International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.9, (2016), pp.177-188
http://dx.doi.org/10.14257/ijsip.2016.9.9.17

Image Inpainting Based on Exemplar and Sparse Representation

Lei Zhang[1,2], Baosheng Kang[1], Benting Liu[1] and Fei Zhang[1]

[1]*School of Information Science and Technology, Northwest University, Xi'an 710127, China*
[2]*Public Computer Teaching Department, Yuncheng University, Yuncheng 044000, China*
inpainting@126.com, bskang@163.com

Abstract

We propose a novel image inpainting approach in which the exemplar and the sparse representation are combined together skillfully. In the process of image inpainting, often there will be such a situation: although the sum of squared differences (SSD) of exemplar patch is the smallest among all the candidate patches, there may be a noticeable visual discontinuity in the recovered image when using the exemplar patch to replace the target patch. In this case, we cleverly use the sparse representation of image over a redundant dictionary to recover the target patch, instead of using the exemplar patch to replace it, so that we can promptly prevent the occurrence and accumulation of errors, and obtain satisfied results. Experiments on a number of real and synthetic images demonstrate the effectiveness of proposed algorithm, and the recovered images can better meet the requirements of human vision.

Keywords: *Image Inpainting; Exemplar; Sparse Representation; K-SVD*

1. Introduction

Image inpainting technique remains a longstanding challenge in image processing and computer vision [1-2], which aims to remove some objects or restore the damaged regions in a way that observers can't notice the flaw [3]. It has been widely applied in restoration of scratched images and films, heritage conservation, removal of objects in images [4], virtual reality, film and television special effects production, and so on [5-6]. Generally speaking, existing image inpainting approaches can be classified into three categories: method based on the Partial Differential Equation (PDE), method based on the Texture Synthesis, and method based on the Sparse Representation.

The most fundamental inpainting approach is based on the Partial Differential Equation, in which the missing region is filled by diffusing the image information from the known region into the missing region at the pixel level [7]. Bertalmio *et. al.*, [8] presented the notion of digital image inpainting. Chan and Shen [9] presented Total Variance (TV) model for inpainting problem, and they also presented the curvature-driven diffusions (CDD) [10] model to fix connectivity problem in TV model. Although these approaches have achieved convincingly excellent results in recovering the small, non-textured region, they tend to induce over-smooth effect or stair-case effect in the larger, textured missing region.

The second category of approaches is exemplar-based inpainting approach. It stems from the texture synthesis technique [11-12], which aims to restore the large or textured missing regions in a visually plausible manner at the patch level [13-14]. Up to now, the famous exemplar-based inpainting approach is proposed by Criminisi *et. al.*, [15]. It uses a predefined priority function to decide the filling order, and selects the target patch with the highest priority, then searches for a patch

ISSN: 2005-4254 IJSIP

which is most similar to the target patch in the source region according to the similarity criterion. At last, the value of each unknown pixel in the target patch is copied from its corresponding position inside the exemplar patch. Compared with diffusion-based approach, exemplar-based approach can obtain more visually reasonable results even in the large missing region [16].

In recent years, researchers have paid more attention in the image inpainting using sparse representations. Using an over-complete dictionary that contains prototype signal-atoms, signals are described by sparse linear combinations of these atoms. Based on sparse representations theory, Elad [17] introduced the K-SVD algorithm [18] to image inpainting in order to improve the restoration effect. Elad *et. al.*, [19] proposed an image inpainting approach that is capable of filling holes in overlapping texture and cartoon image layers. It is a direct extension of a sparse-representation-based image decomposition method called morphological component analysis (MCA) [20].

Compared with the exemplar-based approach, the sparse-representation-based approach takes less risk to introduce undesired objects or artifacts, but the details of recovered image are not as good as the exemplar-based approach, because the sparse representation of an image is just an approximate representation. So our main contribution is to make use of the respective advantages of the two approaches, integrate them organically.

The paper is organized as follows. In Section 2, an overview of the image sparse representation is presented. In Section 3, the details of the proposed algorithm are introduced. The experiments and comparisons are performed in the Section 4. Finally, we conclude this work in Section 5.

2. Review of Related Work

Recently, researchers have paid more and more attention to the sparse representation of signals. It has been widely used in the field of digital image processing. Using an over-complete dictionary matrix $D \in ¡^{n \times K}$ that contains K prototype signal-atoms for columns, a signal $x \in ¡^{n}$ can be represented as a sparse linear combination of these atoms [18]. The representation of x may be exact or approximate:

$$x = D\alpha \tag{1}$$

$$x \approx D\alpha \quad subject\ to \quad \|x - D\alpha\|_2 \le \varepsilon \tag{2}$$

Where, the vector $\alpha \in ¡^{K}$ contains the representation coefficients of the signal x.

Assume that for each image, which contains $\sqrt{n} \times \sqrt{n}$ pixels, ordered lexicographically as column vector $x \in ¡^{n}$, there is a corresponding redundant dictionary $D \in ¡^{n \times K}$ ($K > n$) which can represent the x very sparsely [21]:

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_0 \ \ subject\ to\ \ x \approx D\alpha \tag{3}$$

Where, the solution of the Formula (3) is indeed very sparse, *i.e.*, $\|\hat{\alpha}\|_0 = n$, the notation $\|\hat{\alpha}\|_0$ is l_0-norm, counting the nonzero entries in α. It means that the image can be represented as a linear combination of few atoms from the redundant dictionary D.

Notice that the constraint can be turned into a penalty term, so the constrained optimization is replaced with unconstrained optimization, the above optimization task can be changed to:

$$\hat{\alpha} = \arg\min_{\alpha} \|D\alpha - x\|_2^2 + \lambda \|\alpha\|_0 \quad (4)$$

Donoho [22] proved that the l_1-norm is a good approximation of l_0-norm, so we can estimate the coefficient $\hat{\alpha}$ of the image x over the given dictionary D by:

$$\hat{\alpha} = \arg\min_{\alpha} \|D\alpha - x\|_2^2 + \lambda \|\alpha\|_1 \quad (5)$$

Where, the term $\|\alpha\|_1$ encourages the sparsity of the coefficient vector, the parameter λ controls the tradeoff between the reconstruction error and the sparsity.

While this problem is, in general, very hard to solve, the matching and the basis pursuit algorithms can be used quite effectively to get an approximated solution [23]. In this paper, we use the orthonormal matching pursuit (OMP) [24] because of its simplicity and efficiency.

In addition, choosing an appropriate dictionary is a key step toward a good sparse representation. Aharon *et. al.*, [18] proposed the K-SVD algorithm which can be used to design over-complete dictionary for sparse representation. Given a set of training signals, we seek the dictionary that leads to the best representation for each member in this set, under strict sparsity constraints. It is flexible and can work with any pursuit method, so we make use of the K-SVD algorithm to train the dictionary in our work. For detailed description of the algorithm, please refer to the literature [18].

3. Proposed Algorithm

3.1. Notations

For easy understanding, we adopt same notations used in [15]. As shown in Figure 1, Ω is the target region (*i.e.*, the missing region) which will be removed and filled, Φ is the source region (*i.e.*, the known region), it may be defined as the entire image I minus the target region Ω ($\Phi = I - \Omega$), $\partial\Omega$ 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 point p.

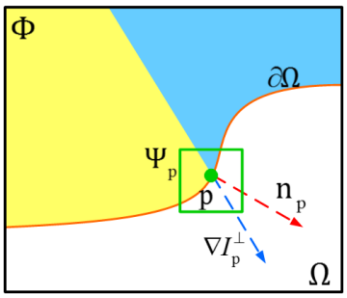


Figure 1. Notations

3.2. Patch Priority Computation

Filling order is crucial and depends entirely on the priority values that are assigned to each patch. Here we follow [15] to compute the patch priority and determine the filling order, because it is biased toward those patches which are on

the continuation of strong edges and which are surrounded by high-confidence pixels, it can reserves the structure information efficiently.

For each patch Ψ_p centered at the point p ($p \in \partial\Omega$) has a patch priority, it is defined as follows:

$$P(p) = C(p) \times D(p) \tag{6}$$

Where, $C(p)$ is the confidence term and $D(p)$ is the data term. The confidence term $C(p)$ indicates how many existing pixels are there in the patch Ψ_p . It is defined as follows:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \bar{\Omega}} C(q)}{\left|\Psi_p\right|} \tag{7}$$

Where, $\left|\Psi_p\right|$ is the area of patch Ψ_p . During the initialization, $C(p)$ is set as follows:

$$C(p) = \begin{cases} 0 & \forall p \in \Omega \\ 1 & \forall p \in \Phi \end{cases} \tag{8}$$

The data term $D(p)$ indicates how strong the isophote hitting the boundary is. It is especially important because it encourages the linear structure to be synthesized first. It is defined as follows:

$$D(p) = \frac{\left|\nabla I_p^{\perp} \bullet n_p\right|}{\alpha} \tag{9}$$

Where, α is a normalization factor. Once all priorities have been computed, we find the target patch $\Psi_{\hat{p}}$ with the highest priority.

3.3. Patch Recovery

We search in the source region for the patch which is most similar to $\Psi_{\hat{p}}$ according to the following formula:

$$\Psi_{\hat{q}} = \arg\min_{\Psi_q \in \Phi} d(\Psi_{\hat{p}}, \Psi_q) \tag{10}$$

Where, $d(\Psi_A, \Psi_B)$ is named DBP, denoting the distance between patches, it is defined as the SSD of the already existing pixels in the two patches:

$$d(\Psi_A, \Psi_B) = \sum_{i,j \in A \cap \bar{\Omega}} \left|\Psi_A(i,j) - \Psi_B(i,j)\right|^2 \tag{11}$$

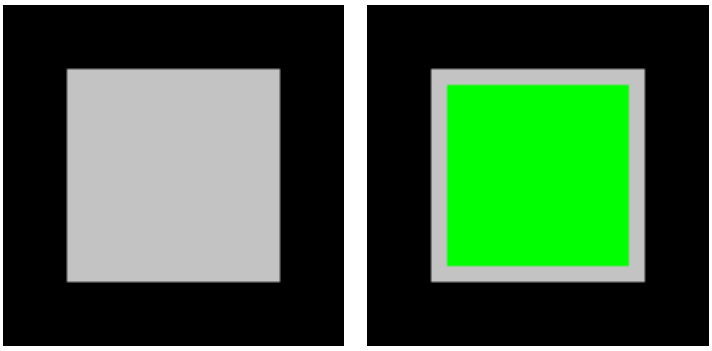
After finding the most similar patch $\Psi_{\hat{q}}$, the target patch $\Psi_{\hat{p}}$ can be recovered as follows:

$$\Psi_{\hat{p}}(i,j) = \Psi_{\hat{q}}(m,n) \tag{12}$$

Where $(i,j) \in \Psi_{\hat{p}} \cap \Omega$, (m,n) is the corresponding position to (i,j) inside $\Psi_{\hat{q}}$.

However, often there will be such a situation: although the SSD of the found exemplar patch $\Psi_{\hat{q}}$ is the smallest among all the candidate patches, it is not appropriate to replace the target patch $\Psi_{\hat{p}}$. To better illustrate, we synthesize a test

image in Figure 2, where (a) is the original image, and the green region in (b) is the target region which will be recovered.



(a) (b)

Figure 2. A Test Image

According to the algorithm described above, we find the target patch $\Psi_{\hat{p}}$ and the exemplar patch $\Psi_{\hat{q}}$. In order to observe more clearly, the patches are amplified because the size of patch is 9×9, and they are shown in Figure 3, where (a) is the target patch $\Psi_{\hat{p}}$, (b) is the exemplar patch $\Psi_{\hat{q}}$, (c) is the recovered patch of $\Psi_{\hat{p}}$ by $\Psi_{\hat{q}}$, but from Figure 2(a) we can easily find that the expected recovered patch of $\Psi_{\hat{p}}$ is (d).

(a) (b) (c) (d)

Figure 3. An Example of Patch Matching Error

As shown from the Figure 3, although we find the exemplar patch (b) according to the matching criterion, the recovered patch (c) is not the desired result. What is worse, as the algorithm progresses, the error will be continually accumulated, which may result in that some undesired objects or artifacts are introduced in the recovered image and the results cannot meet the requirements of the human visual.

Through the analysis we find that, if the DBP (*i.e.*, the value of formula (11)) of the target patch and exemplar patch is too large, it means that there are great differences between the existing pixels in two patches, and the visual discontinuity is likely to appear in recovered images. Besides, if there are great differences between the already existing pixels in target patch and the pixels used to fill in exemplar patch, the same situation described above is also likely to occur, so the difference is named DIP, denoting the distance in patches, and it is defined as follows:

$$s(\Psi_A, \Psi_B) = \sum \left| \frac{\sum_{(i,j) \in \Psi_A \cap \bar{\Omega}} \Psi_A(i,j)}{C_A} - \frac{\sum_{(i,j) \in \Psi_A \cap \Omega} \Psi_B(i,j)}{C_B} \right|^2 \tag{13}$$

Where, C_A is the number of already existing pixels in target patch, and C_B is the number of pixels to be filled in exemplar patch.

Therefore, in order to estimate the occurrence of matching errors accurately, we can set two thresholds β and γ. When the value of DBP is larger than β or the value of DIP is larger than γ, we think that the situation described in Figure 3 happens. Under such circumstance, we skillfully use the sparse representation of image patch to recover the current target patch, instead of replacing it with the exemplar patch. Here we describe in detail how to recover the target patch.

As described in the Section 2, x denotes the target patch, D denotes the redundant dictionary, and x can be sparsely represented over D. From the inpainting point of view, some pixels in x are corrupted, so we use a set v to record the index of corrupted pixels [25]. The new vector x' can be obtained by removing the corrupted pixels from x. In the same way, the new dictionary D' can be obtained by removing the corresponding rows from dictionary D. So the sparse coefficient $\hat{\alpha}'$ can be estimated as follows:

$$\hat{\alpha}' = \arg\min_{\alpha} \left\| D'\alpha - x' \right\|_2^2 + \lambda \left\| \alpha \right\|_1 \tag{14}$$

Then, we recover the target patch according to the following formula:

$$\hat{x}_i = \begin{cases} x_i & i \notin v \\ D\hat{\alpha}' & i \in v \end{cases} \tag{15}$$

In this way, the current target patch can be recovered efficiently. Meanwhile, we can prevent the occurrence of error in a timely manner, making the error cannot be constantly accumulated.

So far, the proposed inpainting algorithm has been described. However, there is a question: how to determine the thresholds? Next, we introduce the determination of the two thresholds in our work.

3.4. Determination of Thresholds

As described above, we use two thresholds β and γ to estimate the occurrence of matching errors. In our work, we have adopted a very simple and effective method to determine the thresholds. We calculate the DBP and DIP of all the matching patches, generate the distribution map of these data, and then we determine the thresholds through the distribution of these data. For example, the DBP and DIP of all the matching patches in the Figure 2(b), is shown in Figure 4, and Figure 5. By analyzing the distribution of these data, the threshold β is set to 4×10^5, and threshold γ is set to 2×10^4.

3.5. Algorithm Description

Here, the proposed algorithm steps are described as follows:

Step1. Identify the target regions. The target regions are extracted manually.

Step2. Compute the patch priorities according to Formula (6), and select the target patch $\Psi_{\hat{p}}$ with the highest priority.

Step3. Search in the source region for the exemplar patch $\Psi_{\hat{q}}$ according to formula (10).

Step4. Compute the values of DBP and DIP, if the values are greater than the threshold values, recover $\Psi_{\hat{p}}$ according to Formula (15), else recover $\Psi_{\hat{p}}$ according to formula (12).

Step5. Update the value of confidence term according to the following formula:

$$C(q) = C(\hat{p}) \quad \forall q \in \Psi_{\hat{p}} \cap \Omega \tag{16}$$

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

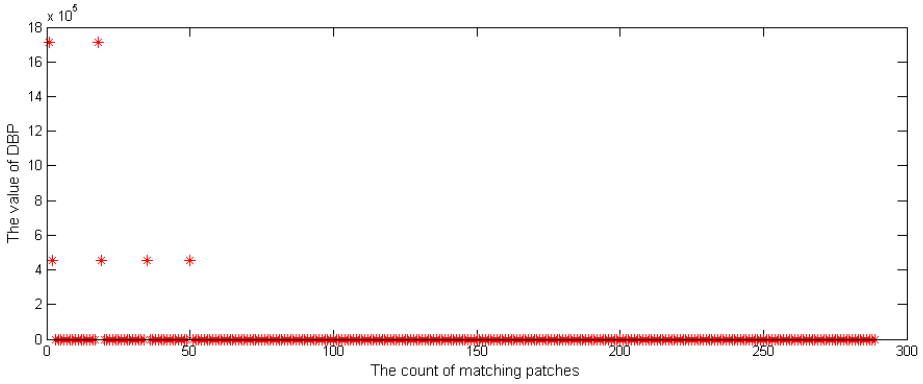


Figure 4. The Distribution of DBP

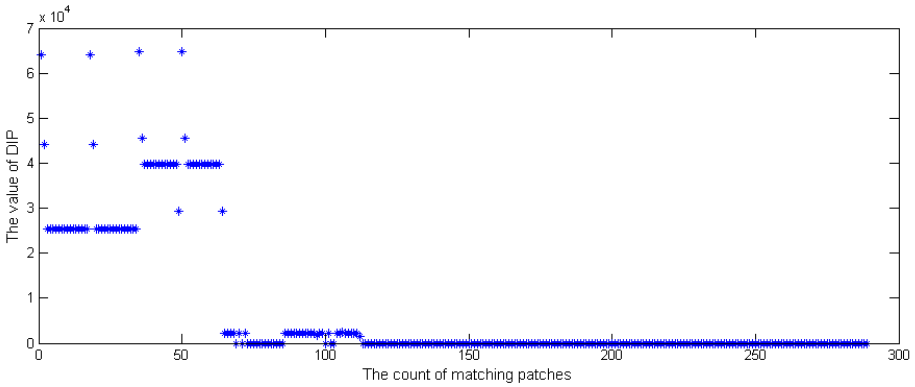


Figure 5. The Distribution of DIP

4. Experimental Results

In this section, we apply the proposed approach on a variety of images, and compare the experimental results with the approach in [15], which is one of the most classical image inpainting approaches. For a better comparison, the experiments are divided into two groups, one is about synthetic test images and the other is about natural images.

In these experiments, the size of patches is 9×9, the size of redundant K-SVD dictionary is 81×256, and all the experiments are run on the computer with the configuration of 2.1GHz processor and 12GB RAM.

4.1. Synthetic Images

In the experiment, we apply the proposed approach on synthetic images, to test whether this approach can obtain the desired results. We synthesize two test images, mark out the green target regions, and recover them using the approach in [15] and

the proposed approach respectively. The experimental results are shown in Figure 6 and Figure 7.

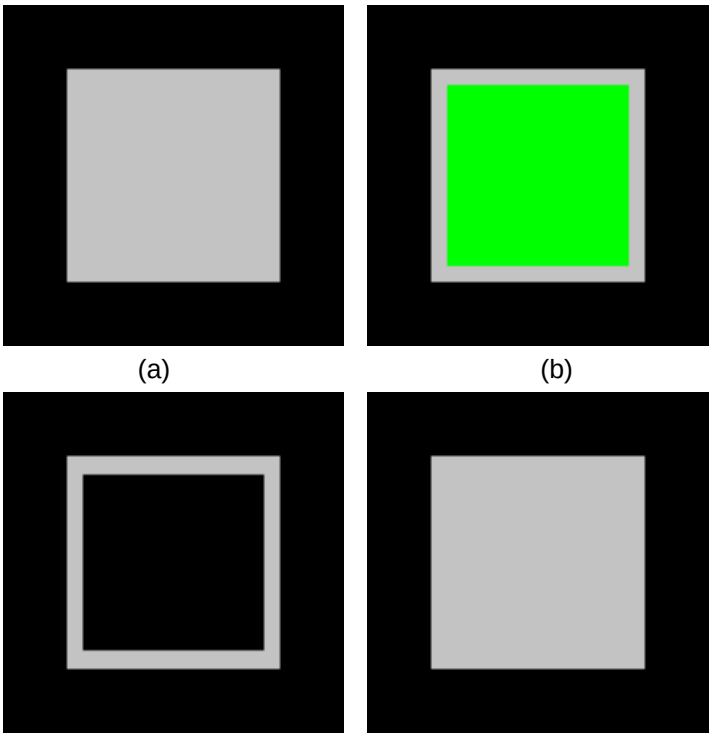


Figure 6. Test Image and Inpainting Results

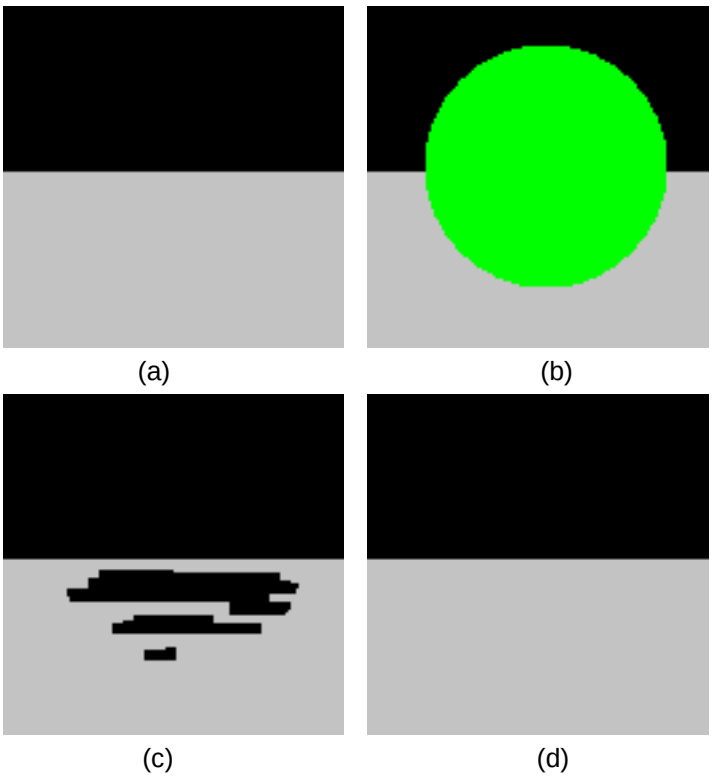


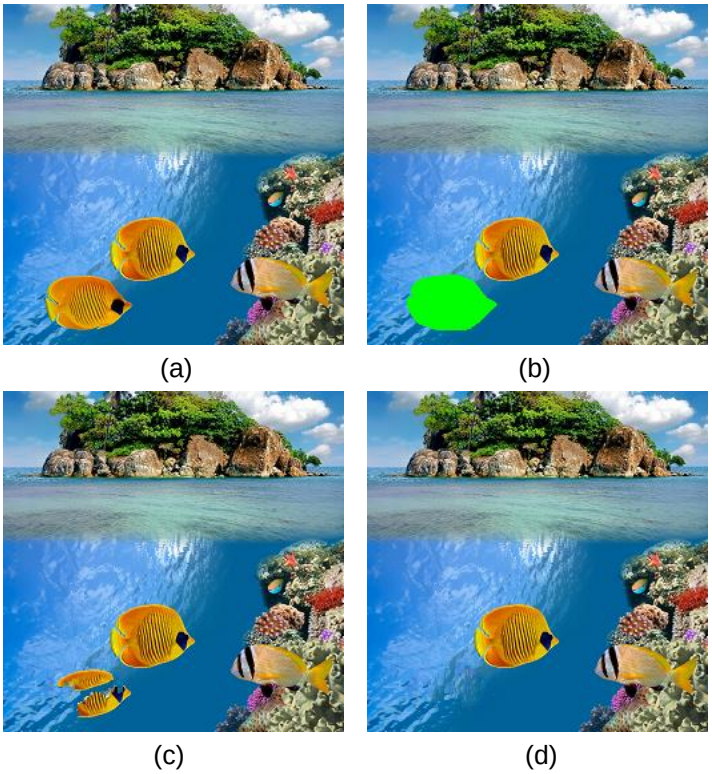
Figure 7. Test Image and Inpainting Results

In Figure 6, and Figure 7, the top-left ones are the original images, the top-right ones are the marked target regions, the bottom-left ones are the results of [15], and the bottom-right ones are the results of proposed approach. As can be seen from the above experiments, the results of the proposed approach is just what we expect, yet the results of [15] introduce some undesired objects in the recovered images.

4.2. Natural Images

In the experiment, we carry out several experiments on natural images, trying to show the practicality of the proposed approach. We first select some objects in the images, and then mark them as green, at last use the approach of [15] and proposed approach to recover the target regions respectively. The experimental results are shown in Figure 8, and Figure 9.

In Figure 8, and Figure 9, the top-left ones are the original images, the top-right ones are the marked target regions, the bottom-left ones are the results of [15], and the bottom-right ones are the results of proposed approach.

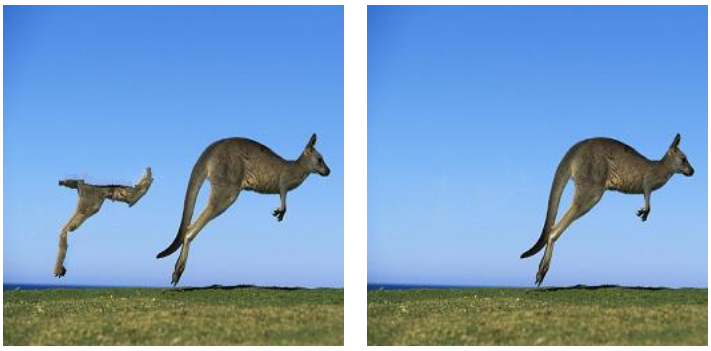


(a) (b)

(c) (d)

Figure 8. Natural Image and Inpainting Results

(a) (b)



(c) (d)

Figure 9. Natural Image and Inpainting Results

From the above results we can see that, in the approach of [15], the target patches are replaced by the unsuitable exemplar patches, and some undesired objects are introduced, so the recovered images are not satisfy the human visual requirements. But in proposed approach, we timely use the sparse representation of target patch to recover it, rather than using the exemplar patch to replace it, so prevent continuous accumulation of errors and the recovered images are more natural.

5. Discussion and Conclusion

5.1. Discussion

We find that the proposed algorithm has a very important relationship with some famous algorithms. A famous exemplar-based inpainting algorithm proposed by [15] can be viewed special case of our algorithm if the thresholds are set to particularly large. Also, a sparse-representation-based inpainting algorithm proposed by [25] can be viewed special case of our algorithm if the thresholds are set to particularly small. Thus we combine the two algorithms together.

5.2. Conclusion

In this paper, we propose a novel image inpainting approach based on exemplar and sparse representation. We use the sparse representation of an image patch to recover the target patch, to timely prevent the mismatch occurs, so that the error cannot be accumulated and the better results can be obtained. In this way, we combine the advantages of exemplar-based approach and sparse-representation-based approach. The experimental results show our approach is efficient to restore damaged images and remove unwanted objects from images.

Acknowledgments

The research is supported by the National Natural Science Foundation of China (No. 61272286), and the Provincial Natural Science Foundation research project of Shaanxi (No. 2014JM8346). All of the authors would like to thank the anonymous referees for their valuable comments and suggestions.

References

[1] M. Banday and A. Sharma, “A Comparative Study of Existing Exemplar based Region Filling Algorithms”, International Journal of Current Engineering and Technology, vol. 4, no. 5, **(2014)**.
[2] M. Wang, B. Yan and K. N. Ngan, “An efficient framework for image/video inpainting”, Signal Processing: Image Communication, vol. 28, no. 7, **(2013)**.

[3] J. Patel and T. K. Sarode, "Exemplar based Image Inpainting with Reduced Search Region", International Journal of Computer Applications, vol. 92, no. 12, **(2014)**.
[4] A. S. Hareesh and V. Chandrasekaran, "Exemplar-based color image inpainting: a fractional gradient function approach", Pattern Analysis and Applications, vol. 17, no. 2, **(2014)**.
[5] C. Guillemot and O. L. Meur, "Image inpainting: Overview and recent advances", Signal Processing Magazine, vol. 31, no. 1, **(2014)**.
[6] Y. T. Kao, L. P. Hung and W. M. Chen, "A new approach of image inpainting in the wireless environment", International Journal of Ad Hoc and Ubiquitous Computing, vol. 14, no. 1, **(2013)**.
[7] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity", IEEE Transactions on Image Processing, vol. 19, no. 5, **(2010)**.
[8] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image inpainting", Proceedings of the 27th annual conference on computer graphics and interactive techniques, New Orleans, LA, USA, **(2000)** July 23-28.
[9] T. F. Chan and J. Shen, "Mathematical models for local nontexture inpaintings", SIAM Journal on Applied Mathematics, vol. 62, no. 3, **(2002)**.
[10] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions", Journal of Visual Communication and Image Representation, vol. 12, no. 4, **(2001)**.
[11] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling", The Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, **(1999)** September 20-27.
[12] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer", Proceedings of the 28th annual conference on Computer graphics and interactive techniques; Los Angeles, CA, USA, **(2001)** August 12-17.
[13] W. Casaca, M. Boaventura, M. P. D. Almeida and L. G. Nonato, "Combining anisotropic diffusion, transport equation and texture synthesis for inpainting textured images", Pattern Recognition Letters, vol. 36, **(2014)**.
[14] Y. Liu and V. Caselles, "Exemplar-based image inpainting using multiscale graph cuts", IEEE Transactions on Image Processing, vol. 22, no. 5, **(2013)**.
[15] A. Criminisi, P. Pérez and K. Toyama, "Region filling and object removal by exemplar-based image inpainting", IEEE Transactions on Image Processing, vol. 13, no. 9, **(2004)**.
[16] J. K. Chhabra and M. V. Birchha, "Detailed Survey on Exemplar Based Image Inpainting Techniques", International Journal of Computer Science & Information Technologies, vol. 5, no. 5, **(2014)**.
[17] M. Elad, "Sparse and redundant representations: from theory to applications in signal and image processing", Springer Science & Business Media, **(2010)**.
[18] M. Aharon, M. Elad and A. Bruckstein, "K-SVD: An Algorithm for Designing Over-complete Dictionaries for Sparse Representation", IEEE Transactions on Signal Processing, vol. 54, no. 11, **(2006)**.
[19] M. Elad, J. L. Starck, P. Querre and D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)", Applied and Computational Harmonic Analysis, vol. 19, no. 3, **(2005)**.
[20] J. Bobin, J. L. Starck, J. M. Fadili, M. Elad and D. L. Donoho, "Morphological component analysis: An adaptive thresholding strategy", IEEE Transactions on Image Processing, vol. 16, no. 11, **(2007)**.
[21] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation", 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition; New York, USA, **(2006)** June 17-22.
[22] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution", Communications on pure and applied mathematics, vol. 59, no. 6, **(2006)**.
[23] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries", IEEE Transactions on Image Processing, vol. 15, no. 12, **(2006)**.
[24] Y. C. Pati, R. Rezaiifar and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition", 1993 Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers; Pacific Grove, CA, **(1993)** November 1-3.
[25] B. Shen, W. Hu, Y. Zhang and Y. J. Zhang, "Image inpainting via sparse representation", 2009 IEEE International Conference on Acoustics, Speech and Signal Processing; Taipei, Taiwan, **(2009)** August 19-24.

Authors

Lei Zhang is currently pursuing a PhD degree at the School of Information Science and Technology, Northwest University, Xi'an, China. His research interests include image processing and pattern recognition.

Baosheng Kang is a professor at the School of Information Science and Technology, Northwest University, Xi'an, China. His research interests include intelligent information processing, image processing and pattern recognition.

Benting Liu is currently pursuing a master's degree at the School of Information Science and Technology, Northwest University, Xi'an, China. His research interests include image processing and pattern recognition.

Fei Zhang is currently pursuing a master's degree at the School of Information Science and Technology, Northwest University, Xi'an, China. His research interests include image processing and pattern recognition.