

PRIOR AND MACRO-FILLING ORDER FOR IMAGE COMPLETION

R. Martínez-Noriega, and A. Roumy

INRIA, Campus Universitaire de Beaulieu, 35042 Rennes-Cedex, France.

ABSTRACT

This paper introduces an algorithm to build *priors* that help image completion to produce better and visually plausible results. The goal of the prior is to seamlessly reconstruct long edges, that exemplar-based inpainting fails to restore. The prior consists in locating the edges across the missing region, and separating the image in regions of relative similar textures. Unlike other techniques, our proposal does not use segmentation or inpainting to obtain the prior, providing a fast technique. Moreover, the technique does not rely on image dependent threshold. Finally, we propose a scheduling for the synthesis of the missing region based on the prior, and Criminisi’s algorithm. The scheduling may be thought as a novel macro-filling order for exemplar-based synthesis. The results are comparable with recent and more complex proposals based on super-resolution or Photoshop. Finally, we emphasize that, our prior is not dedicated to our image completion algorithm, and can be used to drive other image completion or inpainting algorithms.

Index Terms— prior, image completion, inpainting, filling priority

1. INTRODUCTION

Image completion is the process of restoring large removed objects from digital images with visually plausible backgrounds. In the literature, there are two main categories to image completion, diffusion-based [1] and exemplar-based [2]. The former has an excellent performance filling small regions but causes blurring artifacts when the missing region is large. The latter is known to replicate well textures, and therefore is more suitable to restore large regions.

Exemplar-based completion faces two problems, discontinuities on restored edges, and no spatial coherence within a texture region, also known as *error propagation*. In large missing regions, the discontinuities on edges is caused by the *filling order* that is computed locally, and is therefore not aware of the general image structure.

We proposed a novel metric to match the exemplar in [3], assuring spatial coherence, and achieving significantly reduction of error propagation. Nevertheless, since the filling order was locally computed, the discontinuities on edges remained. Others attempts [4, 5], based on improvements of the pioneer filling order of [2], have shown similar discontinuities.

In this paper, we propose to extract high level information from the known part of the image to deal with both problems. The idea is to build a *prior* that helps exemplar-based completion to restore the connection of long edges, and to reduce the error propagation. Indeed, since image completion is an ill-posed problem, the need of prior assumptions about the missing information have been already mentioned in the literature. For example, first proposed in [6] and

then generalize in [7, 8], the authors included the Euler’s elastica as prior to join the level lines arriving to the boundary of the missing region. In [9], the prior consists in comparing histograms of the information at the boundary with the known part to infer the structure of the edge. Unfortunately, all these priors were built to suit only diffusion-based algorithms. In [10] Lagrangian relaxation is used to guide the synthesis. First, a sketch of the image is computed, and the broken level lines are connected with Euler’s elastica. Then, the completed sketch is used as an energy term that is added to the patch metric. The performance of the algorithm is remarkable but still fails to connect some large edges.

An efficient prior for exemplar-based algorithms has been proposed in [11]. This prior consists of lines that localize and join the edges across the missing region. These lines are used to complete the edges solving an optimization problem, and then, the same lines are used to segment the image such that for each label i of the unknown part, there is at least one set with the same label in the known part. Thus, to restore the label i , only exemplars from the known part with the same label are used. This avoids bringing details of one texture into another one. However, in [11], the lines and the labels need to be specified manually. Here, we propose to determine these lines and labels automatically, in Sec. 2, and we call it, the prior. This sort of prior is generic, and can be used in other image completion algorithms like [12] where the authors mentioned the need of this sort of prior to overcome their failure cases. Similar work to ours has been reported in [13, 14]. The authors used segmentation followed by edge detection to build the prior. In our experience, tuning the segmentation to obtain such accurate priors is a very difficult task, because it is image dependent. In contrast, we do not use segmentation, providing a faster algorithm.

In order to show the benefits of our prior, we also propose an image completion algorithm in Sec. 3, based on a popular exemplar-based proposal described in [2]. Our image completion algorithm is basically a scheduling to fill-in the different regions produced by the prior. Unlike [11], we use a low-complex approach purely based on [2] to fill the edges, rather than the more complex optimization with belief propagation proposed by them. Indeed, our scheduling can be understood as a new macro filling-order for exemplar-based algorithms.

2. BUILDING PRIORS

We define the *prior* \mathcal{P} of an image \mathcal{I} as the binary image that contains the locations of salient edge points located at the boundary of the missing region $\Omega \in \mathcal{I}$ and linear edges that join these points and cross Ω . The computation of \mathcal{P} starts from an edge detector, and performs successive pruning such that only the *informative edges* remain. Then, matching and connection of those informative edges is

performed. The algorithm is summarized in Alg. 1.

Algorithm 1: Building the prior

input : Image \mathcal{I} , Missing region Ω

output: Prior \mathcal{P}

- 1 Detect **informative edges**;
 - 1a Perform *edge detection* on \mathcal{I} , and determine a set of objects $\{\beta_i\}$;
 - 1b Select *curve-like* objects;
 - Prune objects with not 2 end-points;
 - Define α_i , as the end point closest to the fill front;
 - 1c Select *almost linear* objects;
 - Prune objects with non-*similar gradients*;
 - 1d Select objects *quasi-orthogonal* to the fill front;
 - Characterize **informative edges**;
 - For each remaining β_i , compute end point α_i and direction of the edge \mathbf{v}_{α_i} ;
 - 2 **Match and connect** informative edges;
-

An informative edge for inpainting is either part of a salient structure, or the border between two different textures, that hits the boundary of the missing region also known as *fill front* $\delta\Omega$. We define an informative edge as the one that satisfies the following properties: located near to the fill front, an informative edge has a simple structure and coherent gradient, and is quasi-perpendicular to the fill front. The first property arises because we are only interested in structures located near $\delta\Omega$ (typically 20 pixel around the missing region). Since we restrict edge detection to a narrow area, then informative edges must be almost linear. More precisely, it has to be a curve and not a tree or a circle (second property) with small curvature (third property). In fact, if an edge had a more complex structure, it would likely be part of a texture. Finally, the last property is motivated by the search for edges which [2] fails to restore, i.e. edges that cross the whole missing region. In fact, edges that are tangent to the fill front can be recovered by [2].

Once the informative edges are found, we compute a characteristic slope for each one, and finally we connect only those pairs of informative edges with similar directions. Then, not all the informative edges will be connected. A non-connected informative edge means that the edge does not cross completely the missing region, and there is not need of connection because the synthesis itself will propagate it. For example, in Fig. 1-Pumpkin-(e) the broken frame in the upper part of the missing region is not connected. However in the final result, column (d), the structure is well preserved.

2.1. Informative Edges

Informative edges are obtained by discarding the information from the edge detector that does not satisfy the statements mentioned before.

Edge detection

Consider Θ as the area of 20 pixels width that surrounds the missing region Ω . Edge detection is applied on Θ . The result is a binary image \mathcal{E} . Pixels of the same edge are usually *connected* between then, forming sort of cluster or sequences. A pixel i is connected to pixel j if j is one of the eight-neighbors of i . Thus, \mathcal{E} can be seen as

a collection of independent sequences that we call *objects* β . Each object β_i represents an edge, and will be further analyzed, to decide if it is an informative edge for the inpainting problem. We use a Canny edge detector with low-high threshold (.04, .1) respectively, and Gaussian filter of $\sigma = \sqrt{2}$.

Curve-like structure

An object β_i has a curve structure if it has only two *end-points*, where an end-point is a pixel in β_i with a single neighbor. Let $\alpha_i = \{\alpha_{i,1}, \dots, \alpha_{i,j}\}$ be the set of edge points associated to the object β_i . More than two endings means structures similar to a tree. No ending means closed shape structure. Therefore, we prune all the objects with a number of end points endings different of two, i.e. $\text{Card}(\alpha_i) \neq 2$. Finally, for each object β_i kept, we define the edge point α_i , as the closest edge point to the fill front.

Almost-linear structure

To test whether the curvature of the edge is small, the gradients $\{\nabla I_{p_j}\}_j$ on the edge i are compared. More precisely, the gradient of neighboring pixels must be almost to parallel. Thus, we prune all β_i with $n^{-1} \sum_j^n |\cos(\nabla I_{p_j}, \nabla I_{p_{j+1}})| < .8$.

Quasi-orthogonal

An edge β_i is perpendicular to the fill front, if its direction vector \mathbf{v}_{α_i} , at the edge point α_i close to the fill front, is parallel to the normal of the fill front \mathbf{n}_{α_i} . Therefore, we prune all the quasi-parallel objects β_i such that $|\mathbf{v}_{\alpha_i} \cdot \mathbf{n}_{\alpha_i}| < .6$ to the fill front.

Depending on the curvature of the edge, we propose two ways to compute the direction \mathbf{v}_{α_i} . If β_i is nearly linear, \mathbf{v}_{α_i} is given by the unit vector of the line that crosses the points $\alpha_{i,1}$ and $\alpha_{i,2}$. Otherwise, the direction vector is defined as $\mathbf{v}_{\alpha_i} = \nabla I_{\alpha_i}^\perp / \|\nabla I_{\alpha_i}\|$ where $\nabla I_{\alpha_i}^\perp$ is the 90°-rotated gradient at point α_i . The reason for these two computations is that the computation of \mathbf{v} is more robust using the end-points α_i than the gradient direction.

A basic and fast measure of linearity is the eccentricity $c_i \in [0, 1]$ of an object β_i . Eccentricity measures the elongation of a disc, and since lines are degenerate discs, this measure can be directly applied to measuring linearity. c_i is zero when the shape of β_i is a disc, and one when it is a line. Formally $c_i = (\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}) / (\mu_{20} + \mu_{02})$, where μ_{pq} are the central moments of the object β_i defined by $\mu_{pq} = A^{-1} \sum_{\rho \in \beta_i} (\rho^x - \rho_c^x)^p \cdot (\rho^y - \rho_c^y)^q$, in which $A = \mu_{00}$ is the area or the number of pixels of β_i , $\rho = [\rho^x, \rho^y]$ are the elements of β_i , and ρ_c is the centroid of β_i .

Notice that not all informative edges are precisely arriving to the fill front. Some of them are separated from the fill front by few pixels because the edge detection is not optimal. We search for them in an area of 5 pixels-width surrounding Ω as previously described. Notice that the normal \mathbf{n} only exist at the fill front. Thus, Ω must be *dilated* to obtain the normal at each point within the area of 5 pixels-width. Dilation is a morphological operation that expands a given shape, in our case the missing region Ω . After all these steps, some images may have small objects β_i with less than 5 pixels. Those objects are pruned as well.

After the pruning, each object β_i that remains is an informative edge. It is characterized by an edge point α_i and a direction vector \mathbf{v}_{α_i} .

2.2. Match and connect informative edges

From the informative edges, we need to find pair of points (α_i, α_j) that belong to the same edge, and draw a line to connect them. Let us define l_i as the line with direction \mathbf{v}_{α_i} that passes through the

point α_i . A point α_j is likely to belong to the edge l_i if the distance d_{ij} between α_j and the line l_i is small. The distance between the point α_j and the line l_i is the distance between the point α_j and the orthogonal projection of α_j on l_i .

However, further constraints must be added to build informative edges. Neighboring edge points should not be connected, since they would either belong to a texture detail, or be a structure close to the boundary, that can be well recovered by classical inpainting methods. Therefore, we impose a minimum separation distance of 15 pixels between connected points. Moreover, only points with similar slopes should be connected, as different slopes would correspond to different edges. Therefore, we impose that the cosine of the angle between the slopes of two connected points is bigger than 0.6. Finally, the algorithm that matches aligned edge points, under the above mentioned constraints, is described Alg. 2.

Algorithm 2: Matching and connecting edge points

input : Set of objects: characterized by edge points $\{\alpha_i\}$, direction \mathbf{v}_{α_i} , and line l_i .

output: Set of connected edge points $\{(\alpha_i, \alpha_j)\}$,
Set of non-connected edge points $\{\alpha_k\}$

- 1 Compute the **constrained distance matrix**;
 - 1a $\forall(i, j), d_{ij} = d(\alpha_j, l_i)$;
 - 1b $\forall i, d_{ii} = +\infty$;
 - 1c $\forall(i, j) \text{ s.t. } |d(\alpha_i, \alpha_j)| < 15, d_{ij} = +\infty$;
 - 1d $\forall(i, j) \text{ s.t. } |\cos(\mathbf{v}_{\alpha_i}, \mathbf{v}_{\alpha_j})| < 0.6, d_{ij} = +\infty$;
 - 2 **Match** edge points;
 - while** d_{ij} contains non- ∞ values;
 - $(i, j) = \arg \min_{k,l} d_{kl}$;
 - add (α_i, α_j) to the connected set;
 - erase rows i, j , columns i, j from $\{d_{ij}\}$;
 - end while**
 - add the remaining α_k 's to the non-connected set;
-

3. IMAGE COMPLETION SCHEDULING

In this section we propose an image completion algorithm that exploits the prior described in Sec. 2. Given an image \mathcal{I} with missing region Ω and known region $\Phi = \mathcal{I} - \Omega$, our algorithm schedules the order in which Ω should be filled-in with a basic exemplar-based synthesis proposed in [2]. This scheduling can be seen as a novel macro-filling order.

The algorithm in [2] chooses the most suitable place $p \in \delta\Omega$ to be filled-in, where $\delta\Omega$ is the fill front. Let Ψ_p be the patch centered at p . Then, the most similar exemplar $\Psi_q \in \Phi$ to Ψ_p , in terms of Euclidean distance, is found. Finally, the missing information in Ψ_p is copied from Ψ_q . This process is repeated until there is no missing information in the image. The most suitable place p is the maximum priority $P(p)$ defined as

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

where $C(p)$ is called the confidence term and measures the reliability of the information around p . $D(p)$ is the data term, and measures the “strength” of the edge that hits $\delta\Omega$, [2].

Since (1) is computed locally, it fails reconstructing non local geometric features. We now detail how to include our prior \mathcal{P} to better reconstruct long edges. For each pair of connected points, we draw a line through the two connected points. These lines cross the whole image and define a partition of the image into regions labeled \mathbf{R}_i . Each \mathbf{R}_i is composed of three parts: a local missing part $\omega_i = \mathbf{R}_i \cap \Omega$, a local known part $\phi_i = \mathbf{R}_i \cap \Phi$, and a boundary B_i given by the pixels that divides the region \mathbf{R}_i from the others. The algorithm will restore each region individually and sequentially, starting by the most outer region \mathbf{R}_i , that contains the more non-connected edge points. This allows to better reconstruct the edges, that starts at the fill front but ends in the middle of the missing region Ω . Finally, for each region \mathbf{R}_i , the algorithm will first synthesized ω_i , and then the boundary $B_i \cap \Omega$.

3.1. Synthesis of ω_i

The algorithm in [2] is used to complete ω_i with two new modifications. First, we allow the synthesis to go beyond the boundary B_i by few pixels. However, when ω_i is fully completed, we remove the information, $\mathbf{R}_i \cap \Omega$, beyond the boundary. This allows a better synthesis of ω_i near the boundary B_i , because the patches Ψ_p cross the boundary. Second, the search of exemplars Ψ_q is constraint to ϕ_i .

3.2. Synthesis of the boundary $B_i \cap \Omega$

[2] is used to complete the edge $B_i \cap \Omega$ but with three modifications. First, we trigger the completion at one end of the line $B_i \cap \Omega$, e.g. the priority (1) is computed at the two extreme points of the line only. Moreover, we dilate $B_i \cap \Omega$ by few pixels in order to have a thin section to be filled in, rather than only a line of one pixel width. This is because the edges in natural images are not sharp, and a single 1-pixel-width line defined by $B_i \cap \Omega$ can not characterize them. Third, the search of exemplars Ψ_q is on the whole known part Φ .

4. RESULTS & DISCUSSION

We compare ours results against other algorithms in Fig. 1. (a) is Criminisi’s exemplar-based algorithm [2]. (b) was obtained with “content aware fill” from Photoshop CS5.1. (c) is a complex algorithm that combines super-resolution and image completion [15]. (d) shows our results and (e) illustrates our prior with red lines and blue squares.

Edge completion is better achieved by our algorithm. Indeed, it is the only one that successfully complete the edges on the four images. Error propagation is highly reduced by our algorithm in comparison to [2] in column (a) because the exemplar search is limited by the prior to regions with similar textures.

In the “Pumpkin” image, our prior does detect the broken frame at the upper part of the missing region. This edge is not connected because there is not a similar edge at the bottom. However, thanks to our macro-filling order, that takes into account the existence of non-connected edge points, the upper part of the missing region is first completed. Therefore, it is well reconstructed. Limitations in computing the prior are for images with very similar textures. For example, in “Bungee jumping”, it is difficult to detect the boundary between the trees and the sea at the bottom part. Limitations on our image completion algorithm are given by the nature of the synthesis algorithm [2]. That is, the more suitable exemplar is not

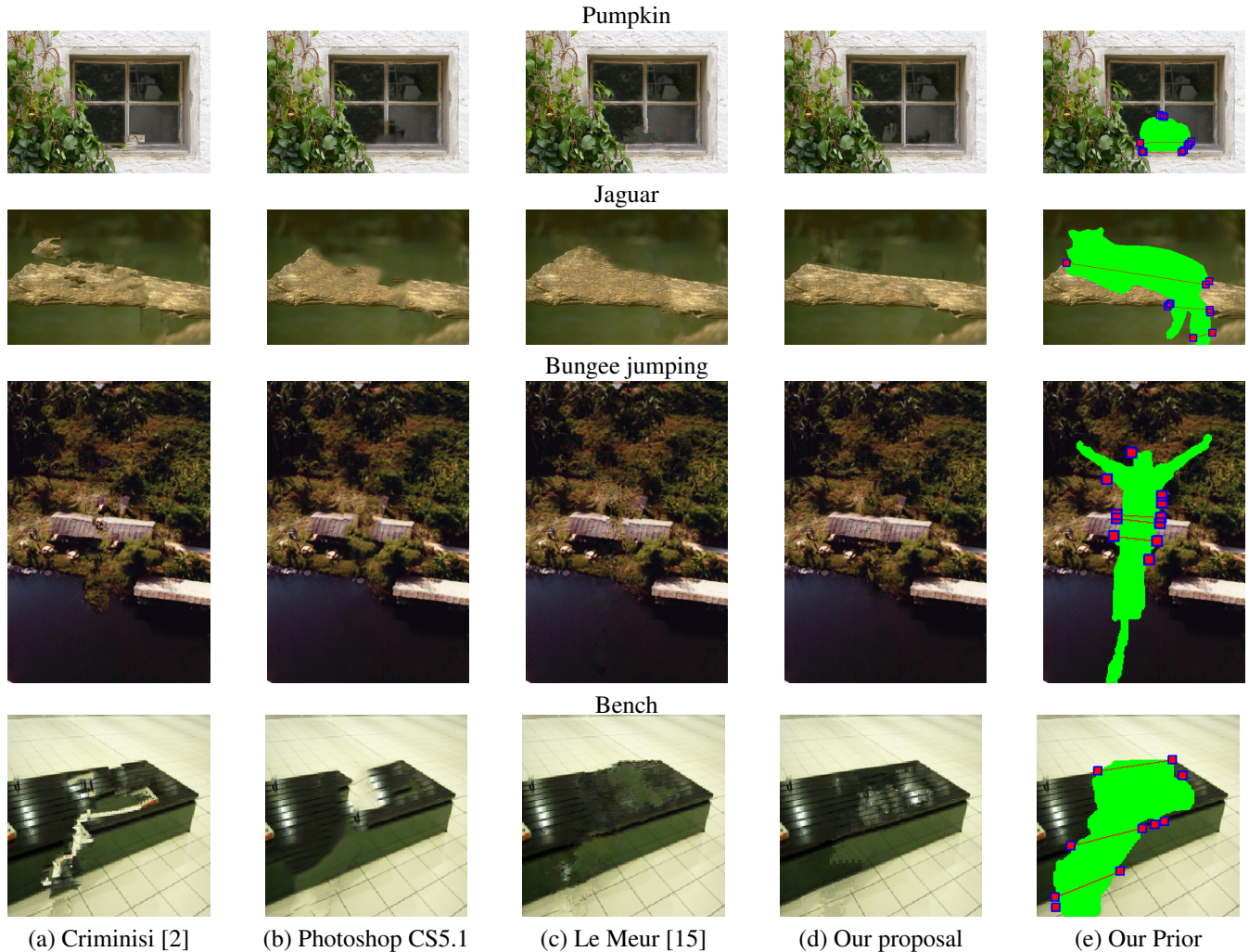


Fig. 1. Comparison of restored images obtained with popular algorithms from literature (a)-(c) and our proposal (d) described on Sec. 3. In column, (e) we show the missing region in green color, the red lines and blue squares represent our prior obtained with the algorithm described in Sec. 2.

always the one with minimum Euclidean distance, and inability to create smooth transition between different colors: green part of the “Bench” in Fig. 1.

5. CONCLUSION

Exemplar-based synthesis can fail restoring long edges because it relies on a filling order, that is computed locally. To overcome this issue, there is a need for a prior providing a global idea of the geometry of the missing region. Previous prior computations were based on either segmentation or inpainting of level lines. Instead, we proposed to detect salient points located near the missing region and eventually match them to build the prior. Our algorithm relies on an efficient pruning of the output of an edge detector and therefore leads to a very fast algorithm. Our prior can be seen as a macro-filling order that turns a simple exemplar-based algorithm into a powerful image completion method that produces better or equal results than current and complex techniques. Currently, we investigate optimization techniques to connect the informative edges when there are T-

junctions. Also, as future work we would like to generalize the prior to non-linear edges and to explore its potential with more powerful synthesis techniques.

6. REFERENCES

- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *SIGGRAPH*, 2000.
- [2] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sept. 2004.
- [3] R. Martinez-Noriega, A. Roumy, and G. Blanchard, "Exemplar-based image inpainting: Fast priority and coherent nearest neighbor search," in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, sept. 2012, pp. 1–6.
- [4] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1153–1165, May 2010.
- [5] M. Turkan, *Novel texture synthesis methods and their application to image prediction and image inpainting*, Ph.D. thesis, Univ. Rennes 1, Dec. 2011.
- [6] S. Masnou and J.-M. Morel, "Level lines based disocclusion," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, oct 1998, pp. 259–263 vol.3.
- [7] S. Masnou, "Disocclusion: a variational approach using level lines," *Image Processing, IEEE Transactions on*, vol. 11, no. 2, pp. 68–76, feb 2002.
- [8] Tony F. Chan, Sung Ha Kang, and Jianhong Shen, "Euler's elastica and curvature-based inpainting," *SIAM Journal on Applied Mathematics*, vol. 63, no. 2, pp. pp. 564–592, 2002.
- [9] Anat Levin, Assaf Zomet, and Yair Weiss, "Learning how to inpaint from global image statistics," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, Washington, DC, USA, 2003, ICCV '03, pp. 305–, IEEE Computer Society.
- [10] F. Cao, Y. Gousseau, S. Masnou, and P. Prez, "Geometrically guided exemplar-based inpainting," *SIAM Journal on Imaging Sciences*, vol. 4, no. 4, pp. 1143–1179, 2011.
- [11] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum, "Image completion with structure propagation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 861–868, July 2005.
- [12] Kaiming He and Jian Sun, "Statistics of patch offsets for image completion," in *Proceedings of the 12th European conference on Computer Vision - Volume Part II*, Berlin, Heidelberg, 2012, ECCV'12, pp. 16–29, Springer-Verlag.
- [13] Jason C. Hung, Chun-Hong Hwang, Yi-Chun Liao, Nick C. Tang, and Ta-Jen Chen, "Exemplar-based image inpainting base on structure construction," *Journal of Software*, vol. 3, no. 8, 2008.
- [14] Jino Lee, Dong-Kyu Lee, and Rae-Hong Park, "Robust exemplar-based inpainting algorithm using region segmentation," *Consumer Electronics, IEEE Transactions on*, vol. 58, no. 2, pp. 553–561, may 2012.
- [15] Olivier Le Meur and Christine Guillemot, "Super-resolution-based inpainting," in *ECCV*, Florence, Italy, Oct. 2012.