平成２９年度 博士論文

Study on Image Inpainting Algorithms
(画像修復アルゴリズムに関する研究)

１４５９５４０３

範　謙

指導教員　張　力峰 准教授

九州工業大学大学院 工学府 電気電子工学専攻

Abstract

For many reasons, photos and videos stored on a digital or an analog media may be destroyed as several regions losing their pixels. Such destruction damages the integrity of the image, thus the digital image inpainting technology has been proposed to solve this problem. This paper proposes improved approaches on the image inpainting algorithm in three fields which contains diffusion-based field, exemplar-based field and dictionary learning based field. By using the remaining pixels information and the redundancy characteristic of the nature photos or artificial pictures, the lost pixels can be complemented by proposed method, and the repaired image gives a good visual effect.

In diffusion-based field, the traditional fast marching method(FMM) has a less calculation time but it is short for contributing an optimal edge result. In this method, FMM uses dir index dst index and lev index to calculate the gray value of the destroyed pixels. An improvement is adapted to the traditional FMM in this paper. The improvement contains two items. One is to change the dir index to improve the inpainting result. The improvement calculates the gradient information in the known region, and uses a pixel selecting method to select more significant pixels to join into the FMM calculation. The other one is to change the inpainting sequence to improve the edge effect. Firstly the proposed improvement detects the edges in the known region. Then use the gradient matrix of the detected edges to predict the edges in the inpainting region and cut the destroyed region into many sub regions. Finally inpaint the sub regions one after another. Through the experiment, the result is shown more reasonable and more similar to the natural scenes. Compared with the traditional FMM, the proposed approach not only remains the advantage of fast processing speed, but also contributes a better estimation result of edges.

Diffusion-based image inpainting method has a quite fast calculation speed in processing and gives a not bad global effect. These advantages make diffusion-based method quite suitable to inpaint the destroyed region which has a small width. But if the neighbor area of the destroyed region has complex textures, or the width of destroyed region becomes large, the inpainting visual effect will reduce. To inpaint this destroyed region, exemplar-based inpainting method is proposed. In exemplar-based field this paper proposed an improvement by changing the inpainting priority, the size of the patch and the patch matching method. First we optimize the priority function by calculating the priority not only focuses on one pixel but also on a patch. By computing the sum of the pixels'

priority in the patch, the patch containing more high priority pixels is to inpaint in the first place. Secondly, adaptive patch method is employed to change the patch size in the inpainting process. This method makes patches adaptively changed in the inpainting process to improve the patch matching accuracy. What' s more, this paper optimizes the patch matching method by adding the rotation invariance. By similarity calculation in different rotation situation the proposed method finally obtains a rotation invariant patch. This improvement makes the matching process contain more structure information and not be confined to the existing information. From the experimental result, it is shown obviously that the improvements make the inpainting match the more suitable patch. The proposed method gives a better global visual effect, especially for the images with more structure contents and the images in which the destroyed region has a large width.

In the traditional exemplar-based inpainting method, it uses the most similar patch to inpaint the destroyed region. But sometimes the destroyed pixels are distributed over images. Under this situation it is hard to match a suitable patch and the exemplar-based method will lead a bad inpainting effect. The dictionary learning based image inpainting method can inpaint the images while training the dictionary to learn from the source region. And it can provide information that does not exist in the source region. These two points are its most prominent advantages. In dictionary learning based field, the proposed method improves the traditional K means-Singular Value Decomposition(K-SVD) method by optimizing the sparse approximation process. To overcome the deficiency of the traditional Orthogonal Matching Pursuit(OMP) method, the proposed method uses an adaptive sparsity instead of the fixed one used in the traditional OMP method. First the method calculates the complexity of the patch in the sparse approximation; According to the value of the complexity, the method adaptively changed the sparsity to improve the sparse approximation result. The experimental results show the proposed method keeps more texture information in the image and obtains a higher PSNR.

In this study, the image inpainting has been strengthened in these three fields: For diffusion-based method, the inpainting effects when the image contains a large destroyed region. For exemplar-based method, the matching accuracy has been improved. For dictionary learning based method, the sparse approximation process has been optimized through the adaptive sparsity.

Contents

Chapter 1

Introduction

1.1 Background

For many reasons, photos and videos stored on a digital or an analog media may be destroyed as several regions losing their pixels. Such destruction damages the integrity of the image, thus the digital image inpainting technology has been proposed to solve this problem. By using the remaining pixels information and the redundancy characteristic of the nature photos or artificial pictures, the lost pixels can be complemented by inpainting algorithm, and the repaired image gives a good visual effect. At the beginning, image inpainting technology is used for repair the images destroyed in some reasons. Like Fig. 1.1 shows:

(a) Destoryed image. (b) Restored image.

Fig. 1.1. Image inpainting for destroyed wall painting.

Fig. 1.1 shows the application on mural restoration. This type of inpainting(Fig. 1.1(a)) often contains a destroyed region that the width of the destroyed region is small. By diffusion-based method it can obtain a quite good result (Fig. 1.1(b)).

Now image inpainting technology is extended widely to apply on other applications. Object removal is also a useful application (Fig. 1.2).

(a) Original picture. (b) Modified picture.

Fig. 1.2. Object removal.

In order to remove all or part of the text content, image inpainting technology sometimes gives a good visual effect. (Fig. 1.3)

(a) Original picture. (b) Modified picture.

Fig. 1.3. Words removal.

Fig. 1.4 shows the application that image inpainting technology applies on super-resolution analysis.

In a word, in the past years image inpainting technology has quite develop and according to the applications image inpainting technology rise into many branches. Each branch has its own advantages and areas of expertise.

(a) Original picture. (b) Processed picture.

Fig. 1.4. Super-resolution analysis.

1.2 Introduction of inpainting method

Image inpainting technology aims to fill the missing region in an image. According to the inpainting theory, it can be divided into two categories, diffusion-based and exemplar-based inpainting methods. Diffusion-based methods inpaint images on pixel-level, while exemplar-based methods inpaint images on patch-level. Image inpainting technology is used for repairing the images destroyed, but now it is extended widely to apply on object removal, image restoration, image compressing and other applications.

The most fundamental inpainting approaches are the diffusion-based inpainting algorithm. Bertalmio [1] firstly introduces the theory of image inpainting and gives this idea in 2000. Bertalmio proposes a model based on isophote to continuously propagate the undamaged information into the destroyed region. They further introduced the Navier-Strokes equation in fluid dynamics into the task of in-painting [2]. Chan and Shen [3] proposed a variational model based on total variation (TV) to repair the destroyed information. And they overcome the deficiency in the TV model that it can not realize the connectivity information by proposing a curvature-driven diffusion equation [4]. A joint interpolation of isophote directions and gray-levels was also designed to incorporate the principle of continuity in a variational framework [5]. Recently, image statistics learned from the natural images are applied to the task of image in-painting [6][7][8]. Image inpainting based on fast marching method (FMM) becomes a research hotspot after Telea' method was proposed [9]. FMM expands the traditional isophote model into three items (the direction, the distance, and the isophote influences) to compute the pixel' s gray value. The advantage of FMM in calculation speed and in inpainting large destroyed region influences other researchers. Yang [10] introduces the anisotropic diffusion theory into the FMM and uses it to improve the visual effect for inpainting the image which has

a large region .These methods mainly focuses on inpainting the missing areas on pixel-level and they performs a good global effect for the image with small width destroyed region. Another advantage of diffusion-based approach is that it has a fast calculation speed. Elad [11] proposes a method based on the image decomposition method called morphological component analysis (MCA). It is designed for the separation of linearly combined texture and cartoon layers in a given image. Ullo [12] employs the Complex Ginzburg-Landau equation to improve the contrast and applies it on restoring the SAR interferograms. Hu [13] proposes a method based on TV model to restore the image generated from the lateral multi-lens video logging device. Andris [14] applies inpainting method on image compressing field. The method develops a proof-of-concept codec which combines inpainting technology based on partial differential equations with the variational optic flow model. The diffusion-based in-painting algorithms have achieved convincingly excellent results for filling the non-textured or relatively smaller missing region. However, they tend to introduce the smooth effect in the textured region and larger missing region.

The second category approaches are the exemplar-based inpainting algorithm. These approaches propagate the image information from the known region into the missing region at the patch level. This idea stems from the texture synthesis technique proposed in [15], in which the texture is synthesized by sampling the most suitable patch from the known region. However, natural images are composed of structures and textures, the structures always constitute one image' s primal sketches (e.g., the edges, corners, etc.) and the textures are always represented the image regions with homogenous patterns or feature statistics (including the flat patterns). Due to Pure texture synthesis technique cannot clearly handle the missing region with composite textures and structures, Bertalmio et al. [16] proposed to firstly divided the image into structure and texture layers and then in-paint them by different method, using diffusion-based method to in-paint the structure layer and using texture synthesis technique to in-paint texture layer. It overcomes the smooth effect disadvantage brought from the diffusion-based in-painting algorithm; however, recovering the larger missing structures also is very hard to achieve. Criminisi et al. [17] designed an exemplar-based in-painting algorithm by propagating the known patches (i.e., exemplars) into the missing patches gradually. To handle the missing region with composite textures and structures, patch priority is defined to encourage the filling-in of patches on the structure. In [18],[19], multiple scales and orientations are introduced to find better matching patches. Barnes[20] proposed PatchMatch as a fast approximate nearest neighbor patch search algorithm. The proposed approach in [21] considered a simple exemplar-based model via global optimization. Wu [22] introduced the concept of cross-isophotes to the exemplar-based in-painting algorithm, in which he designed a cross-isophotes patch priority term based on the analysis of anisotropic diffusion. Wong [23] improved the exemplar-based in-painting algorithm by proposing a nonlocal means approach. The image patch is in-painted by not the most suitable patch but a set of

candidate patches in the known region. Ram [24] focused on the order of the inpainting, by reordering the patches it can obtain a good recovery of the image. In [25], a novel scheme was proposed to use both low-solution and high-solution information to get similar patches. Also a lot of researchers try to use structure information to matching the similar patches. Huang [26] proposed a method to use structure consistency to obtain the most similar patch. Discrete cosine transform [27] was also employed to calculate the similarities between patches, Ou [27] use it to select more significant patch. A lot of researchers also try to inpaint the image in multi-scale patches. Paredes [28] selects patches in multi-scale and calculate the local statistics of them. In this method they calculate the SURF-gradient descriptors in multi-scale patches and use the SURF-gradient descriptors to match the most similar patch. Reel [29] adds the multi-scale theory into the patch matching process, in this method they define the self-similarity in different scales, and segment an image into multiple depth layers, finally they filling the destroyed region by exploiting the self-similarity via nonlocal template-matching. Voronin [30] uses the using local binary patterns (LBP) method to match the patch in the image, LBP method strengthens the influence made by the structure. Martinez-Noriega [31] improves the method in [17] in calculation time. By optimize the priority and limited the matching process in a coherent nearest neighbor it save the calculation time and keep the global visual effect. Compared with diffusion-based in-painting algorithms, the exemplar-based in-painting methods expands the smallest inpainting unit from one pixel to one patch contains several pixels, they keep the texture in the local region and they are more suitable for in-painting the large missing region.

In the past years dictionary learning technology has become more and more mature, so that image inpainting researchers also employed it to optimize the inpainting method and improve the inpainting effect. It can be regard as an approach of exemplar-based inpainting algorithm, exemplar-based inpainting algorithm aims at finding the most similar patch to inpaint the destroyed patch and dictionary learning algorithm tries its best to generate a new patch to inpaint the destroyed one. Elad [32] proposes a sparse representation-based inpainting method. In this paper a dictionary learning method is used for creating a dictionary for sparse representations, via the singular value decomposition (SVD) approach. This method inpaints the images by iteratively alternating between sparse coding the image patches based on the current dictionary, and updating the atoms in the dictionary to better fit the coefficients. In this method Elad also employs orthogonal matching pursuit (OMP) algorithm to calculate the coefficients. This structure also affects a lot of researchers. Rubinstein [53] introduces a concept of double sparsity and use it improve the dictionary learning result. Xu [34] combines exemplar-based inpainting methods with dictionary learning-based inpainting methods after analysis the advantages and disadvantages of them. Xu uses ten most similar patches in the source image as the atoms of the dictionary. And employing the locally linear embedding (LLE) theory to obtain a linear

combination of the candidate patches, finally this method generates the new patch. Hesabi [35] improves the method by add gradient information into the data matrix so that when in the LLE calculation it obtains patches more similar in structure. Smith [36] optimizes the dictionary updating method to find both dictionary atoms and representations; this method employs the coefficient reuse orthogonal matching pursuit (CoefROMP) instead of OMP to obtain a fast coding speed. Ruzic [37] builds the dictionary by dividing the image into variable size blocks according to their context. So not only the patch information, but also their context information was used to improve the inpainting result. Papyan [38] improves the expected patch log likelihood (EPLL) algorithm by considering a multi-scale prior. This method calculates the priority of the patches in different scales and it can provide a smoother priority in inpainting. With the help of dictionary learning technology, there also comes out a lot of applications. Fadili [39] introduces a concept of expectation maximization (EM) method. The method is used in a Bayesian framework and a sparsity promoting prior penalty to improve the inpainting result. Sun [40] proposes a gradient-based discriminative learning method and designs a novel markov random field (MRF) framework to improve the inpaint effect. Wang [41] proposes a novel inpainting model carried on the dictionary learning by method of directions (MOD), and using it to repair missing region of murals in Potala Palace. Munawar [42] employs the higher-order Boltzmann machine and applies it on detecting anomalies on the road. This method tries to inpaint the road patches by training the commonly occurring road features such as lane markings and expansion dividers, depending on the context. Lu [43] applies inpainting technology on depth map reconstruction and proposes a Kinect-based underwater depth map estimation method. More recently, deep neural network is introduced for texture synthesis and image stylization [44, 45, 46, 47, 48, 49]. In particular, Phatak [50] trained an encoder-decoder CNN (Context Encoder) with combined L2 and adversarial loss [51] to directly predict missing image regions.

This paper research on image inpainting algorithm in the three fields contains diffusion-based method, exemplar-based inpainting method and dictionary learning based method. Also this paper proposed improvement to overcome the deficiency in each field and broaden the application in the field. In chapter 2 the paper describes the detail of the diffusion-based inpainting method and the proposed improvement. In chapter 3 the paper shows the overview of the exemplar-based inpainting method and the proposed improvement. In chapter 4 the paper also proposes an improvement of adaptive sparsity to improve the dictionary learning effect. In chapter 5 the paper does more experiment to research the ability of the inpainting methods. Finally, this paper gives the conclusion in chapter 6 and the future work in chapter 7.

Chapter 2

Diffusion-based image inpainting method

2.1 The math model of FMM

In the theory of the FMM, the method regards the Ω as the region need to inpaint. Near the edge of the inpaint region, the method divides the pixels into 3 sets:

Known (the known pixels).This set contains the pixels out of the inpainting region Ω. For these pixels, the gray value and the four neighbor pixels' gray value are known. So the gradient value of the pixel can be computed.

Inside (the unknown pixels). This set contains the pixels inside the inpainting region Ω, the gray value of them are unknown.

Band (the pixels on the $\partial\Omega$). This set contains the pixels on the edge between known region and the inpainting region Ω. The gray value of them are known, but the four neighbour pixels' gray values are unknown. So the gradient value of them can not be computed.

The basic idea is to figure out the distance T of every pixel in the inpainting region Ω, from position of the pixel to the edge $\partial\Omega$. According to the sequence of T from small to large, to inpaint the image, until the whole destroyed region is inpainted.

For every pixel p need to inpaint, selecting its neighborhood $B_\varepsilon(p)$ and using every known pixel q inside to calculate the gray value of p, every q would make an influence for the inpainting(Fig. 2.1). This process is shown as the equation (2.1):

$$I_q(p) = I(q) + \nabla I(q)\,(p - q) \tag{2.1}$$

Here, $\nabla I(q)$ means the gradient of q.

Because of the different influence made by different pixel q, the method employs a weight function $\omega(p, q)$ to weight the relationship between p and q, and finally weighted average

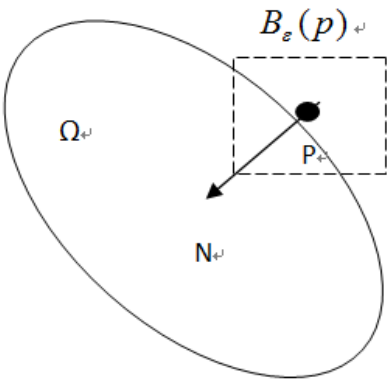


Fig. 2.1. Image inpainting theory

the influence. So the equation improved to be :

$$I_q(p) = \frac{\Sigma_{q \in B_{\varepsilon(p)}} \omega(p,q)(I(q) + \nabla I(q)\,(p-q))}{\Sigma_{q \in B_{\varepsilon(p)}} \omega(p,q)} \tag{2.2}$$

In the equation, the weight function $\omega(p,q)$ depends on the isolux line direction of q, the geometric distance between q and p. Designing the weight function so that it can inpainting the gray value of p and spreading the texture information of region $B_\varepsilon(p)$ at the same time .

In the research of Telea, it sets the weight function $\omega(p,q)$ containing three indexes

$$\omega(p,q) = dir(p,q) * dst(p,q) * lev(p,q) \tag{2.3}$$

These three indexes represent the 3 aspects of influence made by the known pixels to the in-painting process. They calculate the direction, distance and time influence respectively. And they mean if the pixel q is closer to the pixel p, the influence will be more obvious. The three indexes are shown in detail:

$$dir(p,q) = \frac{p-q}{\|p-q\|} \cdot N(p) \tag{2.4}$$

$$dst(p,q) = \frac{d_0}{\|p-q\|^2} \tag{2.5}$$

$$lev(p,q) = \frac{T_0}{1 + \|T(p) - T(q)\|} \tag{2.6}$$

The *dir* index shows the direction influence for the in-paint, it means if the pixel q's direction is close to the pixel p's direction, the more influence it will have. The *dst* index shows the distance influence for the in-paint, it means if the pixel q is near the pixel p, the more influence it will have. The *lev* index shows the time influence for the in-paint, if the pixel q is close to the known information, the more influence it will have.

Fig. 2.2 shows the inpainting results of the traditional diffusion-based method.

(a) Destoryed image.

(b) Inpainting result by PDE method.

(c) Inpainting result by traditional FMM.

Fig. 2.2. The inpainting result of diffusion-based method.

Table 2.1. The inpainting result of main method

Measure parameters and method	Calculating time/s	MSE	PSNR/dB
Based on PDE method（TV model）	3965.6	0.3102	53.2143
Based on FMM	49.0	0.2811	53.6421

Fig. 2.2shows the visual effect of the inpainting result. Fig. 2.2(b)shows that the inpainting result made by PDE method. The PDE method gives a little clearer result, the texture information is better preserved. At the same time the PDE result also can give a better edge effect. The Fig. 2.2(c)shows the result made by current FMM. FMM contributes a little fuzzier result and doesn't give a good edge effect, but for the whole image the FMM preserves the integrity of the image.

From the result shown in Table 2.1, the inpainting quality between these two method makes little difference. Though the PDE result gives a better visual effect, instead the result made by PDE speeds much more time than the result made by FMM. Compared with the origin image, FMM result also has a higher PSNR than the PDE result. These points mean that the method based on FMM algorithm has the potential to be practical.

Although the FMM has a great advantage in computing time, and it can get a similar inpainting effect as the PSNR shown, this method is short in the visual effect, especially for maintaining a good edge effect.

2.2 Improved FMM using gradient matrix

The most important advantage of FMM is this method can spend very little time and give a good global result. But when inpaint one pixel, this method use all the neighbor information so that the method may add any error information into the inpaint pixel. And finally the result shows that if the width of the inpaint area is larger than 20 pixels, the result will be unclear, especially in the isolux line and edge tangent place.

From the sets of equation, the three indexes have the same importance in the inpainting process, but actually in the inpainting the pixels that have the same texture direction with the destroyed pixel will do more contribute to the inpainting. So, if compute under this method, the pixels near the destroyed pixel but with different texture direction will regard had much useful information. In fact, this kind pixels always are the noise pixels. And the inpainting influence made by these pixels will make the edge unclear.

As the disadvantage of the traditional FMM, the FMM algorithm is weak in keeping the edge of the in-painting image. So this paper proposed a new method to remain the fast advantage and give a good edge remaining result at the same time. The gradient information of the remaining pixels is used to make the inpainting process on the texture direction and contribute a better result.

2.2.1 Obtaining the gradient matrix

Regard the $\Omega \subset R^n$ as one-dimensional, the Ω means the image, $n \in N^+$. For the two-dimensional images, $n = 2$. So the gray value of the two-dimensional image can be defined as:

$$u : \begin{cases} \Omega \subset R^n \to R \\ x(x, y) \to u(x) \end{cases} \tag{2.7}$$

the derivatives of the image u to the variable x can be defined as:

$$u_x = \frac{\partial u}{\partial x} \tag{2.8}$$

The gradient is just the derivatives of the image u to the variable x and the variable y, here use ∇u to represent it

$$\nabla u = (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) = (u_x, u_y) \tag{2.9}$$

The modulus of the gradient can be defined as

$$|\nabla u| = \sqrt{{u_x}^2 + {u_y}^2} \tag{2.10}$$

In the specified direction $v \in R^n$, the one-dimensional direction derivatives of the u can be defined as:

$$u_v = \frac{\partial u}{\partial v} = \nabla u \cdot v \tag{2.11}$$

When the image u is a two-dimensional gray image, the one-dimensional direction derivatives of it can be defined as:

$$\begin{cases} v = (v_1, v_2)^T \\ u_v = v_1 u_x + v_2 u_y \end{cases} \tag{2.12}$$

The partial derivative of the two-dimensional gray image u can be defined as:

$$u_{xy} = \frac{\partial u^2}{\partial_x \partial_y} \tag{2.13}$$

In the specified direction $v \in R^n$, the partial derivative of the two-dimensional gray image u can be defined as:

$$u_{vv} = \frac{\partial u^2}{\partial v^2} = \nabla(\nabla \cdot v) \cdot v \tag{2.14}$$

In fact, when calculating the remaining gradient information in the destroyed image, in order to reduce the calculation time, the extraction of the gradient matrix is just focused on the inpainting area and a neighborhood.

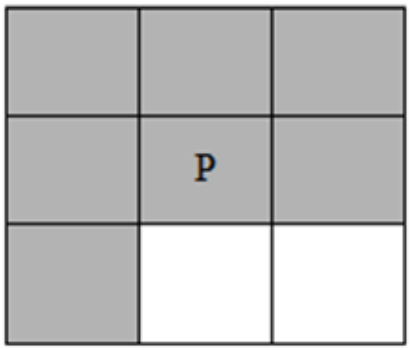


Fig. 2.3. Computing the gradient of the Band pixel

For Known pixels, the gray value of the four neighbor pixels are known, so just using them to compute the gradient by:

$$G_a(x,y) = \sqrt{(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2} \quad (2.15)$$

$$G_d(x,y) = tan^{-1}((I(x+1,y)) - I(x-1,y))/(I(x,y+1) - I(x,y-1))) \quad (2.16)$$

For Band pixels, they are indispensable(because they are quite near the destroyed region) in the computing. But one hand of Band pixels may be the inside pixels, the gray value of the inside pixel is unknown and we cannot get the Band pixels' gradient. So we need to change the equation by using the band pixel itself replace the inside pixel. For the example shown in Fig. 2.3, gray cells represent the known pixels while white cells represent the destroyed pixels. When calculating the gradient of pixel p (in the image the position of p is $I(x,y)$), due to the neighbor pixel $I(x+1,y)$ below is destroyed, the pixel $I(x,y)$ itself is used to replace the pixel $I(x+1,y)$ to compute the gradient.

So changing the gradient function to:

$$G_a(x,y) = \sqrt{(I(x,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2} \quad (2.17)$$

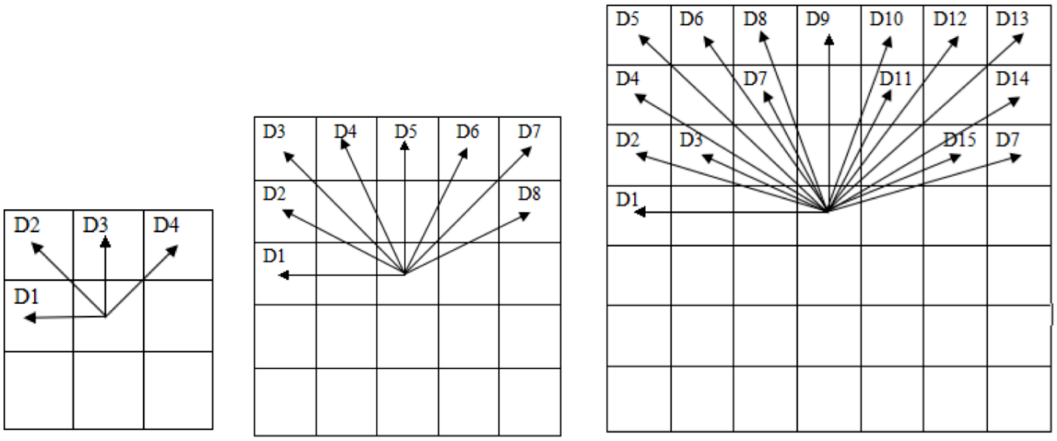
$$G_d(x,y) = tan^{-1}((I(x,y)) - I(x-1,y))(I(x,y+1) - I(x,y-1))) \quad (2.18)$$

Finally we get two matrices, one matrix $G_a(x,y)$ represents the gradient amplitude of every pixel, and the other matrix $G_d(x,y)$ represents the gradient direction of every pixel. For the two matrices, they are both continuous, but the position relationship between two pixels is a discrete value. So it's hard for the matrix $G_d(x,y)$ to do the further processing, and it's necessary to discrete the continuous directions into several discrete directions.

The mainly processing is dividing the direction area from 0 to π evenly into L sub direction areas, if the computed gradient direction $G_d(x,y)$ satisfied :

$G_d(x,y) \in \left(\frac{2(l-1)-1}{2L}\pi, \frac{2l+1}{2L}\pi\right)$

Then define it in the lth sub direction area. The value of l ranges from 1 to L. Different L value means the different discrete direction model. Here we just give 3 discrete direction models, which shows in Fig. 2.4.



(a) 4 directions. (b) 8 directions. (c) 16 directions.

Fig. 2.4. Discrete direction models.

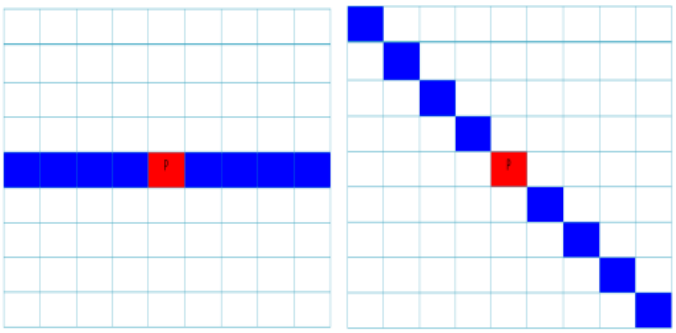
Due to the vertical relationship between the gradient direction and the texture direction, the texture direction of every pixel can be obtained. By the texture direction we can select the pixels that have a strong correlation to calculate in the FMM algorithm.

2.2.2 Select the texture direction by the gradient matrix

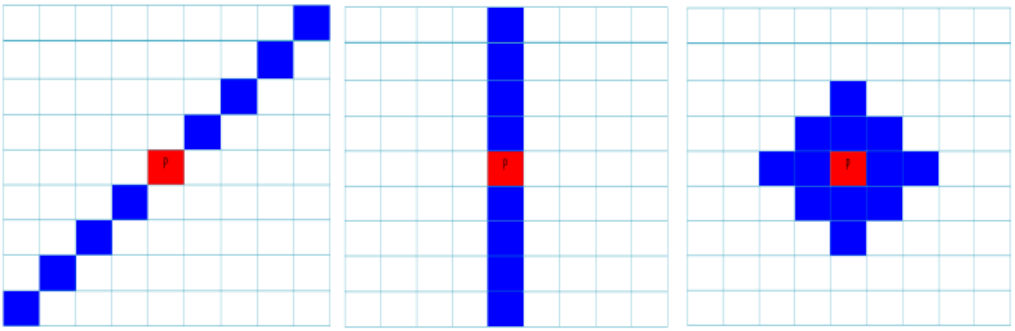
Traditional FMM shows the influence made by gradient through calculating the value of dir index. Now in the improved method, the simple calculation is changed into a pixel selecting processing. In section 2.2.1 the proposed method completed to obtain the gradient matrix and next step is to select more significant pixels into the final gray calculation.

The main idea of the proposed improvement can be described into three steps. Firstly build a neighbor around pixel p and add up the gradient information of all the known pixels in the neighbor. Then set the gradient direction which has the largest gradient amplitude as the pixel p's gradient direction. Finally only select the pixels on this gradient direction into the gray calculation.

In fact, the size of the neighbor and the number of the discrete direction would make a strong influence into the result. On the surface the more discrete directions would bring a positive influence to keep a good edge, but instead more discrete directions would also lead some pixels just part in the discrete direction. If select all these pixels into the calculation, they would also bring a lot of noise into the result. But if don't select them, the method must use a larger neighbor to make sure there're enough pixels in the calculation. What's more, the more discrete directions also would make the processing become more complex and speed more time. So finally the proposed method set the number of discrete directions



(a) dir=1. (b) dir=2.



(c) dir=3. (d) dir=4. (e) dir=5.

Fig. 2.5. Pixels selecting model.

as 4, and the size of the neighbor is set 9 by 9. According to the discrete directions, the proposed method set the number of pixel selecting model also 4, as the Fig. 2.5(a-d) shows us. Besides, a stable gradient condition is created like Fig. 2.5(e). The stable gradient condition means that in pixel p's neighbor, the amplitude of all the discrete directions are very small. The change of the pixels around p is not very large. On this condition, we select the pixels around p into the final calculation.

In the model the red pixel p is pixel need to inpaint, the blue pixels are selected into the calculation.

At last, using the FMM algorithm to compute gray value of the destroyed pixels, due to the new method, here the proposed method just remain the dst index and the lev index to compute, so the $\omega(p,q)$ is changed to be

$$\omega(p,q) = dst(p,q) * lev(p,q) \tag{2.19}$$

In this equation, the dst index and lev index are the same as the traditional method.

2.2.3 The inpainting result the improved FMM algorithm

Destroyed region inpainting and the specified object removal are the most widely used applications. Here the improved method is applied on these two applications.

Firstly, the proposed method tests the application of inpainting destroyed region. As the group of images shown in Fig. 2.6, the author selects an original image (Fig. 2.6(a)) and adds the destroyed region, then generate the destroyed image(Fig. 2.6(b)).by the traditional FMM method ,the inpainting result is shown as Fig. 2.6(c), the result shows that the left inpainting region contains many texture information, when inpainting from the outside to the inside, there're more and more noise mixed into the inpainting calculation, the right destroyed region also have this problem so that the result is a little fuzzy. This problem is well solved by improved FMM and finally the proposed method contributes the result as Fig. 2.6(d) shown. The left region maintains a good edge and does not mixed many noise.

2.2.4 The inpainting result the improved FMM algorithm

Table 2.2 shows parameters of the inpainting result

Table 2.2. Parameters of the inpainting result

	PSNR（db）	Time speeding（s）
Traditional FMM	37.3864	6.076
Dir index changed method	41.0596	8.148

Next the proposed method tests the application of the specified object removal. Firstly a scenery image is selected (Fig. 2.7(a)). In the image the small animal in the left is needed to remove and then set this region as the destroyed region, as Fig. 2.7(b) shown us. Secondly, use the traditional FMM and the improved FMM to inpaint the image. Fig. 2.7(c)shows the inpainting result of the traditional FMM, the result shows that when inpainting into the center region, it brings a lot of noise into the calculation, and then cause to the fuzzy edge of the path, it also leads the lake has a obvious breaking on it. But using the improved FMM which has a changed *dir* index in it, the proposed method contribute the result like the Fig. 2.7(d) shows. The result shows that due to the pixels selecting processing, these pixels whose gradient direction is different from the inpainting pixel would be removed to join into the calculation. So the influence made by the noise can be reduced. In the result, it is hard to believe that there had been a small animal in this region. Not only the path is well connected, but also the wave in the lake is well inpainted. But on the other hand some connection places are a little unnatural.

For the application of the specified object removal, the content of original image has already been changed. This time the inpainting is not to restore the original image, but

(a) Original image. (b) Destoryed image.

(c) Result of traditional FMM. (d) Result of employ the gradient matrix.

Fig. 2.6. Result comparison of FMM when using gradient matrix or not.

to preserve the integrity of the image. So there's no need and no meaning to calculation the MSE and the PSNR. The most important judgment is mainly by visual effect. Here just by comparing the time speeding and the visual effect to show the inpainting result (Table 2.3).

Table 2.3. The measurement of the specified object removal

	Time speeding (s)
Traditional FMM	3.378
Dir improved FMM	4.756

(a) Original image. (b) Destoryed image.

(c) Restored by traditional FMM. (d) Restored by improved FMM.

Fig. 2.7. The object remove inpainting result comparison between the traditional FMM and the improved one.

2.3 Improved FMM by edge prediction

In section 2.2 the author just tries to change the dir index, from the simple gray value calculation to a pixel selecting processing. Although the improvement brings more complexity to the method, the calculation time is just a little increased by reducing the pixels in the calculation.

In fact, the inpainting sequence also makes a great influence to the result. In the traditional FMM, the sequence is not detailed set, just from the outside to the inside. So if the FMM firstly inapint the flat region then inpaint the edge, the inpainting would bring many error information when inpainting the edge. So this section is tried to reset the inpainting sequence to get better inpainting result.

The mainly idea of edge prediction process can be described into three steps. Firstly detecting the edge in the known region and obtaining the breaking regions of the image.

Then using the edge prediction method to extend these breaking edge pixel until to the boundary of the inpainting region, and cutting the inpainting region into many sub regions. Finally using the FMM algorithm to inpaint each sub region one after another and then inpainting the whole image.

2.3.1 Edge detection in known region

To predict a good edge, firstly the method needs to obtain the edge information in the known region. Now the edge detection algorithm has been developed and many methods have been widely used. Now the mainly used methods contain differential edge detection method, sobel method, Laplace method, Canny method and Morphology method. To make the next processing becoming easy. The proposed method selects the Morphology method which contributes a continuous, completed, and clear edge.

The mainly theory of Morphology edge detection method is firstly setting a specific shape structure elements (structure element), and then use it to detect the target image. Through detecting scaling characteristics of the structural elements in the image, the proposed method obtains the structure information of the image, finally completing the analysis and recognition of images.

How to set the structure element is the key to the edge detection processing. Different structure elements would feed back different Geometry information, and make influence to the detecting result. For the structure element, the size and the shape would make a large influence to the detecting result. If the size is too small, it would strong the sensitivity of the edge detail, but it would also make the denoising ability weak. If the size is too large, the denoising ability would become strong but the detected edge would become coarse. The shape also makes an influence to the detecting result.

The calculation of morphology contains erosion and dilation. Regard A is the target image, B is the structure element, both A and B are regarded as the sets in the , then the method defines B erosion A as:

$$A \ominus B = \{x|(B)_x \subseteq A\} \tag{2.20}$$

Here, x means the displacement if the set.

The method defines B dilation A as:

$$A \oplus B = \{x|\tilde{(B)}_x \cap A \neq \phi\} \tag{2.21}$$

Open operation and close operation are also important processing in morphology algorithm, for A and B, the method defines the open operation as

$$A \circ B \quad = \quad (A \ominus B) \oplus B \tag{2.22}$$

Define the close operation as :

$$A \cdot B \quad = \quad (A \oplus B) \ominus B \tag{2.23}$$

When detecting the edge, the Morphology method often use different processing to detect different regions When detecting the inside region, it often by calculating the difference between B erosion A and the image A, then obtain the edge image D_1:

$$D_1 = A - C = A - (A \ominus B) \tag{2.24}$$

When detecting the outside region, it often by calculating the difference between B dilation A and the image A, then obtain the edge image D_2

$$D_2 = (A \oplus B) - A \tag{2.25}$$

For the edge of the gradient, the method needs to calculate the difference between B erosion A and B dilation A, to obtain the edge image D_3:

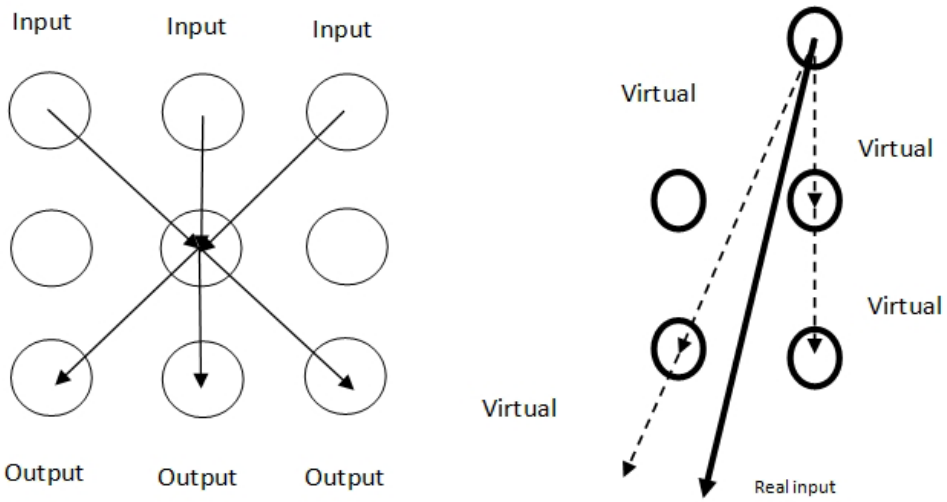
$$D_3 = (A \oplus B) - (A \ominus B) \tag{2.26}$$

After obtaining the edge, the proposed method also needs to use a judgment processing. Due to the weakness of the Morphology method, the result maybe contains wrong edge information, and the gradient of some edge pixels may be not very large. So the proposed method select a neighbor of the edge pixel and judge the gradient information in it, if the gradient is not changed obvious, then the proposed method removes it out of the edge prediction processing.

2.3.2 Edge prediction

After getting the breaking region of the detected edge, then use the gradient information of the breaking region to predict the edge in the destroyed region.

Fig. 2.8 introduces the math model of the edge prediction. Every pixel's gradient contains the amplitude and the direction, the gradient is called the gradient vector. Every gradient vector can equals to effect on the neighborhood in 8 directions. The proposed method regards these gradient vectors as the input vector to the next layer pixels and the output vector from the last layer pixels. So for the spread processing, the proposed method regards it as the pixel making an influence to the others neighbor pixels on the same direction. And the diffusion process of the pixels is regarded as the main pixel make an influence to the neighbor pixels, as the Fig. 2.8(a) shows. For these vectors get by gradient calculation, the proposed method calls them the real vector. Their direction value is continuous, so it's hard for us to see the influence to the neighbor pixels obvious. But after discretization, every real vector can be divided into several virtual vectors, then the influence made by the real vector can be regard as the influence made by different virtual vectors to the corresponding pixels, as the Fig. 2.8(b) shows us. By the transform processing from real vectors to virtual vectors, the influence made by one pixel to the neighbor pixels is obvious to realize and as shown:



(a) Input and output vectors. (b) Real vectors and virtual vectors.

Fig. 2.8. The theory of the edge prediction.

$$Riv = \sum_{i-1}^{N} Viv(i) \tag{2.27}$$

$$Rov = \sum_{i-1}^{N} Vov(i) \tag{2.28}$$

Where, Riv 、 Rov mean the real input vector and real output vector, $Viv(i)$ 、 $Vov(i)$ mean the virtual input vector and virtual output vector, N means the number of the virtual vector divided from the real vector. So in the proposed method regards the real vector as the sum of the virtual vectors. In fact, in the calculation these vectors need to transform to scalars, and the transform function is renewed to the following equation:

$$Riv_{(am)} = \cos(Riv_{(dir)}) = \sum_{i-1}^{N} Viv_{(am)} \cos(Viv(i)_{(dir)}) \tag{2.29}$$

$$Riv_{(am)} = \sin(Riv_{(dir)}) = \sum_{i-1}^{N} Viv_{(am)} \sin(Viv(i)_{(dir)}) \tag{2.30}$$

$$Rov_{(am)} = \cos(Rov_{(dir)}) = \sum_{i-1}^{N} Vov_{(am)} \cos(Vov(i)_{(dir)}) \tag{2.31}$$

$$Rov_{(am)} = \sin(Rov_{(dir)}) = \sum_{i-1}^{N} Vov_{(am)} \sin(Vov(i)_{(dir)}) \tag{2.32}$$

The real vectors and the virtual vectors should satisfy the two group functions. For a smooth edge, the proposed method regards the curvature of the edge changing uniform.

So by detecting the vectors on the same direction, the proposed method contributes the changing rate r of the edge.

$$r = \frac{\sum_{i-1}^{N}(Riv_{i+1} - Riv_i)}{N} \tag{2.33}$$

Here the N is the number of the vectors on the same directions. With the change rate r and the gradient amplitude of virtual input vector Viv_{am} then the proposed method contributes the gradient amplitude of virtual output virtual Vov_{am}:

$$Vov_{am} = Viv_{am} * (1 + r) \tag{2.34}$$

On the surface, these processing just on the gradient amplitude, but in fact due to the gradient amplitude on different virtual directions are changed. So the gradient amplitude and direction of real vector are both already changed, and the pixels' gradient in the inpainting region is then predicted.

The whole processing actually completed to use the pixel's gradient information to predict the gradient of the next pixel on the same direction. Loop this processing until to the detected edge pixel or the known pixel, then the prediction of this breaking region is end. Then predicting the other regions one after another, at last the proposed method is completed to predict all the destroyed regions.

2.3.3 The processing and result of the edge prediction

Fig. 2.9 shows the flow chart of the edge prediction.

For example, Fig. 2.10(a) is a circle image with a destroyed region, the edge is detected and shows in Fig. 2.10(b). After getting the two breaking region. The detail gradient matrix of the breaking region is regarded as an example to show our edge prediction processing. The gradient information of the up breaking field is shown in Fig. 2.11, the values out of the bracket are the gradient amplitudes and the values in the bracket are the gradient directions.

In the Fig. 2.11, the first row and second row show the real gradient value of the breaking region on the top in the Fig. 2.11(b), the values out of the bracket are the gradient directions (from 0 to 0.5, 0.11 means the direction is 0.11π) and the values in the bracket are the gradient amplitudes. The third row is also the real gradient value and the fourth row is the gradient value predicted by the first row and the second row gradient information. The result shows that the proposed method is greatly contributing the gradient information in the inpainting area. Then the proposed method uses this improvement to predict the whole edge in the inpainting area, just like the Fig. 2.11 shows us.

Fig. 2.12(a) shows the result of edge prediction in the edge image, Fig. 2.12(b) shows the result of edge prediction in the gray image(the black part in the circle is the destroyed region). And Fig. 2.13 shows the result of the inpainting. To show the result more obvious,

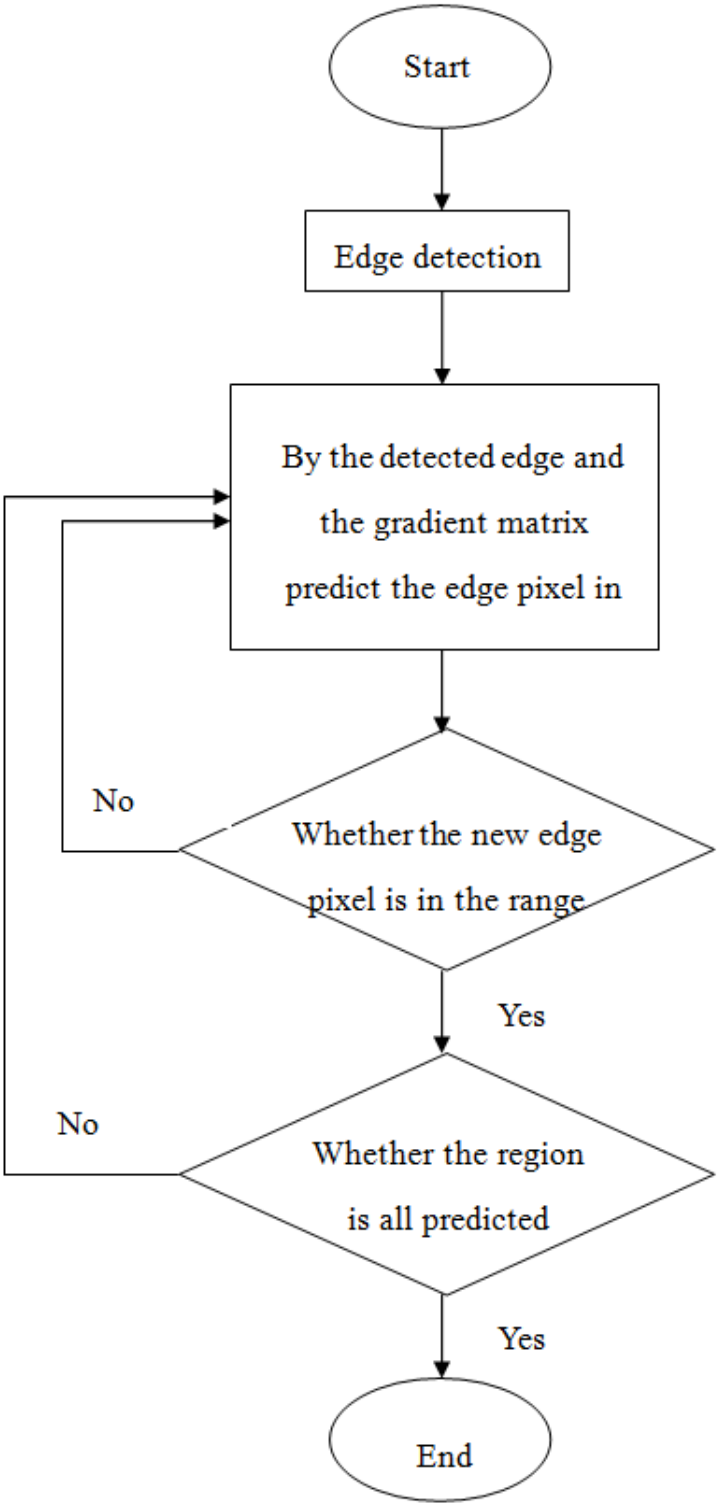
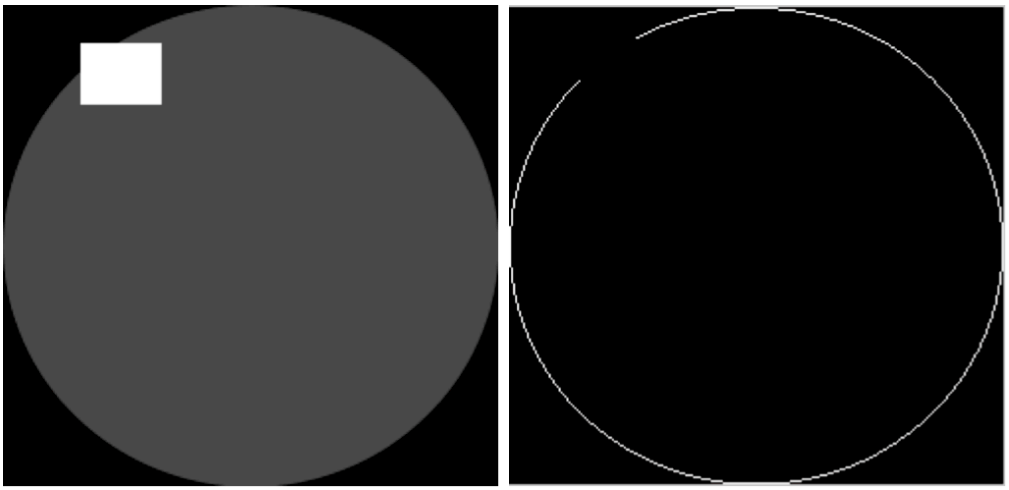


Fig. 2.9. The flow chart of the edge prediction.



(a) Destroyed circle image. (b) Result of edge detection.

Fig. 2.10. The result of the edge detection.

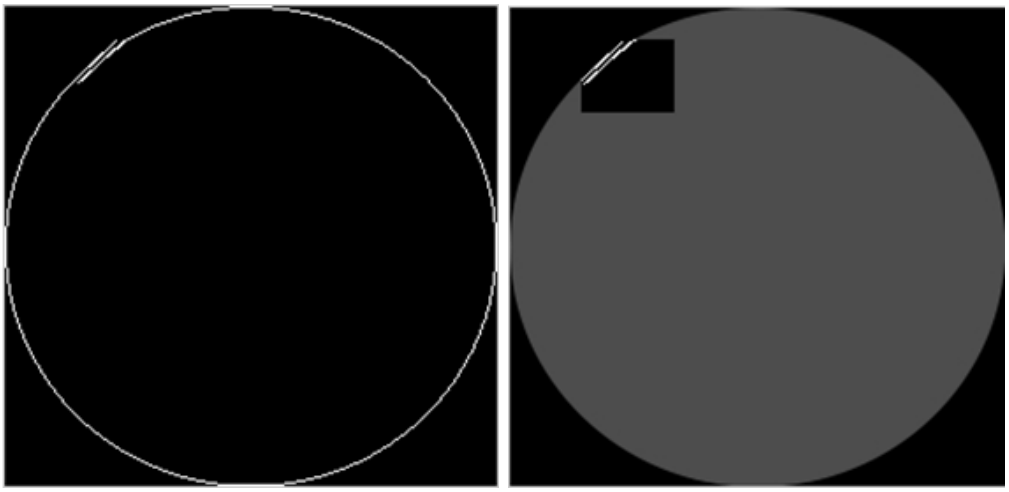
				0.11 (60)	0.21 (101)	0.22 (95)	0.08 (49)
		0.08 (50)	0.20 (95)	0.22 (100)	0.11 (60)		
0.06 (30)	0.19 (86)	0.23 (105)	0.14 (70)				
0.06 (25)	**0.19 (85)**	**0.22 (100)**	**0.16 (60)**				

Fig. 2.11. The difference between the real gradient and the predicted gradient

the proposed method uses the result after binaryzation to show the result. From the result, it can be seen that the proposed method keeps a good edge effect.

The result of the example shows that the proposed method is reasonable. The proposed method completes to maintain a good edge. But the texture in the images usually more complex, so next an image with a little complex texture in it is selected to test the proposed method.

The experiment selects an image with a complex texture in it, like Fig. 2.14(a). After edge detection and breaking region selecting, it is clear that the image has several breaking regions and the gradient in these regions is very complex. After using the edge detecting



(a) Result of edge prediction(in edge image).

(b) Result of edge prediction(in gray image).

Fig. 2.12. The result of edge prediction.

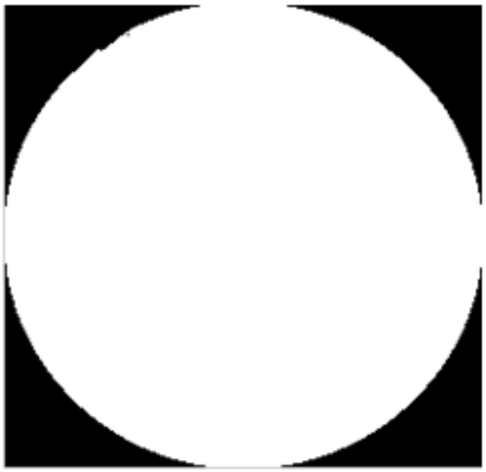
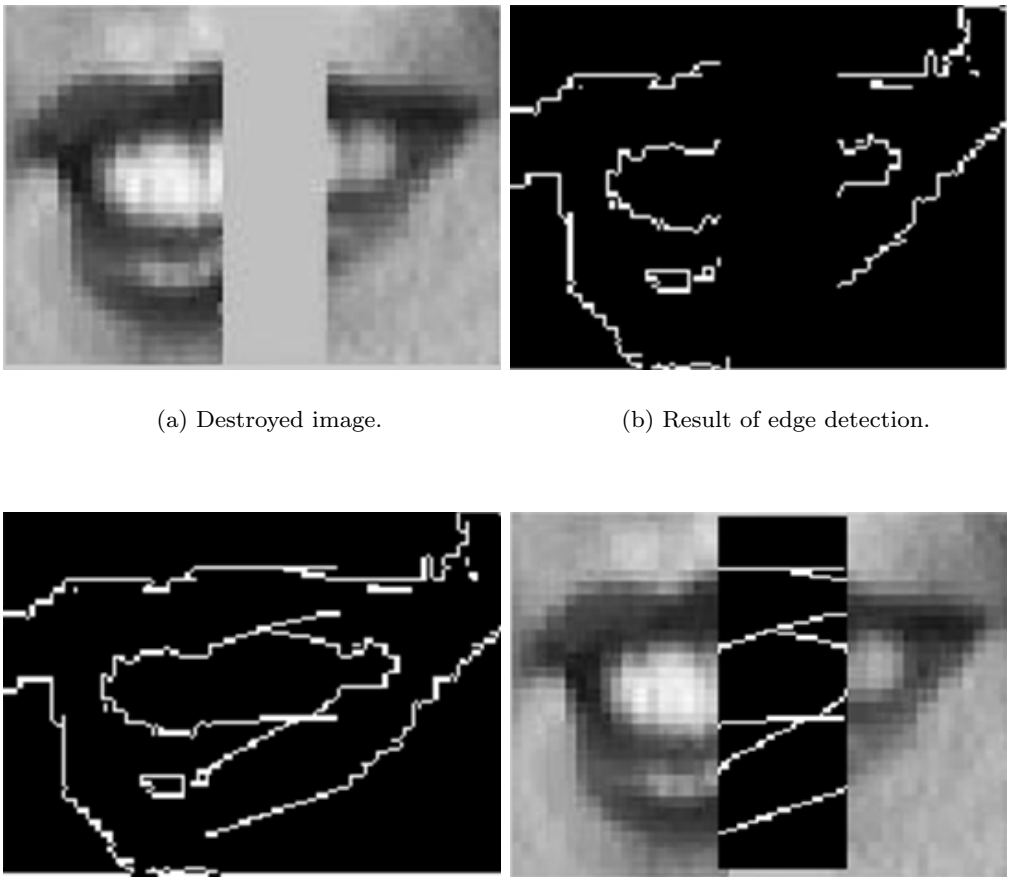


Fig. 2.13. The inpainting result of destroyed circle image

processing introduced in section 2.3.1, and judging the gradient information of the neighbor around the edge breaking region whether the gradient value satisfies the judgment. For this image, the breaking region in the left bottom is not satisfied the judgment, so the proposed method removes it out of the edge prediction processing.

After using the edge prediction introduced in section 2.3.2 for every breaking region, at last the proposed method contributes the result like the Fig. 2.14(b). Fig. 2.14(c) shows the edge prediction result in the edge image and Fig. 2.14(d) shows the edge prediction result in the gray image. Especially in Fig. 2.14(d) the edge prediction processing is well connected the breaking regions and divided the inpainting region into many sub regions.



(a) Destroyed image.

(b) Result of edge detection.

(c) Result of edge prediction(in edge image).

(d) Result of edge prediction(in gray image).

Fig. 2.14. The result of edge prediction in an image.

Though the breaking pixels is not detected, from the gradient information left in the right regions, it well connected to the breaking pixels in the left.

Fig. 2.15 shows the final result. After adding the edge prediction processing, the improved method can keep a good edge effect.

In the next section the improved method is tested on more experiments.

(a) Result of proposed method .

(b) Result of PDE method .

(c) Result of traditional FMM method .

Fig. 2.15. Inpainting result of different methods.

2.4 Experiment and analysis

2.4.1 The preparation of the experiment

This research firstly researches on the traditional FMM inpainting method and the find the disadvantage of it. To overcome the disadvantage, the author uses two items to improve the inpainting method.

Fig. 2.16 shows the main program flow of this improved image inpainting method:

This experiment is mainly come true by matlab2010a, the hardware equipment is by personal computer, which has a CPU of Intel Core $i5-2450M$ and the basic frequency is $2.5GHz \times 2$.

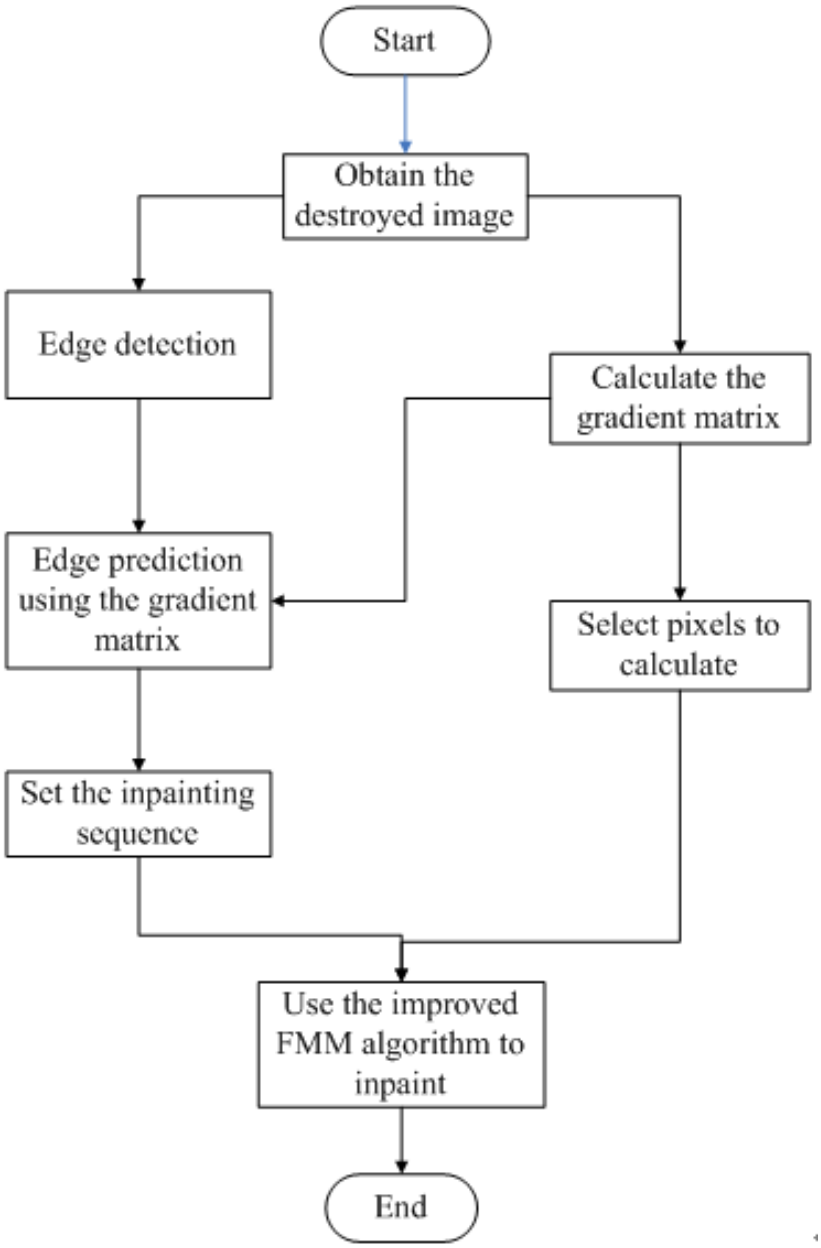


Fig. 2.16. The main flow chart of image inpainting

2.4.2 Measure parameter

To measure the quality of the inpainting result, this research uses two measure parameters. One is using the computing time to show the speed of the inpainting .The other one is the PSNR (Peak Signal to Noise Ratio), it can be calculated by (2.35):

$$PSNR = 10\log(\frac{G_f^2}{MSE}) \tag{2.35}$$

Where, $G_f$means the max gray value of the image, here the G_f is set as 255. The MSE is the mean square error and the equation is:

$$MSE = \frac{1}{N}\sum_{i-1}^{N}(x_i - I_i)^2 \tag{2.36}$$

Where, $x_i$means the gray value of the current pixel in the inpainting image, I_i means the gray value of the corresponding pixel in the original image, N means the number of the destroyed pixels.

To compute the processing time, here use the tic and toc function given by the Matlab to compute the time.

Also, this research uses the subjective judgment by eyes. Sometimes the visual effect may be more important than the parameters.

2.4.3 Experiment result and the analysis

Image inpainting is widely used but it's also mainly used in destroyed pixels inpainting. This experiment would use the time speeding and the PSNR parameter to judge ability of the improved method.

To test the improved method, first select an original image like Fig. 2.17(a). Then add a destroyed region on the center of the main texture, like Fig. 2.17(b).

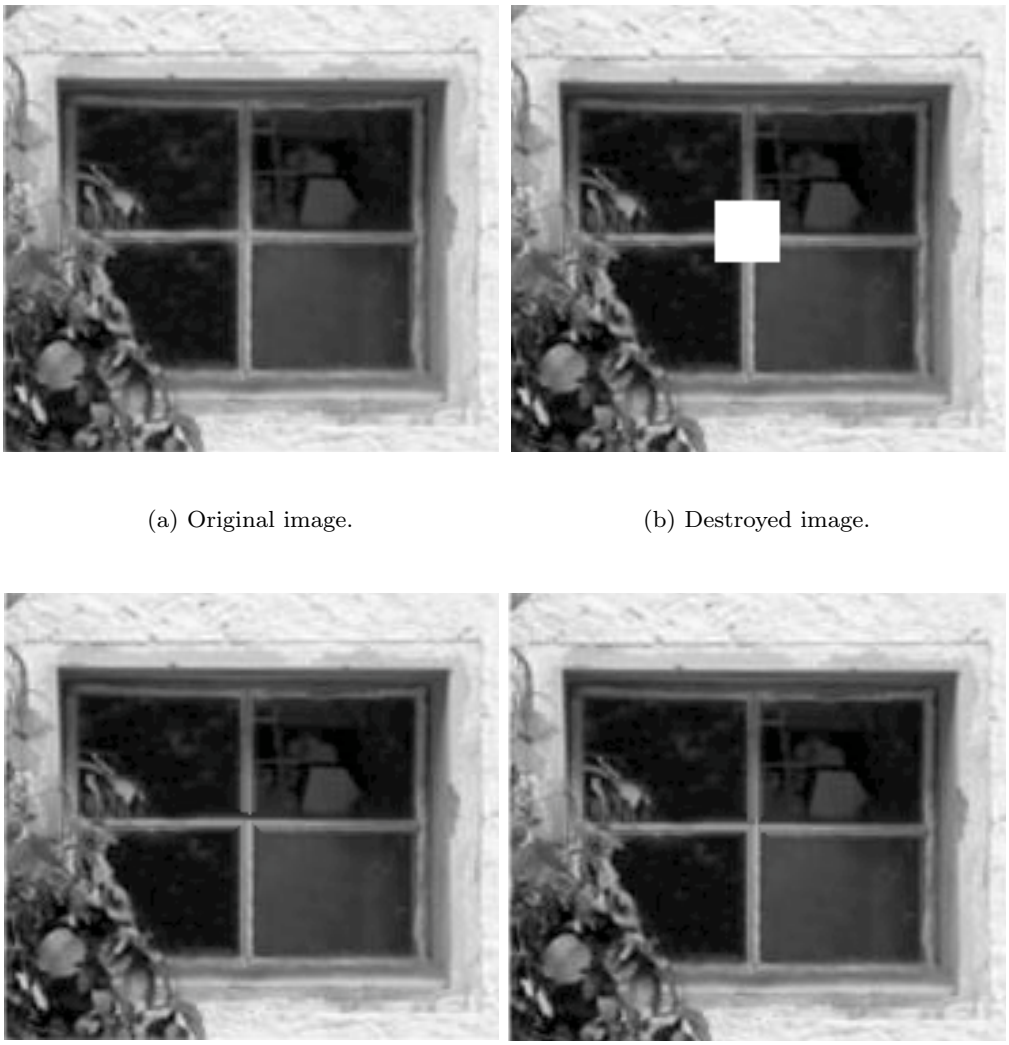
Comparing the traditional method result (Fig. 2.17(c)) and the improved method result (Fig. 2.17(d)). The result shows that the edge of the main texture is better inpainted, and the inpainting result doesn't generate the breaking region. The factual PSNR also shows that it has improved the inpainting quality. By the comparing of time, the improved method cost more time by adding the edge prediction phase. But it is valuable for the improving inpainting result, what's more, the method also has spaces to be further optimized in programing, so the time speeding can be further reduced.

For the original image Fig. 2.18(a), a destroyed region is added on the center of the main texture, like Fig. 2.18(b). By the traditional FMM and the improved FMM, results are shown in figure Fig. 2.18(c) and Fig. 2.18(d).

From the result we can see that two main edges have been improved, the edge on the left is still fuzzy. By the edge prediction result we find the reason is that the left edge is not detected by the edge detection method based on Morphology. The pixels on the both sides of the edge are not changed obvious, so this region is still fuzzy, as the Fig. 2.19 shows us.

At last, the experiment uses a couple of images with the same display but different size of destroyed region, and comparing the result with the traditional method.

From one group images, compared the result of the two methods, it can be seen that the improved method has an advantage on preserving the integrity of image, especially on maintaining the edge. Also the PSNR can show this advantage. Comparing the result of



(a) Original image. (b) Destroyed image.

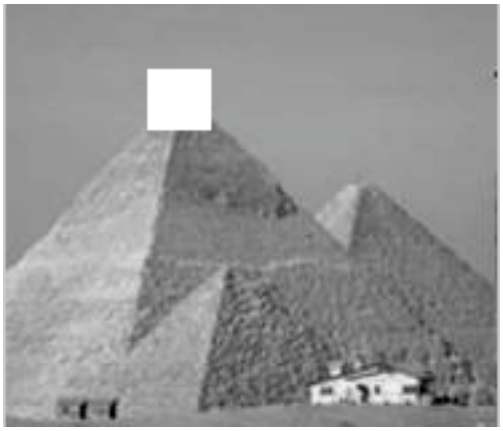
(c) Traditional method result. (d) Proposed method result.

Fig. 2.17. Image inpainting test 1.

different group images, a problem comes out that for the same image, when the size of the destroyed region becomes large, it becomes harder to maintain the integrity of the image. But the improved image is a little better than the traditional method. From this experiment result, it shows that the improved method also has an advantage on large destroyed region inpainting. Every coin has two sides, due to the matrix calculation and the edge prediction process the proposed method added, this time speeding is become longer than the traditional method. But comparing with the improving effect, it is valuable.

Table 2.4 shows the PSNR of these experiments.

(a) Original image.



(b) Destroyed image.

(c) Traditional method result.

(d) Proposed method result.

Fig. 2.18. Image inpainting test 2.

Table 2.4. The PSNR of the experiment

	test1	test2	lena(small region)	lena(large region)
Traditional FMM method	25.45	27.27	26.68	23.89
Proposed method	26.03	28.04	26.59	24.27

(a) Edge detection by method based Morphology.

(b) The result of edge prediction.

Fig. 2.19. The edge prediction result of test 2.

(a) Small destroyed region image.

(b) Traditional method result.

(c) Proposed method result.

(d) Large destroyed region image.

(e) Traditional method result.

(f) Proposed method result.

Fig. 2.20. Inpainting result of the same image with different destroy region.

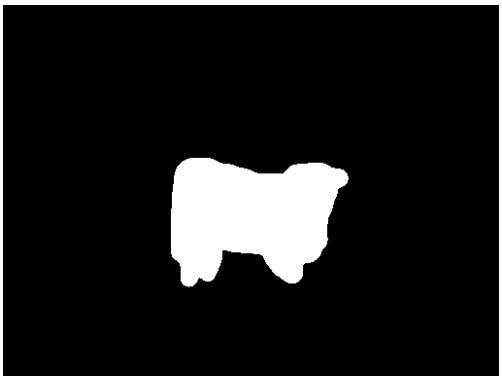
Chapter 3

Exemplar-based image inpainting method

Diffusion-based image inpainting method has a quite fast calculation speed in processing and gives a not bad global effect. These advantages make diffusion-based method quite suitable to inpaint the destroyed region which has a small width. But if the neighbor area of the destroyed region has complex textures, or the width of destroyed region becomes large, the inpainting visual effect will reduce, like the Fig. 3.1 shows us:

The Fig. 3.1 shows the advantage of exemplar-based image inpainting method compared with diffusion-based image inpainting method. It is obvious that the destroyed region contains a complex texture of grass. For the diffusion-based method, it is hard to keep the texture because it inpaints the image in pixel level. Fig. 3.1(c) shows the result become very blurred. But for the exemplar-based method, due to it inpaints the image in patch level, the texture of the grass can be remained in the image Fig. 3.1(d). What's more, the result made by exemplar-based method gives a better visual effect.

(a) Origin image.



(b) Mask image.

(c) Result of diffusion-based method.

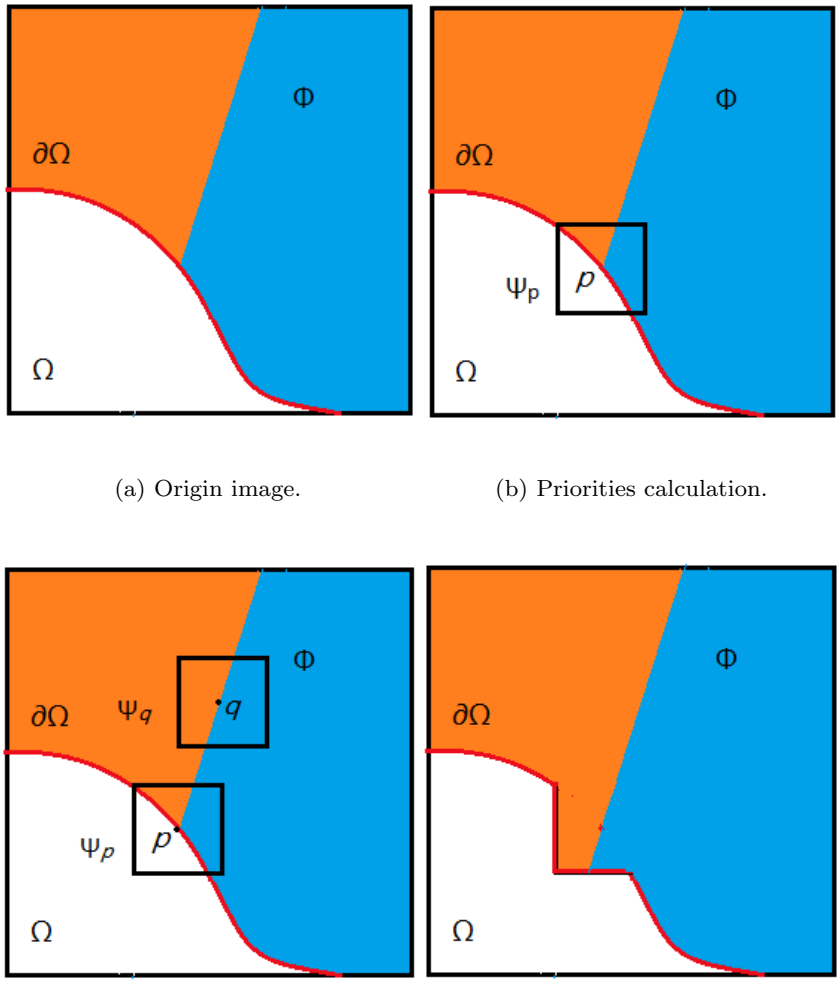
(d) Result of exemplar-based method.

Fig. 3.1. Large destroyed region inpainting result.

3.1 The theory of exemplar-based image inpainting method

The theory of the exemplar-based method can be described as an isophote-driven image sampling process. Fig. 3.2 shows the theory of the exemplar-based inpainting method. From Fig. 3.2, the exemplar-based inpainting method can be concluded in three steps: for an image I, Ωis the region that has already been destroyed, Φ is the source region that used to inpaint Ω. The edge between the two regions is defined as $\partial\Omega$(Fig. 3.2(a)). First, search the pixel p which has the highest priority in $\partial\Omega$, and obtain the patch Ψ_p with p as the center (Fig. 3.2(b)). Second, find the most similar patch Ψ_q from the source region by computing the similarities between Ψ_p and Ψ_q (Fig. 3.2(c)). Third, use the corresponding region in Ψ_q to inpaint Ψ_p and update the edge $\partial\Omega$ (Fig. 3.2(d)). Finally, loop the inpainting processing until the whole destroyed region Ω is inpainted.

For the traditional exemplar-based method, it determines the inpainting priority from



(a) Origin image. (b) Priorities calculation.

(c) Patch matching. (d) Inpainted patch.

Fig. 3.2. The theory of exemplar-based image inpainting.

two items: for one pixel p in the $\partial\Omega$.

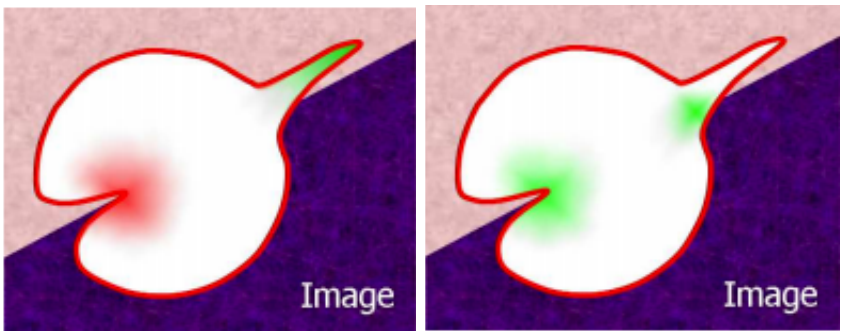
(1) If the corresponding patch Ψ_p has more pixels in the source region, regard the pixel p has a higher inpainting priority. Fig. 3.3(a) shows this item determination, in Fig. 3.3(a) the green pixels has a higher priority while the red pixels has a lower priority.

(2) If the pixel p located near a strong edge of the image, regard the pixel p has a higher inpainting priority. Fig. 3.3(b) shows this item determination, in Fig. 3.3(b) the green pixels has a higher priority. The traditional method employed the priority function $P(p)$ as the product of the two terms:

$$P(p) = C(p) * D(p) \tag{3.1}$$

The confidence term $C(p)$ represents the confidence of patch Ψ_p , it can be defined as:

$$C(p) = \frac{N_s(p)}{D(p)} \tag{3.2}$$



(a) Confidence term. (b) Data term.

Fig. 3.3. The two items of inpainting priority.

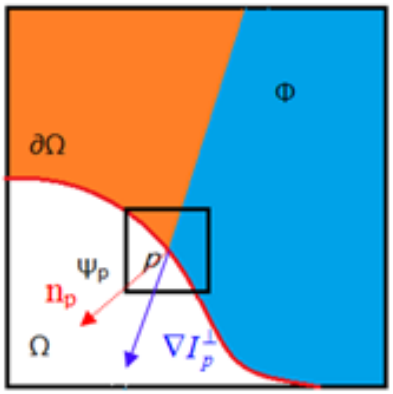


Fig. 3.4. The detail presentation of $D(p)$

In function (3.2), $N_s(p) = \sum_{q \in \Psi_p \cap \bar{\Omega}} c(p)$ $c(p)$ means the confidence of one pixel, if $q \in \Omega$, c(p) equals to 0 while if $q \in \Phi$, c(q) equals to 1. $N_s(p)$ means the number of the pixels both in the source region and patch Ψ_p. $N(p)$means the number of the pixels in the patch Ψ_p. To measure the influence made by the gradient of the pixel p, traditional method use the data term $D(p)$, it is defined as shown in Fig. 3.4:

$$D(p) = \frac{\nabla I_p^{\perp} \bullet n_p}{\alpha} \tag{3.3}$$

In function (3.3), $\nabla I_p^{\perp}$ means the gradient of the pixel p, the n_p means the unit vector orthogonal to the edge in the pixel p, the α means the gray scale of the image.

The data term $D(p)$ is a function of the strength of isophotes hitting the edge at each iteration.

In the procedure of patch matching, to measure the similarities between the destroyed patches and the known patches, patch features play an important role. The difficulty is how to use part of patch and undestroyed patches to predict the whole patch. Gray value

has firstly been used to describe the difference between the destroyed patch and candidate patches. Traditional methods have used SSD to calculate the difference. The most similar patch is obtained by:

$$\Psi_{\bar{q}} = \arg\min_{\Psi_q \in \Phi} ssd(\Psi_{\hat{p}}, \Psi_{\hat{p}}) \tag{3.4}$$

In function (4.4), $\Psi_{\hat{p}} = \hat{M} \bullet \Psi_p$, $\Psi_{\hat{q}} = \hat{M} \bullet \Psi_q$, $\hat{M}$ is the mask image and in the image the pixel equals to 1 when in source region while 0 in destroyed region.

After the patch Ψ_p has been completed inpainted by the patch Ψ_q, the method needs to update the confidence $C(p)$ in the area to make sure the next inpainting processing.

The $C(p)$ is updated in the area delimited by as follows:

$$C(p) = C(q) \forall p \in \Psi_p \cap \Omega \tag{3.5}$$

This simple update rule allows us to measure the relative confidence of patches on the fill front, without image specific parameters. As filling proceeds, confidence values decay, indicating that we are less sure of the values of the pixels near the target region.

The flow of the exemplar-based method is clearly described in Fig. 3.5.

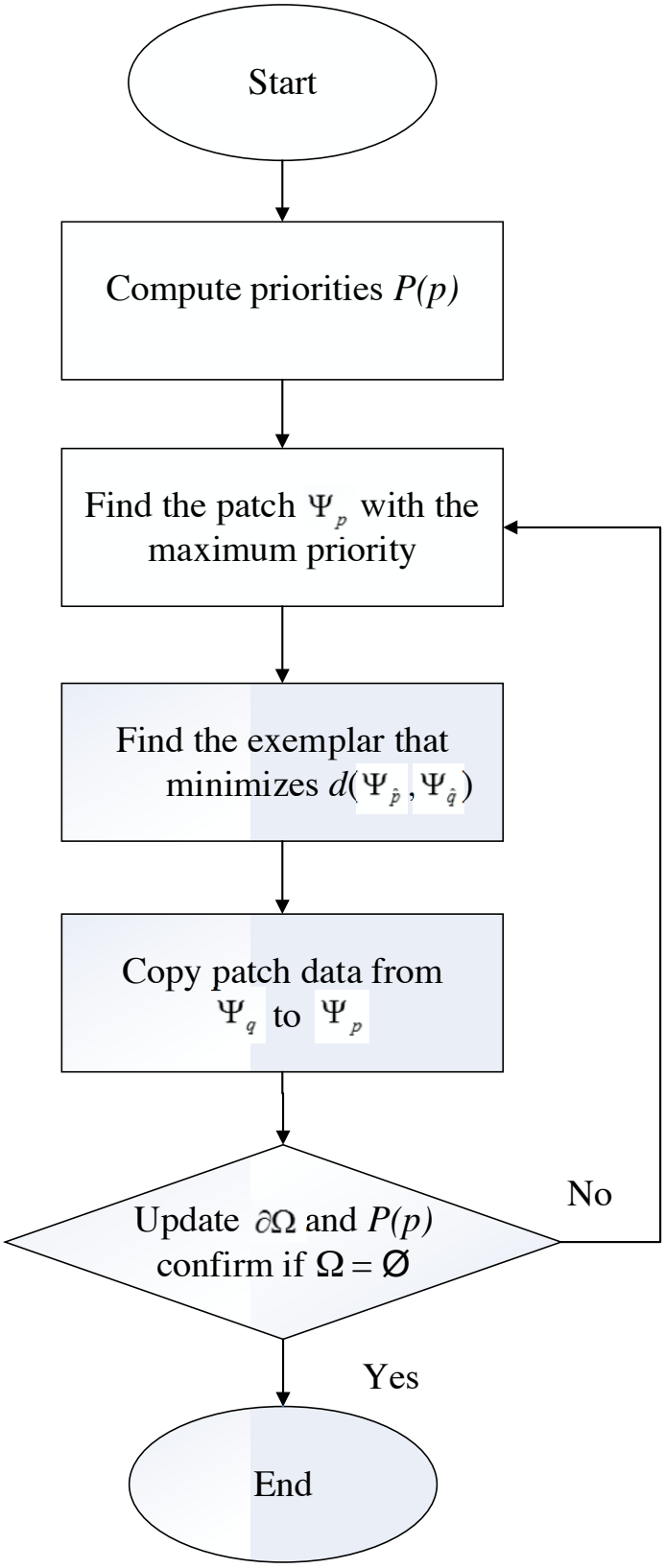


Fig. 3.5. The flow of the traditional exemplar-based method

3.2 Proposed improvement

The traditional exemplar-based method overcomes the shortage of diffusion-based method in principle, but in the inpainting process several deficiencies are still existed. Firstly, due to the limitations of the traditional priority function, it will even makes some noise pixel has a higher priority. Secondly, the traditional exemplar-based method ignored the influence made by the patch size. In the inpainting process it uses a fixed patch size from the beginning to the end. Thirdly, in the patch matching process, the similarity measurement is confined to only use the gray value information. It will lose the structure information. In this paper the proposed method tries to improve the traditional method in these three items.

3.2.1 Inpainting priority

Traditional priority function focused on the single pixel, so noise pixels interfere with the priority calculation. To overcome the deficiency of the traditional priority function, the proposed priority function $P(p)$ is defined as:

$$P(p) = \sum_{q \in \Psi_p \cap \partial\Omega} (C(q) * D(q)) \tag{3.6}$$

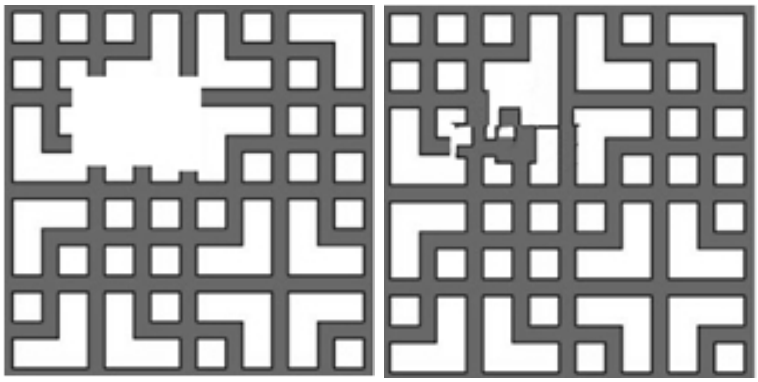
The proposed method expands the focus from single pixel to a neighbor region. By calculating the sum of the $\partial\Omega$ pixels' confidence term and data term in the neighbor, the propose method gives a high priority to the pixel whose neighbor contains more extreme pixels.

3.2.2 Adaptive patch size

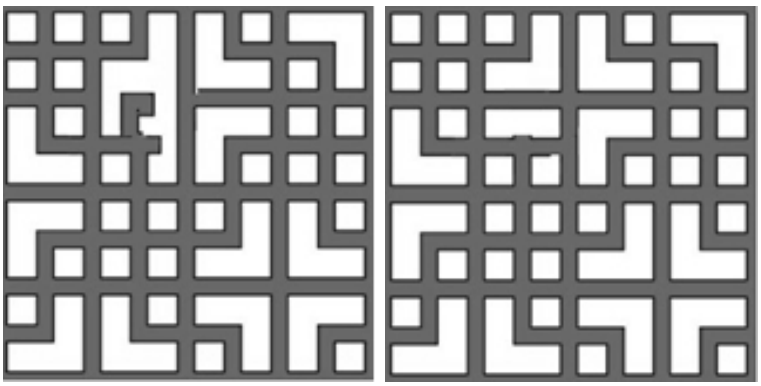
For the patch-level inpainting method, most of them use a fixed patch to inpaint. Even under the same method, the different patch sizes will lead to quite different visual effects. As shown in Fig. 3.6 and Fig. 3.7, it is quite obvious that different patch size makes different result. In Fig. 3.6, it can be find that when inpainting a destroyed triangle to the roof of it, by the matching method can not find a patch completely suit the source region, the last inpainting lead an error to the image, but the smaller patch size can reduce the error. In Fig. 3.7, when inpainting an image with a little complex structure, the patch size difference make a large influence to the result. Especially the smallest one's result is quite worse even it can not be regard as a result. So changing the patch size to meet the needs of different regions is very necessary. The proposed method initializes the patch size before the inpainting process by using an adaptive function. Firstly the method used w_0 as the standard patch size (w_0 depends on the image size and the destroyed region size, here it is regarded as the fortieth of the destroyed region.) to compute the complexity of the patch

(a) Origin image. (b) Result of patch 7*7. (c) Result of patch 13*13.

Fig. 3.6. Inpainting result of different patch size.



(a) Origin image. (b) Result of patch 7*7.



(c) Result of patch 11*11. (d) Result of patch 15*15.

Fig. 3.7. Inpainting result of different patch size.

itself. The concept of patch complexity $S(p)$ is proposed to represent the complexity. The $S(p)$ is defined as the follow:

$$S(p) = \sqrt{(\sum_{p \in N_s(p)} \omega_{p,q}^2) \bullet C(p)} \tag{3.7}$$

In the function (3.7), $\omega_{p,q}$ means the similarity between the patch Ψ_p and patch Ψ_q, and it is defined as:

$$\omega_{p,q} = \frac{1}{Z(p)} exp(\frac{d(\Psi_{\hat{p}}, \Psi_{\hat{p}})}{25}) \tag{3.8}$$

The $Z(p)$ is a normalized constant used to satisfy: $\sum_{q \in N_s(p)} \omega_{p,q} = 1$. Function $d(\Psi_{\hat{p}}, \Psi_{\hat{p}})$ uses Sum of Squared Differences (SSD) to calculate the difference between Ψ_p and patch Ψ_q.

By the analysis of the $S(p)$ value the proposed method can obtain the location information of the patch Ψ_p. A large value of $S(p)$ means that the patch Ψ_p is quite different from most of the patches in the neighbor, often Ψ_p is regard located in an edge region. On the opposite, a small value of $S(p)$ means that the patch Ψ_p is similar with most of the patches in the neighbor, often Ψ_p is regard located in a stable region.

Then the proposed method defined W_m as the matching process patch size and W_p as the inpainting process patches size. So if the proposed method inpainted the patch in an edge region, the proposed method will uses a large patch size in matching process to obtain more information against the mismatch and use a small patch size in inpainting process to reduce the interference to the other inpainting directions. While if the proposed method inpainted the patch in a stable region, the proposed method will just select a large inpainting size to improve the speed of the process.

$$W_m = \begin{cases} 1.5 * w_0 & S(p) \geq \lambda_1(p_{max} - p_{min}) + p_{min} \\ w_0 & \lambda_2(p_{max} - p_{min}) + p_{min} \leq S(p) \leq \lambda_1(p_{max} - p_{min}) + p_{min} \\ 1.25 * w_0 & S(p) \leq \lambda_2(p_{max} - p_{min}) + p_{min} \end{cases} \tag{3.9}$$

$$W_p = \begin{cases} 0.75 * w_0 & S(p) \geq \lambda_1(p_{max} - p_{min}) + p_{min} \\ w_0 & \lambda_2(p_{max} - p_{min}) + p_{min} \leq S(p) \leq \lambda_1(p_{max} - p_{min}) + p_{min} \\ 1.25 * w_0 & S(p) \leq \lambda_2(p_{max} - p_{min}) + p_{min} \end{cases} \tag{3.10}$$

In the functions, p_{max} means the maximum of the $S(p)$ while p_{min} means the minimum of S (p).Threshold λ_1 and λ_2 are set by the experience, in the experiment the proposed method set λ_1 as 0.55 and λ_2 as 0.15.

By the method proposed before, the proposed method is completed to initialize the patch size before the inpainting process.

Before introducing the second improvement, firstly introduce a concept of the pixel difference and the texture difference. These two types of differences are both generated

(a) Matching result of traditional method. (b) Matching result of candidate method.

Fig. 3.8. The theory of candidate method.

between the patches. For a pair of patches, the textures are almost the same, but the distribution of the pixel value is complex and then it generate a large difference, this difference is called the pixel difference. On the other hand, the texture difference means the difference generated by the different texture. The pixel difference is always generated in several regions and each one has a small value while the texture difference is always generated in few regions but has a large value.

By the template matching algorithm, it generally selects the most similar patch to inpaint the destroyed region. This phase sometimes improves the mismatch rate because the accumulation of several pixel differences may larger than one texture difference. But in fact the texture difference always makes a larger influence than the pixels difference in the matching process. So here the candidate method is used to reduce the mismatches. Candidate method is a matching method not only using the most similar one but also using a group of similar patches. In the method the top several most similar patches is sort by the difference. Calculate the distance between these candidate patches, if the distances between them are larger than 10 pixels, these patches will be expand and compared one more time. By this method mismatch rate is reduced. Fig. 3.8 shows an example of the candidate patch system, in Fig. 3.8(a), the right patches are the three most similar patches to the sample patch. For the traditional method, it may select the top patch because it has the minimal difference. But if expand the sample patch and the candidate patches, it is obvious that the third patch is more suitable patch.

3.2.3 Patch matching based on image rotation invariance

In the procedure of patch matching, to measure the similarities between the destroyed patches and the known patches, patch features also play an important role in the image inpainting process. The difficulty is how to use part of patch and undestroyed patches to predict the whole patch. Gray value has firstly been used to describe the difference between the destroyed patch and candidate patches. Traditional methods have used SSD to calculate the difference.

$$\Psi_{\bar{q}} = \arg\min_{\Psi_q \in \Phi} ssd(\Psi_{\hat{p}}, \Psi_{\hat{p}}) \tag{3.11}$$

Gray value only shows the difference between corresponding pixels in two patches. It just compared two patches in pixel level. What's more, to show more structure information of the patches, structural similarity (SSIM) index is used to describe the difference between the destroyed patch and candidate patches.

$$\Psi_{\bar{q}} = \arg\min_{\Psi_q \in \Phi} ssim(\Psi_{\hat{p}}, \Psi_{\hat{p}}) \tag{3.12}$$

The SSIM index is calculated on two patches. The measurement between two patches x and y is defined as:

$$ssim(x, y) = l(x, y) * c(x, y) * s(x, y) \tag{3.13}$$

In function (3.13), $l(x, y)$ shows the difference in luminance between the two patches, $c(x, y)$ shows the difference in contrast between the two patches, $s(x, y)$ shows the difference in structure between the two patches. The detail of these three functions is defined as follow:

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{3.14}$$

$$l(x, y) = \frac{2\sigma_x\sigma_y + c_1}{\sigma_x^2 + \sigma_y^2 + c_1} \tag{3.15}$$

$$s(x, y) = \frac{\sigma_x y + c_3}{\sigma_x\sigma_y + c_3} \tag{3.16}$$

μ_x means the average of x, μ_y means the average of y, σ_x^2 means the variance of x, σ_y^2 means the variance of y, σ_{xy} means the covariance of x and y. $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$, $c_3 = 0.5c_2$ are the two variables to stabilize the division with weak denominator, L is the dynamic range of the pixel values, $k_1 = 0.01$ and $k_2 = 0.03$ are by default.

In the proposed paper, it tries to compare two patches not only on patch level but also adding the rotation information. It can help to predict the result that gray value and structure differences can't give. Traditional SSD has no requirement in the order of the pixels arrangement, just along the coordinate position. So the method proposed a new rotation arrangement. In detail the proposed method transformed the patch into a matrix

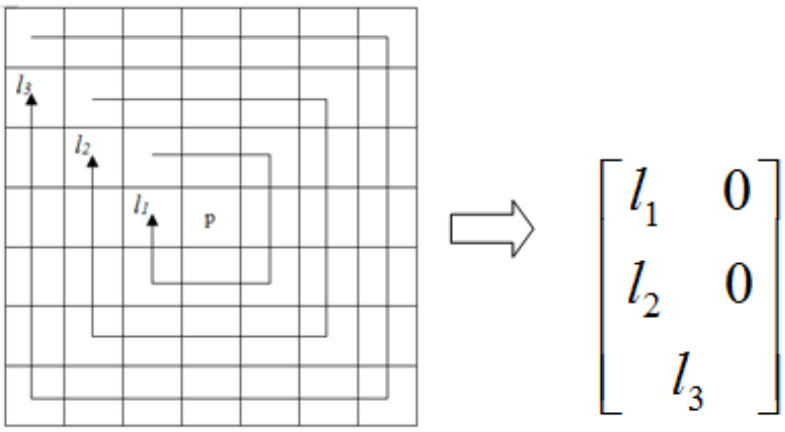


Fig. 3.9. Transformation from patch to rotation vectors

contains several vectors like the Fig.3.8 shows. From the center pixel to the edge pixel, every vector l_i represents one scale of rotation information of the patch. So the rotation of the patch can be equivalent to the movement of the matrix elements. Due to the pixel number of different scale are not equal, here the proposed method add 0 to make it become a matrix.

So that it can achieve the rotation of the patch by the cyclic motion of every layer vector. The proposed method defines a ROR function to represent the cyclic motion of vectors:

For the vector $l_i = (p_1, p_2, ..., p_{(8i-1)}, p_{(8i)})$in layer i, pixel p_i are the pixels in the vector. $ROR(l_i, 1) = (p_{(8i)}, p_1, p_2, ..., p_{(8i-1)})$ means right cyclic moving one position. And

$$ROR(\Psi_{\hat{p}}, s) = ROR_{i=1}^{n}(l_i, num_i) \tag{3.17}$$

In function (3.17), $s = 1, 2 \cdots S$. S means the rotation scale. The relationship between s and num_i is defined as:

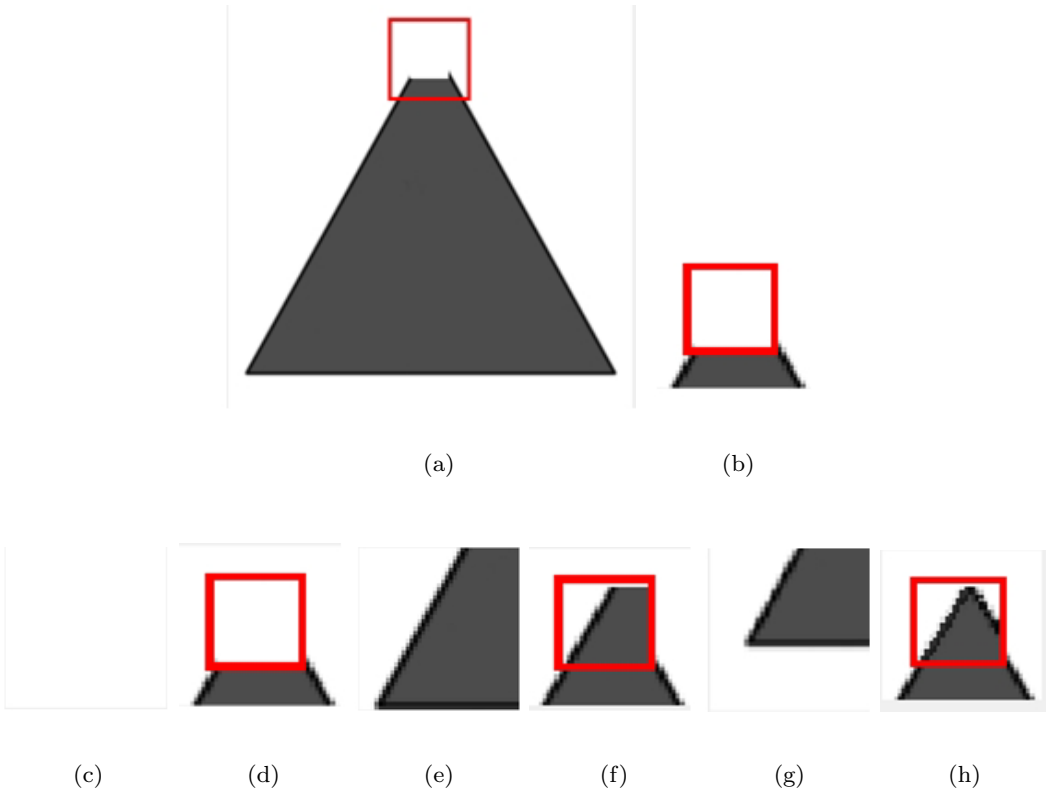
$$num_i = floor(\frac{8 * i * s}{S}) \tag{3.18}$$

The proposed similarity measure function can be defined as:

$$\Psi_{\bar{q}} = \underset{\Psi_q \in \Phi, s=1,2\cdots S}{\arg\min} ssd(\Psi_{\hat{p}}, ROR(\Psi_{\hat{q}}, s)) \tag{3.19}$$

In the experiment the rotation scale S is set as 18.

The optimized similarities function add the rotation invariance, In Fig. 3.10 and Fig. 3.11 the proposed method uses two examples to show the patch matching difference between the proposed matching method and existing matching method in a local region. Fig. 3.10(a) is the Origin destroyed image. Fig. 3.10(b) is the local region expanded from the red rectangle region in Fig. 3.10(a) and the red rectangle region in Fig. 3.10(b) means the destroyed region. Fig. 3.10(c) is the matching patch calculate by SSD method and Fig. 3.10(d) is the inpainting result by this patch. Fig. 3.10(e) is the matching patch calculate by SSIM



(a) (b)

(c) (d) (e) (f) (g) (h)

Fig. 3.10. Matching result compared with algorithm based on SSD and SSIM. (a) Destroyed image. (b) Destroyed patch. (c) Matching result of SSD. (d) Inpainting result of SSD. (e) Matching result of SSIM. (f) Inpainting result of SSIM. (g) Matching result of proposed. (h) Inpainting result of proposed.

method and Fig. 3.10(f) is the inpainting result by this patch. Fig. 3.10(g) is the matching patch calculate by proposed method and Fig. 3.10(h) is the inpainting result by this patch.

Compared with SSD result Fig. 3.10(d) and SSIM result Fig. 3.10(f), the proposed result based on rotation invariance selects more significant patch to inpaint. Also it has a higher SSIM and a higher PSNR in the local region. In Fig. 3.11 the experiment apply the

Table 3.1. The PSNR and SSIM of the Fig.3.9 local region

Feature descriptions	SSD result	SSIM result	Proposed result
PSNR	7.3268	8.1992	16.6082
SSIM	0.0071	0.5336	0.9349

proposed method on a circle, circle graph has a stronger rotational similarity in the image. It is obvious to show the advantage of the proposed method based on image rotation invariance. Fig. 3.11(a) is the Origin destroyed image. Fig. 3.11(b) is the local region expanded from the red rectangle region in Fig. 3.11(a) and the red rectangle region in

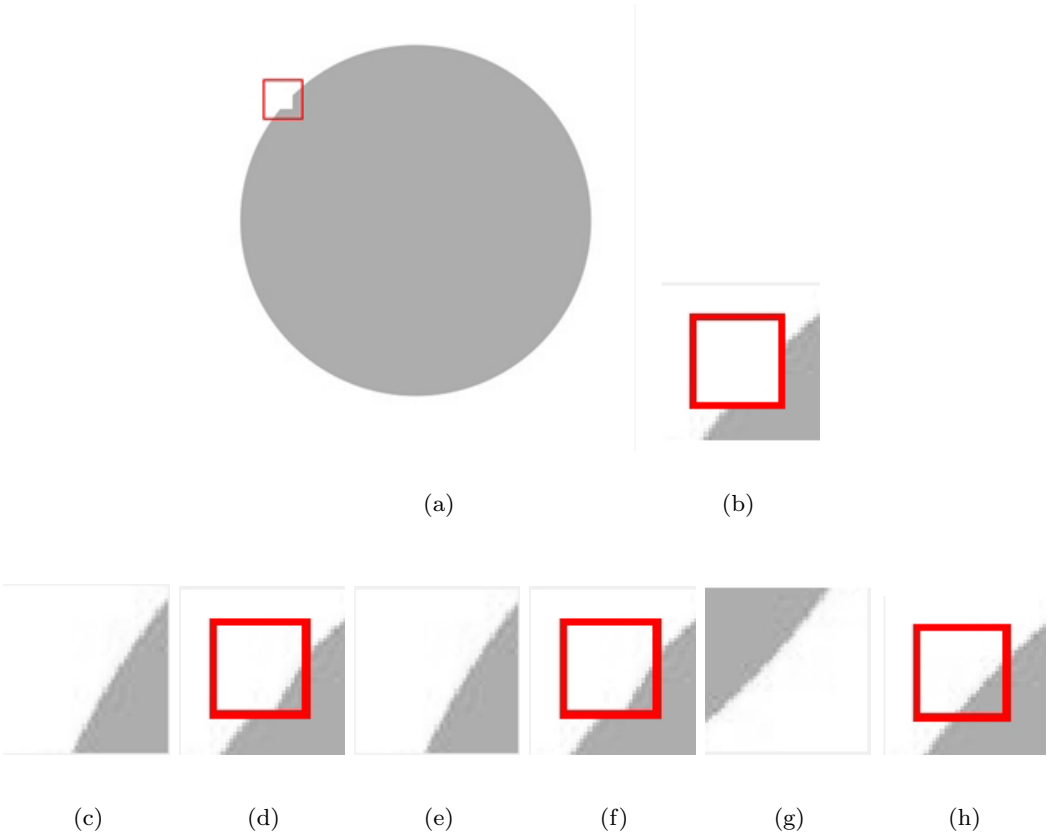


Fig. 3.11. Matching result compared with algorithm based on SSD and SSIM. (a) Destroyed image. (b) Destroyed patch. (c) Matching result of SSD. (d) Inpainting result of SSD. (e) Matching result of SSIM. (f) Inpainting result of SSIM. (g) Matching result of proposed. (h) Inpainting result of proposed.

Fig. 3.11(b) means the destroyed region. Fig. 3.11(c) is the matching patch calculate by SSD method and Fig. 3.11(d) is the inpainting result by this patch. Fig. 3.11(e) is the matching patch calculate by SSIM method and Fig. 3.11(f) is the inpainting result by this patch. Fig. 3.11(g) is the matching patch calculate by proposed method and Fig. 3.11(h) is the inpainting result by this patch.

Table 3.2 shows the PSNR and the SSIM inpainting on a circle. In this experiment the SSD and the SSIM just select the same patch in the matching process. The proposed method based on image rotation invariance has a higher SSIM and a high PSNR in the local region, it also has a better visual effect. From the result, it also can be seen that the

Table 3.2. The PSNR and SSIM of the Fig 3.10 local region

Feature descriptions	SSD result	SSIM result	Proposed result
PSNR	20.9347	20.9347	21.9753
SSIM	0.8076	0.8076	0.8649

proposed matching method improves the matching accuracy.

3.3 Experimental result

In the experiment the proposed method is implemented on a series of destroyed images that contain structure images and normal natural images. Also a comparison with the traditional methods is mentioned. All the experiments run on a computer environment made up of Matlab 2015a, CPU Intel Core i5-2450M 2.5GHz*2, RAM 8G.

Fig. 3.12(a) is the origin image, Fig. 3.12(b) is the mask image, Fig. 3.12(c)is the result made by [17], Fig. 3.12(d) is the result made by[34], Fig. 3.12(e)is the result made by proposed method.

Fig. 3.12 shows a result on a structure image, for the traditional method, the patches was just choose from the source region, so both of them can not completely inpaint the roof of the triangle. The proposed method used the similarity function based on rotation invariance so that it can find a rotation version of the source patch to inpaint. From the result it can seen that the image is quite near the triangle.

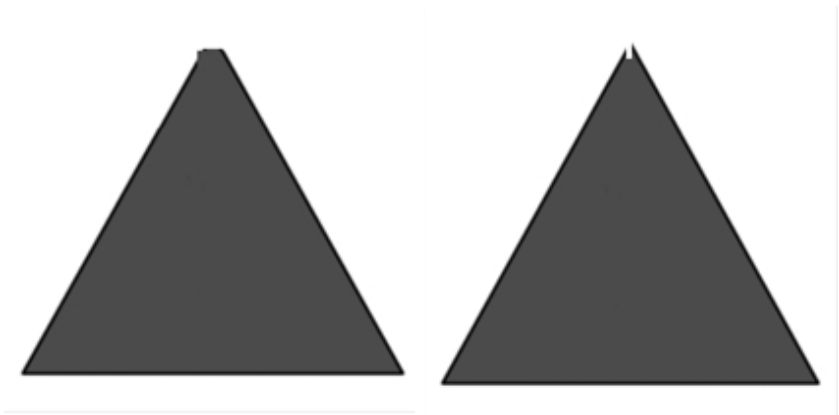
Fig. 3.13(a) is the origin image, Fig. 3.13(b) is the mask image, Fig. 3.13(c)is the result made by [17], Fig. 3.13(d)is the result made by[34], Fig. 3.13(e)is the result made by proposed method.

Fig. 3.13 shows a result on a structure image. For the other method, the patches are just chosen from the source region. When the inpainting process goes into the center of the region, the method makes a mismatch. The proposed method used the adaptive patch so it can adaptively change the patch size and select significant patch. The result shows that the proposed method has a better visual effect.

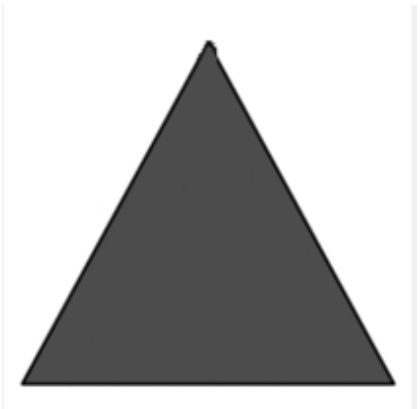
Fig. 3.14 shows an experiment on an image, in which destroyed region locates in a texture region. Fig. 3.14(a) is the origin image, Fig. 3.14(b) is the mask image, Fig. 3.14(c)is the result made by [17], Fig. 3.14(d)is the result made by[34], Fig. 3.14(e)is the result made by proposed method. From the destroyed image it can be seen that although the region around the destroyed region contains a complex texture of grass, the structure information around is only a few. From the result the three exemplar-based methods all completed a good result. It is mainly due to the advantage of the exemplar-based methods. But on the connection with the structure region there also comes few differences. Then expand two part of the region to analysis the result.

Fig. 3.15 show the detail of the inpainting result, Fig. 3.15(d), Fig. 3.15(e), Fig. 3.15(f) is responded to the inpainting result Fig. 3.15(a), Fig. 3.15(b), Fig. 3.15(c). Associate the local region with the whole image the result shows that this local region is located between strong structure and texture of grass. Fig. 3.15(d) leads a mismatch to the patch near the cow. Fig. 3.15(e) eliminates the bad influence by calculation weighted average of a set of patches. Fig. 3.15(f) also eliminates the bad influence by expand the matching size

(a) Origin image. (b) Mask image.

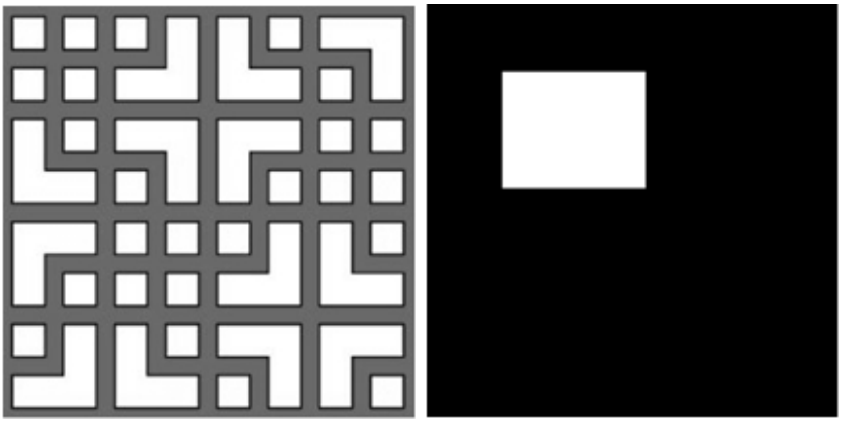


(c) Result of [17] method. (d) Result of [34] method.

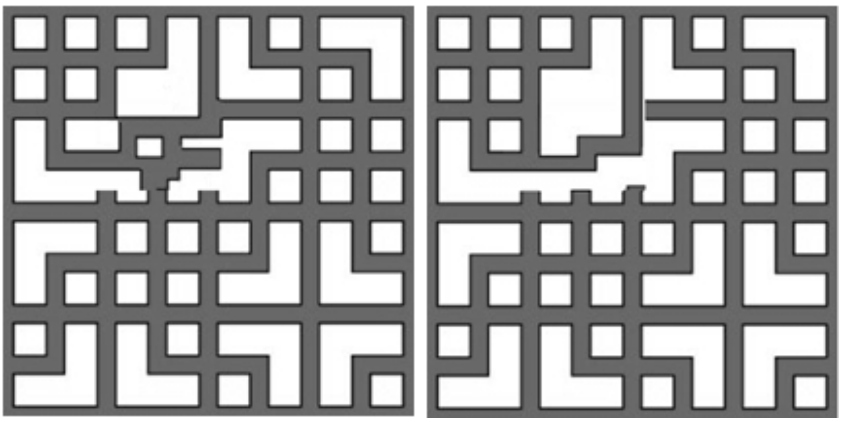


(e) Result of proposed method.

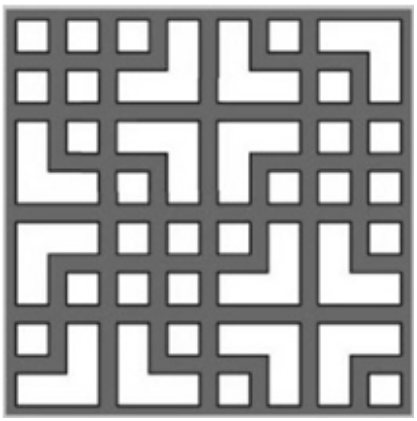
Fig. 3.12. Structure image inpainting result.



(a) Origin image. (b) Mask image.



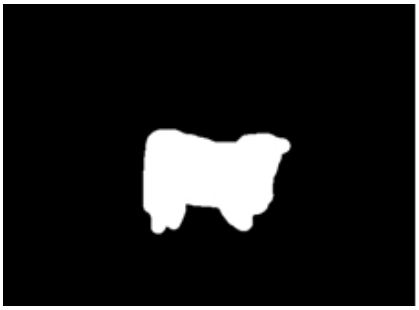
(c) Result of [17] method. (d) Result of [34] method.



(e) Result of proposed method.

Fig. 3.13. Structure image inpainting result.

(a) Origin image.



(b) Mask image.

(c) Result of [17] method.

(d) Result of [34] method.

(e) Result of proposed method.

Fig. 3.14. Natural image inapinting result.

to obtain more significant patch. Fig. 3.16 show the detail of the inpainting result, local region Fig. 3.16(d), Fig. 3.16(e), Fig. 3.16(f) is respond to the inpainting result Fig. 3.16(a), Fig. 3.16(b), Fig. 3.16(c). Associate the local region with the whole image the result shows that this local region is located in the middle of two weak structures. Fig. 3.16(d) blurs the local region Fig. 3.16(e) keeps the structure information but do not connect them. Fig. 3.16(f) not only keeps the structure information but also connects them. Fig. 3.17(a)is the origin image, Fig. 3.17(b)is the mask image, Fig. 3.17(c) is the result made by[17], Fig. 3.17(d) is the result made by[34], Fig. 3.17(e) is the result made by proposed method.

(a) Result of [17] method. (b) Result of [34] method. (c) Result of proposed method.

(d) Local region in Fig. 3.15(a). (e) Local region in Fig. 3.15(b). (f) Local region in Fig. 3.15(c).

Fig. 3.15. Local region of the result in Fig. 3.14.

Fig. 3.17 shows the comparison in normal natural image. The difference is focused on the centre of the destroyed region, especially on the edge of the house. When inpainting into the center region, [17] made a mismatch because the center region contains all directions the information. [34] selects a series of candidate patches to reduce the influence made by the single mismatched patch. The proposed adaptively changed the patch size so that it can use more information to match significant patch and finally keep a good edge. Fig. 3.18(a) is the origin image, Fig. 3.18(b) is the mask image, Fig. 3.18(c) is the result made by[17], Fig. 3.18(d) is the result made by[34], Fig. 3.18(e) is the result made by proposed method.

Fig. 3.18 also shows an inpainting result for destroyed natural image. Compared with the origin image of Fig. 3.17, the content of origin image in Fig. 3.18 is more periodic. And the width of the destroyed region is larger. By the traditional method [17], at the beginning the inpainting result is not bad, but when the inpaint process goes into the middle of the destroyed region, the patch matching accuracy decreased and the result become worse. [34] selects a series of candidate patches to reduce the influence made by the mismatched patches. The proposed method using adaptive changed patch and the optimized similarities function to obtain the patch, so it can add more information to choice the more significant patch. For the visual appearance the proposed method

(a) Result of [17] method.

(b) Result of [34] method.

(c) Result of proposed method.

(d) Local region in Fig. 3.16(a).

(e) Local region in Fig. 3.16(b).

(f) Local region in Fig. 3.16(c).

Fig. 3.16. Local region of the result in Fig. 3.14.

improved the global effect.

(a) Origin image. (b) Mask image.

(c) Result of [17] method. (d) Result of [34] method. (e) Result of proposed method.

Fig. 3.17. Natural image inpainting result.

(a) Origin image.

(b) Mask image.

(c) Result of [17] method.

(d) Result of [34] method.

(e) Result of proposed method.

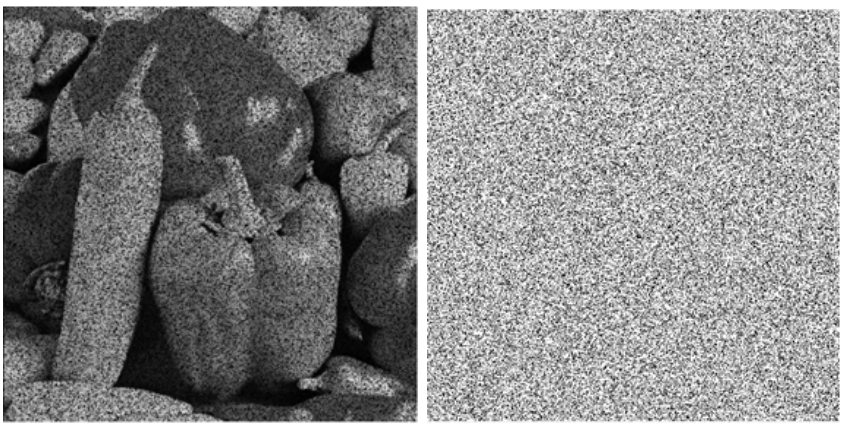
Fig. 3.18. Natural image inpainting result.

Chapter 4

Dictionary learning based Image inpainting method

The exemplar-based inpainting method uses the most similar patch to inpaint the destroyed region. The advantage of the exemplar-based inpainting method is that it can keep the texture of the image and exemplar-based method has a better visual effect. But sometimes the destroyed pixels are distributed over images. In this situation it is hard to match a suitable patch and the exemplar-based method will lead a bad inpainting effect.

The dictionary learning based image inpainting method can inpaint the images while training the dictionary learnt from the source region. And it can provide information that does not exist in the source region. These two points are its most prominent advantages. Fig.4.1 shows the appropriate destroy situation that dictionary learning based inpainting method can inpaint. For this destroy situation the exemplar-based method is impossible to inpaint because in the image even one complete patch is not existed. Fig.4.1(a) is the



(a) Destroyed image. (b) Mask image.

Fig. 4.1. The appropriate situation for dictionary learning based method.

destroyed image and Fig.4.1(b) is the mask image.

4.1 Theory of sparse approximation

Matching pursuit (MP) is a sparse approximation algorithm which involves finding the "best matching" projections of multidimensional data onto the span of an over-complete dictionary D. The basic idea is to approximately represent a signal f from Hilbert space H as a weighted sum of finitely many functions g_{γ_n}(called atoms) taken from D. An approximation with N atoms has the form

$$f(t) \approx \hat{f}_N(t) = \sum_{n=1}^{N} a_n g_{\gamma_n}(t) \tag{4.1}$$

Where a_n is the scalar weighting factor (amplitude) for the atom $g_{\gamma_n} \in D$. Normally, not every atom in D will be used in this sum. Instead, matching pursuit chooses the atoms one at a time in order to maximally (greedily) reduce the approximation error. This is achieved by finding the atom that has the biggest inner product with the signal (assuming the atoms are normalized), subtracting from the signal an approximation that uses only that one atom, and repeating the process until the signal is satisfactorily decomposed, i.e., the norm of the residual is small, where the residual after calculating γ_n and a_n is denoted by

$$R_{N+1} = f - \hat{f_N} \tag{4.2}$$

If R_N converges quickly to zero, then only a few atoms are needed to get a good approximation to f. Such sparse representations are desirable for signal coding and compression. More precisely, the sparsity problem that matching pursuit is intended to approximately solve is

$$min_x \|f - Dx\|_2^2 \; subject \; to \|x\|_0 \leq N \tag{4.3}$$

With $\|x\|_0$ the L_0 pseudo-norm (i.e.the number of nonzero elements of x). In the previous notation, the nonzero entries of x are $x_{\gamma_n} = a_n$, and the γ_n th column of the matrix D is g_{γ_n}. Solving the sparsity problem exactly is NP-hard, which is why approximation methods like MP are used.

For comparison, consider the Fourier series representation of a signal - this can be described in the terms given above, where the dictionary is built from sinusoidal basis functions (the smallest possible complete dictionary). The main disadvantage of Fourier analysis in signal processing is that it extracts only global features of signals and does not adapt to analyzed signals f. By taking an extremely redundant dictionary a signal f can be best matched.

As mentioned above, due to the signal residual is not orthogonal to the chosen atom, the result in each iteration becomes a suboptimal one rather than an optimal one, this situation makes convergence need many iterations. For example: in two-dimensional space,

there is a signal y is represented by $d = [d_1, d_2]$. MP algorithm iteration will be found repeated the iterations on x_1 and x_2, like shown in

$$y = d_1 x_1 + d_2 x_2 + d_1 x_3 + d_2 x_4 \tag{4.4}$$

This result is made by the non orthogonality, which is generated from the vertical projection of the signal residual to the chosen atom. So Orthogonal Matching Pursuit (OMP) algorithm is very necessary to overcome the disadvantage.

The improvement of the OMP algorithm is that each step of the decomposition is orthogonal to all the selected atoms, so that the convergence speed of the OMP algorithm is faster when the accuracy requirements are the same.

The OMP model can be defined as in (4.5), after k order approximation

$$y = \sum_{n=1}^{k} d_n x_n^k + R_k y, with < R_k y, d_n >= 0, n = 1, 2, \cdots, k \tag{4.5}$$

After k+1 order approximation,

$$y = \sum_{n=1}^{k+1} d_n x_n^{k+1} + R_{k+1} y, \ with \ < R_{k+1} y, d_n >= 0, n = 1, 2, \cdots, k+1 \tag{4.6}$$

Use (4.6) minus (4.5),

$$\sum_{n=1}^{k+1} d_n (x_n^{k+1} - x_n^k) + d_{k+1} x_{k+1}^{k+1} + R_{k+1} y - R_k y = 0 \tag{4.7}$$

Suppose the relationship between the atoms in dictionary matrix is:

$$x_{k+1} = \sum_{n=1}^{k} d_n a_n^k + r_k, \ with \ < r_k, d_n >= 0, n = 1, 2, \cdots, k \tag{4.8}$$

Put equation (4.8) into (4.7)

$$\sum_{n=1}^{k+1} d_n (x_n^{k+1} - x_n^k + x_{k+1}^{k+1} a_n^k) + (x_{k+1}^{k+1} r_k + R_{k+1} y - R_k y) = 0 \tag{4.9}$$

So it means that:

$$x_n^{k+1} - x_n^k + x_{k+1}^{k+1} a_n^k = 0 \tag{4.10}$$

$$x_{k+1}^{k+1} r_k + R_{k+1} y - R_k y = 0 \tag{4.11}$$

$$x_{k+1}^{k+1} r_k = R_k y - R_{k+1} y \tag{4.12}$$

Finally the OMP method is described in these phases:

1) Input: the dictionary D contains $d_1, d_2, \cdots, d_n$ in it, the sample vector y sparsity K;
2) Output: the coefficient matrix X

3) Initial: residual $r_0 = y$ Index set $\Lambda_0 = \Phi$, $\Theta_0 = \Phi$, t=1;

4) Loop the phase 1-5

- Phase 1 calculate the $\lambda_t = argmax_{j=1,\cdots,n} | < r_{t-1}, d_j > |$;
- Phase 2 update the $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$, and record the atom $\Theta_t = [\Theta_{t-1}, d_{\lambda_t}]$
- Phase 3 calculate the $\hat{x}_t = argmin_x \|y - \Theta_t \hat{x}\|_2$;
- Phase 4 update the residual $r_t = y - \Theta_t \hat{x}_t, t = t + 1$;
- Phase 5 judge the t if $t > K$, then stop the iteration, else back to phase 1.

4.2 Theory of K-SVD method

In applied mathematics, K-SVD is a dictionary learning algorithm for creating a dictionary for sparse representations, via a singular value decomposition approach. K-SVD is a generalization of the k-means clustering method, and it works by iteratively alternating between sparse coding the input data based on the current dictionary, and updating the atoms in the dictionary to better fit the data.[52][53] K-SVD can be found widely in use in applications such as image processing, audio processing, biology, and document analysis.

K-SVD is a kind of generalization of k-means, as follows. The k-means clustering can be also regarded as a method of sparse representation. That is, finding the best possible codebook to represent the data samples $\{y_i\}_{i=1}^M$ by nearest neighbor, by solving

$$min_{D,X} \|Y - DX\|_F^2 \ subject\ to\ \forall i, x_i = e_k \ for\ some\ k \tag{4.13}$$

Which is equivalent to

$$min_{D,X} \|Y - DX\|_F^2 \ subject\ to\ \forall i, \|x_i\|_0 = 1 \tag{4.14}$$

The letter F denotes the Frobenius norm. The sparse representation term $x_i = e_k$ enforces K-means algorithm to use only one atom (column) in dictionary D. To relax this constraint, the target of the K-SVD algorithm is to represent signal as a linear combination of atoms in D.

The K-SVD algorithm follows the construction flow of the K-means algorithm. However, in contrary to K-means, in order to achieve a linear combination of atoms in D, the sparsity term of the constraint is relaxed so that the number of nonzero entries of each column x_i can be more than 1, but less than a number T_0.

So, the objective function becomes

$$min_{D,X} \|Y - DX\|_F^2 \ subject\ to \forall i, \|x_i\|_0 \leq T_0 \tag{4.15}$$

Or in another objective form

$$min_{D,X} \sum_i \|x_i\|_0 \ subject\ to \forall i, \|Y - DX\|_F^2 \leq \epsilon \tag{4.16}$$

In the K-SVD algorithm, the D is first fixed and the best coefficient matrix X is found. As finding the truly optimal X is impossible, an approximation pursuit method is used. Any algorithm such as OMP, the orthogonal matching pursuit can be used for the calculation of the coefficients, as long as it can supply a solution with a fixed and predetermined number of nonzero entries T_0.

After the sparse coding task, the next is to search for a better dictionary D. However, finding the whole dictionary all at a time is impossible, so the process is to update only one column of the dictionary D each time, while fixing X. The update of the k-th column is done by rewriting the penalty term as

$$\|Y - DX\|_F^2 = \|Y - \sum_{j=1}^{K} d_j x_T^j\|_F^2 = \|(Y - \sum_{j \neq k} d_j x_T^j) - d_k x_T^k\|_F^2 = \|E_k - d_k x_T^k\|_F^2 \quad (4.17)$$

Where x_T^k denotes the k-th row of X. By decomposing the multiplication DX into sum of K rank 1 matrix, the other K-1 terms are assumed fixed and the k-th remains unknown. After this step, solve the minimization problem by approximate the E_k term with a rank-1 matrix using singular value decomposition, then update d_k with it. However, the new solution of vector x_T^k is very likely to be filled, because the sparsity constraint is not enforced. To cure this problem, define ω_k as

$$\omega_k = \left\{ i \mid 1 \leq i \leq N, x_T^k(i) \neq 0 \right\} \quad (4.18)$$

Which points to examples $y_i{}_{i=1}^N$ that use atom d_k (also the entries of x_i that is nonzero). Then, define Ω_k as a matrix of size $N \times |\omega_k|$, with ones on the $(i, \omega_k(i)) - th$ entries and zeros otherwise. When multiplying $x_R^k = x_T^k \Omega_k$ this shrinks the row vector x_T^k by discarding the zero entries. Similarly, the multiplication $Y_k^R = Y\Omega_k$ is the subset of the examples that are current using the d_k atom. The same effect can be seen on $E_k^R = E_k \Omega_k$.

So the minimization problem as mentioned before becomes

$$\|E_k \Omega_k - d_k x_T^k \Omega_k\|_F^2 = \|E_k^R - d_k x_R^k\|_F^2 \quad (4.19)$$

and can be done by directly using Singular Value Decomposition (SVD), which is introduced in the next section. SVD decomposes E_k^R into $U\Sigma V^T$. The first column of U is used to update the dictionary atom d_k, and use the first column of $V \times \Sigma(1,1)$ to update the coefficient vector x_R^k at the same time. After updating the whole dictionary, the process then turns to iteratively solve X, and then iteratively solve D. For one matrix A, the aiming of SVD is to find the full rank decomposition of A, it is shown in Fig.4.2, and SVD can be regard as a factorization of a real matrix. Sometimes the aiming are more often describe as in Fig.4.3: Now suppose there is a m*n matrix A, in fact, the A matrix maps the vectors in n-dimensional space to the k ($k \leq m$) dimensional space, k=Rank (A). The goal now is to find a set of orthogonal bases in n-dimensional space, which keeps still orthogonal after making the A transform. It is assumed that such a set

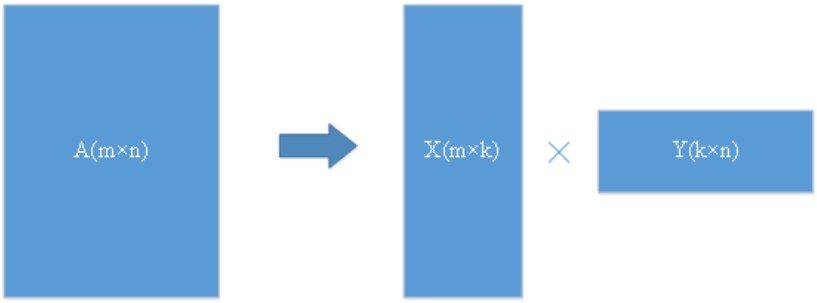


Fig. 4.2. The Aiming of SVD(1)

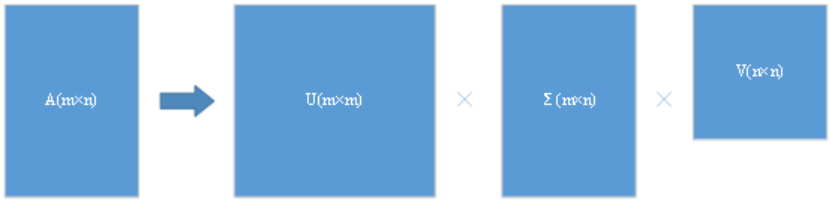


Fig. 4.3. The Aiming of SVD(2)

of orthogonal bases has been found: $\{v_1, v_2, \cdots, v_n\}$ The A matrix maps the set of bases to: $\{Av_1, Av_2, \cdots, Av_n\}$

To make each pair of them orthogonal, it means that:

$$Av_iAv_j = (Av_i)^T Av_j = v_i^T A^T Av_j = 0 \tag{4.20}$$

According to the hypothesis, it exists that $v_i^T v_j = v_i v_j = 0$.

So if the orthogonal basis v is chosen as the eigenvector of A'A, since A'A is the symmetric matrix and v is mutually orthogonal, then...

$$v_i^T A^T Av_j = v_i^T \lambda_j v_j = \lambda_j v_i^T v_j = \lambda_j v_i v_j = 0 \tag{4.21}$$

Thus the orthogonal basis which keeps still orthogonal after the mapping is found, so then unit the orthogonal basis of after the mapping: Because of $Av_iAv_i = \lambda_i v_i v_i = \lambda_i$

So $|Av_i|^2 = \lambda_i \geq 0$

And take the unit vector $u_i = \frac{Av_i}{|Av_i|} = \frac{1}{\sqrt{\lambda_i}} Av_i$

$$Av_i = \sigma_i u_i,\ \sigma_i = \sqrt{\lambda_i}, 0 \leq i \leq k, k = Rank(A) \tag{4.22}$$

Then extended $\{u_1, u_2, \cdots, u_k\}$ to $\{u_{k+1}, \cdots, u_m\}$ according to the condition $k \leq i \leq m$, to satisfy $\{u_1, u_2, \cdots, u_m\}$ become an orthonormal basis in m dimensional space. Similarly, for $\{v_1, v_2, \cdots, v_k\}$ extend to $\{v_{k+1}, \cdots, v_n\}$(this n-k vectors exist in the zero space of A, it means that these vectors are the basis of the solution space for Ax=0), so that $\{v_1, v_2, \cdots, v_n\}$are the set of orthogonal bases in n dimensional space. It means that if selecting $\{v_{k+1}, \cdots, v_n\}$ from the solution space for Ax=0, and define the $\sigma_i = 0, for\ i > k$, then the equation

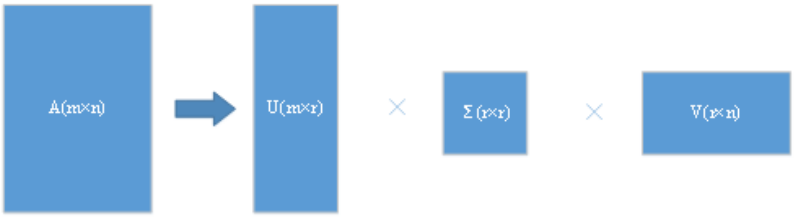


Fig. 4.4. The low order approximation of SVD

(4.22) can extend to :

$$A\,[v_1 v_2 \cdots v_k | v_{k+1} \cdots v_n] = [u_1 u_2 \cdots u_k | u_{k+1} \cdots u_m] \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_k & \\ & 0 & & 0 \end{bmatrix} \tag{4.23}$$

Finally, it is transformed to:

$$A = [u_1 u_2 \cdots u_k | u_{k+1} \cdots u_m] \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_k & \\ & 0 & & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_k^T \\ v_{k+1}^T \\ \vdots \\ v_k^T \end{bmatrix} \tag{4.24}$$

So the representation of SVD is obtained

$$A = U \Sigma V^T \tag{4.25}$$

In the equation (4.25), U is an $m \times m$ unitary matrix (if k = R , unitary matrices are orthogonal matrices),Σ is a diagonal $m \times n$ matrix with non-negative real numbers on the diagonal, V is an $n \times n$ unitary matrix over k. On the other hand, U and V are feature vector of A,Σ is the singular value matrix of A. σ_i in Σ are the singular value of A, it has an advantage of fast convergence speed. So sometimes a low order approximation is used to replace Σ. It is shown in Fig.4.4. In general, the main idea of dictionary learning based image inpainting method is to find the most suitable sparse approximation to the destroyed patch. In the method the overlapped sampled patches are used as the data sample $\{y_i\}_{i=1}^M$. It contains two steps in the method. First step is to use the OMP method to calculate the sparse representation of the patch sample $\{y_i\}_{i=1}^M$ in the source region. In this process fix the dictionary before sparse representation. The second step is updating the each atom $d_l (l = 1, 2, 3 \cdots k)$ in the dictionary after fixing the coefficient matrix. It contains three phase.

1.Find every patch sample y_l that the corresponding coefficient $x_T^l > 0$.

2.Calculate the residual $Er = Y - DX$

3.Use the SVD to update the dictionary atom d_l and the coefficient x_T^l

Loop constant iterates the two steps until to the iteration times T or a threshold ϵ set before.

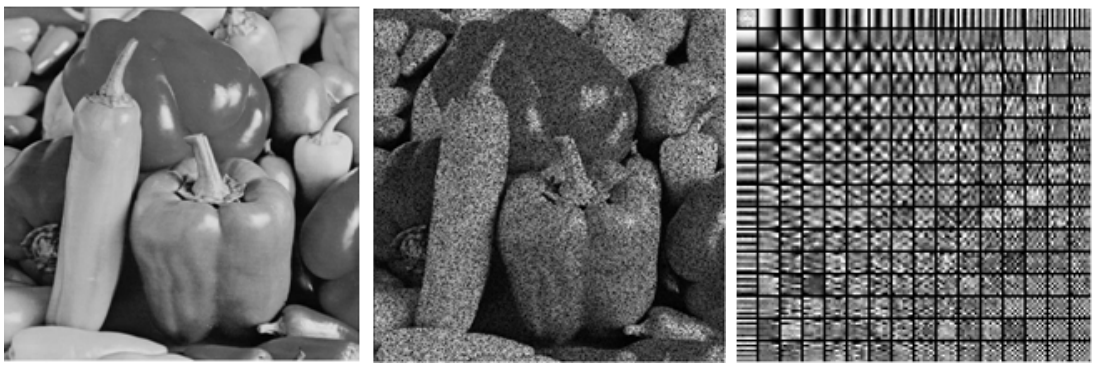
Finally the method obtained the dictionary D and the coefficient matrix X.

In the inpainting process, due to the patches are overlap sampled. So after us obtaining the dictionary D and the coefficient matrix X, it is necessary to transform the patch sample set back to the image by weighted averaging of the influence made by the overlapped patches.

4.3 Adaptive sparsity method

The K-SVD based image inpainting method provides another idea to keep the textures in the image. Not only the most similar one patch, but also a set of dictionary atoms are used to learn more information in a single inpainting process. This method uses a sparse representation of atoms so that it generates a new patch and it is no longer limited to the content of the original image. What's more, the sparse representation makes an influence to the whole patch so it also reduces the noisy pixel's influence. In the sparse approximation process, the K-SVD method uses a fixed sparsity to obtain the sparse approximation. It is easy to ignore the influence made by sparsity but in fact the sparsity also plays an important role in the inpainting. Fig.4.5 shows a sparse approximation example on a natural image. Fig.4.5(a) is the original image. Fig.4.5(b) is the destroyed image and in it 25% pixels lost its gray value. Fig.4.5(c) is the updated dictionary after the inpainting process.

Table 4.1 shows the PSNR of the K-SVD method inpainting result. From the result it shows that in a patch size from 6*6 to 10*10 and the sparse (here the sparse means the



(a) Original image. (b) Destroyed image. (c) Updated dictionary.

Fig. 4.5. Sparse approximation on image "peppers".

Table 4.1. The PSNR of the inpainting result in different patch size and sparsity

Sparsity	6*6 patch	7*7 patch	8*8 patch	9*9 patch	10*10 patch
2	31.4949	32.0009	32.0705	32.1882	32.0826
3	31.7709	32.0063	32.2001	32.4742	32.2497
4	31.6690	31.9919	32.3137	32.3024	32.2999
5	31.5940	31.9815	32.1822	32.5274	32.4290
6	31.4823	31.8755	32.2289	32.2827	32.2586

maximum number of atom in coding process) from 2 to 6, the difference the maximum and the minimum in the form is 1.04. Even in the same patch size, the PSNR difference in different sparsity is about from 0.13 to 0.35. It is almost equal to the difference made by different inpainting methods. So introducing an adaptively changed sparsity is very necessary.

In chapter 3 an improvement of adaptive patch size is proposed; in the method firstly the complexity of the patch itself is calculated. Then by a pair of threshold shown before, the proposed method can obtain the location information of the patch and adjust the patch size according to the location information. Here still use the patch complexity function to realize the location information. If the sparse approximation process the patch in an edge region, a large sparsity is used to keep the structure information and against blurring influence. While if the sparse approximation process the patch in a stable region, a small sparsity is used to make the inpainting result become smooth.

Finally optimize the OMP by adding a sparsity adaptively change process.

1) Input: the dictionary D contains $d_1, d_2, \cdots, d_n$ in it, the sample vector y sparsity K;
2) Output: the coefficient matrix X
3) Initial: residual $r_0 = y$ Index set $\Lambda_0 = \Phi$, $\Theta_0 = \Phi$, t=1, K=K_0=0.5*length(y);
4) Loop the phase 1-7
   - Phase 1 calculate the S(p) introduced in chapter 3.
   - Phase 2 judge the S(p) if $S(p) < 0.15(p_{max} - p_{min}) + p_{min}, K = K_0 + 1$; if $S(p) > 0.55(p_{max} - p_{min}) + p_{min}, K = K_0 - 1$.
   - Phase 3 calculate the $\lambda_t = argmax_{j=1,\cdots,n} | < r_{t-1}, d_j > |$;
   - Phase 4 update the $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$, and record the atom $\Theta_t = [\Theta_{t-1}, d_{\lambda_t}]$
   - Phase 5 calculate the $\hat{x}_t = argmin_x \|y - \Theta_t \hat{x}\|_2$;
   - Phase 6 update the residual $r_t = y - \Theta_t \hat{x}_t, t = t + 1$;
   - Phase 7 judge the t if $t > K$, then stop the iteration, else back to Phase 1.

4.4 Experimental result

In this section the proposed method is applied on several experiments. compared the proposed result with the result made by an inpainting method based on MRF [40] and the result made by K-SVD method [32]. The images used in the experiment are from the Berkeley segmentation dataset.

In the first group, the destroyed images contains 25% pixels lost its gray value.

In this experiment, Fig. 4.6 shows the global visual effect of the inpainting result and Fig.4.7 shows the local visual effect. In Fig.4.6, the proposed method gives the inpainting result with the highest PSNR. In Fig.4.7 the local regions of the inpainting result show the detail difference between these methods. Compared with Fig.4.7(f) and Fig.4.7(g), Fig.4.7(h) gives more natural result especially on the edge because the methods inpaint the destroyed pixels and update the dictionary at the same time. The K-SVD method continues to update the atoms in the dictionary so that the method generate the patch which is more natural. Further more, the proposed method provides more texture than the traditional K-SVD method by adding the adaptive sparse. By detecting the complexity of the patch, if the patch locates in a edge region, the proposed method uses a small sparse to keep the structure, if the patch locates in a stable region, the proposed method uses a big sparse to gain more texture to the result. At last the proposed method gives a result seemed containing more content than the traditional K-SVD method. In the following experiment the inpainting methods are applied on more images with different types.

In the next group, 50% pixels lost its gray value in the image.

Table 4.2 shows the PSNR of the experiments in this chapter. In this chapter 4 images added 25% and 50% destroyed pixels are applied on MRF method, K-SVD method and proposed method.

Table 4.2. The PSNR of the experiments in chapter 4

	MRF result	K-SVD result	Proposed result
'sculpture' (25%destroyed)	25.35	25.35	25.48
'castle' (25%destroyed)	29.21	29.10	29.41
'lion' (25%destroyed)	26.95	26.98	27.05
'tiger' (25%destroyed)	27.82	27.73	27.97
'sculpture' (50%destroyed)	22.63	22.40	22.67
'castle' (50%destroyed)	25.67	25.67	25.89
'lion' (50%destroyed)	24.15	23.91	24.23
'tiger' (50%destroyed)	24.67	24.18	24.63

From the experiment result, the improvement of the proposed method is shown in two aspects.

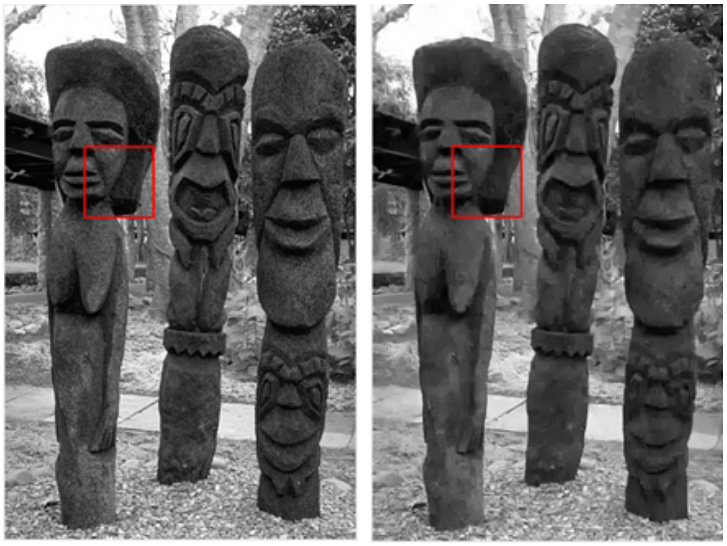
(a) Origin image. (b) Destroyed image.

(c) MRF result. (d) K-SVD result. (e) Proposed result.

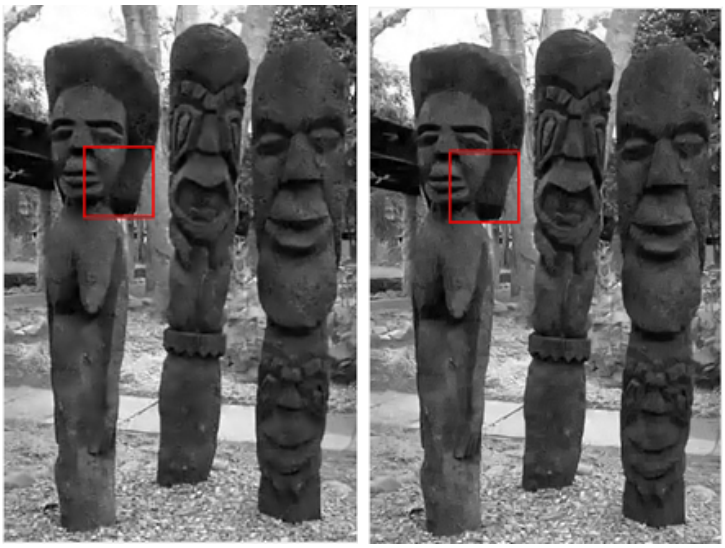
Fig. 4.6. The inpainting result on image ' sculpture'.

Firstly, the result shows that the proposed method has a better visual effect because the proposed method uses the adaptive sparsity in the sparsity approximation. When inpainting the patch near a edge region, the proposed uses a small sparsity to keep the structure information. When inpainting the patch near a texture region, the proposed method uses a large sparsity to gain more texture information in the patch.

Secondly, compared with the K-SVD method, it is obviously that the proposed method has a little improved in PSNR. By using the adaptive sparsity the proposed method can



(a) Origin image. (b) MRF result.



(c) K-SVD result. (d) Proposed result.

(e) Local (a). (f) Local (b). (g) Local (c). (h) Local (d).

Fig. 4.7. Local region of the result in Fig. 4.6.

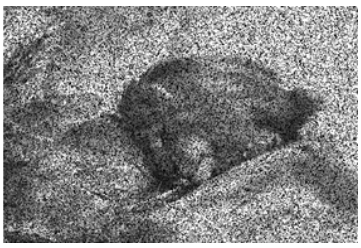
(a) Origin image. (b) Destroyed image.

(c) MRF result. (d) K-SVD result. (e) Proposed result.

Fig. 4.8. The inpainting result on image 'castle'.

get a closer sparsity approximation, so at last the proposed method can get a higher PSNR. Compared with the MRF result, the proposed method also has a small improvement of PSNR in most result.

(a) Origin image.



(b) Destroyed image.

(c) MRF result.

(d) K-SVD result.

(e) Proposed result.

Fig. 4.9. The inpainting result on image 'lion'.

(a) Origin image.

(b) Destroyed image.

(c) MRF result.

(d) K-SVD result.

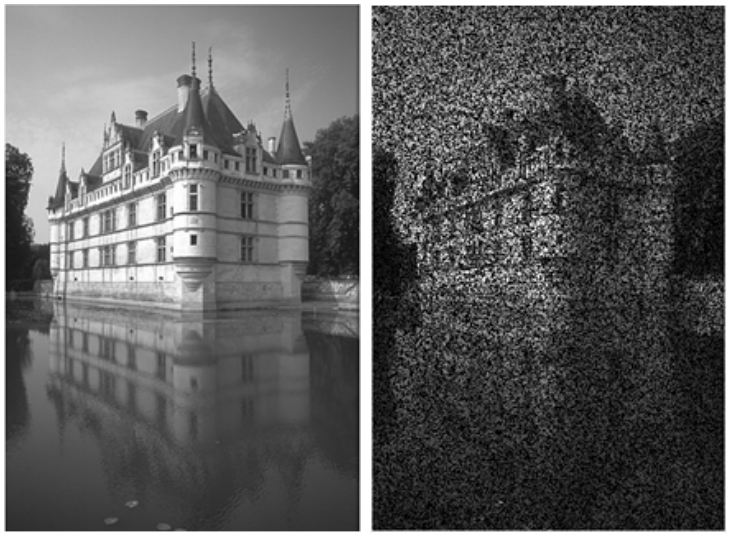
(e) Proposed result.

Fig. 4.10. The inpainting result on image 'tiger'.

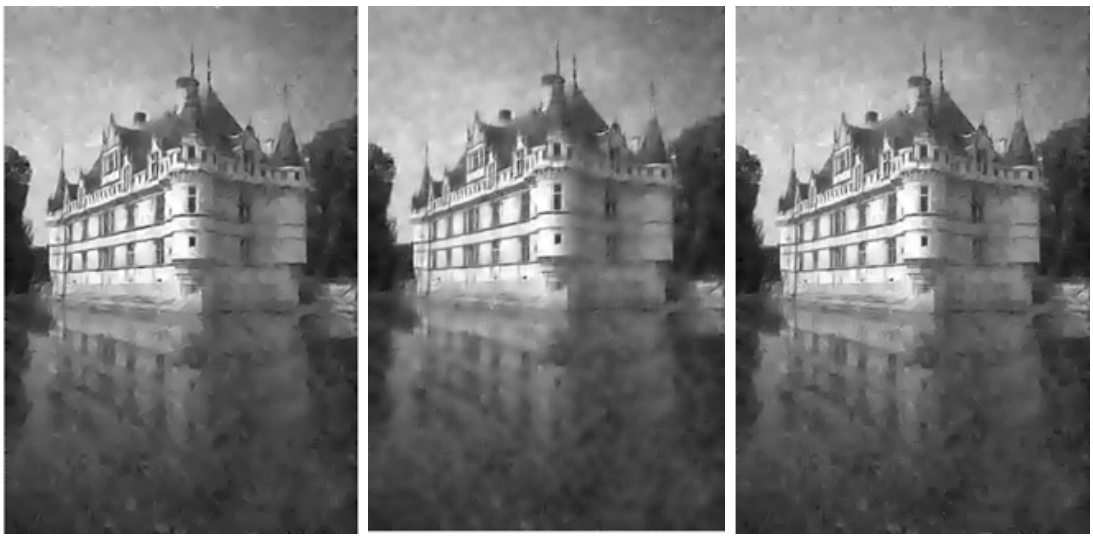
(a) Origin image. (b) Destroyed image.

(c) MRF result. (d) K-SVD result. (e) Proposed result.

Fig. 4.11. The inpainting result on image ’ sculpture’.

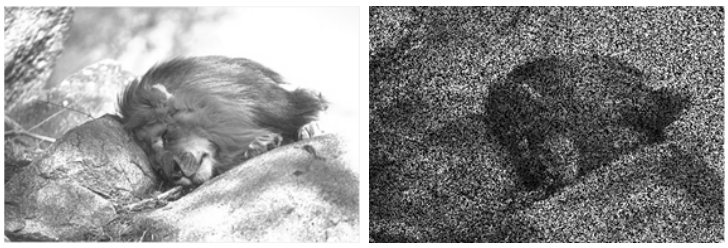


(a) Origin image. (b) Destroyed image.



(c) MRF result. (d) K-SVD result. (e) Proposed result.

Fig. 4.12. The inpainting result on image 'castle'.



(a) Origin image. (b) Destroyed image.

(c) MRF result. (d) K-SVD result. (e) Proposed result.

Fig. 4.13. The inpainting result on image 'lion'.

(a) Origin image. (b) Destroyed image.

(c) MRF result. (d) K-SVD result. (e) Proposed result.

Fig. 4.14. The inpainting result on image 'tiger'.

Chapter 5

Experimental analysis

In this chapter, the three types of the proposed methods are applied on different destroyed type images. Firstly the origin images and their sizes are shown in Fig. 5.1. Fig. 5.1(a) and Fig. 5.1(b) are normal images, containing natural scenery image and person image. These two images are used to show the comprehensive performance of the three types methods. Fig. 5.1(c) has a strong structure in it and this figure is used to show the structure keeping ability of the methods. Fig. 5.1(d) contains a lot of textures and it is used to show the texture keeping ability of the methods.

In this group the proposed method is applied on the pixels type destroyed images. For this type, the destroyed pixels are evenly distributed over the image. Due to the exemplar-based method can not inpaint this type destroyed image. The experiment is just on diffusion-based methods and dictionary learning based method. The Fig. 5.2 shows the inpaint result and the visual effect.

Fig. 5.2 shows the inpainting result for the image which has 20% pixels destroyed.

Fig. 5.3 shows the inpainting result for the image which has 30% pixels destroyed.

Fig. 5.4 shows the inpainting result for the image which has 50% pixels destroyed.

From the result only by the visual effect it is hard to distinguish the difference. On the followed information in Table 5.1 and Table 5.2 shows the ability of these two type method. With the number of the destroyed pixels become larger, dictionary learning based method can remain more details because it divides the image into patches to inpaint so in the inpainting process not only the gray value but also the regional information in the image are used. In the next group the blocks and the line contains pixels will be used to destroy the image. Three types of the methods will be applied on.

Fig. 5.5 shows the inpainting result for the image which has destroyed blocks on it, the size of the block is 6*6.

Fig. 5.6 shows the inpainting result for the image which has destroyed blocks on it, the size of the block is 10*10.

Fig. 5.7 shows the inpainting result for the image destroyed by a line.

Blocks destroyed type and line destroyed type can be regard as the expansion of the pixels destroyed type, not only in quantity but also in quality. This destroyed type loses

(a) (481*321).

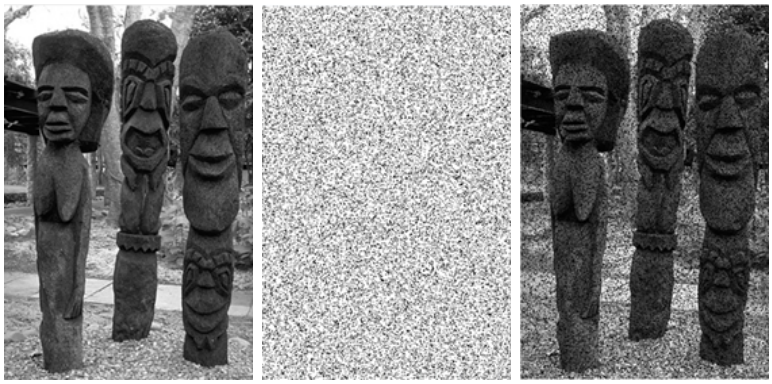
(b) (512*512).

(c) (512*512).

(d) (512*512).

Fig. 5.1. The origin images used in the experiment.

not only the pixels' gray value but also the regional information in the image. Compared the results of these three types methods, it can be seen that the traditional FMM ignores the relationship between pixels so the edges in the result is fuzzy. The proposed diffusion-based method reduces this bad influence and gives a little clear edge, by only selecting the gradient direction pixels in the inpainting process. Dictionary learning based methods realize the regional information between pixels and give a better result. The K-SVD method divide the image into patches to inpaint so that the method can remain more details in the image. Also this type method can generate a clear edge. Because of the destroyed blocks locate in different places, the proposed dictionary learning based method adaptively changes the sparsity in the sparse approximation to respond to different destroyed parts in the image so that the method can give a further similar result. Exemplar-based methods give the most similar results. And compared the traditional and proposed exemplar-based



(a) Origin image. (b) Mask image. (c) Destroyed image.

(d) FMM inpainting result. (e) Proposed diffusion method result. (f) K-SVD result. (g) Proposed dictionary learning result.

Fig. 5.2. 20% pixels destroyed inpainting result.

method, the proposed one gives a further similar result. The proposed one uses adaptively changed patches to inpaint so that the method can better deal with the destroy part in different places and select more significant patch to inpaint.

Fig. 5.8 shows the inpainting result destroyed by a small region(90*90).

Fig. 5.9 shows the inpainting result destroyed by a large region(185*185).

Fig. 5.10 shows the inpainting result destroyed by a small region(90*90).

Fig. 5.11 shows the inpainting result destroyed by a large region(185*185).

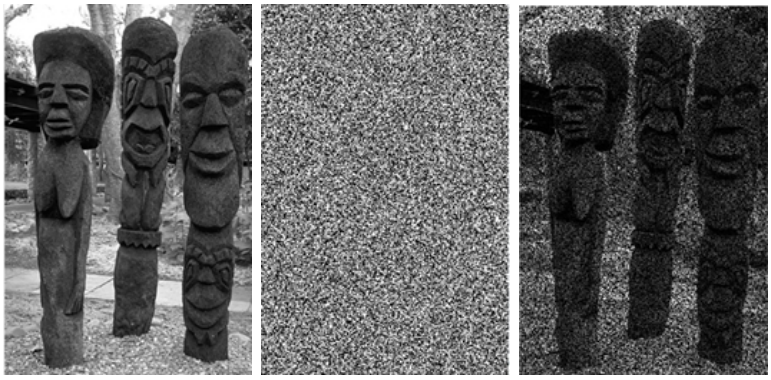
Region destroyed type can be regard as the expansion of the blocks and lines destroyed type. The regional information is further lost, containing structure information and texture information. From the result, traditional FMM selects all the pixels in the neighbor to calculate the destroyed pixels' gray value so that this method leads into a lot of error information and generates a fuzzy edge. Even if the destroyed region is quite large, tradi-

(a) Origin image. (b) Mask image. (c) Destroyed image.

(d) FMM inpainting result. (e) Proposed diffusion method result. (f) K-SVD result. (g) Proposed dictionary learning result.

Fig. 5.3. 30% pixels destroyed inpainting result.

tional FMM almost loses the whole edge. The proposed diffusion-based method keeps the edge by using the edge prediction and direction selection process. The method predicts the edge from the remaining part of the image and uses the prediction to define the inpainting order so that this method can keep the edge. Both of the two diffusion-based method can not keep the texture in the image due to the limitation of diffusion-based method itself. On the other hand, the exemplar-based method overcomes this disadvantage because this type method inpaints the image on patches level. The exemplar-based method calculates the similarity between destroyed patch and candidate patch, and selects the most similar one to inpaint. This process can remain the texture in the patch. Compared the traditional and the proposed exemplar-based method, the proposed one generates a more similar result by matching the patch in different sizes and in different rotation situations. Table 5.1 and Table 5.2 shows the PSNR and the calculation time of all the inpainting



(a) Origin image. (b) Mask image. (c) Destroyed image.

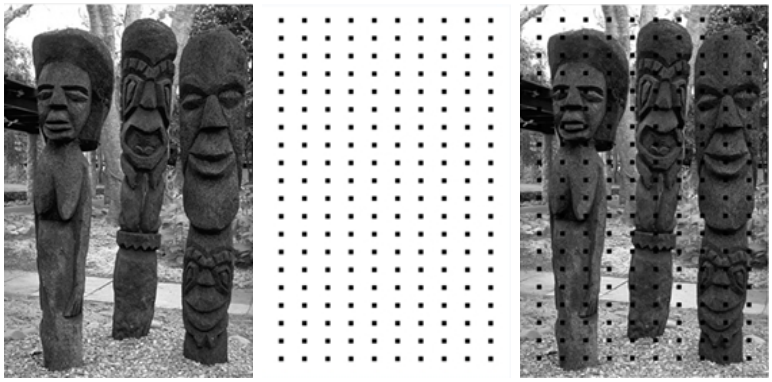
(d) FMM inpainting result. (e) Proposed diffusion method result. (f) K-SVD result. (g) Proposed dictionary learning result.

Fig. 5.4. 50% pixels destroyed inpainting result.

result, With these two measurements each type of inpainting methods is suitable for what type of destroyed images becomes clear.

From the results , it is clear to make these conclusion.

1. To inpaint the pixels type destroyed images, it is clear that both diffusion-based inpainting methods and dictionary learning based methods can give a good visual effect. Diffusion-based inpainting method also has a fast calculation speed. Exemplar-based inpainting methods are hardly to inpaint the image because the source region doesn' t contain a complete patch. Dictionary learning based methods gives a higher PSNR than the diffusion-based inpainting method but instead speed more to calculation time. Higher PSNR means the dictionary learning based methods keep more texture in the image. Also it can be easily seen that the calculation time of diffusion-based inpainting method is grown according to the number of the destroyed pixels, while the calculation time of dictionary



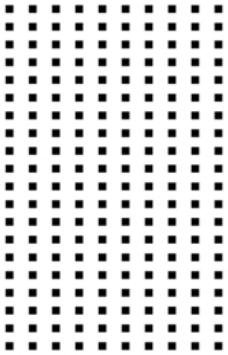
(a) Origin image. (b) Mask image. (c) Destroyed image.

(d) FMM inpainting result. (e) Proposed diffusion method result. (f) Traditional exemplar method result. (g) Proposed exemplar method result.

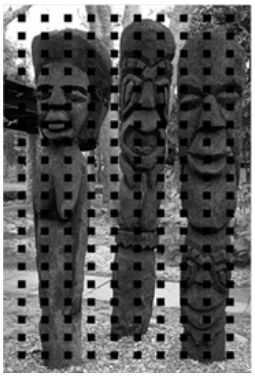
(h) K-SVD result. (i) Proposed dictionary learning result.

Fig. 5.5. 6*6 blocks destroyed inpainting result.

(a) Origin image.



(b) Mask image.



(c) Destroyed image.

(d) FMM inpainting result.

(e) Proposed diffusion method result.

(f) Traditional exemplar method result.

(g) Proposed exemplar method result.

(h) K-SVD result.

(i) Proposed dictionary learning result.

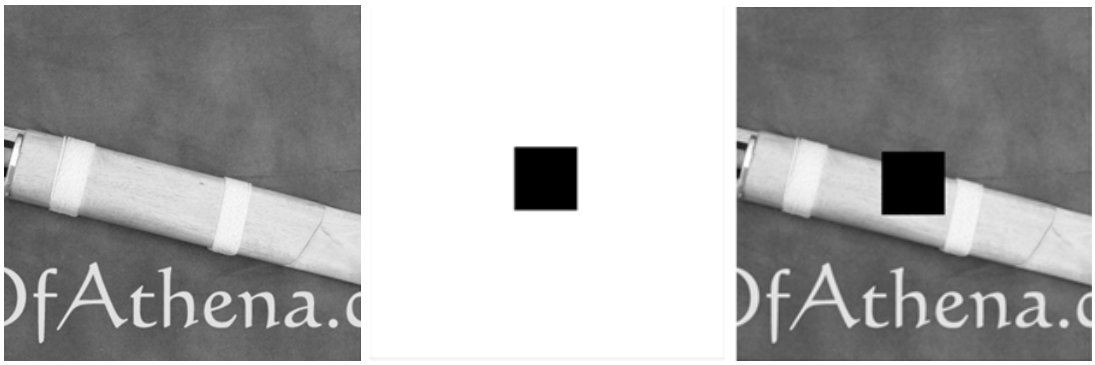
Fig. 5.6. 10*10 blocks destroyed inpainting result.

(a) Origin image. (b) Mask image. (c) Destroyed image.

(d) FMM inpainting result. (e) Proposed diffusion method result. (f) Traditional exemplar method result.

(g) Proposed exemplar method result. (h) K-SVD result. (i) Proposed dictionary learning result.

Fig. 5.7. Line destroyed inpainting result.

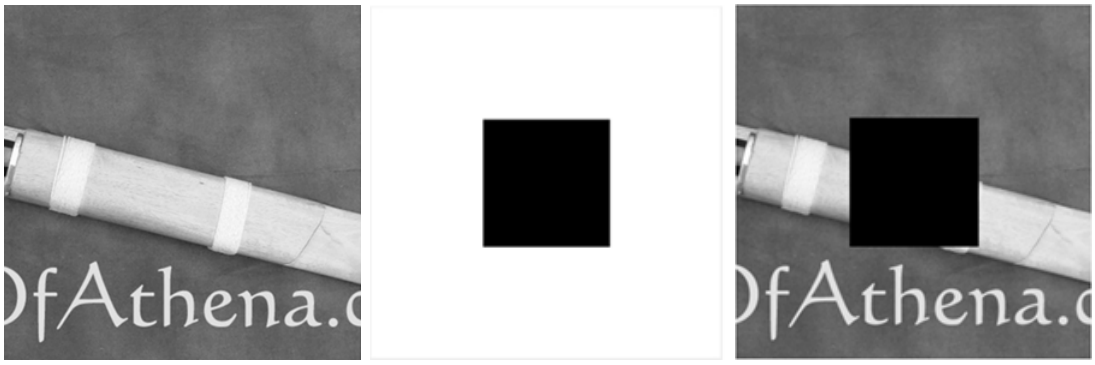


(a) Origin image. (b) Mask image. (c) Destroyed image.

(d) FMM inpainting result. (e) Proposed diffusion method result.

(f) Traditional exemplar method result. (g) Proposed exemplar method result.

Fig. 5.8. Small region destroyed inpainting result.



(a) Origin image. (b) Mask image. (c) Destroyed image.

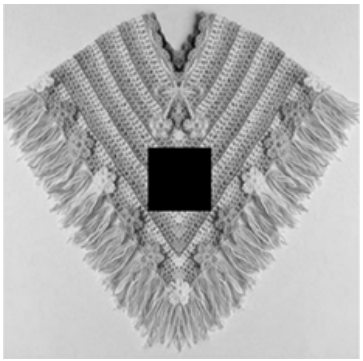
(d) FMM inpainting result. (e) Proposed diffusion method result.

(f) Traditional exemplar method result. (g) Proposed exemplar method result.

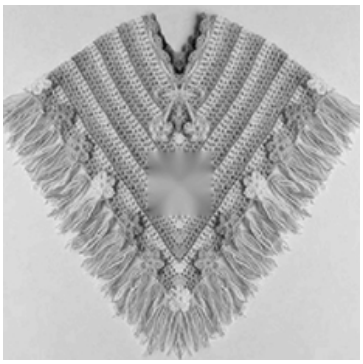
Fig. 5.9. Large region destroyed inpainting result.

(a) Origin image.

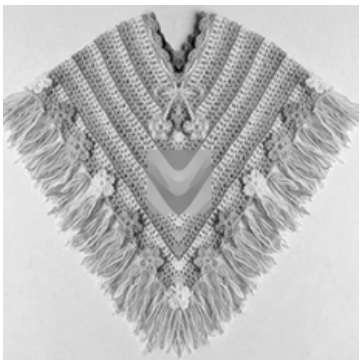
(b) Mask image.



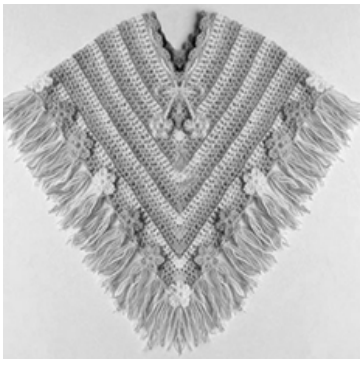
(c) Destroyed image.



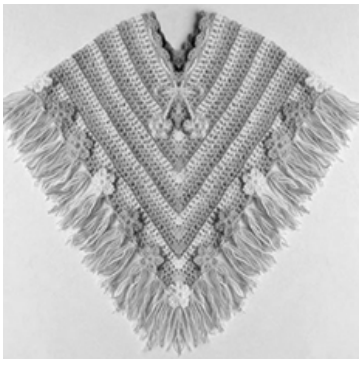
(d) FMM inpainting result.



(e) Proposed diffusion method result.



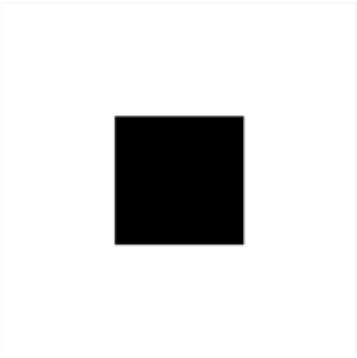
(f) Traditional exemplar method result.



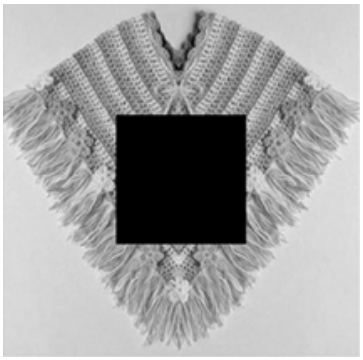
(g) Proposed exemplar method result.

Fig. 5.10. Small region destroyed inpainting result.

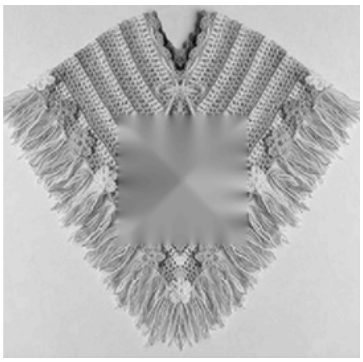
(a) Origin image.



(b) Mask image.



(c) Destroyed image.



(d) FMM inpainting result.

(e) Proposed diffusion method result.

(f) Traditional exemplar method result.

(g) Proposed exemplar method result.

Fig. 5.11. Large region destroyed inpainting result.

Table 5.1. The PSNR of all the inpainting result

	FMM	Proposed Diffusion method	Traditional Exemplar method	Proposed Exemplar method	K-SVD Method	Proposed Dictionary method
pixels 20%	21.36	23.11			25.66	25.75
pixels 30%	20.80	22.73			24.41	24.90
pixels 50%	20.31	22.23			22.90	23.19
blocks 6*6	20.43	20.94	21.79	23.15	21.33	23.38
blocks 10*10	19.27	19.43	20.96	21.80	20.40	23.32
line	23.87	24.82	29.20	30.66	23.27	27.58
region1 90*90	20.66	33.89	34.16	35.72		
region1 185*185	13.24	31.83	32.05	32.83		
region2 90*90	16.63	19.40	21.59	22.96		
region2 185*185	16.62	18.73	18.44	19.33		

learning based methods are almost the same. Because no matter how many the destroyed pixels are the dictionary learning based methods divide the image into patches to inpaint.

2. To inpaint the blocks type and line type destroyed images, when the size of the blocks become 10*10, the result made by diffusion-based inpainting method becomes not clear in some place, but the exemplar-based inpainting method and the dictionary learning based methods keep the edge better.

3. To inpaint the region type destroyed images, diffusion-based inpainting methods almost lost the texture of the image, though in some place the PSNR is not bad. Compared with the FMM, the proposed diffusion-based method improves the edge keeping ability. It has a better visual effect. Exemplar-based methods keep the texture of the image and give a better visual effect. The proposed exemplar-based method selects more significant patches by the improvement and at last gives a better result. Exemplar-based inpainting methods generate a good visual effect and have a higher PSNR.

Table 5.2. The calculation time of all the inpainting result

	FMM	Proposed Diffusion method	Traditional Exemplar method	Proposed Exemplar method	K-SVD Method	Proposed Dictionary method
pixels 20%	16.26	25.33			481.39	582.35
pixels 30%	22.24	32.81			488.71	602.79
pixels 50%	30.47	42.38			502.60	623.53
blocks 6*6	10.48	22.34	59.28	63.46	491.14	606.48
blocks 10*10	21.37	48.98	105.33	127.46	492.80	612.32
line	13.57	24.63	60.29	64.66	610.67	722.18
region1 90*90	7.67	30.25	66.16	87.76		
region1 185*185	27.33	119.67	149.76	186.67		
region2 90*90	7.53	49.22	71.49	102.69		
region2 185*185	28.70	167.52	162.80	269.91		

As a conclusion, the diffusion-based inpainting methods are more suitable to be used in a condition that it is hopeful to get the result in less time. Exemplar-based inpainting methods are suitable to inpaint the images which has a large destroyed region. Dictionary learning based methods also keep the texture of the image and they are more suitable to inpaint the image which contains destroyed pixels all over the images.

Chapter 6

Conclusion

In this paper the research on image inpainting technology is conduct in three fields, diffusion-based field, exemplar-based field and dictionary learning based field.

In diffusion-based field, the traditional FMM method has a less calculation time but it is short for contributing an optimal edge result. Two improving ideas are adapt to the traditional FMM in-painting algorithm. To overcome the disadvantage and remain the advantage of traditional FMM method, Gradient matrix was calculated in the known region to help to select the less but more related pixels to join into the FMM calculation. On the other hand, the gradient matrix of the detected edges makes a contribution to predicting the edges in the destroyed region. And the predicted edges is useful for the inpainting. By the experiment, the resulting images seem more reasonable and more similar to the natural scenes. Compared with the traditional method, the proposed approach not only remains the advantage of fast processing speed, but also contributes a better estimation result of edges.

In exemplar-based field, three aspects have been researched to improve the traditional exemplar-based method. They are the inpainting priority, the size of the patch and the patch matching method. These three aspects are closely correlated to obtain the most similar patch in the inpainting process. From the experimental result it is shown obvious that the improvement increased the accuracy of the patch selection process. The proposed method gives a better global visual effect, especially for the images with more structure contents and the destroyed region which has a large width. In dictionary learning based field, the traditional K-SVD method has been analyzed. And the deficiency that the traditional OMP method using a fixed the sparsity in the whole approximation process has been found. To overcome the deficiency, an optimized OMP method using adaptive sparsity is proposed. The experiment result shows that the proposed method keeps better texture information and prevents the blur influence.

In general the applicability of the image inpainting has been strengthened in these three fields: For diffusion-based method, the inpainting effect when inpainting the image contains a large destroyed region has been strengthened. For exemplar-based method, the matching accuracy in the inpainting has been improved. For dictionary learning

based method, the sparse approximation process has been optimized through the adaptive sparsity.

Chapter 7

Future work

In this paper different patch scales make a strong influences into the inpainting processing has been found. Large scale makes the patch contain more texture and structure information; it will make the matching result more accurate. But instead using a large scale to inpaint will make a strong influence to the followed inpainting process. So in the paper the adaptively patch size method is introduced to improve the inpainting result. In this paper the method focused on finding the most suitable scale and calculated the similarities, but it ignored the association between different scale patches in a single inpainting process. In the future, the association between different scale patches will be researched on and the combination of Multi-scale patches will be used to inpaint images.

What' s more, the exemplar-based methods and dictionary learning based methods both inpaint image in patch level. The edge between the patches from different inpainting processes more or less brings some unnatural feelings. In the research image fusion technology is used to reduce this bad influence but not completed. In the future this attempt will continue.

Acknowledgements

First and foremost, I would like to show my deepest gratitude to my supervisor, Prof. Zhang, a respectable, responsible and resourceful scholar, who has provided me with valuable guidance in every stage of the writing of this thesis. Without his enlightening instruction, impressive kindness and patience, I could not have completed my thesis. His keen and vigorous academic observation enlightens me not only in this thesis but also in my future study.

I shall extend my thanks to Prof. Serikawa for all his kindness and help. I would also like to thank all my teachers and the students who have helped me to develop the fundamental and essential academic competence. My sincere appreciation also goes to the teachers and students of KIT, who participate this study with great cooperation.

Last but not least, I'd like to thank all my friends, for their encouragement and support.

REFERENCES

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image in-painting," in Proc. SIGGRAPH, pp.417-424, 2000.

[2] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "NavierStrokes, fluid dynamics, and image and video in-painting," in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, pp.417-424, 2001.

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

[4] T. Chan and J. Shen, "Non-texture inpainting by curvature-driven diffusions," J. Vis. Commun. Image Represent, vol.4, no.12, pp.436-449, 2001.

[5] C. Bertalmio, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," IEEE Transactions on Image Processing, vol.10, no.8, pp.1200-1211, 2001.

[6] A. Levin, A. Zomet, and Y. Weiss, "Learning how to in-paint from global image statistics," in Proceedings of International Conference on Computer Vision, vol.1, pp.305-313, 2003.

[7] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, vol.2, pp.860-867, 2005.

[8] S. Roth and M. J. Black, "Steerable random fields," in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, pp.1-8, 2007.

[9] A. Telea "An image in-painting technique based on the fast marching method," in Journal of Graphics Tools, vol.9, no.1,pp.23-34, 2004.

[10] Y. Yang, X. Juan, "An improved image in-painting algorithm based on fast marching method," in Journal of Xi'an University of Technology, vol.25, no.2, pp.129-134, 2009.

[11] M. Elad, J. L. Starck, P. Querre, D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," in Applied and Computational Harmonic Analysis, vol.19, no.3, pp.340-358, November 2005.

[12] S. L. Ullo, M. Di Bisceglie, C. Galdi, "A new algorithm for noise reduction and quality improvement in SAR interferograms using inpainting and diffusion," in Geoscience and Remote Sensing Symposium (IGARSS), pp.3602-3605, 2011.

[13] Hongtao Hu, "Application of curvature driven diffusion model in lateral multi-lens video logging image inpainting," in Electronic Measurement and Instruments

(ICEMI), vol.3, pp.1167-1171, 2015.

[14] Sarah Andris, Pascal Peter, Joachim Weickert, "A proof-of-concept framework for PDE-based video compression," in Picture Coding Symposium (PCS), pp.1-5, 2016.

[15] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in Proceedings of International Conference on Computer Vision, pp.1033-1038, 1999.

[16] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image in-painting," in IEEE Trans. Image Process, vol.12, pp.882-889, 2003.

[17] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by examplar-based image inpainting ," in IEEE Trans. Image Process, vol.33, pp. 1200-1212, 2004.

[18] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," in TOG, vol.22, pp.303-312, 2003.

[19] M. Wilczkowiak, G. Brostow, B. Tordoff, and R. Cipolla, "Hole fill through photomontage," in BMVC, pp.492-501, 2005.

[20] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," in TOG, vol.28, pp.1-24, 2009.

[21] B. Wohlberg, "Inpainting by joint optimization of linear combinations of examplars," in IEEE Signal Processing Letters, vol.18, no.1, pp.75-78, Jan 2011.

[22] J. Wu and Q. Ruan, "Object removal by cross isophotes examplar-based image inpainting," in Proc. Int. Conf. Pattern Recognition, pp. 810-813, 2006.

[23] A. Wong and J. Orchard, "A nonlocal-means approach to examplar based inpainting," in the IEEE Int. Conf. Image Processing, pp.2600-2603, 2008.

[24] Idan Ram, Michael Elad, Israel Cohen, "Image Processing Using Smooth Ordering of its Patches," in IEEE Transactions on Image Processing, vol.22, no.7, pp.2764-2774, July 2013.

[25] A. T. Umarani, P. J. Kulkarni, "A novel scheme of color image inpainting by prioritized patches," in Advanced Communication Control and Computing Technologies (ICACCCT), pp.233-237, 2016.

[26] Hui-Yu Huang, Chun-Nan Hsiao, "A patch-based image inpainting based on structure consistence," in Computer Symposium (ICS), pp.165-170, 2010.

[27] Jingsi Ou, Wensheng Chen, Binbin Pan, Yugao Li, "A New Image Inpainting Algorithm Based on DCT Similar Patches Features," in Computational Intelligence and Security (CIS), pp.152-155, 2016.

[28] D. Paredes, P. Rodriguez, "Multi-scale image inpainting with label selection based on local statistics," in the conference proceedings of EUSIPCO, Marrakech, Morocco, pp.1-5, 2013.

[29] Smarti Reel, K. C. P. Wong, Gene Cheung, Laurence S. Dooley, "Disocclusion Hole-Filling in DIBR-Synthesized Images using Multi-Scale Template Matching," in Visual

Communications and Image Processing Conference, pp.494-497, 2014.

[30] V. V. Voronin, A. I. Sherstobitov, K. O. Egiazarian, "Exemplar-based inpainting using local binary patterns," in Proceedings of SPIE - The International Society for Optical Engineering 9019, pp.1-12, January 2014.

[31] R. Martnez-Noriega, A. Roumy, G. Blanchard, "Exemplar-based image inpainting: Fast priority and coherent nearest neighbor search," in Machine Learning for Signal Processing (MLSP), pp.1-6, 2012

[32] Julien Mairal, Michael Elad, Guillermo Sapiro, "Sparse Representation for Color Image Restoration," in IEEE Transactions on Image Processing, vol.17, no.1, pp.53-69, Jan. 2008.

[33] Ron Rubinstein, Michael Zibulevsky, Michael Elad, "Double Sparsity: Learning Sparse Dictionaries for Sparse Signal Approximationin,"in IEEE Transactions on Signal Processing, vol.58, no.3, pp.1553-1564, April 2010.

[34] Zongben Xu, Jian Sun, "Image Inpainting by Patch Propagation Using Patch Sparsity," in IEEE Transactions on Image Processing, vol.19, no.5, pp.1153-1165, May 2010.

[35] Somayeh Hesabi, Nezam Mahdavi-Amiri, "A modified patch propagation-based image inpainting using patch sparsity," in Artificial Intelligence and Signal Processing (AISP), pp.43-48, 2012.

[36] Leslie N. Smith, Michael Elad, "Improving Dictionary Learning: Multiple Dictionary Updates and Coefficient Reuse," in IEEE Signal Processing Letters, vol.20, no.1, pp.79-82, Jan. 2013.

[37] Tijana Ruzic , Aleksandra Pizurica, "Context-Aware Patch-Based Image Inpainting Using Markov Random Field Modeling ," in IEEE Transactions on Image Processing, vol.24, no.1, pp.444-456, Jan. 2015.

[38] Vardan Papyan, Michael Elad, "Multi-Scale Patch-Based Image Restoration," in IEEE Transactions on Image Processing, vol.25, no.1, pp.249-261, Jan. 2016.

[39] M. J. Fadili, J. L. Starck, F. Murtagh, "Inpainting and Zooming Using Sparse Representations," in Comput Journal, vol.52, no.1, pp.64-79, 2009.

[40] J. Sun, M. F. Tappen, "Learning non-local range Markov Random field for image restoration," in Computer Vision and Pattern Recognition, vol.32, no.14, pp.2745-2752, 2011.

[41] Hong Wang, "Inpainting of Potala Palace murals based on sparse representation," in Biomedical Engineering and Informatics (BMEI), pp.737-741, 2015.

[42] Asim Munawar, Clement Creusot, "Structural inpainting of road patches for anomaly detection," in Machine Vision Applications (MVA), 14th IAPR International Conference on, pp.41-44, 2015.

[43] Huimin Lu, Yin Zhang, Yujie Li, Quan Zhou, Ryunosuke Tadoh, Tomoki Uemura, Hyoungseop Kim, Seiichi Serikawa, "Depth Map Reconstruction for Underwater Kinect

Camera Using Inpainting and Local Image Mode Filtering," in IEEE Access, vol.5, pp.7115-7122, 2017.

[44] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," In arXiv:1508.06576v2, 2015.

[45] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks," In NIPS, 2015.

[46] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," In arXiv:1601.04589v1, 2016.

[47] A. J. Champandard, "Semantic style transfer and turning two-bit doodles into fine artwork," In arXiv:1603.01768v1, 2016.

[48] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images." In arXiv:1603.03417v1, 2016.

[49] J. Johnson, A. Alahi, and F. Li, "Perceptual losses for real-time style transfer and super-resolution," In arXiv:1603.08155v1, 2016.

[50] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. Efros, "Context encoders: Feature learning by inpainting," In CVPR, pp.2536-2544, 2016.

[51] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," In NIPS, pp.2672-2680, 2014.

[52] Michal Aharon, Michael Elad, Alfred Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," in IEEE Transactions on Signal Processing, vol.54, no.11, pp.4311-4322, 2006.

[53] Ron Rubinstein, Alfred M. Bruckstein, Michael Elad, "Dictionaries for Sparse Representation Modeling," in Proceedings of the IEEE, vol.98, no.6, pp.1045-1057, 2010.