A Robust and Adaptive Image Inpainting Algorithm Based on a Novel Structure Sparsity

Zhidan Li, Hongjie He, Zhongke Yin, and Fan Chen
Sichuan Key Laboratory of Signal and Information Processing, Southwest Jiaotong University, Chengdu, Sichuan, P.R. China

Abstract—The existing patch sparsity based image inpainting algorithms have some problems in maintaining structure coherence and neighborhood consistence. To address the above problems, a robust and adaptive image inpainting algorithm based on a novel structure sparsity is proposed. The main improvement includes the following three aspects. Firstly, a novel structure sparsity function is defined according to the sparseness of the patch’s nonzero similarities to its neighboring patches to encourage structure propagation preferentially. Secondly, the neighborhood consistence constraint factor is adaptively determined according to the target patch’s structure sparsity value, which aims to reduce block effect and seam effect. Thirdly, to improve computational efficiency, the size of local search region is dynamically determined in accordance with the target patch’s structure sparsity value. Experimental results demonstrate that the proposed algorithm can obtain more pleasurable vision results than that by other similar methods.

Keywords: structure sparsity, image inpainting, adaptive, neighborhood consistence constraint

1. Introduction

With the rapid development of computer technology and multimedia technology in the recent years, image inpainting has become a hot research topic in the field of computer graphics and computer vision. Image inpainting, also known as image completion or image disocclusion, is a research area which uses known information to fill missing region under some rules, with the goal to achieve a visually plausible results. Image inpainting technique can be widely used in various areas, such as image coding and transmission, image and video editing.

Image inpainting methods can be roughly divided into two families: diffusion-based inpainting methods and exemplar-based inpainting methods. The former ones, where the missing area is filled by diffusing the image information from the source area into the missing area slowly, are based on the theory of partial differential equation[1-5]. And they have acquired remarkable achievement for filling the non-textured or reversely smaller damaged area. However, because the diffusion-based approaches implicitly assume that the content of the missing region is smooth and nontextured, they are also inclined to bring in smooth effect in the textured area or large damaged region. Criminisi et al. [6, 7] proposed an exemplar-based inpainting algorithm which is suitable for repairing larger missing regions with different feature of structures and textures. In this approach, priority function and match criterion, which were adopted to determine filling order and find the most similar patch respectively, and firstly introduced into exemplar-based inpainting algorithm. The filling order and match criterion are two key issues of exemplar-based inpainting algorithms, and many researchers study on them to achieve more pleasurable repair results. Cheng et al. [8] amended the definition of priority function to obtain a more robust filling order. Wu [9] proposed a novel exemplar-based completion model which used a cross-isophote diffusion item to decide the filling order. Jemi [10] proposed to use DCT coefficients of exemplar to decide filling order and add the edge information into the match criterion to find the most similar patch. An exemplar-based inpainting based on local geometry was proposed in [11], where structure tensors were employed to define the priority and template matching. Wang [12] improved the exemplar-based inpainting algorithm by using D-S evidence theory to compute priority. Compared with the diffusion-based inpainting methods, the exemplar-based inpainting algorithms have got plausible results for inpainting the larger missing region. However, only the information of the patch located at fill-front was used to compute priority, and it is inadequate because the neighboring information was not considered. To solve the drawbacks, Zhang [13] proposed a novel priority scheme based upon the color distribution. Xu and Sun [14] proposed an inpainting algorithm based on patch sparsity. Patch sparsity was reflected in two aspects: firstly, structure sparsity function was defined to measure the sparseness of nonzero similarities of a patch with its neighboring patches, and it was used to compute priority function to get a more robust filling order; secondly, multiple candidate patches were selected to represent the target patch sparsely and then copy the sparse representation information to the target patch’s missing region. Hesabi et al. [15] used structure sparsity and modified confidence term to compute priority to get a more robust filling order, but they still only used one candidate patch to fill missing region. However, Xu’s algorithm still cannot well maintain structures coherence and textures consistence, because structure sparsity could not well measure confidence of a patch located at structure
region, and neighborhood consistence constraint factor was fixed in different regions.

This paper proposes a robust and adaptive image inpainting algorithm to improve Xu’s algorithm from three aspects: Firstly, a novel structure sparsity function is defined by measuring the confidence of a patch located at structure area instead of texture region. Secondly, the neighborhood consistence constraint factor is adaptively determined on the basis of target patch’s structure sparsity value. Thirdly, local search region size is adaptively determined by target patch’s structure sparsity value. Compared with Xu’s algorithm, a novel structure sparsity function defined in this paper can better encourage the structure region to be filled preferentially; and because neighborhood consistence constraint factor changes with different neighboring features, the algorithm can obtain better guide information of sparse representation to maintain the consistence with neighboring information; also local search strategy can decrease computational complexity.

This paper is organized as follows. In Section 2, Xu’s patch sparsity based image inpainting algorithm is briefly described. In Section 3, the details of the proposed image inpainting algorithm, including the novel structure sparsity, adaptive neighborhood consistent constraint and dynamic local search region are expounded. The experimental results and comparisons with previous algorithms are presented in the Section 4. Finally, we conclude this work in Section 5.

2. Patch sparsity based algorithm

Xu and Sun proposed an image inpainting algorithm based on patch sparsity[14]. In this algorithm, structure sparsity was used to determine filling order. After target patch was selected, multiple candidate patches were found under sum of squared distance (SSD) criterion and were used to represent missing information sparsely. Then the sparse representation information was used to fill missing region. The procedures repeat until all missing pixels were filled. In the following, we briefly describe the structure sparsity and sparse representation.

2.1 Structure sparsity

Given an input image I, the missing region is denoted by \( \Omega \) and its fill-front is \( \delta \Omega \). The source region is indicated by \( \Phi = I - \Omega \). Let \( \Psi_p \) be a square patch centered at the point \( p \in \delta \Omega \), then structure sparsity \( S(p) \) is defined as:

\[
S(p) = \sqrt{\sum_{k \in N_s(p)} \omega_{p,k}^2 \frac{|N_s(p)|}{|N(p)|}}
\]

(1)

where \( N(p) \) is a neighborhood window centered at \( p \), which is set to be larger than the size of patch \( \Psi_p \); \( \omega_{p,k} \) measures the similarity between patch \( \Psi_p \) and its neighboring patch \( \Psi_k \). The terms \( N_s(p) \) and \( \omega_{p,k} \) are defined as follows:

\[
N_s(p) = \{ k | k \in N(p) \text{ and } \Psi_k \subset \Phi \} \tag{2}
\]

\[
\omega_{p,k} = \frac{1}{Z(p)} \exp \left( -\frac{d(\Psi_p, \Psi_k)}{25} \right) \tag{3}
\]

where \( d(\cdot, \cdot) \) measures the mean squared distance and \( Z(p) \) is a normalization constant such that \( \sum_{k \in N_s(p)} \omega_{p,k} = 1 \).

2.2 Patch sparse representation

Let \( \Psi_p \) be the target patch, \( F \) and \( E \) be two matrices to extract already known and missing pixels of \( \Psi_p \), respectively, \( \{ \Psi_q \}_{q=1,\ldots,M} \) be the top \( M \) most similar patches, then \( \Psi_p \) is approximated by the linear combination of \( \{ \Psi_q \}_{q=1,\ldots,M} \), i.e.,

\[
\Psi_t = \sum_{q=1}^{M} \omega_q \Psi_q \tag{4}
\]

Then the unknown pixels in patch \( \Psi_p \) are filled by the corresponding pixels in \( \Psi_t \), i.e., \( E \Psi_p = E \Psi_t \). The constraints for the linear combination in (4) include two aspects:

One is that the estimated patch \( \Psi_t \) should approximate the target patch \( \Psi_p \) over the already known pixels, i.e.,

\[
||F \Psi_t - F \Psi_p||^2 \leq \delta \tag{5}
\]

The other is that newly filled pixels in the estimated patch \( \Psi_t \) should be consistent with the neighboring patches in appearance, i.e.,

\[
||\beta (E \Psi_t - E \sum_{k \in N_s(p)} \omega_{p,k} \Psi_k)||^2 \leq \delta \tag{6}
\]

where \( \omega_{p,k} \) is same as defined in (3), \( \beta \) balances the strength of the constraints in (5) and (6).

The combination coefficients \( \tilde{\alpha} = \{ \alpha_1, \alpha_2, \cdots, \alpha_M \} \) are inferred by minimizing a constrained optimization problem in the framework of sparse representation. Then the linear combination coefficients \( \tilde{\alpha} \) can be inferred by optimizing the constrained optimization problem:

\[
\arg \min \{ ||\tilde{\alpha}||_0 \} \text{ s.t. } ||D \Psi_t - \Psi_t||^2 < \delta \text{ and } \sum_{i} \alpha_i = 1 \tag{7}
\]

where \( D = [ F \quad \beta E ]^T \) and \( \Psi_t = [ F \Psi_p \quad \beta E \sum_{k \in N_s(p)} \omega_{p,k} \Psi_k ]^T \).

3. Proposed algorithm

To solve of the drawbacks of Xu’s approach that structure coherence and texture consistence could not well be maintained, this paper proposed a robust and adaptive inpainting algorithm by introducing a novel structure sparsity function which is used to determine the filling order, neighborhood consistence constraint coefficients and local search region size. The details of our algorithm is expounded in the following.
3.1 Novel structure sparsity

From the definition of structure sparsity in Xu’s algorithm, we can know that structure sparsity value is relate to the similarities between a patch and its neighboring known patches and the ratio between the numbers of known patches and the numbers of patches in the neighborhood (i.e. \( |N_r(p)|/|N(p)| \)). Suppose that a patch locates at structure part, but the numbers of known patches of its neighborhood is relative less, then structure sparsity value will be relative small and the structure patch will not be filled preferentially, which will result in the structures incoherence. Based on the consideration, a novel structure sparsity is proposed in this paper:

\[
S(p) = \frac{\sum_{k \in N_r(p)} \omega_{p,k}^2}{\omega_{p,p}^2}
\]

where \( \omega_{p,k} \) is same defined as (3). From the definition of novel structure sparsity, we can see that its value only relate to similarities. This definition is to avoid structure sparsity value being too small when a patch located at structure part with relative less surrounding known patches. Also we can learn that structure sparsity value increases with respect to the sparseness of patch’s nonzero similarities to its neighboring patches. When a patch locates at structure part, it is saliently distributed within local region, therefore, it has higher structure sparsity value; while a patch locates at smooth part, it has many similar patches in the local neighborhood region, hence, it has smaller structure sparsity value. Under the guidance of structure sparsity, the patches located at structures (e.g., edges and corners) will have higher priority for patch inpainting compared with patches in texture or smooth regions. Since the similarity range is between 0 and 1, structure sparsity will also be in the range of 0 and 1. When a patch locates at smooth region, its structure sparsity value is close to zero hence it cannot lead to a robust filling order. In this paper we use the following transform to avoid structure sparsity value being too small, i.e.,

\[
S(p) = \lambda S(p) + (1 - \lambda)
\]

where 0 < \( \lambda \) < 1. In this paper \( \lambda \) is set to be 0.75.

3.2 Adaptive neighborhood consistent constraint

We still adopt the sparse representation of multiple candidate patches to fill missing information. Differing from Xu’s approach where the neighborhood consistency constraint factor was fixed, we use a varied neighborhood consistency constraint factor. Image patch has different similarities to its neighboring patches, therefore the neighborhood consistency constraint will be different. Neighborhood consistency constraint can be reflected by factor \( \beta \), then different neighborhood consistency constraint can be obtained for different region via adjusting factor \( \beta \). When a patch locates at structure part, its structure sparsity value is relative large. Since the similarity between this patch and its neighborhood is relatively small, we should use relatively small neighborhood consistence constraint to maintain clarity of structure part. On the contrary, when a patch locates at smooth region, its structure sparsity value is relative small. Because the similarity between the patch and its neighborhood is relative large, we should apply relatively large neighborhood consistence constraint to reduce block effect and seam effect. Therefore, we can vary the neighborhood consistence constraint with structure sparsity value. Based on the inversely proportional relationship between neighborhood consistence constraint and structure sparsity value, we adaptively determine the factor \( \beta \) according to:

\[
\beta = 1/(\rho \cdot S(p))
\]

Because structure sparsity value varies between 0.25 and 1, and \( \beta \) should be in the range of [0,1], we multiply a factor \( \rho \) in the denominator. In this paper, \( \rho \) is set to be 6.

3.3 Dynamic local search region

The original global search in Xu’s algorithm is time-consuming. To improve the performance from a efficiency perspective, we propose to adopt a local search method. From the definition of structure sparsity, we know that patch’s location feature can be reflected by its structure sparsity value. The higher structure sparsity value is, the more likely that the patch locates at structure part and the smaller similarity between patch and its neighborhood is, and the search region size should be larger to find similar patch. The smaller structure sparsity value is, the more likely that a patch locates at smooth region, and the higher similarity between the patch and its neighborhood is, hence the local search region can be set smaller to decrease computational complexity rapidly. Here, we adaptively decide the local search region according to the target patch’s structure sparsity value and the search region radius \( W \) is determined by

\[
W = \begin{cases} \gamma \cdot S(p), & \text{if } \gamma \cdot S(p) > 30 \\ 30, & \text{others} \end{cases}
\]

where \( \gamma \) is weight factor, in this paper \( \gamma \) is set to be 60.

3.4 Steps of the proposed algorithm

Let the degraded image indicated by \( I \) and missing region indicated by \( \Omega \), the steps of our painting algorithms are as follows.

Step 1: Compute priority. For any patch \( \Psi_p \) centered at \( p \) for point \( p \in \delta \Omega \), the priority \( P(p) \) is calculated by

\[
P(p) = C(p) \cdot S(p)
\]

where \( S(p) \) is defined in (9) and \( C(p) \) is the confidence term, defined as

\[
C(p) = \sum_{q \in \Psi_p} C(q) / |\Psi_p|.
\]

Step 2: Search candidate patches. After priority values of all patch centered at fill-front are calculated, the patch
with the biggest priority values is selected as the target patch \( \Psi_m \). Then search the top \( M \) most similar patches in the local source region (its size is determined by equation (11)) under SSD criterion \([6,7]\).

**Step 3:** Sparse representation. Use the \( M \) candidate patches to sparse representation the target patch \( \Psi_m \). Before the sparse representation process, neighborhood consistence constraint coefficient is determined by target patch’s structure sparsity value (i.e. equation (10)). Then linear coefficients are inferred by solving constrained optimization equation.

**Step 4:** Fill missing region. Copy sparse representation information \( \Psi_f \) to missing pixels of target patch \( \Psi_m \).

**Step 5:** Updating confidence values. After the patch \( \Psi_m \) has been filled with new pixel values, the confidence \( C(p) \) is updated in the area delimited by \( \Psi_m \), as follows:

\[
C(p) = C(m) \quad \forall p \in \Psi_m \cap \Omega
\]  

(14)

For each newly pixel on the fill-front, compute its patch priority.

**Step 6:** Repeat step 2-step 5 until all missing pixels are filled.

### 4. Experiment Results

In this section, we present simulation results of our approach on natural images, and compare the proposed approach with Xu’s method \([14]\) and Hesabi’s method \([15]\). The algorithms are programmed using Matlab language and executed on a PC with Intel 2.5GHz CPU. In our approach the size of patch is set to \( 7 \times 7 \), the size of neighborhood (i.e., \( N(p) \) around \( p \) in (11)) is set to \( 25 \times 25 \), and the number of candidate patches \( M \) is set to 25.

### 4.1 Scratch and text removal

Figure 1 presents four examples for scratch and text removal. Peak signal-to-noise ratio (PSNR) between inpainted images and original images are measured for qualitative comparison and given in Table 1. As shown in Figure 1, the Xu’s approach produces sharp inpainting results shown in the third column. However, due to the fact that the filling order is not enough robust and the neighborhood consistence constraint factor is fixed, some unpleasant visual discontinuities are introduced. For example, the structure incoherence appears within the red rectangle of Xu’s result. Hesabi’s approach produces less pleasant results because only the most similar patch is used to fill missing region. For example, the structure incoherence appears within the red rectangle of Hesabi’s result in the second row of Figure 1, and the block effect and seam effect emerge within the red rectangle of Hesabi’s result in the first row of Figure 1. For our proposed algorithm, not only the neighborhood can be maintained more consistent, but also the structure part can be kept more coherent. The reason is that in our algorithm, structure sparsity function is defined more reasonably, and the neighborhood consistence constraint factor is adaptively determined according to structure sparsity value, hence the guide information can be obtained more reasonable, which lead to more sharp and consistent repair results with the best PSNR values (presented in Table 1). The inpainting time of Xu’s and our algorithm are present in Table 2. From Table 2, we can see that our method can reduce the compute time dramatically. Therefore, our algorithm not only improve the repair results, but also enhance computational efficiency.

<table>
<thead>
<tr>
<th>Table 1: PSNR (dB) of image inpainting.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1(a)</td>
</tr>
<tr>
<td>Xu’s method([14])</td>
</tr>
<tr>
<td>Hesabi’s method([15])</td>
</tr>
<tr>
<td>Our method</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Execution time (s) of image inpainting.</th>
</tr>
</thead>
<tbody>
<tr>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Figure 1(a)</td>
</tr>
<tr>
<td>Xu’s method([14])</td>
</tr>
<tr>
<td>Our method</td>
</tr>
<tr>
<td>Improved times</td>
</tr>
</tbody>
</table>

### 4.2 Object removal

We also use the proposed algorithm to recover the missing region after object removal. In the results of Xu’s algorithm, the inpainted structures are not always consistent with the surrounding structures. For example the edge of bridge and wall cannot keep linear in the first and second example respectively. The Hesabi’s algorithm uses modified priority function based on structure sparsity to determine filling order, so the results have less effect of structure incoherence, however, there are still some flaws in the results. For example, the unwanted structure appears within the red rectangle of Hesabi’s result in the first row of Figure 2, and the structure incoherence appears within the black rectangle of Hesabi’s result in the second row of Figure 2. As for the proposed algorithm, a novel structure sparsity is adopted and neighborhood consistence constraint is adaptively adjusted, therefore the structure coherence is well maintained and the inpainted patches are more consistent with the surrounding textures. In addition, our algorithm exhaust the less inpainting time, as shown in Table 3.

<table>
<thead>
<tr>
<th>Table 3: Execution time (s) of image inpainting.</th>
</tr>
</thead>
<tbody>
<tr>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Figure 2(a)</td>
</tr>
<tr>
<td>Xu’s method([14])</td>
</tr>
<tr>
<td>Our method</td>
</tr>
<tr>
<td>Improved times</td>
</tr>
</tbody>
</table>
approach with Xu’s method [14] and Hesabi’s method [15]. The algorithms are programmed using Matlab language and executed on a PC with Intel 2.5GHz CPU. In our approach the size of patch is set to 7*7, the size of neighborhood (i.e., $N(p)$ around $p$ in (1)) is set to 25*25, and the number of candidate patches $M$ is set to 25.

4.1 Scratch and text removal

Fig. 1 presents four examples for scratch and text removal. Peak signal-to-noise ratio (PSNR) between inpainted image and original image is measured for qualitative comparison and given in Table 1. As shown in Fig. 1, the Xu’s approach produces sharp inpainting results shown in the third column. However, due to the fact that the filling order is not enough robust and the neighborhood consistence constraint factor is fixed, some unpleasant visual discontinuities are introduced. For example, the structure incoherence appears within the red rectangle of Xu’s result. Hesabi’s approach produces less pleasant results because only the most similar patch is used to fill missing region. For example, the structure incoherence appears within the red rectangle of Hesabi’s result in the second row of Fig. 1, and the block effect and seam effect emerge within the red rectangle of Hesabi’s result in the first row of Fig. 1. For our proposed algorithm, not only the neighborhood can be maintained more consistent, but also the structure part can be kept more coherent. The reason is that in our algorithm, structure sparsity is defined more reasonably, and the neighborhood consistence constraint factor is adaptively determined according to structure sparsity value, hence the guide information can be obtained more reasonable, which lead to more sharp and consistent repair results with the best PSNR values (presented in Table 1). The inpainting time of each algorithm is present in Table 2. From Table 2, we can see that our method can reduce the compute time dramatically. Therefore, our algorithm not only improve the repair results, but also enhance computational efficiency.

![Figure 1: Comparison of inpainted images obtained with Xu’s[14], Hesabi’s[15] and our proposal.](image1)

<table>
<thead>
<tr>
<th></th>
<th>Fig. 1(a)</th>
<th>Fig. 1(b)</th>
<th>Fig. 1(c)</th>
<th>Fig. 1(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xu’s method[14]</td>
<td>35.01</td>
<td>33.13</td>
<td>35.07</td>
<td>29.33</td>
</tr>
<tr>
<td>Hesabi’s method[15]</td>
<td>34.74</td>
<td>31.65</td>
<td>33.41</td>
<td>28.32</td>
</tr>
<tr>
<td>Our method</td>
<td>35.86</td>
<td>34.55</td>
<td>35.91</td>
<td>30.35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Fig. 2(a)</th>
<th>Fig. 2(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xu’s method[14]</td>
<td>1858.1</td>
<td>1182.1</td>
</tr>
<tr>
<td>Our method</td>
<td>248.4</td>
<td>170.8</td>
</tr>
</tbody>
</table>

Improved times: 7.48, 6.92

4.2 Object removal

We also use the proposed algorithm to recover the missing region after object removal. In the results of Xu’s algorithm, the inpainted structures are not always consistent with the surrounding structures. For example the edge of bridge and wall cannot keep linear in the first and second example respectively. The Hesabi’s algorithm uses modified priority function based on structure sparsity to determine filling order, so the results have less effect of structure incoherence, however, there are still some flaws in the results. For example, the unwanted structure appears within the red rectangle of Hesabi’s result in the first row of Fig. 2, and the structure incoherence appears within the black rectangle of Hesabi’s result in the second row of Fig. 2. As for the proposed algorithm, a novel structure sparsity is adopted and neighborhood consistence constraint is adaptively adjusted, therefore the structure coherence is well maintained and the inpainted patches are more consistent with the surrounding textures. In addition, our algorithm exhaust the least inpainting time, as shown in Table 3.

![Figure 2: Comparison of inpainted images obtained with Xu’s[14], Hesabi’s[15] and our proposal.](image2)

<table>
<thead>
<tr>
<th></th>
<th>Fig. 2(a)</th>
<th>Fig. 2(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xu’s method[14]</td>
<td>1858.1</td>
<td>1182.1</td>
</tr>
<tr>
<td>Our method</td>
<td>248.4</td>
<td>170.8</td>
</tr>
</tbody>
</table>

Improved times: 7.48, 6.92
5. Conclusion

This paper proposes a robust and adaptive image inpainting algorithm based on a novel structure sparsity. The major novelty of this work is that a novel structure sparsity, adaptive neighborhood consistence constraint and adaptive local search method are proposed. Experiments and comparisons have showed that the proposed exemplar-based algorithm can better infer the structures and textures of missing region, and produce sharp inpainting results consistent with the surrounding textures. In the future, we will further investigate the sparsity of natural images at multiple orientations, and apply it to image inpainting.

References