International Journal of Scientific & Engineering Research, Volume 5, Issue 8,August-2014 1243
ISSN 2229-5518

Super Resolution Image Reconstruction Using fast Inpainting Algorithm

[1]S.Priyalakshmi, [2]C.K.Sangeetha, [3]S.Vijayalakshmi, [4] T.C.Monica

Abstract--This paper introduces a novel framework for examplar-based inpainting. It consists in performing first the inpainting on a coarse version of the input image. A hierarchical super-resolution algorithm is then used to recover details on the missing areas. The advantage of this approach is that it is easier to inpaint low-resolution pictures than high-resolution ones. The gain is both in terms of computational complexity and visual quality. However, to be less sensitive to the parameter setting of the inpainting method, the low-resolution input picture is inpainted several times with different configurations. Results are efficiently combined with a loopy belief propagation and details are recovered by a single-image super-resolution algorithm. Experimental results in a context of image editing and texture synthesis demonstrate the effectiveness of the proposed method. Results are compared with existing method by the running time calculation of proposed work inpainting methods.

Index Terms-- examplar-based inpainting, single-image super-resolution**.**

1.INTRODUCTION

IMAGE inpainting refers to methods which consist in filling in missing regions (holes) in an image [1]. Existing methods can be classified into two main categories. The first category concerns diffusion-based approaches which propagate linear structures or level lines (so-called isophotes) via diffusion based on partial differential equations [1], [2] and variation methods [3]. The diffusion-based methods tend to introduce some blur when the hole to be filled in is large. The second family of approaches concerns examplar-based methods which sample and copy best matching texture patches from the known image neighborhood [4]–[7]. These methods have been inspired from texture synthesis techniques [8] and are known to work well in cases of regular or repeatable textures. The first attempt to use examplar-based techniques for object removal has been reported in [6]. The authors in [5] improve the search for similar patches by introducing an a priori rough estimate of the inpainted values using a multi-scale approach which then results in an iterative approximation of the missing regions from coarse-to-fine levels.

A recent approach [10] combines an examplar-based approach with super-resolution. It is a two-steps algorithm. First a coarse version of the input picture is inpainted. The second step consists in creating an enhanced resolution picture from the coarse inpainted image.

Although tremendous progress has been made in the past years on examplar-based inpainting, there still exists a number of problems. We believe that the most important one is related to the parameter settings such as the filling order and the patch size.

This problem is here addressed by considering multiple inpainted versions of the input image. To generate this set of inpainted pictures, different settings are used. The inpainted pictures are then combined yielding the final inpainted result. Notice that the inpainting algorithm is preferably applied on a coarse version of the input image; this is particularly interesting when the hole to be filled in is large. This provides the advantage to be less demanding in terms of computational resources and less sensitive to noise and local singularities. In this case the final full resolution inpainted image is recovered by using a super-resolution (SR) method similarly to [10]. SR methods.

The SR problem is ill-posed since multiple high-resolution images can produce the same low-resolution image. Solving the problem hence requires introducing some prior information. This prior information can also take the form of example images or of corresponding LR-HR (Low Resolution - High Resolution) pairs of patches from the input low resolution image itself [13]. This latter family of approaches is known as examplar-based SR methods [12]. An examplar-based super-resolution method embedding *K* nearest neighbours found in an external patch database has also been described in [14]. Instead of constructing the LR-HR pairs of patches from a set of un-related training images, the authors in [13] extract these correspondences by searching for matches across different scales of a multi-resolution pyramid constructed from the input low-resolution image.

In summary, the proposed method improves examplar-based inpainting methods by proposing a new framework involving a combination of multiple inpainting versions of the input picture followed by a single-image examplar-based SR method. Notice that the SR method is used only when the inpainting method is applied on a low resolution of the input picture.

ISSN 2229-5518

2. ALGORITHM OVERVIEW

The proposed method is composed of two main and sequential operations. The first one is a non-parametric patch sampling method used to fill in missing regions. The inpainting algorithm is preferably applied on a coarse version of the input picture. Indeed a low-resolution picture is mainly represented by its dominant and important structures of the scene. We believe that performing the inpainting of such a low-resolution image is much easier than performing it on the full resolution. To give more robustness, we inpaint the low-resolution picture with different settings (patch's size, filling order, etc). By com-bining these results, a final low-resolution inpainted picture is obtained. Results will show that the robustness and the visual relevance of inpainting is improved. The second operation is run on the output of the first step. Its goal is to enhance the resolution and the subjective quality of the inpainted areas. Given a low-resolution input image, which is the result of the first inpainting step, we recover its high-resolution using a single-image super-resolution approach. Fig. 1 illustrates the main concept underlying the proposed method namely:

1) a low-resolution image is first built from the original picture;

2) an inpainting algorithm is applied to fill in the holes of the low-resolution picture. Different settings are used and inpainted pictures are combined;

3) the quality of the inpainted regions is improved by using a single-image super-resolution method.

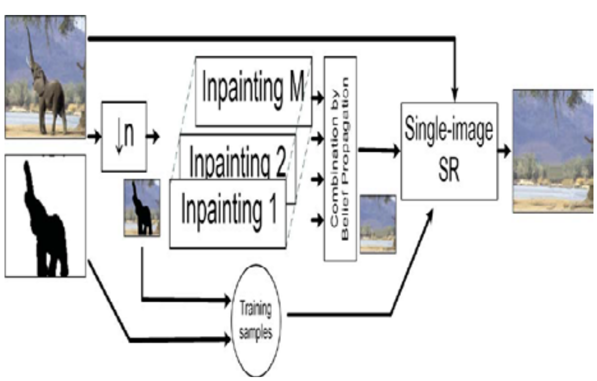

Figure. 1. The framework of the proposed method.

This new method is generic since there is no constraint on both the number and the type of inpainting methods used in the first pass. The better the inpainting of low-resolution images, the better the final result should be. Regarding the number of methods, one could imagine using different settings (patch size, search windows etc) or methods to fill-in the low-resolution images and to fuse results. We believe that it would increase the robustness and the visual relevance of in-Painting.

3. MULTIPLE EXAMPLAR-BASED INPAINTING

This section aims at presenting the proposed inpainting method and the combination of the different inpainted images.

3.1. EXAMPLAR-BASED INPAINTING

The proposed examplar-based method follows the two classical steps as described in [4]: the filling order computation and the texture synthesis. These are described in the next sections.

1) Patch Priority: The filling order computation defines a measure of priority for each patch in order to distinguish the structures from the textures. Classically, a high priority indicates the presence of structure. The priority of a patch centered on $p\mathbf{x}$ is just given by a data and confidence term. term, a tensor-based [9] and a sparsity-based [16] data terms have been used. These terms are briefly described below.

The tensor-based priority

One of the main advantages of a structure tensor is that a structure coherence indicator can be deduced from its eigen values. Based on the discrepancy of the eigen values, the degree of anisotropy of a local region can be evaluated.

The sparsity-based priority

It been proposed recently by. In a search window, a template matching is performed between the current patch $\psi p\mathbf{x}$ and neighbouring patches $\psi p\mathbf{j}$ that belong to the known part of the image. By using a non-local means approach [15], a similarity weight $wp\mathbf{x},p\mathbf{j}$ (i.e. proportional to the similarity between the two patches centered on $p\mathbf{x}$ and $p\mathbf{j}$) is computed for each pair of patches.

2) Texture Synthesis: The filling process starts with the patch having the highest priority. To fill in the unknown part of the current patch $\psi uk\ p\mathbf{x}$, the most similar patch located in a local neighborhood W centered on the current patch is sought. A similarity metric is used for this purpose. The chosen patch $\psi*\ \ p\mathbf{x}$ maximizes the similarity between the known pixel values of the current patch to be filled in $\psi k\ p\mathbf{x}$ and co-located pixel values of patches belonging to W:

$$\psi^*_{p_\mathbf{x}} = \arg\min_{\psi_{p_\mathbf{j}} \in \mathcal{W}} d\left(\psi^k_{p_\mathbf{x}}, \psi^k_{p_\mathbf{j}}\right)$$

$$\text{s.t } Coh\left(\psi^{uk}_{p_\mathbf{x}}\right) < \lambda_{coh}$$

The coherence measure Coh simply indicates the degree of similarity between the synthesized patch $\psi uk\ p\mathbf{x}$ and original patches. Therefore, the constraint in equation prevents pasting in the unknown regions a texture that would be too different from original textures. If none of the candidates fulfil the constraint , the filling process is stopped and the priority of the current patch is decreased. The process restarts by seeking the patch having the highest priority. It is interesting to note that a recent study [19] uses a similar term to predict the quality of the inpainting. Compared to our previous work [10], there is another substantial difference: we only use the best match to fill in the hole whereas a linear combination of the *K* most similar patches is generally performed to compute the patch $\psi^*_{p\mathbf{x}}$ in [10], [15], [16], [20]. In these cases, the estimated patch is then given by:

$$\psi^*_{p_\mathbf{x}} = \sum_{i}^{K} \omega_{p_\mathbf{x},p_\mathbf{i}} \times \psi^k_{p_\mathbf{i}}$$

where *K* is the number of candidates which is often adapted locally so that the similarity of chosen neighbours lies within a range $(1+\alpha)\times d_{min}$, where d_{min} is the distance between the current patch and its closest neighbours. Combining several candidates increases the algorithm robustness. However, it tends to introduce blur on fine textures as illustrated by Fig. 2. In our method, only the best candidate is chosen. Its unknown parts are pasted into the missing areas. A Poisson fusion [23] is applied to hide the seams between known and unknown parts.

Although the proposed method is able to fill in holes in a visually pleasant fashion (as illustrated by Fig. 2, it still suffers from problems of one-pass greedy algorithms. Indeed for most of existing approaches, the setting such as the patch size and the filling order, to name the most important factors, may dramatically impact the quality of results. To overcome this issue, we combine inpainted pictures obtained when different settings are used. In this study, we consider *M* = 13, meaning that the low-resolution picture is inpainted 13 times.

Parameters are given in Table I: the patch size is chosen between 5 × 5, 7 × 7, 9 × 9 and 11 × 11. The filling order is computed by either the sparsity-based or the tensor-based method. The input picture can also be rotated by 180 degrees. This allows changing the filling order.

3.2 COMBINING MULTIPLE INPAINTED IMAGES

The combination aims at producing a final inpainted picture from the *M* inpainted pictures. Before delving into this subject in details, Fig. 2 illustrates some inpainted results obtained for a given setting. We notice again that the setting plays an important role. To obtain the final inpainted picture, three kinds of combination have been considered. The first two methods are very simple since every pixel value in the final

Setting	Parameters
1 (default)	Patch's size 5 × 5 Decimation factor $n = 3$ Search window 80 × 80 Sparsity-based filling order
2	default + rotation by 180 degrees
3	default + patch's size 7 × 7
4	default + rotation by 180 degrees + patch's size 7 × 7
5	default + patch's size 11 × 11
6	default + rotation by 180 degrees + patch's size 11 × 11
7	default + patch's size 9 × 9
8	default + rotation by 180 degrees + patch's size 9 × 9
9	default + patch's size 9 × 9 + Tensor-based filling order
10	default + patch's size 7 × 7 + Tensor-based filling order
11	default + patch's size 5 × 5 + Tensor-based filling order
12	default + patch's size 11 × 11 + Tensor-based filling order
13	default + rotation by 180 degrees + patch's size 9 × 9 + Tensor-based filling order

Table 1
Thirteen configurations used to fill in the unknown parts of the pictures

(a) (b)S1 (c)S2 (d)S3 (e)S4 (f)S5 (e)S6 (f)S7 (g)S9

Figure 2: The low-resolution picture is inpainted with different settings *Sx* (see Table I for the setting details). (a) Original picture with the hole to be filled in. Pictures *S*1 up to *S*8 (from (b) to (i)) are the inpainting results (note the setting sensibility of the inpainting algorithm)

We notice again that the setting plays an important role. To obtain the final inpainted picture, three kinds of combination have been considered. The first two methods are very simple since every pixel value in the final picture is achieved by either the average or the median operator. The advantage of these operators is their simplicity. However they suffer from at least two main drawbacks. The average operator as well as the median one does not consider the neighbours of the current pixel to take a decision. Results might be more spatially coherent by considering the local neighbourhood. In addition, the average operator inevitably introduces blur as illustrated. To cope with these problems, namely blur and spatial consistency of the final result, the combination is achieved by minimizing an objective function. Rather than using a global minimization that would solve exactly the problem, we use a Loopy Belief Propagation which in practice

ISSN 2229-5518
provides a good approximation of the problem to be solved. This approach is described in the next section.

1) ***Loopy Belief Propagation****:* As, the problem is to assign a label to each pixel $p\mathbf{x}$ of the unknown regions T of the picture_ **I**(*). The major drawback of the belief propagation is that it is slow especially when the number of labels is high. Researchers have designed a priority Belief Propagation in order to deal with this complexity bottleneck. Indeed, the number of labels is equal to the number of patches in the source region. Here the approach is simpler since the number of labels is rather small; a label is simply the index of the inpainted picture from which the patch is extracted.
2) ***Comparison of the Combination Methods:*** As expected, when the different inpainted pictures are averaged, the reconstructed areas are blurred. The blur is less striking when the median operator is used to combine pictures. The LBP method provides the best result. The texture is well retrieved and thanks to the global energy minimization results are spatially consistent. In the following, we use the LBP method to combine low-resolution inpainted pictures.
3) ***Possible Extensions:*** Two extensions of the proposed method could be also envisioned. First, rather than combining the low-resolution inpainted pictures, the best inpainting picture could be selected either by a user (semi-supervised by automatically evaluating the inpainting quality using a coherence measure as proposed.

4. SUPER-RESOLUTION ALGORITHM

Once the combination of the low-resolution inpainted pictures is completed, a hierarchical single-image super resolution approach is used to reconstruct the high resolution details of the image. We stress the point that the single-image SR method is applied only when the input picture has been down sampled for the inpainting purpose. Otherwise the SR method is not required. As in [10], [12], the problem is to find a patch of higher-resolution from a database of examples.

The main steps are described below:

1) ***Dictionary building:*** it consists of the correspondences between low and high resolution image patches. The unique constraint is that the high-resolution patches have to be valid, i.e. entirely composed of known pixels. In the proposed approach, high-resolution and valid patches are evenly extracted from the known parts of the image. The size of the dictionary is a user-parameter which might influence the overall speed/quality trade-off. An array is used to store the spatial coordinates of HR patches (*DHR*). Those of LR patches are simply deduced by using the decimation factor equal to
2) ***Filling order of the HR picture:*** The computation of the filling order is similar to the one described in Section III. It is computed on the HR picture with the sparsity-based method. The filling process starts with the patch $\psi HRp\mathbf{x}$ having the highest priority and which is composed of known and unknown parts. Compared to a raster-scan filling order, it allows us to start with the structures and then to preserve them;
3) For the LR patch corresponding to the HR patch having the highest priority, its best neighbour in the inpainted images of lower resolution is sought. This search is performed in the dictionary and within a local neighbourhood. Only the best candidate is kept. From this LR candidate, a HR patch is simply deduced. Its pixel values are then copied into the unknown parts of the current HR patch $\psi HRp\mathbf{x}$. After the filling of the current patch, the priority value is propagated and the aforementioned steps are iterated while

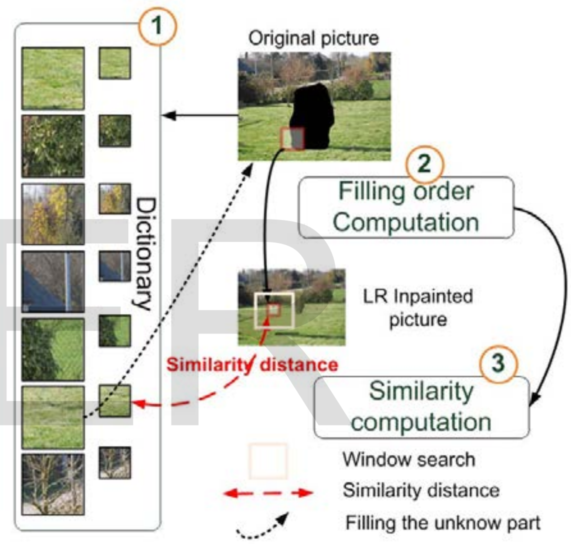

Figure:3 Flowchart of the super-resolution algorithm. The missing parts of the red block are filled in by the best candidate stemming either from the dictionary or from the local neighborhood. The top image represents the original image with the missing areas whereas the bottom one is the result of the low-resolution inpainting. there exist unknown areas.

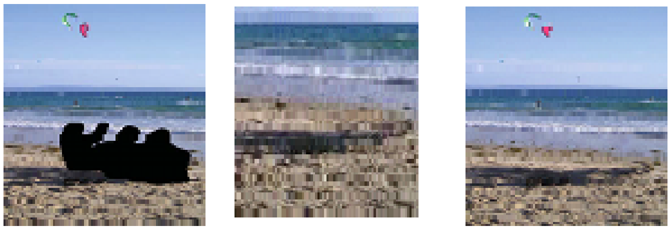

Figure. 4 Results of the proposed method. (a) Input images with unknown regions, (b) inpainted low-resolution images, and (c) final inpainted images (for the sake of visibility, we do not respect the down sampling factor of 4 between pictures (b) and (c))

A Poisson and alpha-blending are again used to hide seams between known and unknown parts and to improve robustness. Compared to [10], the SR method is applied in a hierarchical manner. For instance, if the input picture of resolution *(X, Y)* has been down-sampled by four in both directions, the SR algorithm is applied twice: a first time to recover the resolution *(X*2*, Y*2 *)* and a second time to recover the native resolution.

5. EXPERIMENTAL RESULTS

In this section, the proposed approach is tested on a variety of natural images and compared to five state-of the-art inpainting methods.

5.1 INTRINSIC PERFORMANCE OF THE PROPOSED METHOD

A down sampling factor of 4 in both directions is used. Pictures have a resolution varying in the range 420 × 380 to 720 × 512. The proposed approach faithfully recovers the texture such as the grass, the sand and the snow. Structures are also well recovered. Fig. 4 presents more results which are visually plausible and pleasing in most of the cases.

The less favorable results are obtained when the hole to be filled in is rather small. In this case it might be better to reduce the down sampling factor or even to perform the inpainting at the full resolution.

Figure:4 Results of the proposed method. (a) Input images with unknown regions, (b) inpainted low-resolution images, and (c) final inpainted images (for the sake of visibility, we do not respect the down sampling factor of 4 between pictures (b) and (c))

5.2 PARAMETERS ANALYSIS

1) ***K-NN Patches for Inpainting:*** In the number of K-NN for the examplar-based inpainting to 1 preventing blur apparition as illustrated by Fig. 2. However, as the inpainting is applied on a low-resolution picture, it might make sense to use more than one candidate. We compare results obtained by the baseline approach to those obtained for three different *K* values: 1 (baseline algorithm), 4 and 8. Results are illustrated in Fig.5. Results are similar for *K* = 4. For *K* = 8, the inpainted quality is not as visually pleasing as the one obtained by the two previous settings.

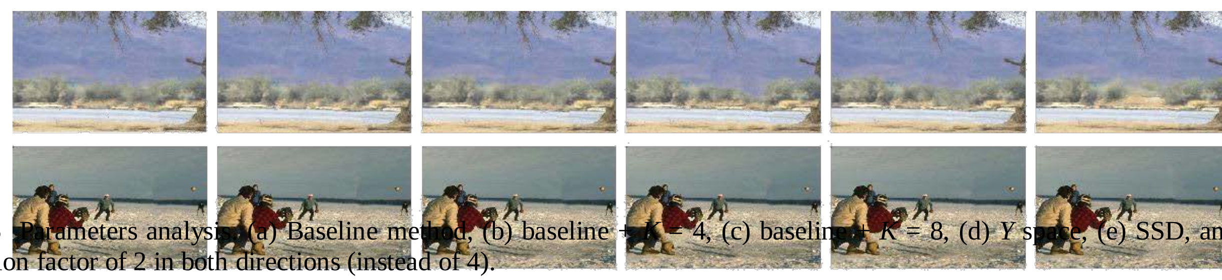

Figure:5 Parameters analysis. (a) Baseline method, (b) baseline + *K* = 4, (c) baseline + *K* = 8, (d) *Y* space, (e) SSD, and (f) with a decimation factor of 2 in both directions (instead of 4).

(a) (b) (c) (d) (e) (f)

2) ***RGB Space Versus Y Component:*** The color space, used to perform the candidate search for instance, might play an important role. By using the luminance component only, results are visually less good than the baseline one, especially for the first picture.
3) ***Downsampling Factor Equal to 2:*** By performing the inpainting at a higher resolution, results are not as good as those obtained by the baseline approach, especially on the first

5.3. RUNNING TIME

Table 2 gives the running time of the proposed approach. Simulation has been performed on a laptop with an Intel Core i7 2.40GHz and 4Go RAM. As the proposed approach is not multithreaded, it just uses one core. In addition, no optimization was made. Table II indicates that the super resolution process is the most time-consuming step (in average it represents 80% of the total time). It is due to the template matching which is not parallelized. To improve the performance, the template matching could be replaced by an approximate nearest neighbour search [7].

Table ii running time in function of the number of missing pixels. The running time is given for the low-resolution inpaintings and the super-resolution

Picture	Resolution	Missing Areas	Inpainting and LBP	SR	Total
Elephant	480 x 320	17%	32s	1m58s	2m30s
SnowBall	480 x 320	13%	22s	1m36s	1m58s
Tiger	480 x 320	28%	59s	2m36s	3m35s
Soldier	320 x 480	30%	51s	2m39s	3m30s
Cow	600 x 400	13%	40s	2m11s	2m51s

Table 2 Running time function

Picture Resolution Missing Inpainting SR Total Areas and LBP Elephant 480 × 320 17% 32s 1m58s 2m30s Snow Ball 480 × 320 13% 22s 1m36s 1m58s Tiger 480 × 320 28% 59s

2m36s 3m35s Soldier 320 × 480 30% 51s 2m39s 3m30s Cow 600 × 400 13% 40s 2m11s 2m51s (top-middle), 21 similar images were found by using *Google Image search*. To guide the LBP, we use the inpainted image from either our baseline inpainting method or the Navier-Stoke diffusion method [33]. These two guides coupled with the similar images already provide quite good results. Result of the method involving multiple inpainting is also shown and further improves the quality. Note that rather than using similar images, it would be possible to use state-of-the-art inpainted pictures and to combine them. This kind of solution might improve performance because of the better complementary of state-of-the-art inpainting methods. It would be possible to mix results from examplar-based, diffusion-based and shift-map inpainting methods. Finally, a semi-supervised approach for which the user is able to choose the best low-resolution inpainted picture could be easily designed. In this case, it is no longer necessary to combine them.

6. EXTENSION

Application to Texture Synthesis: Fig. 6 presents additional results for texture synthesis. A small chunk of texture was placed into the upper left corner of an empty image. The goal is to fill in the missing parts composed here of 256×256 pixels in a visually plausible way. This application is interesting in itself, but it also allows showing a limitation of the proposed approach. When the texture is stochastic as illustrated by Fig. 6, there exist many ways to fill in the holes in a visually plausible fashion. Performing *M* different inpaintings and combining them would lead to a poor quality. To overcome this limitation, the input texture is inpainted just once at a low resolution. The SR method is then applied. For texture synthesis, the patch size is arbitrarily set to 9 × 9 and we down-sample the input chunk by 4 in both directions. In addition, two modifications are also brought to the algorithm. First we limit the search for the closest candidate to the input picture (here it is the original texture chunk). Second the filling order is the raster scan order.

The two first texture chunks illustrated by Fig. 6 are complex to synthesize. Some artefacts are visible on the first two results whereas for the two last results the synthesised texture has a very good quality.

(a)

(b)

Figure. 6. Examples of texture synthesis. (a) Chunk of texture (256 × 256) with the unknown parts in black (256 × 256). (b) Results obtained by the proposed approach.

7. CONCLUSION

A novel inpainting approach has been presented in this paper. The input picture is first down sampled and several inpaintings are performed. The low-resolution inpainted pictures are combined by globally minimizing an energy term. Once the combination is completed, a hierarchical single image super resolution method is applied to recover details at the native resolution. Experimental results on a wide variety of images have demonstrated the effectiveness of the proposed method. One interesting avenue of future work would be to extend this approach to the temporal dimension. Also, we plan to test other SR methods to bring more robustness to the method. But the main important improvement is likely the use of geometric constraint and higher-level information such as scene semantics in order to improve the visual relevance.

8.REFERENCES

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpaint-ing,” in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Tech.*, Jul. 2000, pp. 417–424.

[2] D. Tschumperlé and R. Deriche, “Vector-valued image regularization with PDEs: A common framework for different applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 506–517, Apr. 2005.

[3] T. Chan and J. Shen, “Variational restoration of non-flat image fea-tures: Models and algorithms,” *SIAM J. Appl. Math.*, vol. 61, no. 4,
pp.1338–1361, 2001.

[4] A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by examplar-based image inpainting,” *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[5] I. Drori, D. Cohen-Or, and H. Yeshurun, “Fragment-based image com-pletion,” *ACM Trans. Graph.*, vol. 22, no. 2003, pp. 303–312, 2003.

[6] P. Harrison, “A non-hierarchical procedure for re-synthesis of complex texture,” in *Proc. Int. Conf. Central Eur. Comput. Graph., Vis. Comput. Vis.*, 2001, pp. 1–8.

[7] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “Patch-Match: A randomized correspondence algorithm for structural image editing,” *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, Aug. 2009.

[8] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proc. 7th IEEE Comput. Vis. Pattern Recognit.*, Sep. 1999, pp.1033–1038.

[9] O. Le Meur, J. Gautier, and C. Guillemot, “Examplar-based inpainting based on local geometry,” in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 3401–3404.

[10] O. Le Meur and C. Guillemot, “Super-resolution-based inpainting,” in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 554–567.

[11] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. Katsaggelos, “SoftCuts: A soft edge smoothness prior for color image super-resolution,” *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 969–981, May 2009.

[12] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar.–Apr. 2002.

[13] D. Glasner, S. Bagon, and M. Irani, “Super-resolution from a single image,” in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 10. Oct. 2009, pp.349–356.

[14] H. Chang, D.-Y. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in *Proc. IEEE Comput. Vis. Pattern Recognit.*, vol. 1. Jun.–Jul. 2004, pp. 275–282.

[15] Y. Wexler, E. Shechtman, and M. Irani, “Space-time video comple-tion,” in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun.–Jul. 2004, I-120–I-127.