International Journal of Engineering

Journal Homepage: www.ije.ir

# Improving Image Inpainting based on Structure and Texture Information Using Quadtree

K. Peyvandi, F. Yaghmaee*

*Computer and Electrical Engineering Department, Semnan University, Semnan, Iran*

*P A P E R   I N F O*

*Paper history:*
Received 07 February 2020
Received in revised form 27 March 2020
Accepted 28 March 2020

*Keywords*:
*Image Inpainting*
*Image Decomposition*
*Principal Component Analysis*
*Quad Tree*
*Local Steering Kernel*
*Structure*
*Texture*

*A B S T R A C T*

In this paper, we present a novel and efficient algorithm for image inpainting based on the structure and texture components. In our method, after decomposing the image into its texture and structure components using Principal Component Analysis (PCA), these components are inpainted separately using the proposed algorithm. Finally, the inpainted image is simply acquired by adding the two inpainted images. For structure inpainting we used quadtree concept to identify the importance of each pixel located on the boundary of the target region. Subsequently, we detect the correct path for filling so that this path demonstrates an orientation for the better structure inpainting. It is noteworthy that structure inpainting is more important because human vision is sensitive to the coherence of structure. For texture inpainting, we use Euclidean distance in the texture component for patch selection. Also, the geometric feature is considered by Local Steering Kernel (LSK) in the original image to assist chooing a better patch candidate. The experimental results of our algorithm demonstrate the effectiveness of the proposed method.

***doi**: 10.5829/ije.2020.33.05b.27*

**NOMENCLATURE**

| | | | |
|---|---|---|---|
| I | Original image | $\delta\Omega^+$ | 1 pixel thick boundary region outside |
| Mask | Binary image for determination missing pixels | $\delta\Omega$ | Boundary region |
| $\Omega$ | Target region contains missing pixels | $\Psi_p$ | A target patch centered at pixel P |
| $\Phi$ | Source region contains known pixels | $\delta\Omega^-$ | 1 pixel thick boundary region inside |

## 1. INTRODUCTION

Image inpainting is an important field of image processing in recent years. The purpose of image inpainting is to fill the target region which is characterized by a binary image named mask. The missing region may be logo, subtitle, text or any undesirable objects in the image. The unknown pixels must be estimated such that the inpainting result appears visually plausible to the human observer. Generally, the target region should be filled by the source region information which is named as the inpainting problem.

Formally an inpainting problem can be represented in this way: image I is given, one or multiple regions are characterized to be inpainted by a binary mask image so that the source region is denoted by $\Phi$ and the target region is $\Omega$ ($\Omega$ = I− $\Phi$), $\delta\Omega$ represents the boundary of the target regions all of the notation can be seen in NOMENCLATURE.

Many methods exist for image inpainting which can be categorized in four groups: (1) diffusion-based inpainting, (2) patch-based inpainting, (3) low-rank methods and (4) sparsity-based algorithms so that each of these groups assumes prior knowledge for inpainting, which is described completely in Section 2.

Furthermore, many of image inpainting methods apply their algorithm on the whole of the image, while an image consists of two components, structure and texture. Each component has some useful features so that the result of the image inpainting can be improved by considering these attributes.

Generally, some of the inpainting algorithms are appropriate for structure and others are suitable for texture. Algorithms which are efficient for structure

*Corresponding Author Institutional Email: f_yaghmaee@semnan.ac.ir (F. Yaghmaee)

Please cite this article as: K. Peyvandi, F. Yaghmaee, Improving Image Inpainting based on Structure and Texture Information Using Quadtree, International Journal of Engineering (IJE), IJE TRANSACTIONS B: Applications Vol. 33, No. 5, (May 2020) 940-948

inpainting may produce blur artifact while texture synthesis cannot recover geometric information.

In this paper, we propose a new inpainting algorithm based on the structure and texture separately. For this purpose, an image decomposition is essential, so that we apply the Principal Component Analysis (PCA) technique on the image to obtain the two components. One of these components is the low rank which is considered as texture and another is the sparse image that is assumed as structure. Subsequently, each of them would be inpainted individually and then the final result can be acquired by combining them.

It is noticeable that the novelty of our method is to present a new image inpainting for structure component based on quadtree and after that we correct the filling orientation by our new approach. Also for texture component, our new patch-based inpainting uses two criteria for patch selection. One of these criteria is the Euclidean distance in texture and another criteria is the geometric feature for inpainting the texture component. After all, the inpainted image is acquired by adding the two inpainted components.

The paper is organized as follows: the related work is reviewed in Section 2. Then in Section 3 our method is described in details, and the result of our method is presented in Section 4. Finally, the conclusion is provided in Section 5.

## 2. RELATED WORKS

There are different image inpainting methods in literature. The existing methods can be classified into four categories: diffusion-based, patch-based, rank minimization algorithm, and sparsity-based method. The diffusion-based inpainting methods were introduced by Bertalimo in 2000 [1]. This category methods use some smooth local assumption as prior knowledge. The diffusion-based methods are appropriated for the non-texture or relatively small missing region. Bertalimo introduced an approach based on the linear partial differential equation, some of the algorithms from this group are proposed in following. For example, a method was presented by Chan and Shen which uses Curvature Driven Diffusion (CCD) model by Total Variation (TV) algorithm to improve the inpainting [2]. In another method, Euler's elastic concept was applied for the diffusing curvature structures [3].

Ballester et al. presented a new equation which was introduced as a partial differential equation to fill the missing region [4]. It is noticeable that fractional-order anisotropic diffusion can be considered as a generalization of the integer-order derivation. So Pu et al. used fractional derivation and considered texture details of the images to improve the result of inpainting [5].

The patch-based inpainting methods are common categories [6, 7] which consider non-local self-similarity as prior knowledge. This category was introduced by Criminisi [8], in which a target patch is attended and filled by the surrounding area. In this method, patch priority is proposed based on data and confidence term. In another algorithm, Chen et al introduced structure complexity factor for a patch, and presented the structure-aware patch-based inpainting [9].

Isogawa et al. proposed a new patch-based inpainting algorithm that converts an image to a low dimensional feature space, then inpaint it in the new space, and finally the inverse conversion is applied to the original space [10]. A new patch-based inpainting method was proposed by Liu, which uses the optimized patch size instead of a fixed size [11]. The majority of the patch-based inpainting use the Euclidian distance as a criterion for the patch similarity, while some of them describe a new approach for detecting the similar patches. For example, Zhang et al. presented the Jaccard similarity coefficient as a new measurement for the patch comparison [12]. Also, another method was proposed in [13] which considers the color feature and space distance between two patches and determines an optimized patch scale.

The low rank based inpainting approaches were proposed in recent years. In this category, an image is considered as a matrix or tensor [14, 15], and then the missing pixels in the target region must be estimated. So the low-rank matrix (tensor) is obtained as a result. It is noticeable that the computation of the tensor rank is an NP-hard problem [16], thus applying nuclear norm is the best choice for tensor rank minimization [17]. Generally, matrix or tensor nuclear norm can be acquired by SVD decomposition, however, other definitions were recently proposed for the nuclear norm [18].

In the majority of the image inpainting algorithms which use the low-rank concept, pixels are distributed in the whole image. Thus, this category methods are appropriated for text removal, small object subtraction or noise reduction. But for massive object removal a combination of low-rank and patch-based inpainting is necessary. For example, Guo et al. presented a patch-based and low-rank inpainting algorithm so that in each step N similar patches are selected and then each patch is vectored as a column of a matrix [19]. This matrix is low rank, thus the missing pixels can be estimated based on low-rank matrix approximation.

Sparsity prior is considered in the sparsity-based inpainting so that this group can be combined with the low-rank algorithms [20]. Furthermore, patch-based algorithms by sparsity-based can produce high-quality results [21]. For example, a patch priority filling approach is proposed in [22] which uses the structure sparsity as patch priority [22].

## 3. PROPOSED METHOD

We mentioned earlier that an image can be decomposed to the structure and texture so that each of them can represent different features of the image. The majority of the available image inpainting algorithms are appropriated for the structure while these algorithms create artifacts texture.

Consequently, attention to both of them can be impressive in the inpainting process. In this paper, firstly we decompose the image to structure and texture using the PCA which is described in detail in Section 3.1. Then a new method is presented to inpainting the structure component that is explained in Section 3.3 and finally, texture inpainting algorithm is proposed in Section 3.4. It is noticeable that our new structure inpainting is based on the quadtree, thus in Section 3.2, the quadtree concept is explained. Furthermore, the flowchart of our algorithm is shown in Figure 1.

**3. 1. Image Decomposition based on PCA** It is noteworthy to mention, PCA can be used for the data analysis and pattern recognition. So it maps the data from a higher-dimensional space to a lower-dimensional space. One of the PCA applications is in the image processing field. So applying PCA on the image can decompose it into its components, low rank, and sparse component which are shown via Equation (1).

$$I = L + S \tag{1}$$

where I is the original image, L is the low-rank component and S is the sparse component. An example of applying PCA on the image can be seen in Figure 2. Generally, the majority of the image decomposition algorithms that extract the structure and texture could present an obstacle. So the original image can't be recovered by their components. Formally an image can be decomposed based on Equation (2).

$$I = U \oplus V \tag{2}$$

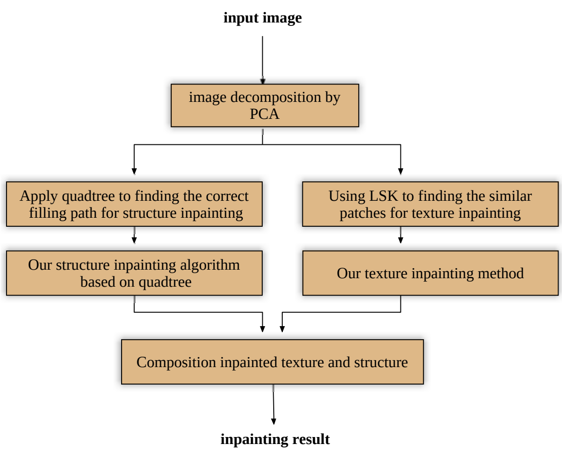


**Figure 1.** Flowchart of the proposed method

where I is the original image, U is the structure image and V is referred to as the texture component. Operator $\oplus$ is different in various decomposition methods. In PCA operator $\oplus$ is the same as $+$, subsequently, the original image is recovered from the structure and texture.

In Figure 3, another result can be seen so that decomposition is done for a grayscale image.

**3. 2. Quadtree Concept** A quadtree is a tree data structure that each internal node has four children. Quadtree works by dividing a square image into four equal-sized square blocks, and then testing each block to see if it meets some criterion of homogeneity (in this paper we used the standard quadtree function in Matlab which is using a threshold in the variance range for block

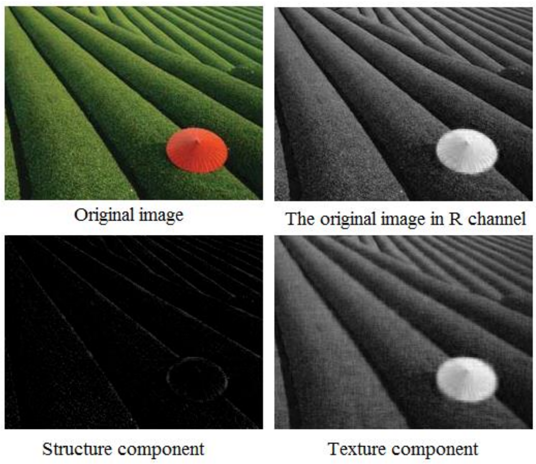


**Figure 2.** Image decomposition is done based on PCA for RGB image

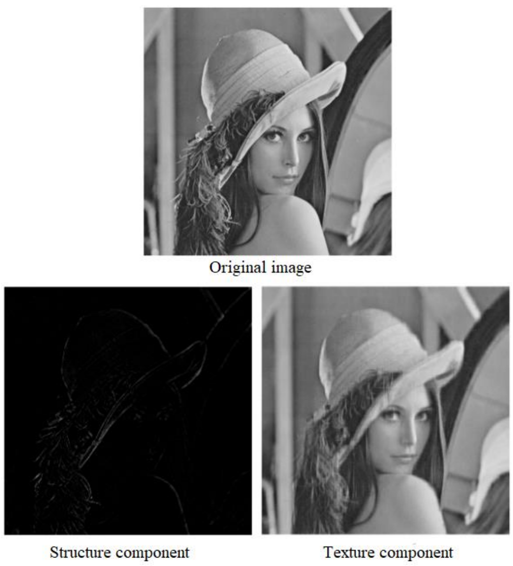


**Figure 3.** Image decomposition using PCA on the grayscale image

decomposition). If a block meets the criterion, it is not divided any further. If it does not meet the criterion, it is subdivided again into four blocks, and the test criterion is applied to those blocks. This process is repeated iteratively until each block meets the criterion. The result might have blocks of several different sizes.

We use quadtree in an image to specify the importance of different regions and decompose an image to variable size patches. So pixels which belong to a smaller size patch is more important, and large size patches consist of worthless pixels. Subsequently, the importance of each pixel is characterized based on the patch size in which the pixel is belonged. An example of applying quadtree and decomposing an image into the different size regions is presented in Figure 4.

It is noticeable that quadtree only can be applied in a square image. Furthermore, the result, which is acquired by the quadtree, and the original image are the same size generally. To apply the quadtree on the non-square image, we resize an image to square size by padding the black pixels as can be seen in Figure 4. In this condition, the size of the original image and quadtree result are not equal. In Figure 4 we demonstrate a rectangular shape created based on padding.

To apply the quadtree on the image, a threshold is necessary which bounded between 0 and 1. This threshold determines the similarity of the pixels located in a region. In one case, each pixel is considered as a region in the lower bound, while the whole of the image can be remarked as a region in the upper bound. The experimental result shows the appropriated value for the threshold is 0.6.

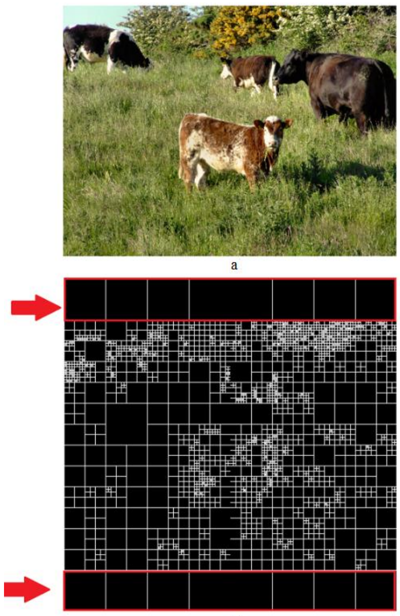


**Figure 4.** Applying quadtree on the image, (a) the original image, (b) the quadtree image

**3. 3. Inpainting by Quadtree** As previously mentioned, the original image and a binary image (mask) are the inputs of the image inpainting problem. The purpose is filling the target region based on the source area so that the result must be consistent with the human vision. In this section, our new method for inpainting the structure component is proposed.

We apply quadtree on the original image so that different size squares are obtained. These variable sizes in the boundary of the target region show which part of the target region is important and has higher priority for filling. Therefore the orientation of filling the missing area is acquired based on quadtree. The detail of our algorithm for determining the correct orientation by quadtree is described as follows.

Let $\Psi_p$ is a target patch centered at pixel P so that φ is the source area of this patch. We start with pixel P and consider 8-neighbors of a given pixel P so that each of these pixels might be related to a region obtained by the quadtree. We select one of the 8-neighbors and named $P_1$ which is related to the smallest region. Because whatever a region is small, pixels located are more significant, while a pixel that is placed on a large area belongs to a smooth region. In the following, we select the next pixel which is 8-neighbors of pixel $P_1$ so $P_1$ is more important than others. In the same way $P_2$, $P_3$, $P_4$, … are chosen, subsequently. Then a path is acquired which shows the accurate direction of filling. We demonstrate an example of detecting the correct direction for filling by our method in Figure 5.

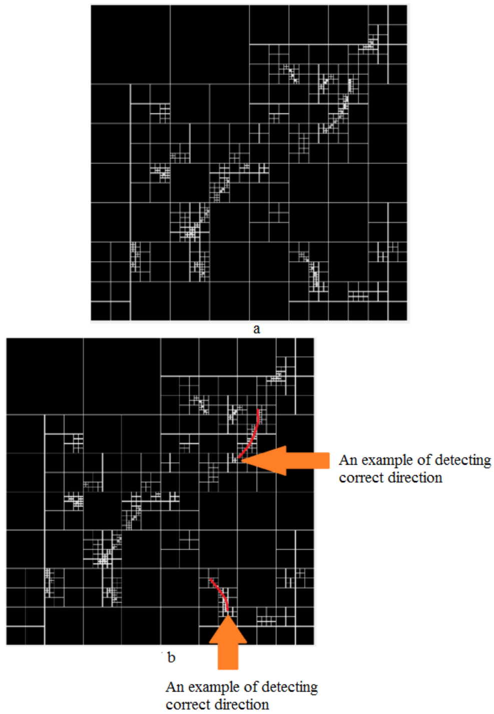


**Figure 5.** Detecting the correct path for filling using our method

**3. 4. Structure Filling Process** In this section, we describe the detail of the structure filling process based on the quadtree. As mentioned previously, we detect the accurate path for structure inpainting based on Section 3.3. These pixels located on the correct path have high weight for filling the missing pixels. Our notation diagram is shown in Figure 6 so that an image I is given, the target area is Ω and the source region is Φ.

Generally in the inpainting process, the target patch $\Psi_p$ is a patch centered at pixel P, that pixel P is located on δΩ and is a known pixel. But for structure inpainting, we consider target patch Ψp which pixel P located on δΩ-. In the other words, target patch Ψp is a patch centered at pixel P which P is an unknown pixel that should be estimated. We want to calculate the pixel P based on the neighborhood known pixels while neighborhood pixels have different weight. Indeed known pixels that are located on the correct path, as discussed in Section 3.3, have a high weight and other neighborhood pixels have a low weight.

Therefore pixel P is estimated based on the weighted average of known pixels in the patch which is centered at pixel P. Formally the approximation of pixel P is accomplished using Equation (3).

$$I_P = \frac{\sum_{r\in\varphi} W_i(r).I_p(r)}{\sum_{r\in\varphi} W_i(r)} \tag{3}$$

where P is a pixel that must be estimated and located on $\delta\Omega^-$. $I_P$ is a 5×5 patch from I centered at pixel P, $W_i$ is a 5×5 weight matrix and φ is the source region of the patch $I_P$. Here, we should note that $W_i$ can be acquired based on the correct path which was described in Section 3.3. Therefore, pixels located on the correct path is used for filling the target patch by a higher weight than the other pixels placed in the patch. For color images, the estimation of a missing pixel must be done in three channels (R, G and B). However, the other color spaces can be applied in this step such as CIE Lab [23], CMC, and YCbCr. Figure 7 demonstrates the examples of our structure inpainting algorithm.

**3. 5. Texture Inpainting** In the next step, the texture component should be inpainted so the texture image is extracted based on the method described in Section 3.1. We propose new patch-based inpainting for texture filling. In the patch-based inpainting, the unknown pixels are filled via searching for similar patches for each of the target patches.

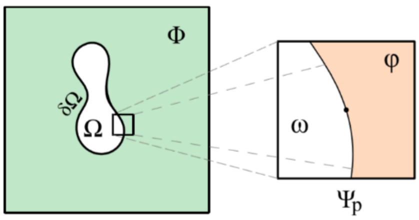


**Figure 6.** The notation diagram [33]

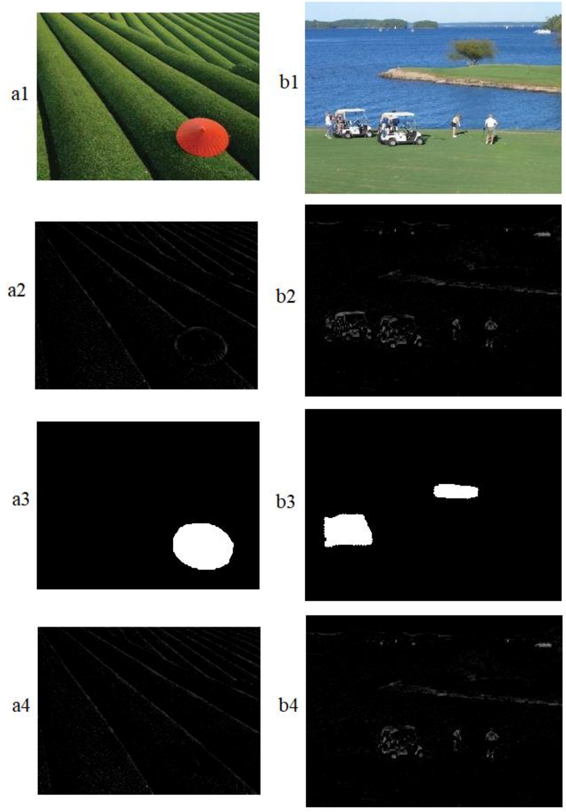


**Figure 7.** Our structure inpainting ($a_1$,$b_1$) Original images, ($a_2$,$b_2$) Structure components, ($a_3$,$b_3$) related masks, ($a_4$,$b_4$) the results of structure inpainting by our algorithm

In this section, we describe new criteria for patch selection. For choosing candidate patches, a similarity measurement is necessary. Many methods use Euclidean distance as a similarity criterion. It is noticeable that this criterion may cause wrong patch selection. To avoid this mistake, we use the LSK feature [38] together with the Euclidean distance. Various applications apply the LSK such as image denoising [24-26], image and video restoration [27], Detail Enhancement [28], image inpainting [29] and image super-resolution [30]. Steering kernel is a nonlinear filter that estimates image gradient in one step, and then gradient information is used for determining the dominant orientation of the local gradients in the image.

We extract the geometric feature based on steering kernel regression (SKR) [36] to computes LSK at each pixel of the image within a neighborhood. In the SKR method, each pixel is recognized as texture, flat, strong edge, corner or weak edge. It is noteworthy; we apply the SKR method on the original image instead of the texture image, thus extracted feature in the original image facilitates the texture inpainting. Therefor in the texture inpainting, the similar patches are detected by Euclidean distance then the patch which is the best similar to the target patch based on the geometric feature is selected. Indeed the geometric feature is obtained based on the SKR method that applied to the original image. Two examples of texture inpainting by our method can be seen in Figure 8.

Extracting geometric features on the original image instead of the texture component is important. Because the original image is appropriate for identification of a pixel as texture, flat, strong edge, corner or weak edge. Subsequently, Euclidean distance is used in texture component and the geometric feature is extracted in the original image so that the best patch is selected to fill the target patch in the texture component. In Figure 8, examples of the texture inpainting by our algorithm is presented.

**3. 6. Original Image Inpainting** In previous sections, we inpainted the texture and structure separately so that a combination of these components is necessary. One of the advantages of using PCA as image decomposition is that the result can be easily acquired by adding the two components. While in the majority of the image decomposition methods combination of components is a difficult task.

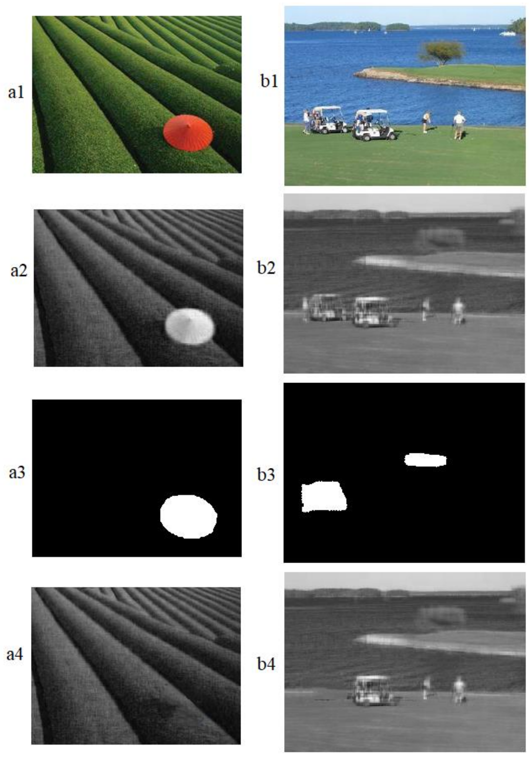


**Figure 8.** Texture inpainting results by our method, ($a_1$,$b_1$) Original images, ($a_2$,$b_2$) texture components, ($a_3$,$b_3$) related masks, ($a_4$,$b_4$) the results of texture inpainting by our method

## 4. RESULTS AND DISCUSSION

In this section, we demonstrate the result of the proposed method on various test images. In our method, in the first step, the image decomposition is accomplished, and then each of the components is inpainted separately and finally the combination of the results is done.

Briefly the following steps have done in our algorithm: image decomposition, components inpainting, and structure and texture combination for each of the color channels (red, green, blue) in RGB color space. So Figure 9 demonstrates examples of experimental results.

In Figure 9, we demonstrate the results of our algorithm and state-of-the-art inpainting methods to validate our method performance. Figure 9 ($a_1$,$b_1$,$c_1$)

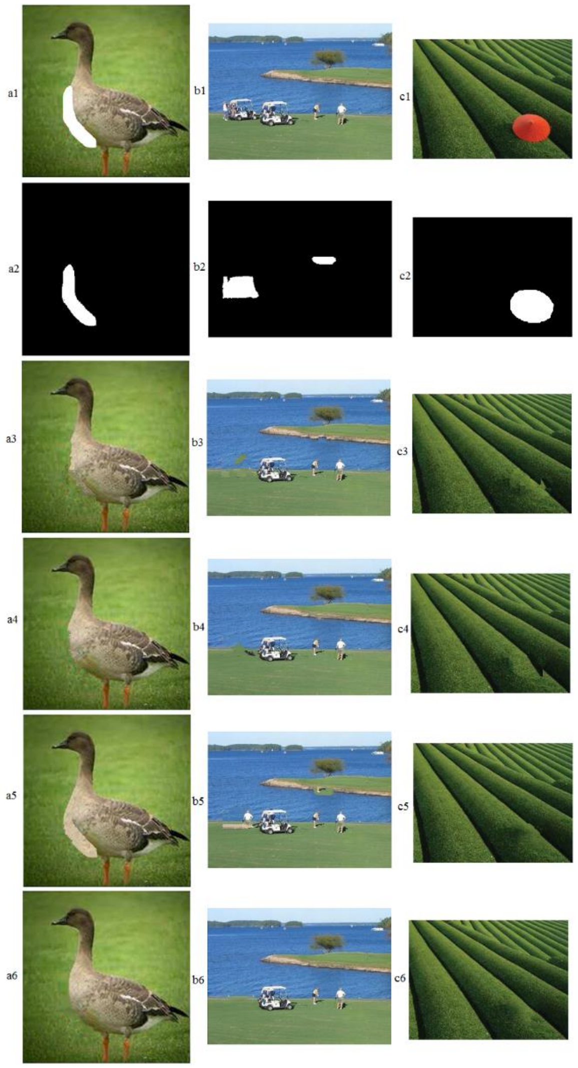


**Figure 9.** ($a_1$, $b_1$, $c_1$) Original Images, ($a_2$,$b_2$,$c_2$) masks, ($a_3$,$b_3$,$c_3$) image inpainted by Approximation algorithm [33], ($a_4$,$b_4$,$c_4$) inpainted images by Tensor SVD [34], ($a_5$,$b_5$,$c_5$) the results of Sparsity method [21], ($a_6$,$b_6$,$c_6$) are the result of our method

represent the original images, Figure 9 ($a_2$,$b_2$,$c_2$) are the related masks. The results of approximation matrix [33] can be seen in Figure 9 ($a_3$,$b_3$,$c_3$), also Figure 9 ($a_4$,$b_4$,$c_4$) are the results of HOSVD algorithm [34]. Figure 9 ($a_5$,$b_5$,$c_5$) demonstrates the Sparsity method results [21] and finally the results of our method can be seen in Figure 9 ($a_6$,$b_6$,$c_6$).

To evaluate the inpainting results we have used two objective criteria, PSNR [31] and SSIM [32] which are used commonly in similar works. It is noteworthy; PSNR and SSIM can be utilized when the reference image is available.

As mentioned earlier, in Figure 9 three images ($a_1$,$b_1$,$c_1$) were inpainted by different methods while the reference image is available for Figure 9 (b1) only. Subsequently for this image, objective measurement such as PSNR and SSIM can be calculated and objective criteria can't be computed for Figure 9 ($a_1$,$c_1$). We demonstrate the acquired PSNR and SSIM in different methods for inpainting Figure 9 ($b_1$) as can be seen in Table 1.

As previously mentioned, the image decomposition and the two-stage inpainting are necessary for our algorithm. Thus the time of decomposition and the time of the two-stage inpainting increase the runtime complexity. However the quality of our method is plausible. Another example for the result of our algorithm can be seen in Figure 10. Because the reference is not available for this image, we should use subjective criterion for the comparison. As can be seen, artificial effects are created in the sky by the other methods (Figure 10 (a3,a4,a5)) while the result of our algorithm (Figure 10 (a6)) looks better than the others.

Our novel method for structure inpainting which uses the quadtree is impressive to improve image quality. Because human vision is sensitive to the coherence of structure [35] and many methods attempt for recovering structure properly [13]. Consequently, the structure inpainting is very important for improving the final result. In our method, the quadtree assists to identify the correct path for a better inpainting.

It is noteworthy; our texture inpainting is a patch-based inpainting. The novelty of our method is applying LSK to the original image for better patch selection. Indeed the patch selection is accomplished in both of the texture components and the original image. So Euclidean distance is considered in the texture component while the geometric feature which acquired by LSK is attended in the original image. As a result patch selection by our method is highly efficient for texture inpainting.

**TABLE 1.** Comparison of different methods based on PSNR and SSIM (for inpainting the Figure 9 ($b_1$) )

| Method | PSNR | SSIM |
|---|---|---|
| Approximation matrix [33] | 27.014 | 0.960 |
| HOSVD [34] | 27.486 | 0.964 |
| Sparsity [21] | 25.435 | 0.958 |
| Our method | 28.021 | 0.971 |

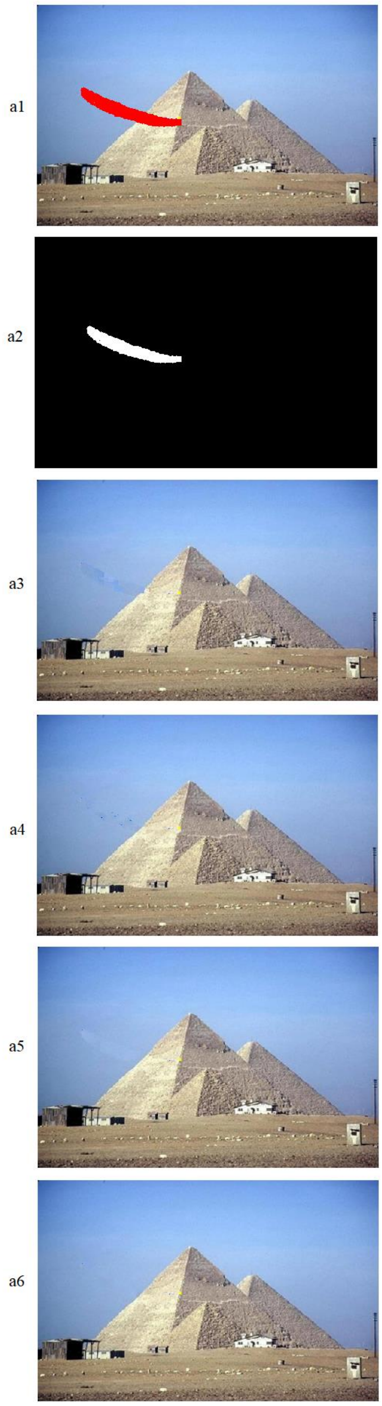


**Figure 10.** (a1) Original Image, (a2) mask, (a3) the inpainted image by Approximation algorithm [33], (a4) the inpainted image by Tensor SVD [34], (a5) the result of Sparsity method [21], (a6) the result of our algorithm

Another matter that should be considered is the ability of PCA as image decomposition in our method. Two components are obtained by PCA and each of them inpainted separately and then the final inpainted result can be easily acquired by adding two inpainted components. While in the majority of the image decomposition the original image can't be acquired by its components. For example, we test SVD instead of PCA in decomposition step. If SVD applied instead of PCA three matrix U, ∑ and V are gained [37]. Formally the SVD decomposition of a matrix $I_{(m\times n)}$ is $I = U\Sigma V^{*}$ where $\Sigma$ is a diagonal matrix and contains singular values. As a decomposition, we can hold k higher singular values and replace others with zero which is considered as a component. Furthermore, we set zero the k higher singular values and hold others so that the acquired image will be the other component. Although we can decompose an image and obtain two components, the result of inpainted components can't be combined comfortably. According to our experiment, when an image is inpainted, the number and values of its singular values are changed. Subsequently, the inpainted original image can't be obtained based on its components. The outputs of our research show that PCA is appropriate for image decomposition in the inpainting process.

## 5. CONCLUSION

In this paper, we proposed a new method for image inpainting based on structure and texture separately. In the first step, Image decomposition is accomplished based on the PCA to extract structure and texture components. Then the correct path for structure inpainting is identified by the quadtree. Also, texture image based on patch selection is inpainted by using Euclidian distance in texture and geometric feature in the original image.

Finally, the result can be obtained by adding two inpainted components. The proposed method can generate visually plausible results in comparison with state-of-the-art algorithms.

## 6. REFERENCES

1. Bertalmio, M., Sapiro, G., Caselles, V. and Ballester, C., "Image inpainting", 27th annual conference on Computer graphics and interactive techniques, (2000), 417-424.

2. Chan, T. F. and Shen, J., "Nontexture Inpainting by Curvature-Driven Diffusions", ***Journal of Visual Communication and Image Representation,*** Vol. 12, No. 4, (2001), 436-449.

3. Shen, J., Kang, S. H. and Chan, T. F., "Euler's Elastica and Curvature-Based Inpainting", ***SIAM Journal on Applied Mathematics***, Vol. 63, No. 2, (2003), 564-592.

4. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G. and Verdera, J., "Filling-in by joint interpolation of vector fields and gray levels", ***IEEE Transactions on Image Processing,*** Vol. 10, No. 8, (2001).

5. Pu, Y., Wang, W., Zhou, J. and Wang, Y., "Fractional Derivative Detection of Digital Image Texture Details and Implementation of Fractional Derivative Filter", ***Science in China Series F: Information Sciences,*** Vol. 38, No. 12, (2008), 1319-1339.

6. Ruzic, T. and Pizurica, A., "Context-aware patch-based image inpainting using Markov random field modeling", ***IEEE Transactions on Image Processing,*** Vol. 24, No. 1, (2015), 444-456.

7. Wang, H., Jiang, L., Liang, R. and Li, X. X., "Exemplar based image inpainting using structure consistent patch matching", ***Neuro Computing***, Vol. 269, (2017), 90-96.

8. Criminisi, A., Pérez, P. and Toyama, K., "Region filling and object removal by exemplar-based image inpainting," ***IEEE Transactions on Image Processing,*** Vol. 13, (2004), 1200-1212.

9. Chen, Z., et al, "Structure aware image inpainting using patch scale optimization," ***Journal of Visual Communication and Image Representation,*** Vol. 40, (2016), 312-323.

10. Isogawa, M., Mikami, D., Takahashi, K. and Kojima, A., "Image and video completion via feature reduction and compensation," ***Multimedia Tools and Applications,*** Vol. 76, (2017), 9443-9462.

11. Bi, X., Liu, H., Lu, G., Wei, J., Chao, Y., "Exemplar-Based Image Inpainting using Automatic Patch Optimization", International Conference on Video and Image Processing, (2018), 128-133.

12. Zhang, N., Ji, H., Liu, L. and Wang, G., "Exemplar-based image inpainting using angle-aware patch matching", ***EURASIP Journal on Image and Video Processing,*** Vol. 70, (2019). https://doi.org/10.1186/s13640-019-0471-2

13. Chen, Z., Dai, C., Jiang, L., Sheng, B., Zhang, J., Lin, W., & Yuan, Y. "Structure aware image inpainting using patch scale optimization", ***Journal of Visual Communication and Image Representation,*** Vol. 40, (2016), 312-323.

14. Liu, J., Musialski, P., Wonka, P. and Ye, J., "Tensor completion for estimating missing values in visual data," ***IEEE Transactions on Pattern Analysis and Machine Intelligence,*** Vol. 35, No. 1, (2013), 208–220.

15. Liang, X., Ren, X., Zhang, Z. and Ma, Y., "Texture repairing by unified low rank optimization," ***Journal of Computer Science and Technology,*** Vol. 31, No. 3, (2016), 525-546.

16. Calvi, G. G., Dees, B. S. and Mandic, D. P., "Tight lower bound on the tensor rank based on the maximally square unfolding," (2019).

17. Fazel, M., "Matrix rank minimization with applications," PhD thesis, Stanford University, (2002).

18. Bengua, J. A., Phien, H. N., Tuan, H. D. and Do, M. N., "Efficient tensor completion for color image and video recovery: low-rank tensor train", ***IEEE Transactions on Image Processing,*** Vol. 26, (2017), 2466-2479.

19. Guo, Q., Gao, S., Zhang, X., Yin, Y. and Zhang, C., "Patch-based image inpainting via two-stage low rank approximation", ***IEEE Transactions on Visualization and Computer Graphics,*** Vol. 24, (2018), 2023-2036.

20. Jin, X., et al, "Sparsity-based image inpainting detection via canonical correlation analysis with low-rank constraints", ***IEEE Access,*** Vol. 6, (2018), 49967-49978.

21. Xu, Z. and Sun, J., "Image inpainting by patch propagation using patch sparsity", ***IEEE Transactions on Image Processing,*** Vol. 19, No. 5, (2010), 1153-1165.

22. Jin, D. and Bai, X., "Patch-sparsity-based image inpainting through a facet deduced directional derivative", ***IEEE Transactions on Circuits and Systems for Video Technology,*** Vol. 29, (2019), 1310-1324.

23. Melgosa, M., Testing CIELAB-based color difference formulas", ***Color Research & Application***, Vol. 25, (2000), 49-55.
24. Zhang, M. and Gunturk, B. K., "Multi resolution bilateral filtering for image denoising", ***IEEE Transaction on Image Processing,*** Vol. 17, No. 12, (2008), 2324-2333.
25. Zhang, B. Y. and Allebach, J. P., "Adaptive bilateral filter for sharpness enhancement and noise removal", ***IEEE Transaction on Image Processing,*** Vol.17, No. 5, (2008), 664-678.
26. Wang, J., Chen, Y., Li, T., Lu, J. and Shen, L., "Residual-based kernel regression method for image denoising", ***Mathematical Problems in Engineering,*** (2016).
27. Zhang, H., Yang, J., Zhang, Y. and Huang, T. S., "Image and video restorations via nonlocal kernel regression", ***IEEE Transactions on Cybernetics,*** Vol. 43, No. 3, (2013), 1035-1046.
28. Sun, Z., Han, B., Li, J., Zhang, J. and Gao, X., "Weighted guided image filtering with steering kernel", ***IEEE Transactions on Image Processing,*** Vol. 29, (2019), 500-508.
29. Ghorai, M., Samanta, S., Mandal, S. and Chanda, B., "Multiple pyramids based image inpainting using local patch statistics and steering kernel feature", ***IEEE Transactions on Image Processing,*** Vol. 28, No. 11, (2019), 5495-5509.
30. Zhang, K., Gao, X., Tao, D. and Li, X., "Single image super-resolution with non-local means and steering kernel regression," ***IEEE Transactions on Image Processing,*** Vol. 21, No. 11, (2012), 4544-4556.
31. Ying, H., Kai, L. and Ming, Y., "An improved image inpainting algorithm based on image segmentation," ***Computer Science,*** Vol. 107, (2017), 796-801.
32. Ogawa, T. and Haseyama, M., "Image inpainting based on sparse representations with a perceptual metric," ***EURASIP Journal on Advances in Signal Processing,*** No. 1, (2013), 179.
33. Alilou, V. K., Yaghmaee, F., "Exemplar-based image inpainting using svd-based approximation matrix and multi-scale analysis," ***Multimedia Tools and Application,*** Vol. 76, No. 5, (2017), 7213-7234.
34. Ghorai, M., Mandal, S. and Chanda, B., "A two-step image inpainting algorithm using tensor svd", Computer Vision-ACCV 2014 Workshops, (2014), 63-77.
35. Vese, L., Sapiro, G. and Osher, S., "Simultaneous structure and texture image inpainting", ***IEEE Transactions on Image Processing,*** Vol. 12, No. 8, (2003), 882-889.
36. Takeda, H., Farsiu, S. and Milanfar, P., "Kernel regression for image processing and reconstruction," ***IEEE Transactions on Image Processing,*** Vol. 16, No. 2, (2007), 349-366.
37. Kasana, G., Singh, K. and Singh Bhatia, S., "Singular value decomposition based steganography technique for JPEG2000 compressed images" , ***International Journal of Engineering (IJE) Transaction,*** Vol. 28, No. 12, (2015), 1720-1727.
38. Priyadharshini, R. A. and Arivazhagan, S., "Object recognition based on local steering kernel and SVM," ***International Journal of Engineering, Transaction B: Applications***, Vol. 26, No. 11, (2013), 1281-1288.


## Persian Abstract


**چکیده**

در این مقاله، یک روش جدید برای ترمیم تصویر با استفاده از ساختار و بافت و بر اساس quadtree آورده شده است. به طور کلی یک تصویر قابل تجزیه به دو تصویر ساختار و بافت می‌باشد که هر کدام از این اجزاء می‌توانند ویژگی‌های خاصی از تصویر را نشان دهند. در این مقاله ابتدا به کمک روش PCA تجزیه تصویر انجام شده و سپس هر یک از اجزاء به طور جداگانه ترمیم می‌شوند. در بخش ترمیم ساختار از مفهوم quadtree استفاده شده است. با اعمال quadtree تصویر به نواحی متمایزی افراز می‌شود به طوری که پیکسل‌های هر ناحیه بر اساس یک حد آستانه‌ای به هم شباهت دارند و نسبت به ناحیه دیگر متفاوت هستند. اندازه هر ناحیه بدست آمده پس از اعمال quadtree می‌تواند اهمیت آن ناحیه را نشان دهد و ما از این امر در ترمیم ساختار استفاده می‌نماییم. همچنین در ترمیم بافت یک روش ترمیم مبتنی بر وصله مطرح نمودیم که از LSK برای تشخیص وصله‌های مشابه در ترمیم تصویر استفاده می‌نماید. پس از ترمیم هر یک از اجزاء، تصویر نهایی از ترکیب دو جزء ترمیم شده بدست می‌آید. نتایج روش پیشنهادی با استفاده از معیارهای کمّی چون PSNR و SSIM نشان می‌دهد که استفاده از روش پیشنهادی ما می‌تواند به کیفیت ترمیم تصویر کمک قابل توجهی نماید.