 and 4, respectively.

Judging from Fig. 8, the restoration effect of Haar wavelet outperformed that of Bior4.4 wavelet, and the restoration effect of level-2 wavelet decomposition was better than that of level-4 wavelet decomposition. This was because when these data were set to zero during the restoration process in the damaged area, an artificial boundary was generated around the restoration area. When the wavelet was decomposed, the artificial boundary was caused by extension effect, which was related to the support length of base wavelet. The shorter the support length of base wavelet was, the smaller the elongation effect became, and the more accurate the restoration of the damaged area became. Therefore, in order to

minimize the extension effect of the artificial boundary, the support length of base wavelet should be as short as possible. In addition, the wavelet decomposition series J was greater, and the low frequency energy was more concentrated. The low frequency subgraph restoration error would have great influence on the quality of the reconstructed image. The decomposition level J was preferably set at the level of 2 or 3, and they could respectively provide a subgraph of 1/4 or 1/8 of the original image. The decomposition level could basically meet the requirements of most restoration images.

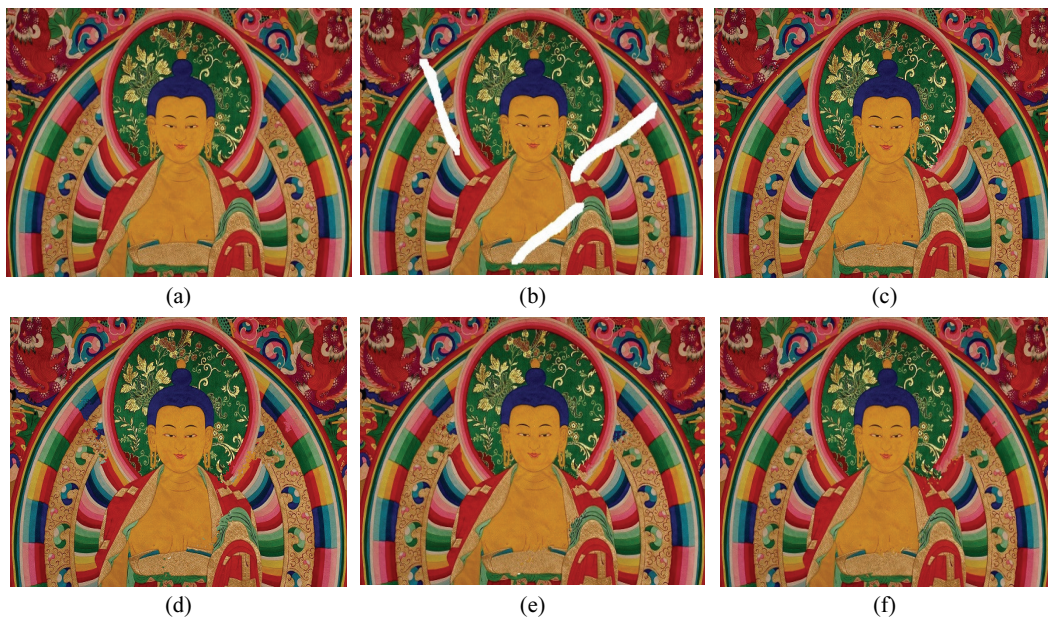


Fig. 8. Comparison of inpainting effects: (a) original image, (b) broken image, (c) level-2 decomposition of Harr (PSNR=39.2567 dB), (d) level-2 decomposition of Bior4.4 (PSNR=38.8153 dB), (e) level-4 decomposition of Harr (PSNR=37.9457 dB), and (f) level-4 decomposition of Bior4.4 (PSNR=39.0243 dB).

5.3 Damaged Image Inpainting

In order to verify the effectiveness of this algorithm, the repair experiments of some actually damaged thangka images were carried out on computer. As shown in Figs. 9 and 10, Figs. 9(a) and 10(a) were actually damaged thangka images, and Figs. 9(b) and 10(b) were the images of damage marks. Figs. 9(c) and 10(c) were the inpainting results using literature [18]. Mismatches resulted in defects in the inpainting results. Figs. 9(d) and 10(d) were the restoration results using literature [19]. There were bulges and fractures in the restoration results. Figs. 9(d) and 10(e) were the restoration results using literature [20]. Although the damaged edges could be connected, the smooth area was blurry. Figs. 9(f) and 10(f) were the restoration results using the method presented in this paper. The results based on this algorithm displayed original visual effect. This algorithm enjoyed good restoration effect in both smooth and edge parts.

Because Figs. 9 and 10 were both actually damaged images, their PSNR values could not be calculated. Objective judgment could only rely on the inpainting time of algorithms, and the repair time of the three algorithms was listed in Table 2.

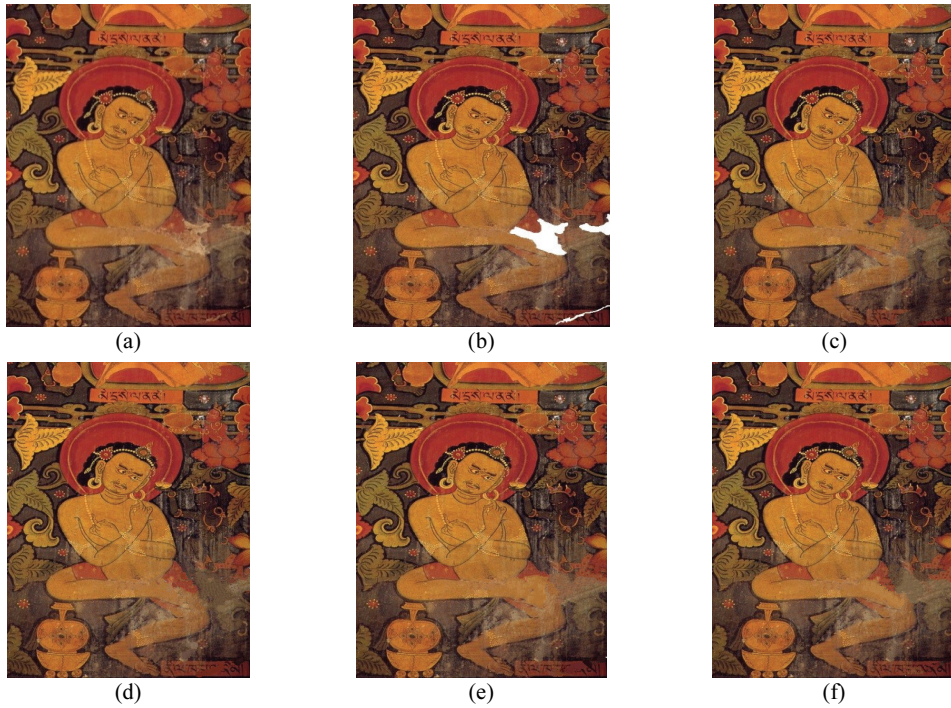


Fig. 9. Inpainting results for image_4: (a) original image, (b) marked image, (c) results by [18], (d) results by [19], (e) results by [20], and (f) results by the proposed algorithm.

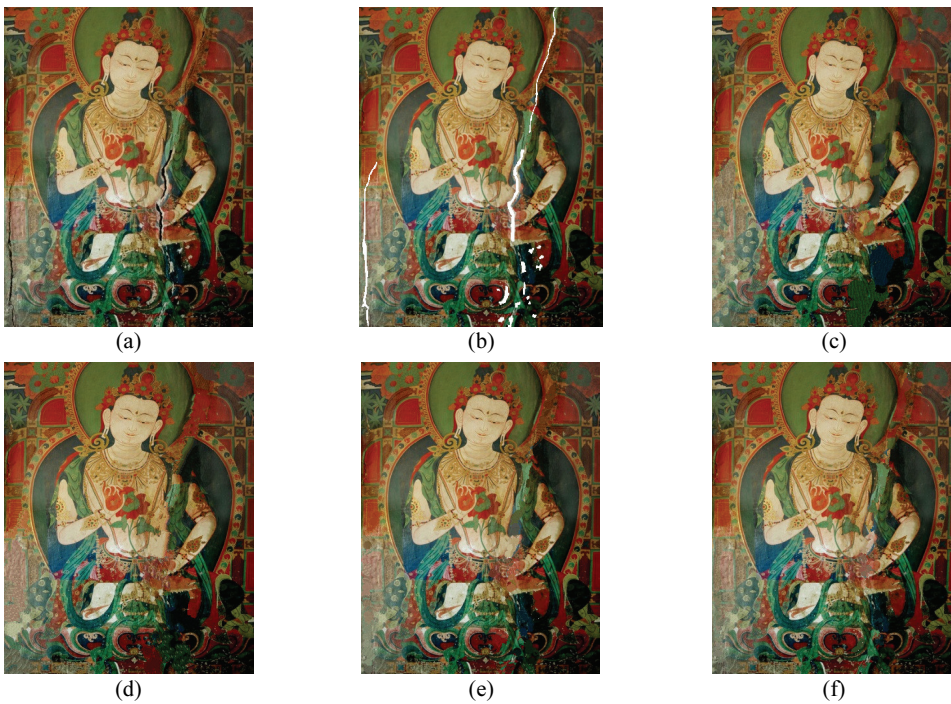


Fig. 10. Inpainting results for image_5: (a) original image, (b) marked image, (c) results by [18], (d) results by [19], (e) results by [20], and (f) results by the proposed algorithm.

It could be seen from Table 2 that the restoration time of this method was slightly longer than that of other algorithms. The reason was that when the high frequency subgraph was restored, the consumed time of the algorithm increased due to increased curve fitting, edge factor and improved search method.

Table 2. Inpainting time (s) of four algorithms

	Image size	Inpainting time (s)			
		[18]	[19]	[20]	Proposed
Fig. 9	654×845	176.83	179.45	178.56	220.87
Fig. 10	750×948	196.87	208.45	198.47	286.31

6. Conclusion

When some damaged edges are restored, it is difficult to take into account the structural integrity of an image and good visual effect on the basis of traditional algorithms. In this paper, an inpainting algorithm combining wavelet transform with structure constraint has been proposed for thangka images. Thangka images were decomposed into subgraphs with different resolutions and components by wavelet transform, and then the edge contour information was extracted and restored. Moreover, the coefficient relations of different components were organically combined with texture synthesis, and the edge contour information was extracted and repaired to constrain the repair of the high frequency subgraph structure. By introducing edge change factor and improving the matching block search method, the repair methods of layering and classification were adopted to enhance the ability to restore edges and textures. Experimental results showed that this algorithm could effectively restore damaged images with strong edges and rich texture, and had a better restoration effect which was more in line with the visual effect of human eyes. However, the algorithm took longer to restore images, and it was not ideal to restore large texture damaged images. Therefore, how to improve the restoration accuracy of low frequency subgraphs is still worth further discussion.

Acknowledgement

This work was supported by Scientific Research Planning Project of Shaanxi Province Educational Commission, China (No. 19JK0891), Major cultivation project of Xizang Minzu University, China (No. 19MDZ03), and Natural Science Foundation of China (No. 62062061).

References

- [1] S. T. Pohlmann, E. F. Harkness, C. J. Taylor, and S. M. Astley, "Evaluation of Kinect 3D sensor for healthcare imaging," *Journal of Medical and Biological Engineering*, vol. 36, no. 6, pp. 857-870, 2016.
- [2] M. Imtiyaz, A. Kumar, and S. Sreenivasulu, "Inpainting an image based on enhanced resolution," *Journal of Computer Technology & Applications*, vol. 6, no. 1, pp. 23-26, 2015.
- [3] S. Feng, R. Murray-Smith, and A. Ramsay, "Position stabilisation and lag reduction with Gaussian processes in sensor fusion system for user performance improvement," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 4, pp. 1167-1184, 2017.
- [4] F. Li and T. Zeng, "A new algorithm framework for image inpainting in transform domain," *SIAM Journal on Imaging Sciences*, vol. 9, no. 1, pp. 24-51, 2016.

- [5] F. Chen, T. Hu, L. Zuo, Z. Peng, G. Jiang, and M. Yu, "Depth map inpainting via sparse distortion model," *Digital Signal Processing*, vol. 58, pp. 93-101, 2016.
- [6] W. Chen, H. Yue, J. Wang, and X. Wu, "An improved edge detection algorithm for depth map inpainting," *Optics and Lasers in Engineering*, vol. 55, pp. 69-77, 2014.
- [7] F. Wang, D. Liang, N. Wang, Z. Cheng, and J. Tang, "An new method for image inpainting using wavelets," in *Proceedings of 2011 International Conference on Multimedia Technology*, Hangzhou, China, 2011, pp. 201-204.
- [8] H. Zhang and S. Dai, "Image inpainting based on wavelet decomposition," *Procedia Engineering*, vol. 29, pp. 3674-3678, 2012.
- [9] A. S. Deshmukh and P. Mukherji, "Image inpainting using multiresolution wavelet transform analysis," in *Proceedings of 2012 International Conference on Communication, Information & Computing Technology (ICCICT)*, Mumbai, India, 2012, pp. 1-6.
- [10] K. He, R. Liang, and T. Zhang, "Fast texture image completion algorithm based on dependencies between wavelet coefficients," *Journal of Tianjin University*, vol. 43, no. 12, pp. 1093-1097, 2010.
- [11] Z. Xiao and W. Zhang, "Wavelet-domain fast inpainting algorithm for texture image," *Chinese Journal of Scientific Instrument*, vol. 29, no. 7, pp. 1422-1425, 2008.
- [12] M. Xiao, G. Li, Y. Tan, and J. Qin, "Image completion using similarity analysis and transformation," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 4, pp. 193-204, 2015.
- [13] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23-34, 2004.
- [14] P. Buysseens, M. Daisy, D. Tschumperle, and O. Lezoray, "Exemplar-based inpainting: technical review and new heuristics for better geometric reconstructions," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1809-1824, 2015.
- [15] L. Cai and T. Kim, "Context-driven hybrid image inpainting," *IET Image Processing*, vol. 9, no. 10, pp. 866-873, 2015.
- [16] J. Patel and T. K. Sarode, "Exemplar based image inpainting with reduced search region," *International Journal of Computer Applications*, vol. 92, no. 12, pp. 27-33, 2014.
- [17] L. Tang, Y. Tan, and Z. Fang, and C. Xiang, and S. Chen, "An improved Criminisi image inpainting algorithm based on structure component and information entropy," *Journal of Optoelectronics: Laser*, vol. 28, no. 1, pp. 108-116, 2017.
- [18] H. M. Patel and H. L. Desai, "A review on design, implementation and performance analysis of the image inpainting technique based on TV model," *International Journal of Engineering Development and Research (IJEDR)*, vol. 2, no. 1, pp. 191-195, 2014.
- [19] H. Zhang and S. Dai, "Image inpainting based on wavelet decomposition," *Procedia Engineering*, vol. 29, pp. 3674-3678, 2012.
- [20] B. Wang, L. Hu, J. Cao, R. Xue, and G. Liu, "Image restoration based on sparse-optimal strategy in wavelet domain," *Acta Electronica Sinica*, vol. 44, no. 3, pp. 600-606, 2016.



Fan Yao <https://orcid.org/0000-0001-5982-7262>

She is working as an Associate Professor in Department of Information Engineering, Xizang Minzu University, Xian Yang, China. Her research interests include the digital protection of Tibet culture heritage and image processing.