

Research on Algorithm for Fast Image Inpainting Based on Exemplars

Shengfeng Zhu

Hanjiang Normal University, Shiyao 442000, China

Keywords: Image inpainting; exemplars-based; inpainting algorithm;

Abstract: An exemplars-based two-step image inpainting algorithm is presented. Firstly, the input image is inpainted synchronically with its coarse version via the patch-GST-based inpainting priority and best-match patches based on the weighted SSD. This inpainted image is plausible except some artifacts in the areas that contain edges. Then, the second-time inpainting proceeds employed only on these areas. In the searching of the best-match patches, patch structural distance and dynamic search window techniques are proposed to improve the searching speed. Experimental results show that the inpainted images obtained by the proposed algorithm can effectively recover the structural continuity of the input image. In addition, the processing speed of the proposed algorithm is much faster than the existing competing algorithms.

1. Introduction

Image inpainting, which aims to recover the missing regions of an image, is a hot topic in the fields of computer vision and image processing. It has been widely used in many applications like image restoration, image editing and encoding, etc. There are basically two categories of image inpainting schemes: PDE (Partial Derivative Equation)-based schemes [1] and exemplar-based ones [2]. The former is founded on the theory of PDE and variational methods. Some blur artifacts are easily introduced in this kind of algorithms, especially when the missing region is large. The latter relies on the small image patches to propagate textures and structures from the known regions into the unknown regions, without causing blur effects. Its basic idea is that missing pixels in one patch can be inpainted via one or several best-match patches. Unfortunately, finding the best match patch for every patch to be inpainted is a time consuming task. It takes several minutes for traditional exemplar-based inpainting methods to complete an image with common size. In real applications, we need a more fast way to do this.

In this paper, a novel fast two-step exemplar-based inpainting algorithm is proposed. Firstly, one level of Digital Wavelet Transform (DWT) is performed on the image to be inpainted. Only the LL sub-band of the wavelet domain coefficients is used as the coarse version of the input image such that the original missing regions are then inpainted with an exemplar-based approach like that in [2] synchronously. Secondly, to overcome the artifacts of the first inpainted original image, which appears in the regions that contain edges, a second-time inpainting is performed only in those regions. Through this two-step coarse-to-fine inpainting process, both the structures and the textures of the original input image can be well propagated from the known regions into the missing regions. Moreover, in order to speed up the best-match patch searching process, two techniques (i.e., the dynamic search window and the patch structure distance test) are proposed. The proposed algorithm can achieve plausible results very close to those of the state-of-the-art approaches, but in a much faster way with lower computational cost.

The remainder of this paper is organized as follows. Section 2 describes the proposed method in detail. Section 3 presents the performance of the presented method and comparisons with other approaches. Section 4 concludes the paper.

2. Algorithm Description

2.1 Exemplar-based inpainting

Given an image I with the unknown/missing region Ω and known region Ω' , the goal of the proposed algorithm is to complete Ω via the information in Ω' . Like the classical exemplar-based inpainting algorithms, two main tasks of our approach are to determine the filling order and to find the best-match patches for the inpainting.

1) GST(Gradient Structure Tensor)-based filling order:

Denoting the contour of Ω as $\partial\Omega$, we determine the filling order by the priority of each pixel on $\partial\Omega$. For a patch Ψ_p centered at p , its priority $P(p)$ is defined as follows:

$$P(p) = C(p) \times D(p) \quad (1)$$

$C(p)$ is the confidence term [2] defined as

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \Omega'} C(q)}{|\Psi_p|} \quad (2)$$

$|\cdot|$ calculates the number of pixels in the patch. The intention of $C(p)$ is to favor those patches which have more of their pixels already filled [2]. $D(p)$ is the data term to increase the priority of those patches with structures. Some researchers proposed different ways to compute the data term [3], [4]. In this paper, like [4], the data term is computed based on GST, which can well describe the image structure as well as its orientation and has been widely used in image processing fields [5].

2) Finding best match via weighted SSD: To find the best match patch of Ψ_p , [2] resorts to the Sum of Squared Distances (SSD) of the known region between the patches.

In [6], a gradient-based prefilling method is proposed to estimate the values of pixels in the unknown region of a patch. Based on the same prefilling method, [7] uses the WSSD (Weighted SSD) in both known region and unknown region to find the best-match patch. Their prefilling method is based on the assumption that the gradients of the pixels in the unknown region are equal to zero. Under this assumption, the unknown region can be estimated by the least square solution of the gradient equations. This simple method only takes into account the local gradients within the patch, so the result cannot be plausible. In [3], the authors observed that the newly filled pixels in the inpainted patch should be consistent with the neighboring patches in appearance. Based on this observation, we propose a new prefilling method. We almost use the same denotations as [3].

2.2 Smart best-match patch searching

In the exemplar-based inpainting algorithm, the computation complexity concentrates on searching of the best-match patch. To make the searching process faster, a KD-tree based approximate nearest neighbor searching algorithm has been introduced in [6]. In this section, we consider this problem in a different way. Two techniques are proposed to make the searching process smarter and faster.

1) Patch structure distance test: SSD computing is the most time consuming in the best-match patch searching process. As mentioned in 2.1.1, the patch GST histogram represents the structural information of the patch. If SSD values of two patches are small, their structural information must be similar, and so are their patch GST histograms.

2) Dynamic search window: In traditional exemplarbased methods, the search area Φ in (3) is the whole unknown region or a fixed window centered at the center of each patch to be inpainted [2]-[4].

In Fig. 1, for example, the blue patch is to be inpainted, the neighbor window and the local window are marked by red and green colors respectively, and the found best-match patch is marked in black. In this way, the whole search scope can be narrowed down dynamically in the inpainting process. Since such a neighbor window is introduced, the size of the local window becomes much smaller. In this paper, the sizes of the local window and the neighbor window are set to 8~10 times and 5~7 times of the patch size, respectively. But when the neighbor window is null, the size of the local

window can be larger.

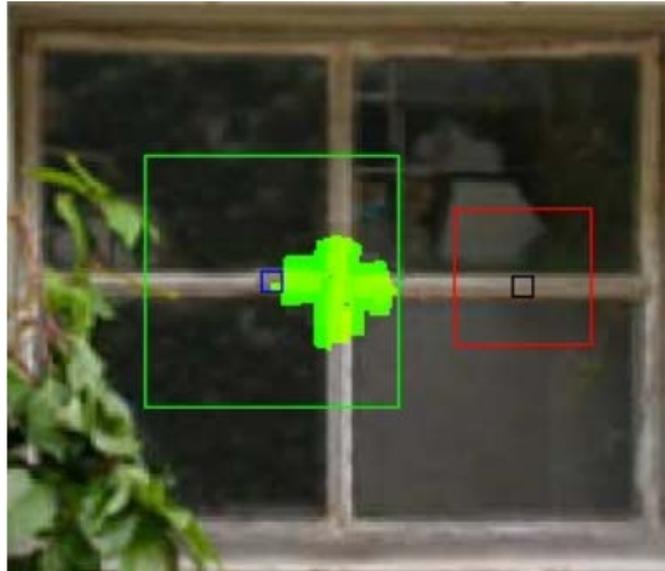


Figure 1. Local window (green) and its neighbor window (red)

C. Two-step coarse-to-fine inpainting In image inpainting, two main difficulties are to preserve the continuous structures and textures of the input image and to reduce the computation time of the inpainting operation as much as possible. We propose a two-step coarse-to-fine inpainting algorithm based on the following consideration: Not only can a down sampled image be inpainted much faster, but also through DWT based multi-resolution analysis, the structure information of the original image can be well preserved so as to make the inpainting result more plausible. In our two-step inpainting algorithm, the coarse version of the input image is acquired through an one-level DWT, so the coarse-version image is down sampled by 2 from the original one.

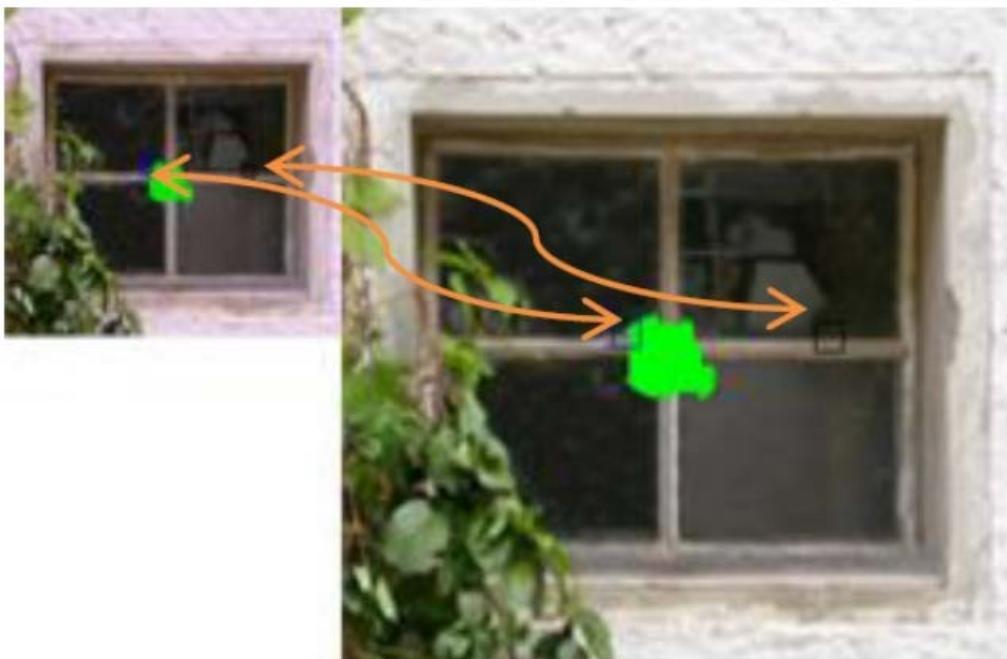


Figure 2. Image inpainting based on the coarse-version image

We observed that the original input image can be inpainted synchronously with its coarse version. In Fig. 2, for example, the left image C is a coarse version of input image, and the right one is the corresponding original image O ; if for the patch Ψ_c in C located at (x, y) , its best-match patch $\hat{\Psi}_c$ is found to be located at (i, j) , then the corresponding patch Ψ_o in O located at $(2x, 2y)$ can be inpainted via the corresponding match patch Ψ_o located at $(2i, 2j)$. The flow e chart of the proposed

two-step inpainting algorithm is shown in Fig.3. In the first inpainting, both the coarse-version and the original input image can be inpainted synchronously, but only the inpainting result at original resolution is required for the following step.

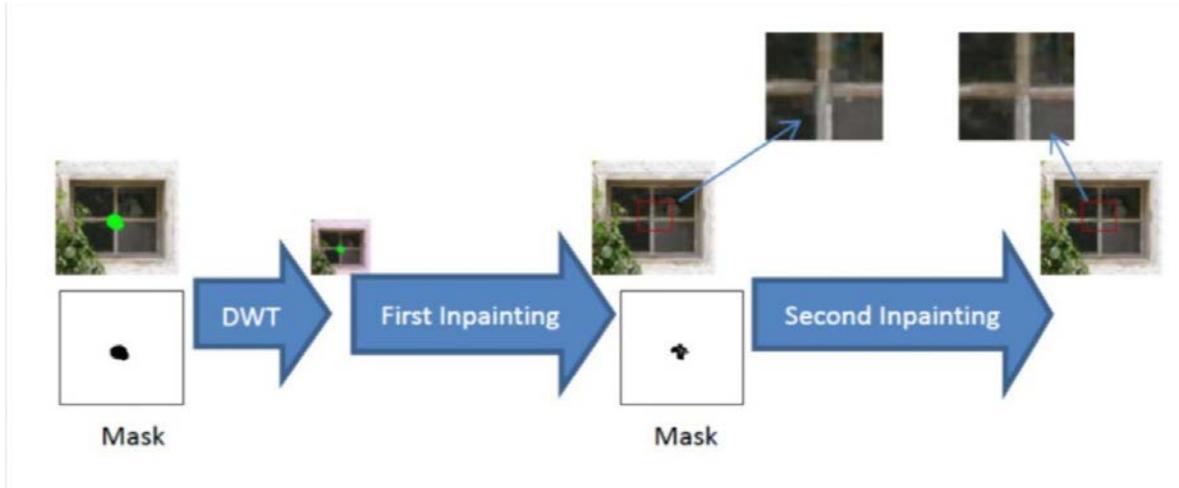


Figure 3. Flow chart of two-step inpainting

We can see that the first-time inpainted image (shown in the middle of Fig.3) can recover most of the structures and textures, but some artifacts exist in the edge regions (marked by the red rectangle with a zoomed-in version shown in the right-top). To overcome this defect, the second inpainting is performed only in these areas. The edge regions with artifacts can be located through PSE. The pixels that need to be inpainted twice are defined as $\forall p \in \Omega, Pe(p) > ET$ hres. In this paper, ET hres is set to $0.015 * M \times q \in I (Pe(p))$. In the middle of Fig.3, we can see that the updated mask for the second inpainting is smaller than the original one, and only covers those areas with artifacts. In the first inpainting, the missing regions are firstly estimated by the prefill method, then the output of the first inpainting will be used as the input of the second inpainting.

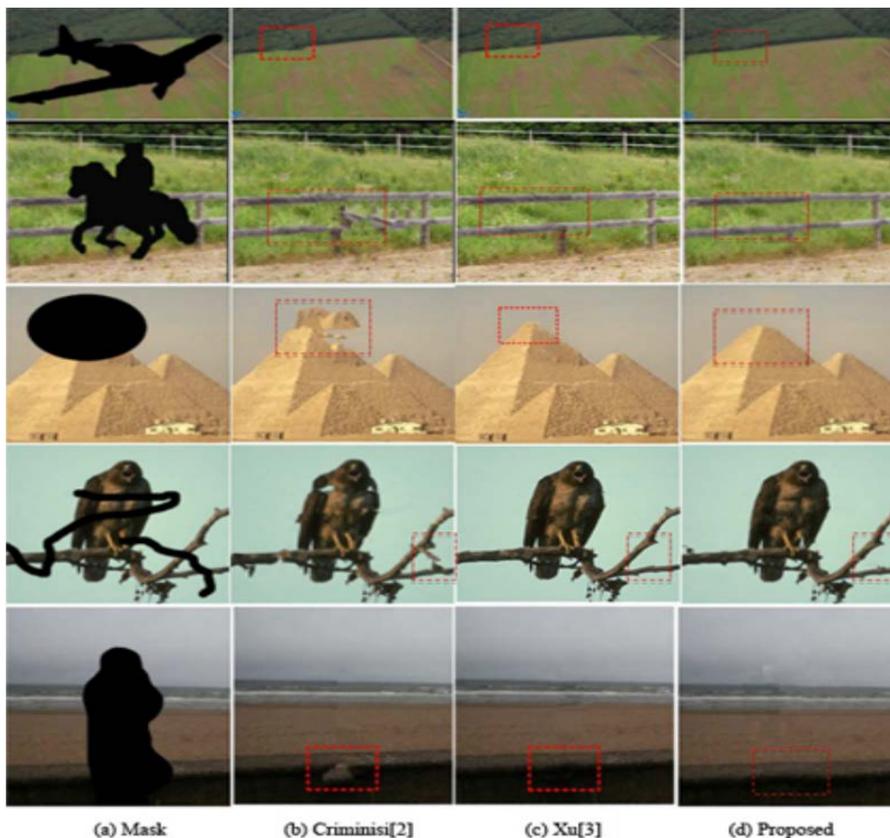


Figure 4. Inpainted results and comparison with Criminisi[2] and Xu[3]

3. Results

Our algorithm is tested on a variety of natural images. In Fig.6, some of the results are demonstrated and compared with the existing state-of-the-art approaches [2], [3]. In the following examples, the patch size is set to 7×7 in the coarse-version images and 9×9 in the original ones. The size of $N_s(p)$ is set to 51 as in [3]. The algorithm is implemented with C++ programming language. All the experiments are run on a PC equipped with an Intel 2.67 GHz CPU.

In Fig. 4, we can see that in the images inpainted by our method, the missing regions can be effectively filled with continuous structures and textures. Our results outperform those inpainted by [2], and are close to the results in [3]. In our implementation of [2]'s method, the patch size is set to 9×9 , some results are different from those given in [3]. The comparison of the computation time is shown in Tab.1, where the images are corresponding to those of Fig.6 from top to down. Because of the lower computational complexity, our algorithm is much faster than both [2] and [3], thanks to the multi-resolution two-step inpainting, the dynamic search window and PSD test speed up the schemes.

4. Conclusion

In this paper, a fast two-step exemplar-based image inpainting algorithm is presented. The first inpainting is based on the coarse version of the input image, in which the highfrequency details are filtered by DWT. Then the second inpainting is performed only in the areas that contain edges at the original resolution. In the second inpainting, results of first inpainting are treated as prefill result, and are computed as a part of the WSSD, in such a way that the structural information of the known regions of the input image can be well propagate into the missing regions. The main merits of this algorithm are in the following four points: • first, the DWT based two-step inpainting scheme can quickly obtain the inpainted results at the original resolution; • second, the GST based filling order enables sharp structures to be preserved in the inpainted result; • third, the prefilling scheme based on the image locality and continuity is more effective and accurate; • forth, using the dynamic search window and PSD test, the best-match searching process can be smarter and faster. Compared with the state-of-the-art algorithms, our method is simpler and easier to implement, and the results are comparable, but the computation cost is much lower. In some applications, real-time image inpainting is required and can be done via parallelization and GPU computing [6]. We will consider this in the future work.

References

- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/AddisonWesley Publishing Co., 2000, pp. 417–424.
- [2] A. Criminisi, P. Prez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, pp. 1200–1212, 2004.
- [3] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *Trans. Img. Proc.*, vol. 19, no. 5, pp. 1153–1165, May 2010.
- [4] O. Le Meur, J. Gautier, and C. Guillemot, "Exemplar-based inpainting based on local geometry," in *ICIP*, 2011, pp. 3401– 3404.
- [5] U. Kothe, *Edge and junction detection with an improved structure tensor: Pattern Recognition, Proc. of 25 th DAGM Symposium, Springer LNCS 2781*. Springer, 2003, pp. 25–32.
- [6] T.-H. Kwok, H. Sheung, and C. Wang, "Fast query for exemplar-based image completion," *Image Processing, IEEE Transactions on*, vol. 19, no. 12, pp. 3106 –3115, dec. 2010.
- [7] D. J. Florinabel, S. E. Juliet, and V. Sadasivam, "Combined frequency and spatial domain-based patch propagation for image completion," *Computers and Graphics*, pp. 1051–1062, 2011.