An image inpainting method for object removal based on difference degree constraint

Lei Zhang¹ · Minhui Chang¹

Received: 21 February 2020 / Revised: 11 August 2020 / Accepted: 9 September 2020 /
Published online: 30 September 2020
© The Author(s) 2020

Abstract

In the inpainting method for object removal, SSD (Sum of Squared Differences) is commonly used to measure the degree of similarity between the exemplar patch and the target patch, which has a very important impact on the restoration results. Although the matching rule is relatively simple, it is likely to lead to the occurrence of mismatch error. Even worse, the error may be accumulated along with the process continues. Finally some unexpected objects may be introduced into the target region, making the result unable to meet the requirements of visual consistency. In view of these problems, we propose an inpainting method for object removal based on difference degree constraint. Firstly, we define the MSD (Mean of Squared Differences) and use it to measure the degree of differences between corresponding pixels at known positions in the target patch and the exemplar patch. Secondly, we define the SMD (Square of Mean Differences) and use it to measure the degree of differences between the pixels at known positions in the target patch and the pixels at unknown positions in the exemplar patch. Thirdly, based on MSD and SMD, we define a new matching rule and use it to find the most similar exemplar patch in the source region. Finally, we use the exemplar patch to restore the target patch. Experimental results show that the proposed method can effectively prevent the occurrence of mismatch error and improve the restoration effect.

Keywords Image inpainting · Object removal · Exemplar · Difference degree

1 Introduction

Image inpainting derives from the restoration of damaged artworks [5]. Its basic idea is to use the undamaged and effective information to restore the damaged regions according to certain rules [11, 30]. Its main purpose is to make the restored image meet the requirements of human vision, so that people who are not familiar with the original image cannot notice the restoration trace [17, 33]. With the rapid development of computer and multimedia technology, the inpainting technology has been widely used in many fields [20], such as

✉ Lei Zhang
zhanglei@ycu.edu.cn

¹ School of Mathematics and Information Technology, Yuncheng University, Yuncheng, China

scratches restoration of old photos and precious literatures, protection of cultural relics [8, 18], robot vision, film and television special effects production, and so on [3, 37].

At present, the existing inpainting approaches can be classified into three categories. The first is based on the Partial Differential Equation (PDE). Its basic idea is that the missing region is filled smoothly by diffusing the effective information from the undamaged region into the damaged region at the pixel level [35]. The representative approaches include the BSCB model [5], the TV model [7], and the CDD model [6]. For the small and non-textured damaged regions, these approaches can achieve convincingly excellent result, but for the large and textured missing regions they tend to induce over-smooth effect or stair-case effect.

The second is based on the exemplar, which is also commonly used method at present. Its basic idea is that the missing region is restored using the similar patch in a visually plausible manner at the patch level. The representative approaches include the non-parametric sampling approach [12], the exemplar-based approach [9], the patch-sparsity-based approach [39], and the nonlocal-means approach [38]. The advantage of these approaches is they can obtain satisfactory results in restoring the large damaged regions. However, they also have some disadvantages, such as unreasonable restoration order, mismatch error and error accumulation, low efficiency due to greedy search, and so on.

The third is based on the sparse representation. Its basic idea is to calculate the sparse representation coefficient of the damaged patch on the over-complete dictionary, and then restore the damaged patch according to the sparse representation coefficient and the over-complete dictionary. The representative approaches include the K-SVD-based approach [1] and the MCA-based approach [13]. These approaches can achieve convincing results in filling missing pixels and restoring scratches. However, due to the fact that the methods based on sparse representation are still in the initial stage, and the over-complete dictionary has a great influence on the restoration results, the adaptability of over-complete dictionaries needs to be further improved.

From the perspective of image inpainting, removing object from the image means that we need to restore the large-scale damaged regions [14, 46], and the most commonly used method is based on exemplar [4, 10]. For each damaged patch, it searches for the similar exemplar patch in the undamaged region according to the matching rule, and then uses the exemplar patch to replace the damaged patch. A series of studies have shown that, the method can achieve satisfactory restoration effect. However, it also suffers from some shortcomings. For example, it uses the SSD (Sum of Squared Differences) to measure the degree of similarity between target patch and exemplar patch. Although the matching rule is simple, it may lead to a mismatch error, i.e., the damaged patch is replaced by an unsuitable patch. Even worse, the error will be continually accumulated along with the process progresses. Finally some undesired objects will be introduced into the restored region, and the restoration results cannot meet the requirements of human vision.

In view of these problems, we propose an image inpainting method for object removal based on difference degree constraint. Compared with the previous methods, our main contributions are as follows:

1. We define the MSD (Mean of Squared Differences) between the target patch and the exemplar patch, and use it to measure the degree of differences between corresponding pixels at known positions (i.e., the pixels that already exist) in the target patch and the exemplar patch.
2. We define the SMD (Square of Mean Difference) between the target patch and the exemplar patch, and use it to measure the degree of differences between the pixels at

- known positions (i.e., the pixels that already exist) in the target patch and the pixels at unknown positions (i.e., the pixels that will be used to fill) in the exemplar patch.
3. Based on MSD and SMD, we define a new matching rule, and the exemplar patch with the smallest sum of MSD and SMD is selected as the most similar patch. In this way, we can effectively prevent the occurrence and accumulation of mismatch error, and improve the restoration effect.

2 Related works

The PDE-based methods solve the partial differential equations and make the effective information smoothly propagate to the damaged region along the direction of isophote. Rathish et al. [31] used the square of the L2 norm of Hessian of the image as regularization term, used convexity splitting to solve the resulting semi-discrete scheme in Fourier domain. Yang et al. [40] utilized the newly defined fractional-order structure tensor to control the regularization process. The new model can inherit genuine anisotropism of tensor regularization, and better handle subtle details and complex structures. Theljani et al. [36] based on a fourth-order variational model, used an adaptive selection of the diffusion parameters to optimize the regularization effects in the neighborhoods of the small features. Mousavi et al. [28] considered the effect of spectrum and phase angle of the Fourier transform, generated two regularization parameters and had two degree of freedom, so as to restore an image. These methods can be used to restore small-scale damaged regions, such as removing scratches, removing text coverage, filling holes, and so on.

The exemplar-based methods mainly search for the most similar exemplar patch in the undamaged source region, and then use it to restore the corresponding damaged target patch. These methods are commonly used to remove object from images. Liu et al. [23] used structural similarity index measure (SSIM), obtained the best candidate patch by four cases of rotation and inversion, so as to find the most similar exemplar patch. Isogawa et al. [21] noted the important effect of the mask on the restoration results, proposed a mask optimization method and used it to automatically obtain good results. Wong et al. [38] introduced the idea of non-local mean in image denoising into image inpainting, and used the mean of a number of exemplar patches to replace the damaged patch. However, it may lead to the loss of texture details, and induce the over-smooth phenomenon in restored regions. Shen et al. [34] directly selected some patches from original image, formed the over-complete dictionary, and restored the damaged images based on sparse representation. For smooth images, the method can obtain satisfactory restoration effect. But for texture images, it may lead to loss of some texture details. Liu et al. [24] modified the confidence term into an exponential form, computed the sum of confidence term and data term to make the filling order more reasonable. Zhang et al. [47] used the information of curvature and gradient to replace the data term, to improve the filling order. However, they did not improve the matching rule, so there may be a mismatch error between patches in the inpainting process. Nan et al. [29] set different weights for data item and confidence item according to the golden section, so that the restoration order is more reasonable, but it cannot effectively prevent the occurrence of mismatch error, and the restoration effect needs to be improved. Yao [41] introduced the correlation between the target patch and the neighborhood patch into the priority calculation, and modified the multiplication to addition. Besides, she defined a new similarity calculation function to improve the restoration effect. Ghorai et al. [15] proposed a Markov Random Field (MRF)-based image inpainting method. They used a novel group

formation strategy based on subspace clustering to search the candidate patches in relevant source region only, and adopted an patch refinement scheme using higher order singular value decomposition to capture underlying pattern among the candidate patches. Zhang et al. [43] used a surface fitting as the prior knowledge, and used a Jaccard similarity coefficient to improve the matching precision between patches. These exemplar-based methods have attracted the attention of many researchers, and various improved methods have been continuously proposed.

The sparse-representation-based methods mainly use over-complete dictionary and sparse coding to reconstruct damaged pixels. Zhang et al. [44] classified the patches according to local feature. For smooth patch, it was restored by over-complete dictionary and sparse coding. For texture patch, it was restored by exemplar patch. Mo et al. [27] used self-adaptive group structure and sparse representation to solve the problems that object structure discontinuity and poor texture detail occurred in image inpainting method, and achieved better restoration results. Hu et al. [19] combined Criminisi method and sparse representation method, used sparse representation instead of searching for the most similar exemplar patch in Criminisi algorithm. Zhang et al. [45] monitored the restoration process. When there was a mismatch between the exemplar patch and the target patch, they calculated the sparse coding of the target patch on the discrete cosine dictionary, then reconstructed the target patch using the over-complete dictionary and sparse coding, and obtained a better restoration effect. However, the application of sparse representation in image inpainting is still in its infancy, its representation model and the adaptability of over-complete dictionary needs to be further improved.

It should be noted that in recent years, with the rapid development of deep learning, researchers have applied it to various fields of computer vision [25], such as object segmentation [26], object detection, saliency detection and so on. In terms of image inpainting, Goodfellow et al. [16] proposed a Generative Adversarial Networks (GAN). They used a large number of real images to train the generative model and the discriminative model, so that the deep network can learn the feature distribution of the real images. Finally, the generator can be used to automatically generate images that are very similar to the real images. Sagong et al. [32] proposed a fast PEPSI (Parallel Extended-decoder Path for Semantic Inpainting) model. It can reduce the number of convolution operations by adopting a structure consisting of a single shared encoding network and a parallel decoding network with coarse and inpainting paths. Zeng et al. [42] proposed a PEN-Net (Pyramid-context ENcoder Network). It used a U-Net structure to restore an image by encoding contextual semantics from full resolution input, and decoding the learned semantic features back into images. Besides, it used a pyramid-context encoder to progressively learn region affinity by attention from a high-level semantic feature map and transfer the learned attention to the previous low-level feature map. Jiang et al. [22] used a generator, a global discriminator, and a local discriminator to design the network model, and generated more realistic restoration results. In addition, there are seven papers on image inpainting based on deep learning at the CVPR 2020.

3 Definitions of MSD and SMD

3.1 Notations

For easy understanding, we adopt same notations used in [9]. As shown in Fig. 1, Ω is the target region (i.e., the damaged region) which will be removed and filled, Φ is the source

Fig. 1 Notation diagram

region (i.e., the undamaged region), it may be defined as the entire image I minus the target region (i.e., $\Phi = I - \Omega$). $\partial\Omega$ denotes the boundary of the target region Ω . Suppose that the patch Ψ_p centered at the point $p (p \in \partial\Omega)$ is to be filled. Given the patch Ψ_p , n_p is the unit vector orthogonal to the boundary $\partial\Omega$ and $\nabla I_p^\perp$ is the isophote at the point p .

3.2 Mismatch error

In the traditional exemplar-based method, when the target patch Ψ_p is determined, it searches for the exemplar patch which is most similar to Ψ_p according to the following matching rule [9]:

$$\Psi_{\hat{q}} = \arg \min_{\Psi_q \in \Phi} ssd(\Psi_p, \Psi_q) \quad (1)$$

where $ssd(\Psi_p, \Psi_q)$ is defined as the SSD (Sum of Squared Differences) of the already existing pixels between the two patches.

Although the matching rule is simple, there is a risk that the target patch is restored by an unsuitable exemplar patch. Even worse, the error will be continually accumulated along with the process progresses, which may introduce some undesired and unexpected objects into the target regions. To better illustrate, in Fig. 2 we show the restoration process when removing a target from an image according to this matching rule, where the first (i.e., the first row and the first column) is the original image. The second is the object to be removed, which is marked in green. The third is the inpainting mask. From the fourth to the last one (i.e., the fourth row and the fifth column) are the restoration process of object removal. It should be noted that we have saved a total of 371 images in the simulation experiment. In order to simplify the display, we only show 17 of them here. But from these images, we can still clearly see the restoration process of the target region.

From Fig. 2 we can see that, at the beginning the target region where the object is located is gradually reduced along with the process progresses. However, in the image in row 2 and column 3, a small part of the arm of another contestant is copied into the target region, which means that the target patch has been replaced by an unsuitable exemplar patch, i.e. there is a mismatch error between the two patches. After that, the arm part in the target region is gradually enlarged, which means that the mismatch error has been accumulated along with the process progresses. Finally, an unexpected object is introduced into the restoration image, and make the result cannot meet the requirements of human vision, as shown in the last image.

Fig. 2 Restoration process when removing a target from an image according to the original matching rule

Through the above analysis, we think the situations in which mismatch error is likely to occur can be summarized into two categories. The one is the differences between the pixels that already exist in target patch and the corresponding pixels in exemplar patch are relatively large. The other is the differences between the pixels that already exist in the target patch and the pixels that will be used for filling are relatively large. Under these situations, if the target patch is replaced by the exemplar patch, it is likely to lead to the occurrence of mismatch error.

3.3 Definition of MSD

On the one hand, if the differences between the already existing pixels in target patch and the corresponding pixels in exemplar patch are relatively large, the mismatch error is likely to occur. According to this situation, we define the MSD and use it to measure the degree of differences between the two patches. It is defined as:

$$MSD(\Psi_p, \Psi_q) = \frac{\sum |\bar{M}\Psi_p - \bar{M}\Psi_q|^2}{\sum \bar{M}} \quad (2)$$

where Ψ_p is the target patch and Ψ_q is the exemplar patch. M is the binary mask, it uses 1 to indicate the pixels that need to be filled, and uses 0 to indicate the pixels that already exist. $\bar{M}\Psi_p$ extracts the pixels that already exist in the target patch Ψ_p , $\bar{M}\Psi_q$ extracts corresponding pixels in the exemplar patch Ψ_q . In short, MSD calculates the average of the squared differences between corresponding pixels at known positions in two patches, which is used to measure the degree of difference between the two patches.

According to the (2), if the value of MSD is small, it means the already existing pixels in the two patches are very similar. On the contrary, if the value of MSD is large, it means the already existing pixels in the two patches are very different. In this situation, when using the exemplar patch to restore the target patch, the mismatch error is likely to occur.

3.4 Definition of SMD

On the other hand, if the differences between pixels that already exist in the exemplar patch and pixels that will be used for filling are relatively large, the mismatch error is still likely to occur. According to this situation, we define the SMD and use it to measure the degree of differences between the two patches. It is defined as:

$$SMD(\Psi_p, \Psi_q) = \left| \frac{\sum \bar{M}\Psi_p}{\sum \bar{M}} - \frac{\sum M\Psi_q}{\sum M} \right|^2 \quad (3)$$

where Ψ_p is the target patch and Ψ_q is the exemplar patch, M is the binary mask. $\sum \bar{M}\Psi_p / \sum \bar{M}$ calculates the average of pixels that already exist in the target patch. $\sum M\Psi_q / \sum M$ calculates the average of pixels that will be used for filling, these pixels are located at unknown positions in the exemplar patch. In short, SMD is used to measure the degree of difference between the pixels that already exist and the pixels used to fill.

According to (3), if the value of SMD is small, it means the already existing pixels in target patch and the pixels that will be used for filling are very similar. On the contrary, if the value of SMD is large, it means the already existing pixels and the pixels that will be used for filling are very different. In this situation, when using the exemplar patch to restore the target patch, the two parts of the restored patch are very different, and the mismatch error is still likely to occur.

3.5 Definition of matching rule

Based on the above definitions of MSD and SMD, we define a new matching rule in the proposed method, and use it to measure the degree of similarity between exemplar patch and target patch. It is defined as:

$$\Psi_{\hat{q}} = \arg \min_{\Psi_q \in \Phi} MSD(\Psi_p, \Psi_q) + SMD(\Psi_p, \Psi_q) \quad (4)$$

To better illustrate the effectiveness of the matching rule defined in (4), we show a simple example in Fig. 3. We synthesize a binary test image, which consists of 9 small squares, and each small square is 9×9 in size, as shown in (a). In (b), we specify a target patch in this image, which is located in the center of the image. In addition, we also specify the damaged region in the target patch, which is marked by green and its size is 3×9 .

According to the matching rule defined in (1), the similar exemplar patch can be found, which is marked by red dotted line in (c), because the SSD of the patch is 0. Finally, using the exemplar patch to restore the target patch, we obtain the restored image, as shown in (e). It can be seen that significant visual inconsistencies appear in the result image.

According to the matching rule defined in (4), the MSD of the exemplar patch which is marked by red dotted line in (c) is 0. However, the SMD of the patch is 65025. Therefore, the

Fig. 3 Comparison of matching rules defined in (1) and (4)

sum of MSD and SMD of the patch in (c) is 65025. In comparison, the MSD of the exemplar patch which is marked by blue dotted line in (d) is 0, and the SMD of the patch is still 0. Therefore, the sum of MSD and SMD of the patch in (d) is 0. Since 0 is less than 65025, the exemplar patch in (d) is determined to be a best similar exemplar patch. Using the exemplar patch to restore the target patch, we obtain the restored image, as shown in (f). It can be seen that the restoration result can meet the requirements of human visual consistency.

4 Proposed method

4.1 Priority computation

In the image inpainting, the filling order is determined by the priority values. In the proposed method, we use the method in [9] to compute the patch priority, because it can make the patches which are on the continuation of strong edges get higher priority values, and maintain the continuity of the structure.

For each patch Ψ_p centered at the point p has a patch priority, it is defined as:

$$P(p) = C(p) \times D(p) \quad (5)$$

where $C(p)$ is the confidence term and $D(p)$ is the data term. Confidence term $C(p)$ indicates how many existing pixels are there in the target patch. It is defined as:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \Phi} C(q)}{|\Psi_p|} \quad (6)$$

where $|\Psi_p|$ is the area of the patch, that is, the number of pixels contained in the patch. Therefore, the more pixels already exist in a patch, the greater its confidence term. During the initialization, $C(p)$ is set as:

$$C(p) = \begin{cases} 0 & \forall p \in \Omega \\ 1 & \forall p \in \Phi \end{cases} \quad (7)$$

Data term $D(p)$ indicates how strong the isophote hitting the boundary is. It is defined as:

$$D(p) = \frac{|\nabla_p^\perp \cdot n_p|}{\alpha} \quad (8)$$

where α is a normalization factor. Therefore, the smaller the angle between the vector $\nabla_p^\perp$ and the vector n_p of a patch, the larger its data term.

4.2 Patch matching

Once all priorities have been calculated, we find the target patch Ψ_p with the highest priority, and search for the exemplar patch which is most similar to it in the source region according to the matching rule.

In the proposed method, we use the matching rule in (4) instead of (1) to search for the exemplar patch $\Psi_{\hat{q}}$.

4.3 Patch restoration

When the exemplar patch $\Psi_{\hat{q}}$ is found, we use it to restore the current target patch Ψ_p as follows:

$$M\Psi_p = M\Psi_{\hat{q}} \quad (9)$$

4.4 Algorithm description

In order to describe the proposed method more clearly, we show its flow chart in Fig. 4.

Here, the proposed algorithm steps are described in detail as follows:

- Step 1. Identify the target regions. The target regions are indicated and extracted according to the inpainting mask;
- Step 2. Calculate the patch priorities according to (5), and select the target patch Ψ_p with the highest priority;

Fig. 4 Flow chart of the proposed method

- Step 3. For each exemplar patch, calculate the MSD and the SMD according to (2) and (3), and find the most similar patch $\Psi_{\hat{q}}$ according to (4);
- Step 4. Use exemplar patch $\Psi_{\hat{q}}$ to restore the target patch Ψ_p according to (9);
- Step 5. Update the value of confidence term according to the following equation:

$$C(\hat{p}) = C(p) \quad \forall \hat{p} \in \Psi_p \cap \Omega \quad (10)$$

The algorithm iterates the above steps until all pixels in target region have been filled.

5 Experimental results

In order to verify the effectiveness and feasibility of the proposed method, we select a number of natural images from the BSDS dataset [2] in the experiment. All the experiments are run on the computer with the configuration of 2.7GHz processor and 2GB RAM.

For each image, we specify a target object, and then restore the region where it is located, to achieve the goal of removing the object. For better comparison and analysis, in the experiment, we use the method in [9], the method in [34], the method in [24], the method in [47], the method in [29], the method in [41], and the proposed method to restore each image, respectively. Finally, we compared and analyzed the restoration results qualitatively and quantitatively.

In the following figures, we show the restoration results of different methods in two groups: smooth images and texture images. For each Figure, (a) is the original image, (b) is the object to be removed, which is marked by green, (c) is the result of method in [9], (d) is the result of method in [34], (e) is the result of method in [24], (f) is the result of method in [47], (g) is the result of method in [29], (h) is the result of method in [41], and (i) is the result of the proposed method. For better comparisons, in the restoration results of each method, we marked the target region with a white rectangle.

5.1 Qualitative analysis

In Figs. 5, 6 and 7, we show the restoration results of three smooth images. To better distinguish, we name each image separately. In Fig. 5, we remove an eagle from the image, so we name the image as “eagle”. In Fig. 6, we remove the polar bear from the image, and we name the image as “polar bear”. In Fig. 7, we remove a goose from the image, and we name the image as “goose”.

In Figs. 8, 9 and 10, we show the restoration results of three texture images, which contain bright colors and rich textures. In Fig. 8, we remove the rider on the left side of the image, and we name the image as “rider”. In Fig. 9, we remove the stone column from the image, and we name the image as “stone column”. In Fig. 10, we remove the tree from the hillside, and we name the image as “hillside”.

As can be seen from the above figures, in the results of method in [9], there are often some undesired and unexpected objects in the target regions. For example, a small portion of another eagle was copied to the target region in Fig. 5 (c); an unexpected black object appeared in the target region in Fig. 6 (c); the head of the other goose was copied to the target region in Fig. 7 (c); part of the red coat was copied to the target region in Fig. 8 (c); a small part of the stone bench was copied to the target region in Fig. 9 (c); part of the blue lake appeared in the target region in Fig. 10 (c). The reason is that it uses the SSD to measure the degree of similarity between two patches, which may result in that the target patch

Fig. 5 Restoration results of “eagle”

Fig. 6 Restoration results of “polar bear”

Fig. 7 Restoration results of “goose”

is restored by an unsuitable exemplar patch, and the mismatch error may be continually accumulated along with the process progresses, eventually lead to the occurrence of the above situation.

In the method in [34], it used the sparse representation to restore the damaged patches, for smooth images, it can obtain satisfactory restoration effect, as shown in Figs. 5 (d), 6 (d), and 7 (d). However, for texture images, texture details may be lost, resulting in over-smooth phenomenon in the target region. For example, lots of grass details are lost in the target region in Fig. 8 (d), there is a serious over-smooth phenomenon in the target region in Fig. 9 (d), and the details of the grass on the hillside are also lost in the target region in Fig. 10 (d). The reason is that the over-complete dictionary lacks adaptability to different textures, and this method only approximately reconstructs of image patches, so the restoration effect of texture image needs to be further improved.

The method in [24] modified the confidence term into an exponential form and computed the sum of confidence term and data term to make the filling order more reasonable, which can improve the restoration effect to some extent, as shown in Figs. 6 (e), 7 (e), and 8 (e), there are no isolated, trivial objects in the target regions. However, a small portion of another eagle was introduced into the target region in Fig. 5 (e). Parts of the stone bench were copied to the target region in Fig. 9 (e). Part of the blue lake appeared in the target region in Fig. 10 (e). The reason is that this method only improves the filling order and does not improve the matching rules, so sometimes the mismatch error occurs during restoration process.

Fig. 8 Restoration results of “rider”

The method in [47] can obtain better restoration results. For example, no superfluous objects are introduced into the target area in Figs. 5 (f), 6 (f), 8 (f), and 10 (f). The reason is that it used the information of curvature and gradient to replace the data term, which make the filling order more reasonable to a certain extend. However, it still has some problems. For example, part of another goose’s wings was copied into the target region in Fig. 7 (f), and part of stone bench was copied into the target region in Fig. 9 (f). The reason is that this method can make the filling order more reasonable, but it cannot effectively prevent occurrence of mismatch error.

The method in [29] changed the weights of data item and confidence item according to the golden section, which can make the restoration order more reasonable and obtain better results, as shown in Figs. 6 (g) and 8 (g). However, a small portion of another eagle was copied to the target region in Fig. 5 (g). A small part of the head of another goose was copied to the target region in 7 (g). A small part of the stone bench was copied to the target region in Fig. 9 (g), and part of the blue lake appeared in the target region in Fig. 10 (g). The reason is that this method only improves the filling order, but does not further improve the matching rule.

The method in [41] modified the priority calculation and defined a new matching rule, so it can achieve better restoration effect, and no undesired objects were introduced into target region in all these images. However, it added a distance factor constraint in the matching rule, which makes it always tend to select those exemplar patches that are closer to the target patch. Therefore, for the image with rich texture, those texture details that are closer to the target region may be duplicated many times, resulting in excessive texture repetition in the target region. For example, it achieved satisfactory results in Figs. 5 (h), 6 (h), and 7 (h). But

Fig. 9 Restoration results of "stone column"

Fig. 10 Restoration results of "hillside"

in Fig. 8 (h), there are some continuous repetitions of grass in the target region. In Fig. 9 (h), there is excessive repetition of some branches in the target region. In Fig. 10 (h), there are multiple repetitions of a part of the lawn in the target region.

Compared with the other methods, the proposed method can achieve satisfactory results, and there are no undesired and unexpected objects in the target regions. The reason is that, it not only measures the degree of differences between corresponding pixels at known positions in the target patch and the exemplar patch, but also measures the degree of differences between the pixels at known positions in the target patch and the pixels at unknown positions in the exemplar patch, which can effectively prevent the occurrence of mismatch error, and make the results meet the requirements of human vision.

5.2 Quantitative analysis

Through the analysis in Section 3.2, we know that if the MSD of exemplar patch is relatively large or the SMD of exemplar patch is relatively large, we consider the mismatch error is likely to occur. So in the proposed method, we define a new matching rule as shown in Eq (4), the exemplar patch with the smallest sum of MSD and SMD is selected as the most similar exemplar patch. Therefore, in order to quantitatively analyze and compare various methods, we separately calculated the sum of MSD and SMD of each matched patch of each method during the restoration process, and used it to verify whether the sum of MSD and SMD in our method was relatively small compared with other methods.

However, for each image, each method contains a lot of matched patches in the restoration process. Moreover, in each method, the sum of MSD and SMD of the vast majority of matched patches are very small. Therefore, for each image, if the values of all matched patches of all methods are placed in the same chart, the distribution of data will be relatively centralized and cannot be clearly distinguished. For example, in the restoration process of rider, the sum of MSD and SMD of each matched patch of each method are shown in Fig. 11. As can be seen from it, the vast majority of the values in each method are relatively small, and their distribution is relatively concentrated, so we cannot distinguish the values of each method well and cannot effectively perform quantitative comparisons.

Based on the above analysis, in order to better distinguish the data distribution of each method and compare the performance of each method more clearly, we extracted the largest 10 values of each method for quantitative analysis, as shown in Fig. 12, where (a) is the

Fig. 11 Sum of MSD and SMD of each matched patch of each method in the restoration process of “rider”

Fig. 12 Quantitative comparison of each method in each image, where each sub-figure shows the largest 10 values (sum of MSD and SSD) of each method for each image

largest 10 values of each method in restoration process of “eagle”. (b) is the largest 10 values of each method in restoration process of “polar bear”. (c) is the largest 10 values of each method in restoration process of “goose”. (d) is the largest 10 values of each method in restoration process of “rider”. (e) is the largest 10 values of each method in restoration process of “stone column”. (f) is the largest 10 values of each method in restoration process of “hillside”.

In Fig. 12 (a), the values of method in [9, 24], and [29] are relatively large, the values of method in [41, 47], and proposed method are relatively small. The distribution of these data coincides with the restoration results shown in Fig. 5. In Fig. 5 (c), (e), and (g), unexpected objects are introduced into the target region, while in Fig. 5 (f), (h), and (i), no undesired objects are introduced into the target region.

In Fig. 12 (b), only the values of method in [9] are relatively large, and the values of the other five methods are relatively small. What exactly confirms with this situation is that in Fig. 6 (c), an unexpected black object appeared in the target region, and the restoration effect of the other five methods can meet the requirements of human visual consistency.

In Fig. 12 (c), the values of method in [9, 47], and [29] are relatively large while the values of method in [24, 41], and proposed method are relatively small. The distribution of these data is exactly consistent with the restoration results in Fig. 7. Parts of the body of another goose were copied to the target region in Fig. 7 (c), (f), and (g), while no undesired objects were introduced into the target region in Fig. 7 (e), (h), and (i).

In Fig. 12 (d), only the values of method in [9] are relatively large, while the values of the other five methods are relatively small. What exactly confirms with this situation is that, part of the red coat was copied to the target region in Fig. 8 (c), while the other methods can obtain better restoration results.

In Fig. 12 (e), the values of method in [47] are the largest, followed by the values of methods in [9, 24, 29, 41], and the values in proposed method are the smallest. Corresponding to this situation is that, the size of unexpected object in Fig. 9 (f) is the largest, followed by the size of unexpected object in Fig. 9 (c), (e), (g), and (h). There is no unexpected object appears in the target region in Fig. 9 (i).

In Fig. 12 (f), the values of method in [9] are the largest, followed by the values of methods in [24, 29], and [41], the values of method in [47] and proposed method are the smallest. Corresponding to this situation is that, the size of unexpected object in Fig. 10 (c) is the largest, followed by the size of unexpected object in Fig. 10 (e), (g), and (h). There are no unexpected objects appear in the target region in Fig. 10 (f) and (i).

Through the above analysis, we can see that the quantitative results are consistent with the qualitative visual results. If the sum of MSD and SMD are relatively high, some undesired and unexpected objects are introduced into the target region. If the sum of MSD and SMD are relatively small, then no extra objects are introduced into the target region and the restoration images can satisfy the requirements of subjective visual consistency. The experimental results can effectively illustrate the effectiveness of the new matching rule defined in our method.

5.3 Discussion

It should be noted that we did not compare the proposed method with the method based on deep learning. The reason is that we think the basic principles of these two types of methods are completely different. The exemplar-based method mainly searches for similar exemplar patch in the undamaged region of the image, and then uses it to restore the damaged pixels, while the deep learning method mainly uses a large number of real images to train the deep network model, so that the network model can learn the feature distribution of the real images, finally it can be used to automatically generate images that conform to the feature distribution to achieve the purpose of restoring the damaged region. Compared with the deep learning method, the exemplar-based method does not need to use a large number of samples and spend a lot of time to train the deep network. However, we have realized the powerful capabilities of deep learning in image inpainting. The papers presented at CVPR conferences in 2019 and 2020 have shown that deep learning methods have achieved amazing and satisfactory results. Up to now, we have carried out in-depth research on inpainting methods based on deep learning, hoping to further improve the effect of image restoration.

In addition, we also need to note that although the proposed method has achieved good restoration results, its restoration process takes more time than other methods. We think there are two main reasons: First, for each target patch, we need to traverse the whole source region to search for the most similar exemplar patch, which takes a certain amount of time. Second, for each exemplar patch, we need to calculate MSD and SMD separately, and then compare them according to the sum of MSD and SMD. This process also takes a certain amount of time.

6 Conclusions

In this paper we propose an image inpainting method for object removal based on difference degree constraint. For the problems of mismatch error and error accumulation during the restoration process, we define the MSD and the SMD between the target patch and the exemplar patch, and use them to measure the degree of difference between the two patches. Based on the MSD and the SMD, we redefine a new matching rule to prevent mismatch error and error accumulation in time. Experimental results show the effectiveness of the proposed method. In the following research, we will study the inpainting method based on the Generative Adversarial Networks, hoping to further improve the restoration effect of object removal.

Acknowledgements The research is supported in part by National Natural Science Foundation of China (Grant: 61703363), in part by Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (Grant: 2019L0855), in part by Scientific Research Project of Yuncheng University (Grant YQ-2017027, XK-2018034, CY-2019025). All of the authors would like to thank the anonymous referees for their valuable comments and suggestions.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Aharon M, Elad M, Bruckstein A (2006) K-SVD: an algorithm for designing over-complete dictionaries for sparse representation. *IEEE Trans Signal Process* 54(11):4311–4322
2. Arbelaez P, Maire M, Fowlkes C, Malik J (2011) Contour detection and hierarchical image segmentation. *IEEE Trans Pattern Anal Mach Intell* 33(5):898–916
3. Bahat Y, Schechner YY, Elad M (2015) Self-content-based audio inpainting. *Signal Process* 111:61–72
4. Bandy M, Sharma A (2014) A comparative study of existing exemplar based region filling algorithms. *Int J Current Eng Technol* 4(5):3532–3539
5. Bertalmio M, Sapiro G, Caselles V, Ballester C (2000) Image inpainting. In: *Proceedings of the 27th annual conference on computer graphics and interactive techniques*, pp 417–424

6. Chan TF, Shen J (2001) Nontexture inpainting by curvature-driven diffusions. *J Vis Commun Image Represent* 12(4):436–449
7. Chan TF, Shen J (2002) Mathematical models for local nontexture inpaintings. *SIAM J Appl Math* 62(3):1019–1043
8. Chang IC, Wun ZS, Yeh HY (2019) An image inpainting technique on chinese paintings. *J Comput (Taiwan)* 29(3):121–135
9. Criminisi A, Prez P, Toyama K (2004) Region filling and object removal by exemplar-based image inpainting. *IEEE Trans Image Process* 13(9):1200–1212
10. Deng LJ, Huang TZ, Zhao XL (2015) Exemplar-based image inpainting using a modified priority definition. *PLOS one* 10(10):1–18
11. Dong X, Dong J, Sun G, Duan Y, Qi L, Yu H (2019) Learning-Based Texture synthesis and automatic inpainting using support vector machines. *IEEE Trans Ind Electron* 66(6):4777–4787
12. Efros AA, Leung TK (1999) Texture synthesis by non-parametric sampling. In: *The Proceedings of the Seventh IEEE international conference on computer vision*, (2), pp 1033–1038
13. Elad M, Starck JL, Querre P, Donoho DL (2005) Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Appl Comput Harmon Anal* 19(3):340–358
14. Gaonkar SR, Hire PD, Pimple PS, Kotwal YR (2014) Image inpainting using robust exemplar-based technique. *Int J Comput Sci Eng* 2(4):176–179
15. Ghorai M, Mandal S, Chanda B (2018) A group-based image inpainting using patch refinement in mrf framework. *IEEE Trans Image Process* 27(2):556–567
16. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial networks. *Adv Neural Inform Process Syst* 6:1–9
17. Guo Q, Gao S, Zhang X, Yin Y, Zhang C (2018) Patch-based image inpainting via two-stage low rank approximation. *IEEE Trans Vis Comput Graph* 24(6):2023–2036
18. Hu W, Li Z, Liu Z (2014) Fast morphological component analysis Tangka image inpainting based on nonlocal mean filter. *J Comput-Aided Design and Comput Graph* 26(7):1067–1074
19. Hu G, Xiong L (2017) Criminisi-based sparse representation for image inpainting. In: *2017 IEEE third international conference on multimedia big data*, pp 389–393
20. Ishi MS, Singh L, Agrawal M (2013) A review on image inpainting to restore image. *IOSR J Comput Eng* 13(6):8–13
21. Isogawa M, Mikami D, Iwai D, Kimata H, Sato K (2018) Mask optimization for image inpainting. *IEEE Access* 6:69728–69741
22. Jiang Y, Xu J, Yang B, Xu J, Zhu J (2020) Image inpainting based on generative adversarial networks. *IEEE Access* 8:22884–22892
23. Liu H, Bi X, Lu G, Wang W (2019) Screen window propagating for image inpainting. *IEEE Access* 6:61761–61772
24. Liu Y, Wang F, Xi X, Liu Z (2014) Improved algorithm for image inpainting based on texture synthesis. *J Chinese Comput Syst* 35(12):2754–2758
25. Lu X, Wang W, Ma C, Shen J, Shao L, Porikli F (2019) See more, know more: unsupervised video object segmentation with co-attention siamese networks. *CVPR* pp 3623–3632
26. Lu X, Wang W, Shen J, Tai YW, Crandall D, Steven CH (2020) Hoi, learning video object segmentation from unlabeled videos *CVPR*
27. Mo J, Zhou Y (2019) The research of image inpainting algorithm using self-adaptive group structure and sparse representation. *Clust Comput* 22(3):7593–7601
28. Mousavi P, Tavakoli A (2019) A new algorithm for image inpainting in fourier transform domain. *Comput Appl Math* 38(1):22–30
29. Nan A, XI X (2014) An improved criminisi algorithm based on a new priority function and updating confidence. In: *International conference on biomedical engineering and informatics*, pp 885–889
30. Ogawa T, Haseyama M (2017) Exemplar-based image completion via new quality measure based on phaseless texture features. In: *IEEE International conference on acoustics speech and signal processing*, pp 1827–1831
31. Rathish Kumar BV, Halim A (2019) A linear fourth-order pde-based gray-scale image inpainting model. *Comput Appl Math* 38(6):1–21
32. Sagong M, Shin Y, Kim S, Park S, Ko S (2019) PEPSI:Fast image inpainting with parallel decoding network. *CVPR* pp 11360–11368
33. Sangeetha K, Sengottuvelan P, Balamurugan E (2013) Performance analysis of exemplar based image inpainting algorithms for natural scene image completion. In: *2013 7th international conference on intelligent systems and control*, pp 276–279
34. Shen B, Hu W, Zhang Y, Zhang YJ (2009) Image inpainting via sparse representation. In: *IEEE international conference on acoustics, speech and signal processing*, pp 697–700

35. Sridevi G, Kumar SS (2017) Image inpainting and enhancement using fractional order variational model. *Def Sci J* 67(3):308–315
36. Theljani A, Belhachmi Z, Kallel M, Moakher M (2017) A multiscale fourth-order model for the image inpainting and low-dimensional sets recovery. *Math Meth Appl Sci* 40(10):3637–3650
37. Wagh PD, Patil DR (2015) Text detection and removal from image using inpainting with smoothing. In: 2015 international conference on pervasive computing, pp 1–4
38. Wong A, Orchard J (2008) A nonlocal-means approach to exemplar-based inpainting. In: 15th international conference on image processing, pp 2600–2603
39. Xu Z, Sun J (2010) Image inpainting by patch propagation using patch sparsity. *IEEE Trans Image Process* 19(5):1153–1165
40. Yang X, Guo B (2017) Fractional-order tensor regularization for image inpainting. *IET Image Process* 11(9):734–745
41. Yao F (2019) Damaged region filling by improved criminisi image inpainting algorithm for thangka. *Clust Comput* 22(6):13683–13691
42. Zeng Y, Fu J, Chao H, Guo B (2019) Learning pyramid-context encoder network for high-quality image inpainting. *CVPR* pp 1486–1494
43. Zhang N, Ji H, Liu L, Wang G (2019) Exemplar-based image inpainting using angle-aware patch matching. *EURASIP J Image Video Process* 70:1–13
44. Zhang L, Kang B, Liu B, Bao Z (2016) A new inpainting method for object removal based on patch local feature and sparse representation. *Int J Innovat Comput Inf Control* 12(1):113–124
45. Zhang L, Kang B, Liu B, Zhang F (2016) Image inpainting based on exemplar and sparse representation, international journal of signal processing. *Image Process Pattern Recog* 9(9):177–188
46. Zhang L, Kang B, Zhang F, Li X (2015) Exemplar-based image inpainting using local feature constraints, *ICIC express letters. Part B: Appl* 6(11):2983–2988
47. Zhang S, Wang K, Zhu X (2014) Improved criminisi algorithm constrained by local feature. *Comput Eng Appl* 50(8):127–130

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.