A perceptual image completion approach based on a hierarchical optimization scheme

T.T. Dang[1], A. Beghdadi[1], M-C. Larabi[2]

[1] *L2TI, Institut Galilée, Université Paris 13, France*
[2] *XLIM, Dept. SIC, Université de Poitiers, France*

Abstract

This paper aims to introduce a novel efficient approach for high-quality and fast image restoration by combining a greedy strategy and a global optimization strategy based on a pyramidal representation of the image. A coarse version of the input image is first restored by exemplar-based method using a greedy strategy. From the low-resolution inpainted image, higher resolutions are interpolated and refined by a global optimization strategy. Experimental results on natural images demonstrate the effectiveness of the proposed method. Moreover, a comparison with some methods of the state-of-the-art confirms the superiority of the proposed method in terms of image quality and computational time.

Keywords: image inpainting, image completion, hierarchical representation, greedy algorithm, graph cuts.

1. Introduction

Image inpainting, also known as blind image completion, refers to the action of filling missing parts or objects in an image. During the last decade, it becomes a very important research topic in the fields of computer vision and image processing because of its suitability for plenty of professional and consumers applications such as: i) application of digital effects (*e.g.* removing undesired objects or logos), ii) image restoration (*e.g.* deleting scratches or blotches in old photographs), iii) image coding and transmission (*e.g.* recovering missing blocks, error concealment), etc.

Since, the targeted completion is performed blindly and without any cue about what would be the original content, the focus is put on restoring the damaged parts/objects by maintaining as much as possible the naturalness of the image. Moreover, the restored parts should not be visible or perceptually annoying to human viewers and the used algorithm needs to be robust, efficient and, requiring minimal user interactions or quick feedbacks.

To date, several approaches of image in painting have been proposed in literature [1]. The most fundamental inpainting approach is the diffusion based in which the image is modeled as a function of smoothness. The completion is performed by interpolating the image information from the known region into the missing region at the pixel level. In their pioneering work, *Bertalmio et al.* [2] proposed an algorithm for object removal that inwardly propagates information from the boundaries of the selected object, in a smoothly manner. This approach reproducing real techniques performed by professional restorators has been constructed using a third order Partial Differential Equations (PDE) and improved using fluid dynamics knowledge [3]. In their work, *Bornemann and Marz* [4] proposed an improvement of the Bertalmio's approach by modifying the weight function and replacing the edge-oriented transport direction method by the coherence direction. In the same vein, *Chan and Shen* proposed an inpainting model relying on a variational framework based on the total variation to recover the missing information [5]. They also introduced a new PDE-based inpainting approach exploiting curvature driven diffusion. The literature on inpainting algorithms involving variational or PDE approaches is relatively rich and various. Nevertheless, it is commonly agreed that the one proposed by *Tschumperlé et al.*, based on an efficient second-order anisotropic diffusion model for multi-valued image regularization [6], is one of the most efficient when parameters are carefully selected. Finally, even though the methods falling in this approach are very efficient for untextured and relatively small region, they show some important drawbacks due to their incapacity to restore texture and tend to introduce blurring effect when the missing region is large.

The second category of approaches is the exemplar-based algorithms, in which texture is modeled through probability distribution of the pixel brightness values. This approach is inspired from texture synthesis techniques proposed by *Efros et al.* [7] and improved by *Ashikhmin* [8] with the aim of reducing the computational cost of patches matching based on the notion of coherence or Kwatra et al[9, 10] for patch synthesis based on graphcut. However, natural images are composed of structures and textures, in which the structures refer to edges or contours and the textures are image regions with homogeneous patterns or feature statistics (including flat patterns). This is why pure texture synthesis techniques cannot be efficiently applied to missing regions/objects with composite textures and structures. Authors in [11] decomposed the image into structure and texture components. Then, the restoration is performed simultaneously and independently on each component by means of geometry and texture oriented methods, respectively. The developed approach rely on a number of parameters such as the contour preservation, the structure anisotropy and the number of iterations. Indeed this approach allows to avoid blurring effect observed with the diffusion approaches. But it is still hard to recover missing structures of a large size.

In such a process, neighboring known pixels around the missing region or the object to be removed is an important source of the most relevant information regarding the target region. So, the natural way suggests to start filling the target region inwardly in an onion-peel fashion. Instead, a patch propagation based on patch priority is proposed in [12] to encourage the filling-in of patches on the structures. Several improvements have been introduced for patch priority such as cross-isophotes patch priority in [13], color distribution priority in [14], patch sparsity in [15] as well as for patch synthesis such as non-local means [15, 16]. Generally, these approaches are known as greedy strategies and they have acceptable computation time in comparison to diffusion approaches. Moreover, it tries to take into account human perception features. For instance, priority is designed based on the salient structures considered as important for human perception. However, these approaches show some common problems related to local optimization, patch priority, patch selection and so on.

Besides the aforementioned technique, inpainting can be considered as a global optimization problem that can be solved by minimizing a coherence measure [17] or energy functions of smoothness [18, 19] or bidirectional similarity [20, 21]. Global optimization strategies often provide better results in comparison to other strategies but at the cost of a higher computational complexity. The latter is mainly due to the fact that time complexity increases linearly with both the number of source pixels and unknown pixels.

Recently, more general sparse image representations using dictionaries have proven their efficiency in the context of inpainting [22, 23, 24]. The idea of this approach is to represent an image by a sparse combination of an over complete set of transforms. Then, missing pixels are inferred by adaptively updating this sparse representation. However, similar to the diffusion-base approach, this one may fail in recovering structures or may introduce a smoothing effect when filling large missing regions.

Although tremendous progress has been achieved on image inpainting over recent years, there are still significant challenges. For example, image completion of large missing regions is one of major open problems. The computaional complexity and time running is also one of the main issues to consider in image inpainting. In this study, we propose a novel approach for high-quality and fast image completion by combining and leveraging the benefits of both greedy and global optimization strategies using a pyramidal representation of the image.

The proposal is directed by the observation that the human visual system is more sensitive to salient structures being stable and persistent at different scales. Therefore, a hierarchical image inpainting scheme is developed in order to control and preserve salient features during the completion process. This scheme allows to restore the missing regions in a visually plausible way. A top-down completion is implemented from the top level (the lowest resolution) to the bottom level (the original resolution). A greedy algorithm, based on the idea introduced in [25], is developed for the lowest resolution to restore the damaged region. It provides a good initialization accounting for the human perception at the higher resolution. It is worth noticing that, since the low-resolution inpainted image has a critical impact on the quality at the final output, caution should be taken when using the algorithm proposed in [12]. Therefore, the inpainting algorithm in [12] is first improved by evaluating the filling-in priority with all pixels in the patch. This makes it different from simply using the gradient-based or cross-isophotes-based priority as proposed in [12, 13] while taking advantage of a lower complexity than those proposed in [14, 15]. The impact of patch synthesis terms on the quality of the inpainted images is also studied. The similarity measurement based only on color channels is insufficient to propagate accurate linear structures into the target region and leads thus to garbage growing. This comes from the observation that the HVS is sensitive to not only the intensity of a spectral color but also to the context in which it is observed. To maintain this variation, a new term representing image gradient is introduced as a weighting parameter in the computation of the similarity measure. In addition, patch selection based on standard deviation of variances of neighboring source patches is introduced. For higher resolution, an offset map defining the relationship between pixels to be filled and pixels in the known region is applied instead of using directly pyramidal images. First, an offset map is extracted from inpainted image at lowest resolution. Then, it is interpolated for adjacent higher resolution as an initial guess. The offset map is refined and optimized by a global optimization algorithm, i.e. multi-label graph-cuts [26, 27] with operators defining both data and smoothness constraints. Finally, the inpainted image is derived based on the offset map for highest resolution, the original resolution. Experimental results highlight a noticeable improvement in both implementation performance and visual quality of the inpainted images.

The remainder of this paper is organized as follows. Section 2 is dedicated to the description of th proposed inpainting framework describing the adopted strategy and the different tuning made to improve visual quality and performance. The experimental results are given and discussed deeply in section 3 by giving objective and visual comparison with the very representative state-of-the-art. Finally, this paper ends with some conclusions and gives future directions.

2. The proposed method

Image completion in the case of images with large missing regions is a very challenging task. As mentioned in the introduction, several solutions have been developed to tackle this problem. In this section, we introduce a novel framework based on hierarchical representation in order to propose a solution for the described problem. Therefore, our idea will be described and our proposal justified with regards to literature.

2.1. Algorithm overview

The proposed approach is composed of two main and successive operations. The first one is a greedy strategy, exemplar-based method, used to fill in missing regions. This approach is applied on a coarse version of the input image rather than the original image itself. The proposal is inspired from the observation telling that the human visual system is sensitive to salient structures being stable and persistent through different scales. In other words, the most relevant structural information of an image remains visible and attractive at different levels of resolution. Therefore, this observation is exploited when performing image inpainting. Indeed, at low resolution the inpainting would be less sensitive to local singularities and noise effect. Furthermore, it is much less computationally demanding than when processing the original full resolution image and it cope with the problem of large regions inpainting.

The second operation consists of restoring the damaged region on the original high resolution image by exploiting spatial information contained in the lowest resolution image provided at the first step. At higher resolution, the inpainting problem is modeled as an optimal graph labeling where an offset-map represents the selected label for each unknown pixel. The offset map could be determined and refined by optimizing an energy function using multi-label graph cuts [26, 27]. Because an unknown pixel in the damaged region could originate from any pixel in the source region, the global optimization strategies can be computationally unfeasible. On the one hand, for inpainting quality, fair assignments may lead to unexpected bias for optimization since they consider fairly possible label assignments but this does not fit with human perception. On the other hand, for speed, a huge label set requires high computational load.

In this paper a new strategy is proposed to overcome these limitations. A hierarchical approach is developed in order to reduce the memory and computational load and improve the image inpainting quality. A hierarchical strategy could provide enough-good results for the inpainting problem, even though optimality cannot be guaranteed.

For the sake of clarity and comparability, some notations that are similar to those in [25] are adopted. The whole image domain, I, is composed of two disjoint regions: the inpainting region (or target region) Ω, and the source region Φ ($\Phi = I - \Omega$). The proposed algorithm is summarized through the pseudo-code given below (Table 1).

At the beginning of the proposed algorithm, a gaussian pyramid is constructed from the image to be inpainted. This step consists of low-pass filtering and downsampling images of the preceding level of the pyramid. According to the above idea, a set of images $G_0, G_1, ..., G_N$ with various levels of details is generated with $G_0 = I$ is the input or original image. The number of pyramid levels is depending on two criteria *i.e.* the original size of the image and the smallest allowed resolution. The former depends on the user's image but for the latter $min(width(G_N), height(G_N)) \geq 32$ to avoid missing important details.

2.2. Inpainting lowest resolution image (G_N)

In order to integrate some low-level features of the HVS and simulate its hierarchical perceptual properties, a greedy strategy is applied for the lowest resolution G_N. In the proposed scheme, an extension of [25] is developed to complete the coarse resolution. Fig. 1 illustrates the different steps of the exemplar-based inpainting strategy. Generally, an exemplar-based method

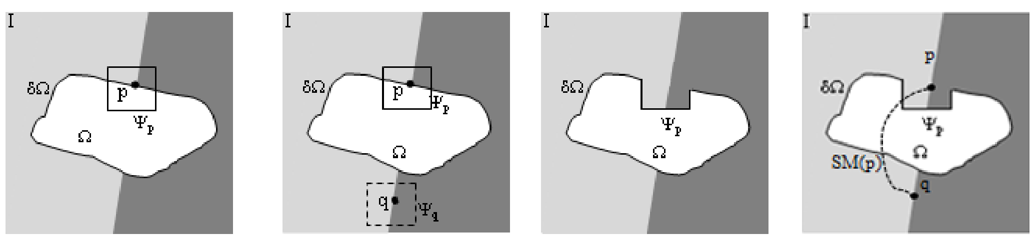
Table 1: Our algorithm

Input : I - input image;
N - pyramid levels;

Algorithm :

$G_0 = I$;
$\{G_1, G_2, ..., G_N\} = buildPyramid(G_0)$;
Complete G_N with the scheme in [12] using window-based priority;
$SM_N = generateShiftMap(G_N)$;
for $i = N - 1$ ***downto*** 0 ***do***
  $SM_i = interpolate(SM_{i+1})$;
  $SM_i = optimize(SM_i, G_i)$;
end for
$G_0 = restore(SM_0)$;
Output $= G_0$;

consists of two main steps: i) determination of the filling order and ii) selection of the best matching patch. These steps are analyzed in details in the next sections.



(a) Priority estimation (b) Patch match (c) Inpainting (d) Offset definition

Figure 1: Illustration of the examplar-based inpainting.

2.2.1. Window-based priority

For the greedy strategy, an appropriate definition of the priority is essential since the decision taken with it is irreversible at the next stages. Otherwise, error may accumulate continuously because no improvement is applied on the previous stages. Many formulations have been built for priority [12, 13, 14, 28]. In this work, we used the window-based priority proposed in [25] which is considered as more robust and efficient than the others.

Classically, a priority is composed of two terms: i) confidence term and ii) data term. Here, we concentrate on the analysis of the data term which distinguishes the structures from the textures (or flat regions). The confidence term is not mentioned here since it does not provide any additional improvement. However, a data term with a high value indicates the presence of a structure. In [12], authors introduced a pixel-based data term, $D(p)$, depending proportionally on the isophote direction or gradient of the known region. Thus, if the gradient at pixel p is large, the priority will be high. In other words, when the gradient values of the texture component are greater than those of the structure component or when the regions are affected by noise, the isophote-driven priority method may violate the requirement of an appropriate priority rule and yields to bad results. To overcome this difficulty, a patch-based data term is introduced as done in [14, 15]. This solves to some extent the problem but at the cost of an increased computational time.

Reasonably, the data term should be estimated from all neighborhood pixels in an exemplar centered on the current pixel to account for the neighboring influence and spatial coherence. Hence, we propose a first improvement by introducing a better definition of the data term. It is based on the local changes of pixel intensities in each window, W_p, centered at pixel p with shifted windows in different directions. The local change of intensity at each pixel $p(x, y)$ is characterized by the following second-moment matrix or structure tensor [25]:

$$M(p) = \sum_{W_p} G_{W_p}(x, y) \begin{pmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{pmatrix} \tag{1}$$

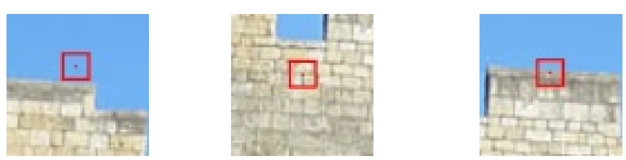
where G_{W_p} is a windowed weighting Gaussian function. This structure tensor is a 2×2 symmetric and semi-positive matrix which captures the intensity structure of the local neighborhood. The 2D structure tensor and its eigenvalues $\lambda_1 \geq \lambda_2 \geq 0$ summarize the distribution of the gradient within the defined window. The two corresponding eigenvectors represent two orthogonal directions directed along the local maximum and minimum variation of image intensities. Whereas, the eigenvalues measure the effective variations (strength of contours) of image intensities along these vectors. Therefore, our data term is defined as the follows:

$$D(p) = \frac{\lambda_1}{\lambda_2 + \epsilon} \tag{2}$$

where ϵ is a very small positive value introduced to ensure computation stability (in our experiment, $\epsilon = 10^{-10}$). This data term is related not only to the geometry features such as contours or edges but also texture features. There are three cases to be considered for each window as those defined by *Beaudet* [29]:

- If the data term is much greater than one ($D(p) \gg 1 \Leftrightarrow \lambda_1 \gg \lambda_2$), the local shifts in one direction cause little changes and significant change in the orthogonal direction; then the window is considered as a region with strong edges;
- If the data term is close to one ($D(p) \approx 1 \Rightarrow \lambda_1 \approx \lambda_2$), there are two possible cases:
  - If both eigenvalues are high, the shift in any direction will result in a significant change, this indicates a texture or complex structures such as corners.
  - If both eigenvalues are small, the shift in any direction will cause a little change, the patch is of approximately constant intensity (flat region).

With this priority, the patches would be classified in a more robust way and the computation time will be acceptable. Table 2 gives some numerical values of the proposed priority in comparison to those described in [12, 13] for patches illustrated on Figure 2. Since our approach is based on contour preservation, the priority of associated pixels should be higher than other pixels; which is demonstrated Table 2.



(a) Flat region (b) Texture region (c)Edge or contour

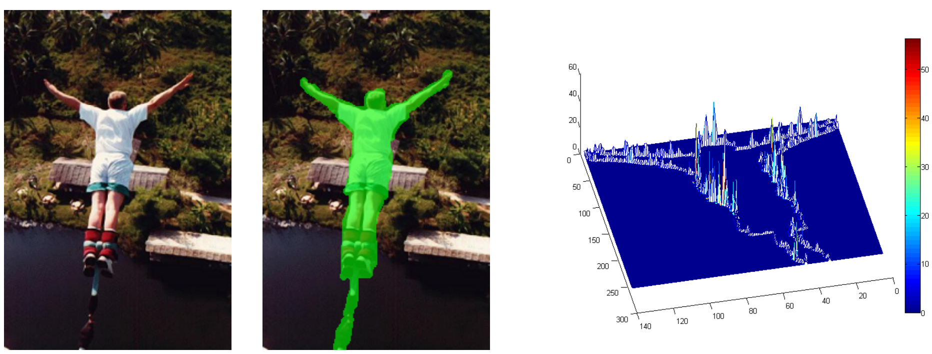
Figure 2: Window-based priority of different regions.

Figure 3 illustrates the priority of pixels on the boundary of inpainting regions where the red color refers to higher value and the blue color refers to lower value. One can notice that higher values can be observed on contours

Table 2: Priority associated with different patches illustrated on Figure 2.

Patch	Priority in [12]	Priority in [13]	Our Priority (λ_1, λ_2)
Flat region (Fig. 2a)	0.001	3.269	0 (*0*, *0*)
Texture region (Fig. 2b)	**0.032**	**3.916**	1.374 (*0.067*, *0.049*)
Edge or contour (Fig. 2c)	0.024	3.130	**4.059** (*0.025*, *0.006*)

as explained previously. Figure 4 illustrates the performance of the proposed method on various low-resolution images. These images are downsampled versions of the original ones (the down sampling factor is set to 4 in both directions). Four inpainting methods, namely *Criminisi et al.* [12], *Wu et al.* [13], *Zhang et al.* [14] and *Cheng et al.* [28], belonging to the second group of approaches, are implemented with the same size of patch to avoid bias. Visual inspection of the results of Figure 4 shows that the proposed approach achieves reasonable results in most cases in comparison with the others.

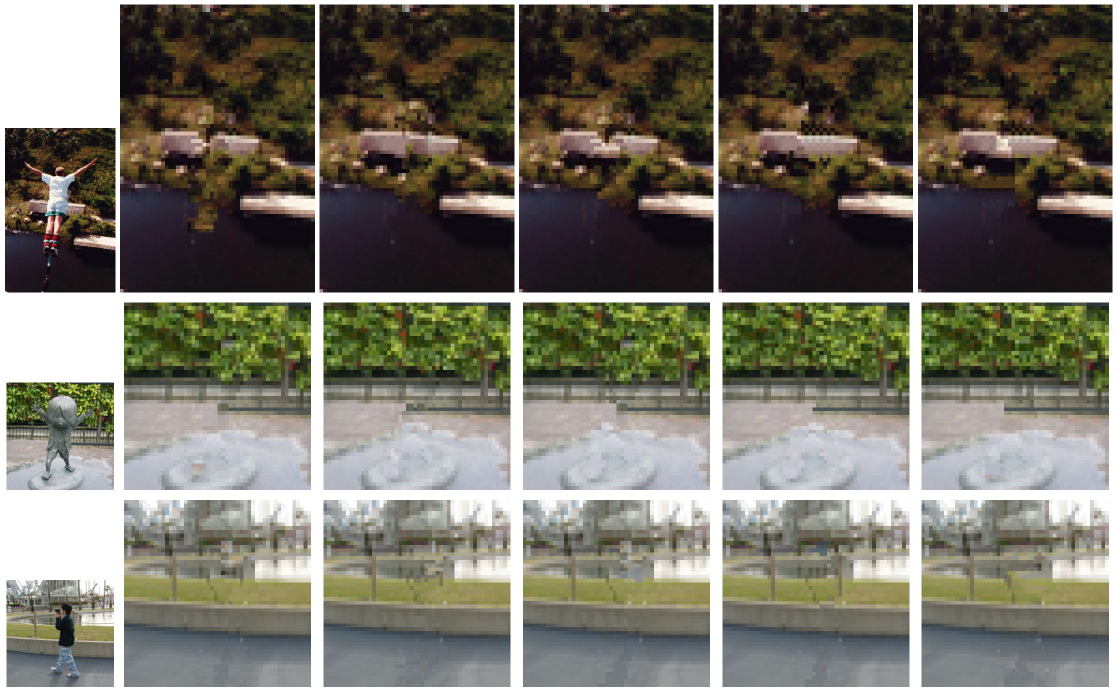


(a) Original image (b) Mask image (c) Our priority

Figure 3: Illustration of the proposed priority focusing on edge preservation.

2.2.2. Patch selection

The second step of the proposed algorithm consists in finding the suitable patch for the filling process. The similarity measurement, computed based on all known pixels in the patch, should be consistent with human perception. When based only on color, it is insufficient to propagate accurate linear structures into the target region and leads to uncontrolled and incoherent growing. This is mainly due to the fact that the perceptual color appearance depends not only on the color of the observed patch but also on



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

Figure 4: Completion of low resolution images. (a) image to be inpainted. Ouput when using priority adopted in (b) [12]; (c) [13]; (d) [14]; (e) [28] and (f) our proposal.

the surrounding and the context on which the patch is perceived. Consequently, to solve this problem, a similarity measure which takes into account the difference in colors and gradients is proposed and given below.

$$d(\Psi_p, \Psi_q) = \sum_i (\theta(I_p^i - I_q^i)^2 + (1-\theta)(\nabla I_p^i - \nabla I_q^i)^2) \tag{3}$$

where I_p, I_q are the corresponding RGB vectors; ∇I_p, ∇I_q represent the image gradient vectors. θ is a user defined weight balancing the two terms, fixed for the following as $\theta = 0.67$. The target patch with the minimal distance to the source one, Ψ_p, is the one to be chosen as described by the following equation:

$$\Psi_{\widehat{p}} = argmin_{\Psi_q \in \Phi}\{d(\Psi_p, \Psi_q)\} \tag{4}$$

As mentioned in [13], a major problem of local neighborhood search is its tendency to get stuck at a particular place in the same image and to produce verbatim copying. This region may generate blocking artifacts. In order to address this problem, we proposed an improvement for the patch selection step. The idea is based on the fact that patches that are neighbors in the input image should remain neighbors in the output image. First, K most similar patches obtained by the local neighborhood search patches are used as candidates. Second, for each patch, Ψ_p, a standard deviation describing the variability of neighboring source patches is formulated as follows:

$$V(\Psi_p) = \sum_{E\in\{R,G,B\}} \sqrt{\frac{\sum(E(\Psi_p) - \bar{E}(\Psi_{N(p)}))^2}{|N(p)|}} \tag{5}$$

where $N(p)$ is the neighborhood centered at p, $E(\Psi_p)$ is the variance of pixel values at the neighboring patch Ψ_p in each of the RGB channels and $\bar{E}(\Psi_p)$ is the mean variance of $|N(p)|$ neighboring patches in RGB channels. The size of $N(p)$ is a global parameter that should be chosen larger than the patch size. consequently, the chosen patch must satisfy the following equation:

$$\Psi_{\widehat{p}} = \underset{\Psi_q\in\Phi}{\arg} min\{|V(\Psi_p) - V(\Psi_q)|\} \tag{6}$$

Figure 5 illustrates an example for patch selection *i.e.* a completion of the *Kanizsa* triangle. One can notice that unsuitable patch selection as shown on subfigure 5-b, may lead to unexpected results or artifacts (*see* subfigure 5-b). On the contrary, an appropriate selection as shown on subfigure 5-e, produces satisfactory results (*see* subfigure 5-f)

Finally, the missing pixels are copied from the corresponding pixels in the selected patch. An offset map defining the relationship between pixels to be inpainted and pixels in the known regions is obtained by keeping a track of the copy process. The latter is of a high importance since it is used as an initial guess for inpainting higher resolution in the pyramid.

2.3. Inpainting higher resolution images

Once the inpainting of the lowest resolution image G_N is completed, an offset map is generated and used as the initialization to reconstruct higher resolutions. This offset map defining the relationship between pixels to be inpainted and pixels in the known regions (*see* Figure 6) is given below:

$$SM(p) = \begin{cases} (\triangle x, \triangle y) & p(x,y) \in \Omega \\ (0,0) & \text{otherwise} \end{cases} \tag{7}$$

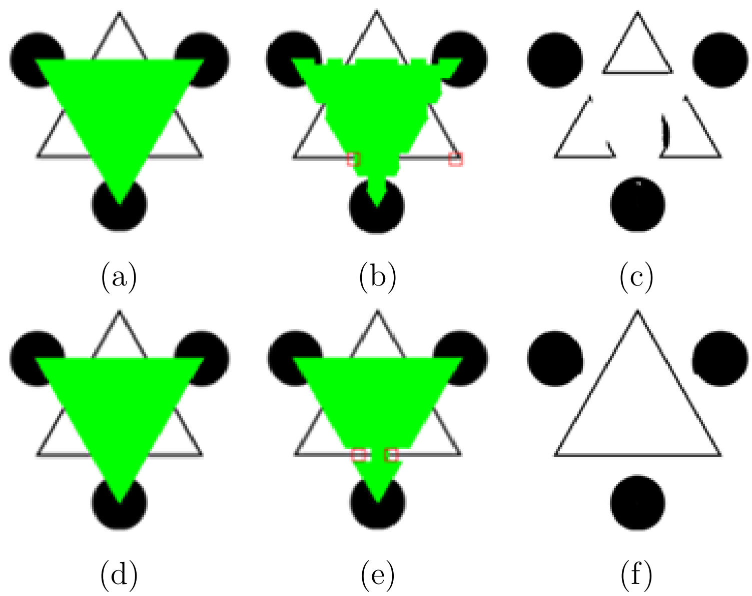
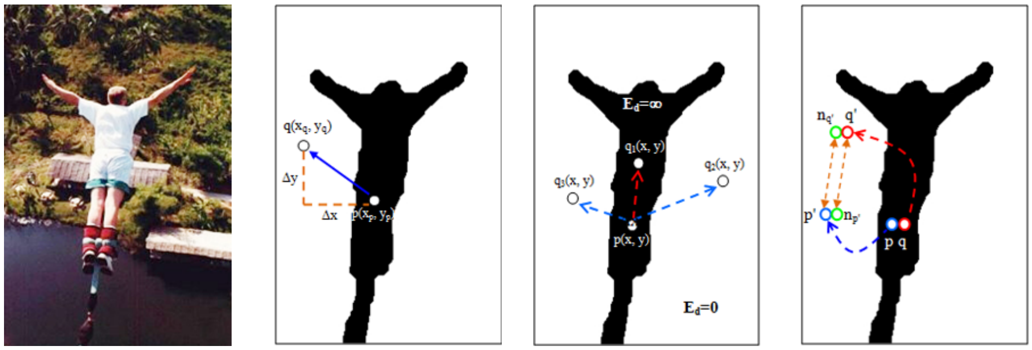


Figure 5: A restoration of the *Kanizsa* triangle using inappropriate and appropriate patch selection. (a) & (d) image to be inpainted; (b) & (e) a patch selection respectively with and without improvement; (c) & (f) Final results.

Therefore, the offset map a previous resolution is interpolated for a higher resolution. However, the inpainted results derived directly from this map may contain annoying artifacts affecting the naturalness of the resulting image. Authors of [27] proposed an energy function composed of two terms, data and smoothness, in order to refine the offset map. The energy function is defined as follows:

$$EM = \alpha \sum_{p \in \Omega} E_d(SM(p)) + (1-\alpha) \sum_{(p,q) \in NB} E_s(SM(p), SM(q)) \qquad (8)$$

Where E_d is the data term related to external requirements and E_s is the smoothness term defined over a set of neighboring pixels, NB. The parameter α is a user defined weighting factor, fixed to $\alpha = 0.5$ in our case, allowing to balance the two terms. One objective when using equation 8 is to minimize the energy related to both data and smoothness terms. Several optimization approaches exist in the literature. In the proposed approach, a global optimization based on graph-cuts is used because its recognized efficiency.



(a) An original image (b) An offset (c) Data term (d) Smoothness term

Figure 6: Illustration of the operators used in the proposed approach.

2.3.1. Data Term

The data term, E_d, is linked to external constraints that measures how appropriate is a label, or an offset. During the completion process, for each pixel in the target region an offset is assigned to the pixel in the known regions. This offset is used in the computation of the data term to avoid including pixels from the missing region. This data term is defined by equation (9).

$$E_d(SM(p)) = \begin{cases} \infty & (x + \triangle x, y + \triangle y) \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

In some cases, the specific pixels in the input image can be forced to appear or disappear in the output image by setting the value of E_d. For example, a saliency map can be used to weight the data term. Therefore, a pixel with a high saliency value should be kept and a pixel with a low saliency value should be removed. Figure 6-c illustrates visually how to adapt the value of the data term.

2.3.2. Smoothness Term

The second component of the energy function is the smoothness term representing the discontinuity between two neighboring pixels $p(x_p, y_p)$ and $q(x_q, y_q)$. In [27], authors proposed an effective formula, expressed by by equation (10), for the smoothness term accounting for both color and gradient differences between corresponding spatial neighbors in the output and input images to create a coherent stitching.

$$E_s(SM(p), SM(q)) = \begin{cases} 0 & SM(p) = SM(q) \\ \beta\delta M(SM(p)) + \gamma\delta G(SM(p)) & otherwise \end{cases} \quad (10)$$

where β and γ are weighting factors balancing these two terms, set to $\beta = 1$, $\gamma = 2$ for our experiment. ΔM and ΔG denote the differences of magnitude and gradient and they are defined as the follows:

$$\begin{aligned} \Delta M(SM(p)) &= ||I(n_{p'}) - I(q')|| + ||I(n_{q'}) - I(p')|| \\ \Delta G(SM(p)) &= ||\nabla I(n_{p'}) - \nabla I(q')|| + ||\nabla I(n_{q'}) - \nabla I(p')|| \end{aligned} \quad (11)$$

where I and ∇I are the magnitude and gradient at these locations. $p' = p + SM(p)$ and $q' = q + SM(q)$ are locations used to fill for pixels p and q, respectively. $n_{p'}$ and $n_{q'}$ are two 4-connected neighbors of p' and q', respectively. Figure 6-d depicts an intuitive way for evaluating the smoothness term. The main idea is based on the fact that a pixel is used for filling, then its neighbors should be also filled as neighbors in the inpainted regions. Moreover, the difference between filled pixels and their neighbors in the target region and known region should be as small as possible.

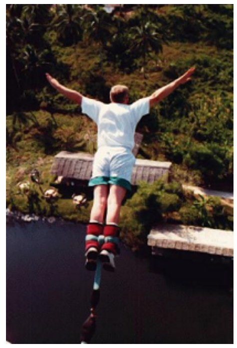
Figure 7 provides an example of the offset map at the original resolution image after the graph-cuts optimization. Each offset is the 2-D coordinates including horizontal and vertical relationships. The output generated by two corresponding offset maps is shown in the Figure 7-b.

2.4. Interpolation of successive levels

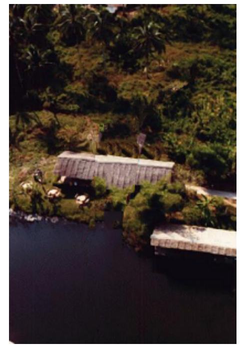
A full offset map is first inferred from a completion at the lowest level of the pyramidal representation of the input image. Then, it is interpolated to higher resolutions using the *nearest neighbor algorithm*, and the offset-map values are *upscaled* by simply doubling each value to match the higher image resolution.

At the highest level, only small shifts relative to the initial guess are examined. It means that only some parent neighbors are considered. In our implementation, the shift relative to each coordinate varies in the range $[-a, a]$, so it takes $(2a+1)^2$ labels for both direction. It is important to note that the data and smoothness terms are always computed with respect to the actual shifts and not to the labels.

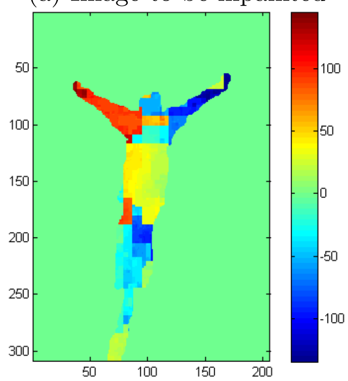
Figure 8-a illustrates an example of the gaussian pyramidal decomposition and the associated reconstruction scheme (Figure 8-b).



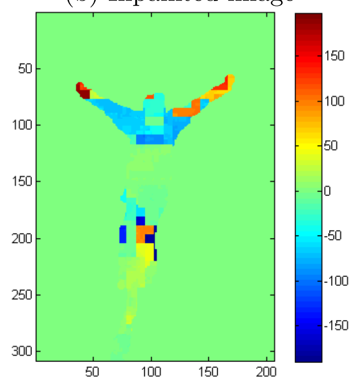
(a) Image to be inpainted



(b) Inpainted image

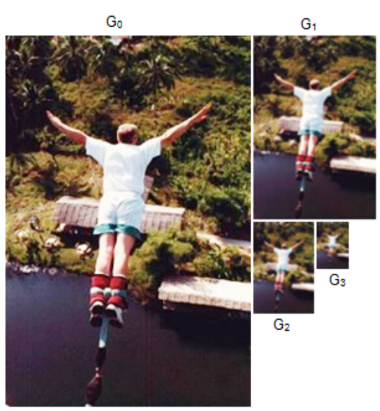


(c) Horizontal offset map

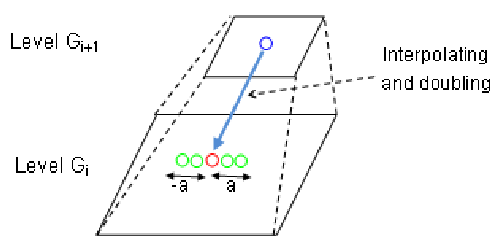


(d) Vertical offset map

Figure 7: Illustration of the offset values for a commonly used image.



(a) Gaussian pyramid of the input image



(b) Interpolation between two successive levels

Figure 8: Illustration of the gaussian pyramid and interpolation of the Shift-Map.

3. Comparison with the state-of-the-art approaches

This section is dedicated to the performance evaluation of the proposed algorithm and comparison with the state-of-the-art. The parameters of the algorithm are kept constant for the test presented in this paper.

3.1. Comparative study

Figure 9 illustrates the results of the proposed method in comparison with some state-of-the-art methods belonging to the same category. In this paper, some methods of the second group introduced by *A. Criminisi et al.* [12], *J. Wu et al.* [13], *Q. Zhang et al.* [14] and *W. Cheng et al.* [28] are considered. Four input images, commonly used for in painting evaluation, of size 200×200 respectively *yokoya*, *kidstatue*, *cameraman* and *student* are chosen for the experiment. The obtained results shown in Figure 9 demonstrate the efficiency of our method in terms of visual quality and in comparison to the state-of-the-art.

In order to avoid the limitations of subjective evaluation. We employed an objective metric in [30, 31] to objectively measure the quality of the inpainting results. The chart 10 illustrate a comparison between our inpainting quality with other methods in the second group and the chart in figure 11 is to compare with state-of-the-art methods including both other greedy algorithms and other global optimization approach. These charts again objectively confirmed that the quality of our restoration is not lower than quality of others in most cases.

3.2. Comparison with unoptimized version

Our method is based on previous idea we developed in [25] but with the use of a multi-resolution patch matching instead of an offset-map optimization that we call here *unoptimized* version. Results obtained with the latter are compared to those of the proposed approach on Figure 12. It is clearly shown that our method (Figure 12-c) is slightly better than the method proposed in [25]. This makes sure by the objective comparison included in chart 11. However, the performance in terms of computational complexity of our proposal is rather better than the unoptimized approach. This issue is discussed in more details in the next section.

-a- -b- -c- -d- -e- -f-

yokoya kidstatue cameraman student

Figure 9: Inpainting results for 4 commonly used image: a) original images; results when using method of b) *A. Criminisi et al.* [12], c) *Wu et al.* [13], d) *Zhang et al.* [14], e) *Cheng et al.* [28] and f) Our proposal.

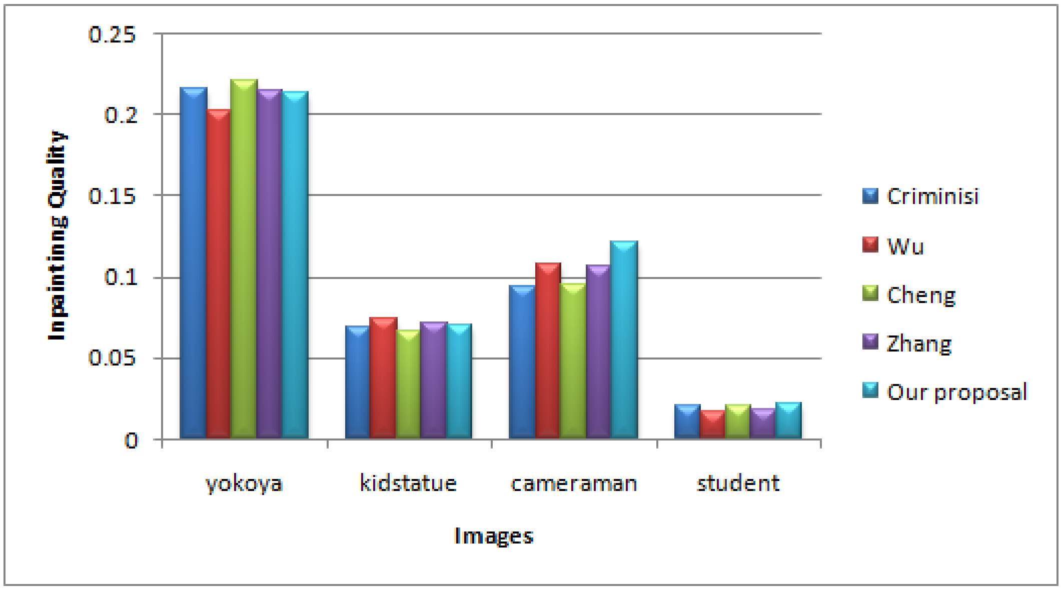


Figure 10: A comparison of inpainting quality with the methods in the second groups.

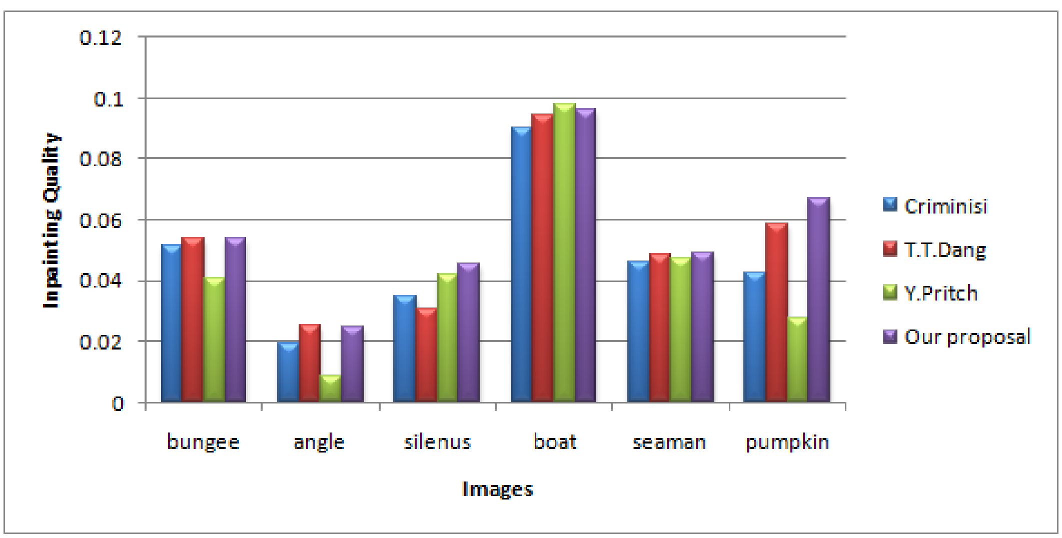
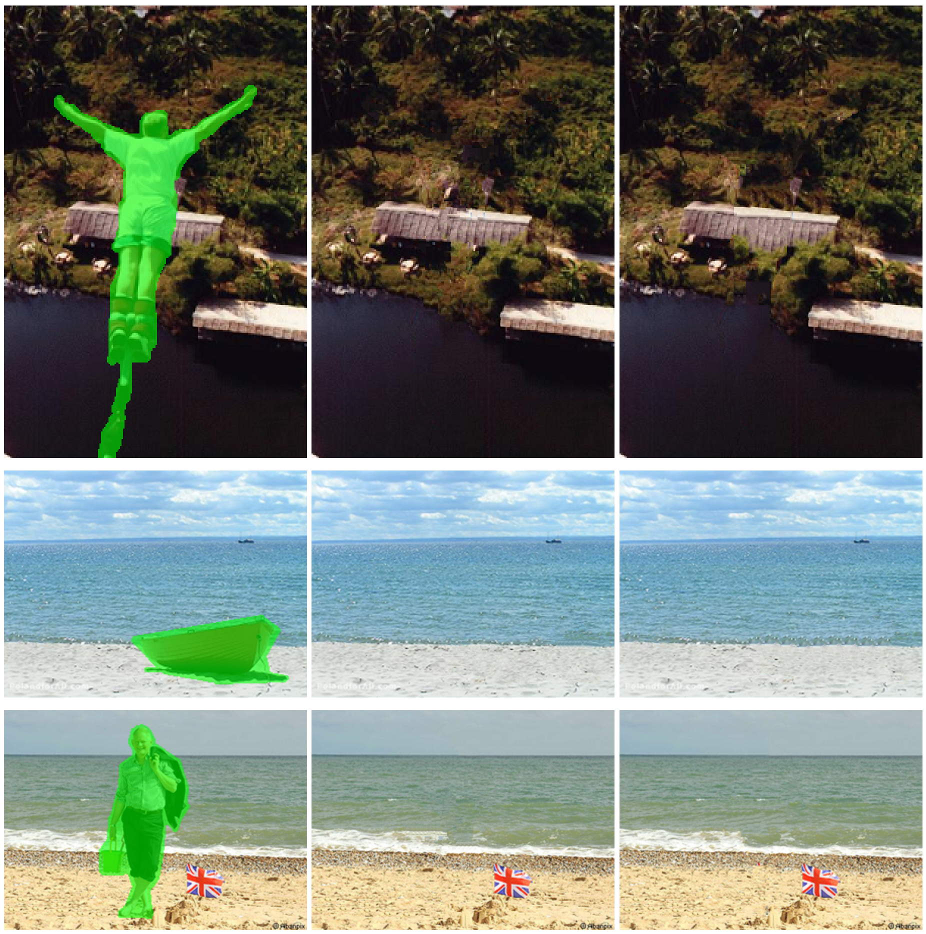


Figure 11: A comparison of inpainting quality with state-of-the-art methods.



(a) (b) (c)

Figure 12: A comparison between our proposal and the unoptimized version described in [25]. (a) Image to be inpainted; Inpainting results using (b) unoptimized version and (c) our proposal.

4. Performance evaluation

The visual quality of the proposed approach has been clearly demonstrated in the previous sections. To confirm the efficiency of our proposal, a complete evaluation of performance is considered in term of the computational complexity and discussed based on both two aspects: i) locally (compare among different stages of the algorithm) and ii) globally (compare our algorithm with others).

4.1. Local performance evaluation

Locally, we first carried out an analysis of the performance of each stage in our algorithm including: greedy strategy (window-based priority, patch-match search) and global optimization strategy.

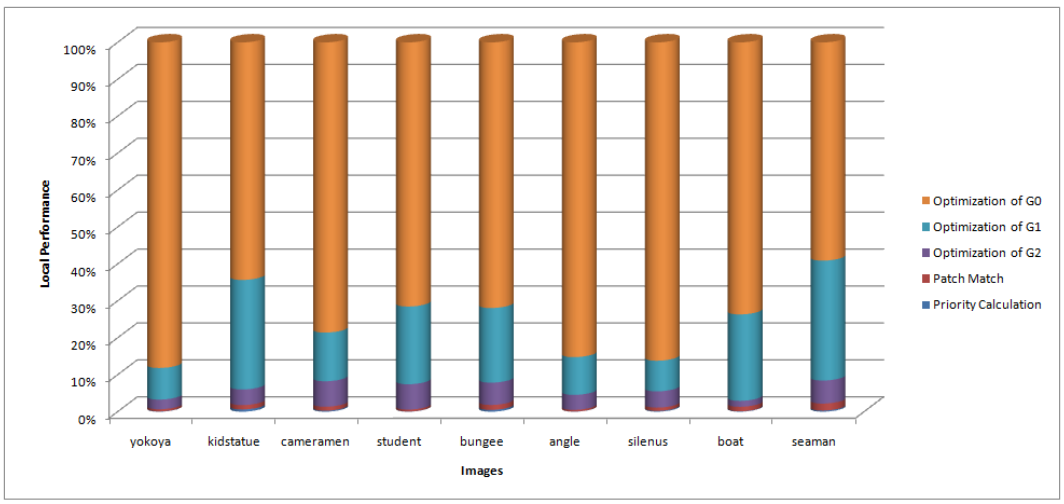


Figure 13: An analysis of local performance in our proposal.

Figure 13 shows a chart of comparison between stages in our algorithm for nine images. Each column displays total time in percent of each stage. In this case, the level of pyramid is set to 4 (depending on the size of input image). The greedy strategy is applied at the lowest version which have the smallest size. Thus, the time consuming for this strategy is very low. There are two main sub-stages in this strategy: window-based priority and patch-match search. From the figure 13, it is very clear to see that these steps account for a very small portion in whole process (average time about 0.18% for window-based priority and 0.78% for patch-match searching).

The execution time is mainly due to global optimization. The higher levels are, the more time consume to complete. This is illustrated evidently in the chart. The average time of global optimization strategy for the levels G_2, G_1, G_0 corresponds to 3.68%, 14.2% and 58.53%. Totally, the greedy strategy occupies less than 1% of whole execution time, whereas the global optimization consumes nearly 76.4%. This analysis shows that combining the greedy strategy with the global optimization algorithm not only guarantees the quality output based on the feature of human visual system but also is an important step to improve the performance of restoration process.

In addition, the use of hierarchical representation also significantly reduces the execution time because of the reduction of image size.

4.2. Global performance evaluation

To increase the reliability of our framework, we go on analyzing the global performance by comparing with other approaches. The performance is evaluated from two points of view: i) between methods of the second group and ii) between methods belonging to greedy and global optimization strategies. For the sake of fair comparison, all source codes of the compared methods have been written in C/C++ language and implemented on the same PC with an Intel Core i5 2.8GHz CPU and 4GB RAM .

Firstly, a number of approaches from the second group are chosen for evaluating the performance of our method including *A. Criminisi* [12], *J. Wu* [13], *Q. Zhang* [14] and *W. Cheng* [28]. In this part of the experiment, four images of size 200 × 200 pixels, shown in Figure 9, are used. The computation time obtained for the considered methods is given in Table 3 in seconds.

Table 3: Computational time (in seconds) for comparison of methods of the second group

Image	*yokoya*	*kidstatue*	*cameraman*	*student*
Damaged Area	6.53%	19.86%	12.78%	7.8%
A. Criminisi [12]	**3.22**	10.09	5.75	4.74
J. Wu [13]	3.64	10.29	6.05	5.98
W. Cheng [28]	3.58	11.60	6.91	5.64
Q. Zhang [14]	38.38	251.74	111.84	59.52
Our proposal	3.71	**4.53**	**3.29**	**3.84**

Secondly, for the sake of larger comparison with literature, three inpainting methods corresponding to algorithms proposed by *A. Criminisi et al.*

[12] representing greedy strategy; *T.T. Dang et al.* [25] for the unoptimized approach; and *Y. Pritch et al.* [19] for global optimization strategy have been implemented. Five images, shown in Figure 8 were chosen for this experiment (including *bungee* (206×308), *angle* (300×252), *silenus* (256×480), *boat* (300×225) and *seaman* (300×218)).

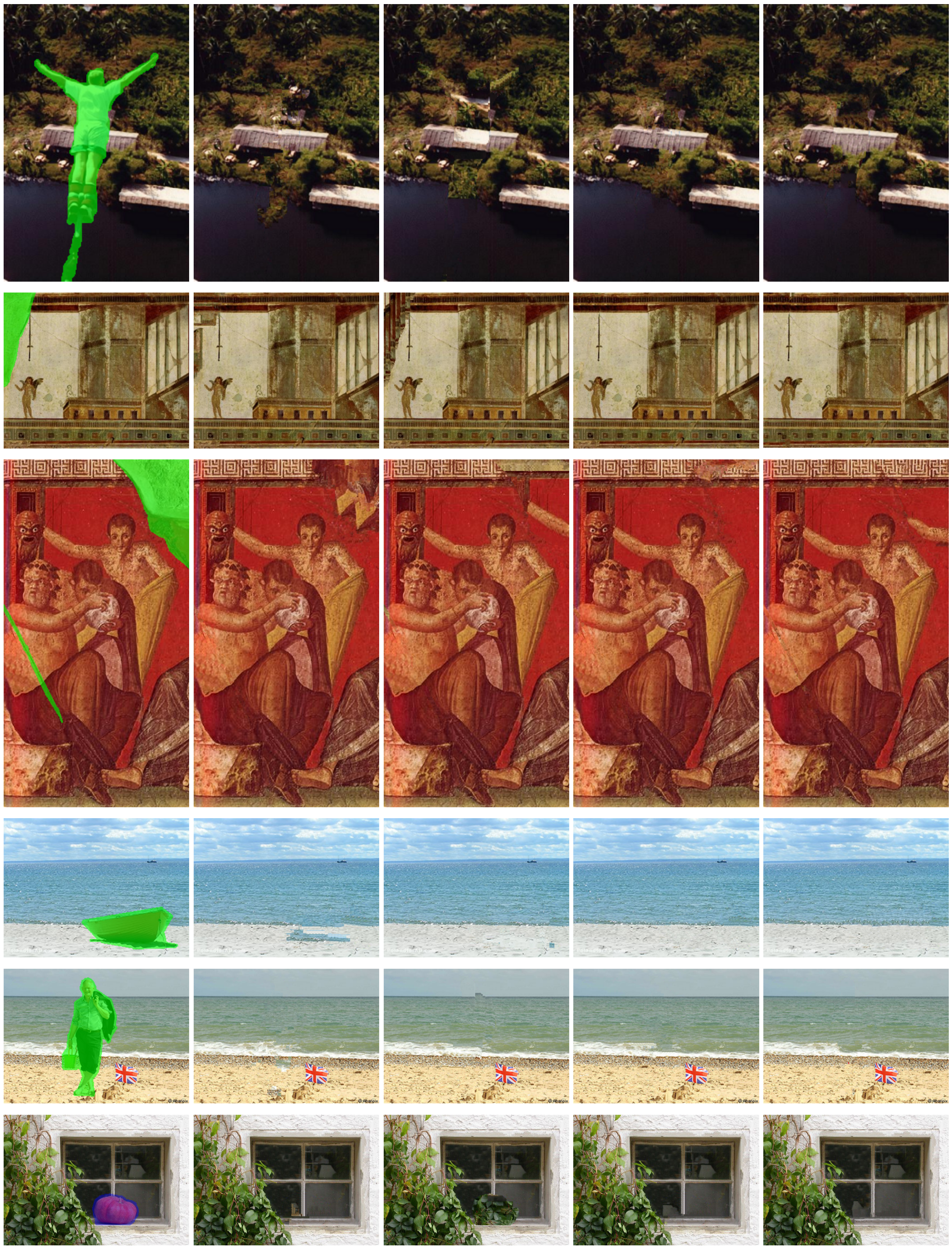
Figure 14 illustrates the results obtained with the proposed approach in comparison to the others. Figure 14-a gives images to be inpainted where inpainting areas cover respectively *12.6%*, *5.83%*, *7.74%*, *10.73%* and *14.87%* of the whole image. As it can be seen from these results, the visual quality of our proposal is confirmed again. Indeed, our results look more natural and more coherent than those of other approaches. Moreover, the values in Table 4 depicting the computational time in seconds for the selected methods show an outstanding performance of our approach in comparison with the others.

Table 4: Computational time in global comparison

Image	*bungee*	*angle*	*silenus*	*boat*	*seaman*	*pumpkin*
Size	(206×308)	(300×252)	(256×480)	(300×225)	(300×218)	(473×332)
Damaged Area	12.6%	5.83%	7.74%	10.73%	14.87%	5.1%
A. Criminisi [12]	16.30	8.20	38.29	24.54	27.31	28.98
T. T. Dang [25]	15.92	16.36	63.18	50.18	55.16	54.57
Y. Pritch [19]	35.39	13.24	57.68	21.18	15.50	37.35
Our proposal	**3.32**	**5.81**	**7.53**	**7.25**	**5.97**	**6.31**

5. Conclusion

In this paper, a novel inpainting approach has been introduced by combining both greedy and global optimization strategies based on a pyramidal representation of the image. We first propose an extension of a well-known exemplar-based method where major improvements are linked to the introduction of window-based priority and patch selection. This extension is applied only at the lowest resolution in order to generate an appropriate initialization accounting for human perception. Then, an offset is produced and interpolated for higher resolution images. In order to achieve visually smooth result in the final output image, the offset map is refined by a global optimization algorithm based on multi-label graph-cuts. A comparison with some representative approaches from literature belonging to the second group (i.e. global optimization) is carried out and inpainting results show that our approach not only produces better visual quality of output images but also



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

Figure 14: Performance evaluation. (a) Image to be inpainted; Outputs when using methods in (b) [12]; (c) [19]; (d) [25] and (e) our proposal;

implements noticeably faster. A subjective validation with an adapted protocol will help in characterizing the visual improvement of the propose approach. Finally, As future directions, we believe that the proposed approach may be extended for video completion purposes. This kind of application is indeed very time consuming and our proposal could drastically reduce the computational complexity.

[1] P. Arias, G. Facciolo, V. Caselles, G. Sapiro, A variational framework for exemplar-based image inpainting, International Journal of Computer Vision (2011) 1–29.

[2] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, Proceedings of the 27th annual conference on Computer graphics and interactive techniques (2000) 417–424.

[3] M. Bertalmio, A. Bertozzi, G. Sapiro, Navier-stokes, fluid dynamics, and image and video inpainting, In Proceeding IEEE International Conference on Computer Vision and Pattern Recognition (2001) 355–362.

[4] F. Bornemann, T. Marz, Fast image inpainting based on coherence transport, Journal of Mathematical Imaging and Vision 28 (2007) 259–278.

[5] T. F. Chan, J. Shen, Nontexture inpainting by curvature-driven diffusions, Journal of Visual Communication and Image Representation 12 (2001) 436–449.

[6] D. Tschumperle, Fast anisotropic smoothing of multi-valued images using curvature-preserving pdes, International Journal of Computer Vision 68 (2006) 65–82.

[7] A. Efros, T. Leung, Texture synthesis by non parametric sampling, Proceeding of the Seventh IEEE International 2 (1999) 1033–1038.

[8] M. Ashikhmin, Synthesizing natural textures, The proceedings of 2001 ACM Symposium on Interactive 3D Graphics (2001) 217–226.

[9] V. Kwatra, A. Schodl, I. Essa, G. Turk, A. Bobick, Graphcut textures: image and video synthesis using graph cuts, ACM Transactions on Graphics (TOG 22 (2003) 277–286.

[10] V. Kwatra, I. Essa, A. Bobick, N. Kwatra, Texture optimization for example-based synthesis, ACM Transactions on Graphics (TOG 23 (2005) 795–802.

[11] A. Bugeau, M. Bertalmio, Combining texture synthesis and diffusion for image inpainting, Proceeing International Conference Computer Vision Theory and Applications (2009) 26–33.

[12] A. Criminisi, P. Perez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Transaction of Image Process 13(9) (2004) 1200–1212.

[13] J. Wu, Q. Ruan, Object removal by cross isophotes exemplar based image inpainting, Proceeding of International Conference of Pattern Recognition (2006) 810–813.

[14] Q. Zhang, J. Lin, Exemplar-based image inpainting using color distribution analysis, Journal of Information Science and Engineering (2011) 1–12.

[15] X. Zongben, J. Sun, Image inpainting by patch propagation using patch sparsity, Image Processing, IEEE Transactions on 19(5) (2010) 1153–1165.

[16] A. Wong, J. Orchard, A nonlocal-means approach to exemplar-based inpainting, Image Processing, IEEE Transactions on (2008) 2600–2603.

[17] Y. Wexler, E. Shechtman, M. Irani, Space-time video completion, IEEE Transactions Pattern Analysis and Machine Intelligence 29 (March 2007) 463 – 476.

[18] N. Komodakis, G. Tziritas, Image completion using global optimization, IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1 (2006) 442 – 452.

[19] Y. Pritch, E. Kav-Venaki, S. Peleg, Shift-map image editing, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009) 151–158.

[20] C. Barnes, E. Shechtman, A. Finkelstein, D. Goldman, Patchmatch: a randomized correspondence algorithm for structural image editing, ACM Transactions on Graphics-TOG 28 (2009) 24–34.

[21] D. Simakov, Y. Caspi, E. Shechtman, M. Irani, Summarizing visual data using bidirectional similarity, Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (2008) 1–8.

[22] M. Elad, J. Starck, P. Querre, D. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis, Applied and Computational Harmonic Analysis 19 (2005) 340–358.

[23] Z. Xu, J. Sun, Image inpainting by patch propagation using patch sparsity, Image Processing, IEEE Transactions on 19 (2010) 1153–1165.

[24] M. Fadili, J. Starck, F. Murtagh, Inpainting and zooming using sparse representations, Image Processing, IEEE Transactions on 52 (2009) 64–79.

[25] T. T. Dang, M. C. Larabi, A. Beghdadi, Multi-resolution patch and window-based priority for digital image inpainting problem, 3rd International Conference on Image Processing Theory, Tools and Applications (2012) 280 – 284.

[26] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 1222–1239.

[27] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, M. Cohen, Interactive digital photomontage, Proceedings of SIGGRAPH (2004) 294–302.

[28] W. Cheng, C. Hsieh, S. Lin, C. Wang, J. Wu, Robust algorithm for exemplar-based image inpainting, Proceeding of International Conference on Computer Graphics, Imaging and Visualization (2005) 294–302.

[29] P. Beaudet, Rotationally invariant image operators, in: International Joint Conference on Pattern Recognition, volume 579, 1978, pp. 579–583.

[30] T. T. Dang, A. Beghdadi, M. Larabi, Perceptual quality assessment for color image inpainting, IEEE International conference on image processing, ICIP (2013).

[31] T. T. Dang, A. Beghdadi, M. Larabi, Perceptual evaluation of digital image completion quality, 21st European signal processing conference, EUSIPCO (2013).