Exemplar-based Image Inpainting using Multiscale Graph Cuts

Yunqiang Liu and Vicent Caselles

***Abstract*—We present a novel formulation of exemplar-based inpainting as a global energy optimization problem, written in terms of the offset map. The proposed energy functional combines a data attachment term to ensure the continuity of the reconstruction at the boundary of the inpainting domain and a smoothness term that ensures a visually coherent reconstruction inside the hole. This formulation is adapted to obtain a global minimum using graph cuts algorithm. In order to reduce the computational complexity, we propose an efficient multiscale graph cuts algorithm. Moreover, to compensate for the loss of information at low resolution levels we use a feature representation computed at the original image resolution. This permits to alleviate the ambiguity induced by comparing only color information when the image is represented at low resolution levels. Our experiments show the good performance of the proposed algorithm when compared with other recent algorithms.**

***Index Terms*—Image inpainting, offset map, graph cuts, feature vector**

I. Introduction

IMAGE inpainting, also known as image completion or dis-occlusion, is an active area of research in image processing. It aims to obtain a visually plausible interpolation in a region of the image where the data is missing or we want just to modify it. It has become a standard tool in image and video processing with many applications to image restoration (e.g., scratch or text removal in photographs), new view synthesis [1], [2] (e.g. filling the dis-occluded regions), object removal, image coding [3], [4] and transmission [5] (e.g., recovery of the missing blocks), etc.

Most inpainting methods found in the literature can be classified in two main categories: geometry- and exemplar-based methods.

In geometry-based methods, images are usually modeled as functions with some degree of smoothness, for instance, expressed in terms of the curvature of the level lines or the total variation of the image. They take advantage of the smoothness assumption and interpolate the inpainting domain by continuing the geometric structure of the image (its level lines, or its edges), usually as the solution of a (geometric) variational problem, or by means of a partial differential equation (PDE). Such PDE can be derived from variational principles, as is the case for instance in [6], [9], [11], or inspired by phenomenological modeling [8], [10], [13], [12]. These methods show good performance in propagating smooth level lines or gradients, but fail in the presence of texture. They are often referred to as structure inpainting methods.

In most cases, geometry-oriented methods are local in the sense that they are based on PDEs. An implication of this is that, among all data available from the image, they only use the information at the boundary of the inpainting domain. This frequently tends to introduce blurring artifacts in textured or large missing regions.

Exemplar-based methods were initiated by the work of Efros and Leung [17] on texture synthesis using non-parametric sampling techniques. In this context, texture is modeled as a two dimensional probabilistic graphical model, in which the gray, or color, value of each pixel is conditioned by its neighborhood. These methods rely directly on a sample of the desired texture to perform the synthesis. The value of each target pixel p is copied from the center of a (square) patch in the sample image, chosen to match the available portion of the patch centered at p. Exemplar-based methods have been adapted to inpainting [20], [19], [15]. As opposed to geometry-based methods, the exemplar-based approaches are non-local: to determine the value at a pixel, the whole image may be scanned in the search for a matching patch. They provide excellent results in recovering textures or repetitive patterns, but they may have difficulties in interpolating the image geometry if there are no examples of it.

Different strategies have been proposed to combine geometry and texture inpainting. Bertalmio et al. [18] decomposed the image into structure and texture components, inpainting each of them separately using geometry- and texture-based techniques, respectively. The final result is the composition of the two completed layers. Criminisi et al. [19] extended the texture synthesis approach by gradually propagating the information of the known region into the hole according to a priority order. This order is determined by the strength of the incoming edges and by a confidence map that takes into account the amount of known information in the patch. In this way it encourages the propagation and continuity of strong edges and the concomitant information into the hole. Xu et al. [21] improved the method of [19] by defining a patch priority order based on structure sparsity that can better distinguish between texture and structure, and is more robust to the continuation of edges. They also use a sparse linear combination of exemplars to infer patches in a framework of sparse representation, improving the consistency of the selected patches with their surroundings. Sun et al. [22] proposed to improve visual consistency by first propagating the structure, and then

Y. Liu is with Barcelona Media - Innovation Center, 08018 Barcelona, Spain (email: liuyunq@yahoo.com).

V. Caselles is with the Departament de Tecnologies de la Informació i les Comunicacions, Universitat Pompeu-Fabra, 08018 Barcelona, Spain (email: vicent.caselles@upf.edu)

filling-in the remaining unknown regions using a patch-based texture synthesis technique. This method requires the user to specify the curves that define, or approximate, the most salient missing structures. Jia et al. [23] propose a two-step method based on performing first a texture-based segmentation of the input image and extrapolating the region boundaries by tensor voting to generate a complete segmentation of the image. In a second step missing colors are synthesized using tensor voting. Tensor voting is a very attractive method for maintaining curvature; however it cannot perform well on complex structures. Moreover, image segmentation of natural images is a difficult task.

Most of the above exemplar-based approaches are greedy procedures where each target pixel is visited only once, and the results are very sensitive to the order in which pixels are processed. Contrary to the greedy methods, some approaches [27], [28], [29], [24], [25] formulate image inpainting as a discrete global optimization problem, where the image is modeled using a Markov Random Field (MRF) with pairwise interactions. In [27], the objective function is minimized using efficient belief propagation. In [30], [28], expectation-maximization (EM) schemes are used for texture synthesis or image completion. The EM-like algorithms require an initialization which, as it turns out, may get easily stuck into poor local minima. A multi-resolution approach is applied for fixing this problem. Starting at the coarsest level, the inpainting technique is then applied sequentially to each scale, using the upsampled result of the previous scale as initialization. This has become a common strategy for energy optimization in image inpainting methods.

Variational models are appropriate for combining the main features of geometry- and exemplar-based inpainting methods. A variational formulation of Efros-Leung method was proposed in [31], where the inpainting problem was written in terms of a *correspondence map* assigning to each point of the inpainting domain a corresponding point in the known part of the image, although the proposed numerical method was still greedy. The authors also proposed a total variation regularization of the correspondence map formulation, but did not study this model numerically. Demanet et al.'s model [31] is further analyzed by Aujol et al. [32], which also considered several extensions that include a regularization term on the reconstructed image (e.g. penalizing the total curvature of the level lines). Arias et al. [24], [25] proposed a variational framework for non-local image inpainting that permits to include features of geometry-based approaches by a proper choice of the similarity criterion. This is the case of the non-local Poisson and gradient medians methods which are based on the comparison of gradient patches [30], [24], [25]. Their energy is written in terms of the unknown part of the image and the correspondence map. The computational complexity is reduced by the use of the Nearest Neighbor Field (NNF) [42]. In [34], the author proposed a variational formulation of Efros-Leung method where the reconstructed image minimizes the sum of the distances of each patch to the manifold of patches determined by the known parts of the image. An equivalent formulation was also proposed in [43]. Bugeau et al. [33] iteratively applied texture synthesis and geometric methods in order to combine features of both of them.

The image inpainting method we propose here is based on a reformulation of the exemplar-based model in [31] as a discrete global optimization problem. Our inpainting algorithm relies on the global minimization of an energy function that enforces the structure and texture consistency. The proposed energy functional is written in terms of the offset map and combines a data attachment term to ensure the continuity of the reconstruction at the boundary of the inpainting domain and a term (also derived from a reduced patch comparison) that ensures a spatially coherent reconstruction inside the hole. This formulation is adapted to obtain a global minimum using graph cuts algorithm [37], [39], and can also be interpreted as a MAP-MRF (Maximum A Posteriori in the Markov Random Field) estimation problem. In order to reduce its computational complexity, graph cuts are applied using a multiscale scheme. Moreover, to compensate for the loss of structure and texture information at low resolution levels we use a feature representation that is computed at the original image resolution. This permits to improve the precision of correspondences when the image is represented at low resolution levels, and alleviates the ambiguity induced by comparing only color information. This is a novel feature of our method.

Although, according to our previous review, the optimization of an energy functional is a common approach for exemplar-based image inpainting [31], [30], [27], [43], [34], [24], there are significant differences with our method, both in the energy definition and the optimization strategy. Let us discuss them in more detail. As we said above, our energy function is defined in terms of the offset map, comparing full patches at the boundary of the inpainting domain to ensure the continuity of the reconstructed image on it, and using a reduced patch comparison (the four nearest neighbors) inside the hole that is able to produce a visually coherent reconstruction. In [27], the authors also consider an objective function that corresponds to the energy of a discrete MRF whose nodes are a subgrid of the image and whose labels are the possible values of the correspondence map. The energy consists of a boundary term similar to ours and a (pairwise potential) term measuring how well two patches assigned to neighboring nodes agree in their overlapping region. If written in terms of the same variable (either the correspondence or the offset maps), the two energies would coincide if the energy in [27] is applied to the full image grid, and the size of the patches used in the pairwise potential is reduced as in our method. The formulation of the energy function in terms of the offset map permits us to use the graph cuts algorithm to optimize the energy satisfying the assumptions that guarantee the convergence of α-expansions. The use of the four nearest neighbors in the pairwise potentials reduces significantly the number of edges of the graph, and therefore its complexity. When combined with the multiscale scheme, our experiments show that the performance times compare advantageously to the ones of [27]. Our experiments also show the added flexibility in recovering a coherent reconstruction when using all image pixels in the hole as nodes of the graph, instead of a subgrid. As shown in Figure 5(e), some structure inconsistency is generated using [27].

The relation of the proposed method with [31] has been mentioned above and will be explained in more detail in Section II. In contrast to [31] that uses the correspondence map as unknown, and a greedy optimization algorithm, the use of the offset map permits us to optimize the energy efficiently by means of graph cuts. Essentially the same can be said when we compare it to [24], [25]. In these papers, the authors alternatively optimize their energy with respect to the unknown image and the correspondence map (using for that the Patch Match algorithm) converging to a critical point [26]. The efficiency of our algorithm compares favorably to the one in [24], [25].

When addressing the inpainting problem in [41], the authors consider an energy with a data term and a smoothness term written also in terms of the offset map (called there the shift-map). Their data term is used to force the shift-map to look for pixels in the hole's complement, and therefore it cannot always ensure the continuity of the reconstructed image at the boundary of the hole (an example of this is shown in our experiment in Figure 10). Thus, the method proposed here can be seen as a combination of the ideas in [27] and [41], using in addition an efficient multiscale graph cuts algorithm that incorporates a feature representation at the lower resolution scales.

Summarizing, the main contributions of this paper are:

1) The inpainting problem is formulated using an exemplar-based approach as an energy optimization problem for the offset map. The energy function consists of a data attachment term that ensures the good continuation of the reconstructed image at the boundary of the inpainting domain and a term that favors spatial coherence in the image completion. The formulation is adapted to obtain a global optimum using graph cuts.
2) We propose a multiscale graph cuts algorithm to efficiently solve the energy minimization problem in which a feature vector representation is introduced to compare patches at low resolution, to compensate the information loss. This representation can significantly eliminate ambiguities and improve the accuracy of the offset map.

Let us finally explain the organization of the paper. In Section II we introduce our energy functional, written in terms of offset map. In Section III we propose a multiscale graph cuts algorithm and discuss some details of the algorithm. Experimental results and comparison with other algorithms are given in Section IV. In Section V we summarize the conclusions of the paper.

II. A model for exemplar based image inpainting

A. The Demanet-Song-Chan formulation for Inpainting

In the paper [31], the authors proposed a reformulation of the texture synthesis method by Efros and Leung [17] as an optimization problem, which is well adapted for image inpainting. Since it is at the basis of our formulation, let us briefly review it.

Let $\mathcal{D} = \{1,\dots,M\} \times \{1,\dots,N\}$ denote the image domain, and let $u : \mathcal{D} \to I\!R^n$ be a given image, where $n = 1$ for gray level images and $n = 3$ for color ones. To fix ideas, let us work in the color case and write $u = (u_i)_{i=1}^3$. Let us denote by Ω a hole in the image, that is, a subset of $\mathcal{D}$ where the image u is not known. The objective of image inpainting is to fill-in the hole Ω using the information from the known part of the image $\Omega^c = \mathcal{D} \setminus \Omega$, see Figure 1. We denote by $\partial\Omega$ the boundary of the hole. Since we are at the discrete level, we consider the boundary points as part of Ω.

Given a pixel $p \in \Omega$, we denote by Ψ_p the patch of the image centered at p, that is, the restriction of the image u to a neighborhood $Q_p = p + Q_0$ of p. We have denoted by Q_0 a neighborhood of 0, typically a square domain centered at 0.

Interpreting Efros and Leung [17] method as the solution of an optimization problem, Demanet, Song and Chan [31] proposed to find an optimal correspondence map $T : \Omega \to \Omega^c$ that minimizes the criterion

$$\mathcal{E}_c(T) = \sum_{p\in\Omega} \sum_{h\in Q_0} |u(T(p+h)) - u(T(p)+h)|^2. \quad (1)$$

$T(p) \in \Omega^c$ represents the point from which we copy its color at $p \in \Omega$. Notice that $T(p+h)$ is not defined when $p+h \notin \Omega$ and $u(T(p)+h)$ is also not defined whenever $T(p)+h \in \Omega$. But, let us keep this notation avoiding the technicalities of the more correct one that would exclude these cases. Notice also that, once a minimizer T of $\mathcal{E}_c$ has been found, the interpolated image can be defined in Ω as $u(p) = u(T(p))$, for all $p \in \Omega$.

The authors acknowledge the difficulty in minimizing (1) and propose an algorithm based on local minimizations close to the algorithm in [17]. They select a pixel p near the border of Ω and find $T(p)$ as the minimizer of the Euclidean distance $\|\Psi_p - \Psi_{T(p)}\|_2$, where only the known parts of the neighborhoods are compared. Then they define $u(p) = u(T(p))$. They repeat this process for all pixels of Ω using an onion peeling strategy. After this step, all pixels of Ω have been visited and the image u reconstructed in Ω. Then the previous algorithm is repeated until the energy does not decrease any more.

Since it will be convenient for our purposes, let us write $T(p) = p + m(p)$, where $m : \Omega \to I\!Z^2$ represents the offset map. Then

$$\begin{aligned}
\mathcal{E}_c(T) &= \sum_{p\in\Omega} \sum_{h\in Q_0} |u(T(p+h)) - u(T(p)+h)|^2 \\
&= \sum_{q\in\Omega+Q_0} \sum_{h\in Q_0} |u(T(q)) - u(T(q-h)+h)|^2 \\
&= \sum_{q\in\Omega+Q_0} \sum_{h\in Q_0} |u(q+m(q)) - u(q+m(q-h))|^2,
\end{aligned}$$

where we wrote $q = p+h$ to obtain the second equality. Thus, by minimizing this energy we aim to impose that the offset of the neighbors of q, say $m(q-h)$, has also to be valid for q. That is, both values $u(q+m(q))$ and $u(q+m(q-h))$ should be visually consistent. This would be the case if the offset map is constant or locally constant. Together with the above computation, this observation serves as a starting point for our formulation.

B. The proposed inpainting model

Let us formulate the inpainting problem as the computation of an offset map $m : \Omega \to \mathbb{Z}^2$ that minimizes the energy

$$\mathcal{E}(m) = \mathcal{E}_d(m) + \lambda \mathcal{E}_s(m), \tag{2}$$

where $\lambda > 0$, and $\mathcal{E}_d(m)$ and $\mathcal{E}_s(m)$ stand for the data and smoothness terms of the energy, respectively. Knowing the offset map, we reconstruct the image u in Ω by $u(p) = u(p + m(p))$, for all $p \in \Omega$. The choice of the offset map m as an optimization variable will be justified in Remark 3.2.

The data term is defined by comparing the patches Ψ_p and $\Psi_{p+m(p)}$ for the pixels p near the boundary of Ω. The patches are compared only using their known parts. This term ensures the continuity of the reconstruction at $\partial\Omega$. Thus, we define the data term as

$$\mathcal{E}_d(m) = \sum_{p \in \partial\Omega} D(\Psi_p, \Psi_{p+m(p)}), \tag{3}$$

where

$$D(\Psi_p, \Psi_{p+m(p)}) = \frac{\sum_{q \in Q_p^*} w_g(p,q) d_u(q, q+m(p))^2}{\sum_{q \in Q_p^*} w_g(p,q)} \tag{4}$$

measures the similarity between the target patch Ψ_p and the source patch $\Psi_{p+m(p)}$. We have denoted by $d_u(q, q')^2$ the Euclidean distance between the color vectors $u(p)$ and $u(q')$, i.e.

$$d_u(q, q')^2 = \textstyle\sum_{i=1}^{3} (u_i(q) - u_i(q'))^2, \;\; q, q' \in \mathcal{D}. \tag{5}$$

The weights $w_g(p, q)$ give more importance to pixels near the patch center and are defined as

$$w_g(p, q) = e^{-k_1 |p-q|^2}, \qquad p, q \in \mathcal{D},$$

with $k_1 > 0$ (we have taken $k_1 = 0.25$ in our experiments). Notice that the sum in (4) is extended to the common part of the two patches, that is

$$Q_p^* = \{q \in Q_p \cap \Omega^c : q + m(p) \in \Omega^c\}.$$

When both patches have no common part, the term $D(\Psi_p, \Psi_{p+m(p)})$ is set to a constant value. In practice, since the constant value does not affect the results obtained by minimizing $\mathcal{E}$, we use the value zero.

The smoothness term aims to impose the visual consistency of pixel values pointed by the offset map. Given a pixel $p \in \Omega$, we denote by $\mathcal{N}_p$ a neighborhood of p (e.g. the 4-neighborhood formed by the horizontal and vertical nearest neighbors). Then we define the smoothness term as

$$\mathcal{E}_s(m) = \sum_{p \in \Omega \setminus \partial\Omega} \sum_{q \in \mathcal{N}_p} d_u(p + m(p), p + m(q)). \tag{6}$$

Notice that $d_u(p+m(p), p+m(q))$ in (6) is zero if the offset map is constant in the neighborhood of p.

The energy function $\mathcal{E}$ tries to keep the boundary continuity using the data term $\mathcal{E}_d$, and get a consistent reconstruction inside the hole thanks to the smoothness term $\mathcal{E}_s$, that says that two neighboring pixels in the hole should be similar to some neighboring pair in the known region, as used for photo montage [40] or image retargeting [41].

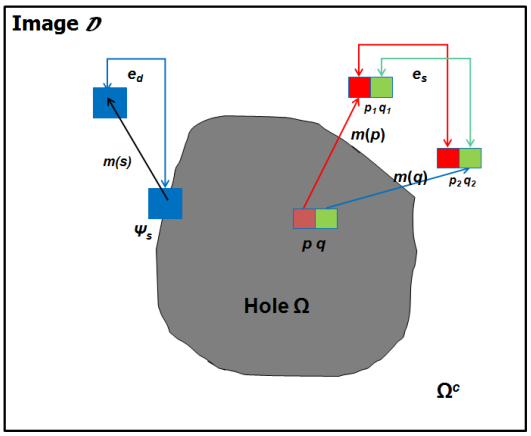


Fig. 1. Illustration of the energy function.

Figure 1 illustrates both terms of the energy function. To illustrate the smoothness term, let p and q be two adjacent pixels, marked in red and green colors in Figure 1. If they have the same offset vector, then they are copied from two adjacent pixels in the known region, and the term $|u(p + m(p)) - u(p + m(q))| = 0$. However, this difference can be zero while p, q do not necessarily share the same offset vector. This is also illustrated in Figure 1. Suppose that p and q are mapped to pixels p_1 and q_2 in Ω^c through their corresponding offsets $m(p)$ and $m(q)$. Let p_2 and q_1 be the neighbors of pixels p_1 and q_2, respectively. The choice of the offsets $m(p)$ and $m(q)$ is optimal if the pixels $p_1 = p + m(p)$ and $p_2 = p + m(q)$ are similar in color (they are compared in the term $d_u(p + m(p), p + m(q))$), and also $q_1 = q + m(p)$ and $q_2 = q + m(q)$ are similar (they are compared in the term $d_u(q + m(p), q + m(q))$). In that case the pixels p, q are filled-in in a consistent way. Some artifacts may appear if we cannot find a neighboring pair in Ω^c similar to this filled-in neighboring pair in the hole.

III. Optimization of $\mathcal{E}$ using multiscale graph cuts

A. Minimizing the energy $\mathcal{E}$ via graph cuts

Notice that the energy (6) can be written as

$$\mathcal{E}(m) = \sum_{p \in \partial\Omega} e_d(p; m(p)) + \lambda \sum_{(p,q) \in E} e_s(p, q; m(p), m(q)), \tag{7}$$

where $E = \{(p, q) : p \in \Omega \setminus \partial\Omega, q \in \mathcal{N}_p\}$ is the set of edges connecting each pixel $p \in \Omega \setminus \partial\Omega$ to its neighbors $\mathcal{N}_p$ in Ω. The term $e_d(p; m(p))$ is given by (4) and measures the cost of associating the offset $m(p)$ to pixel p. By writing $e_d(p; m(p)) = 0$ for pixels $p \in \Omega \setminus \partial\Omega$, the first sum may be extended to Ω. The term $e_s(p, q; m(p), m(q)) = |u(p + m(p)) - u(p + m(q))|$ evaluates the consistency cost of assigning the offsets $(m(p), m(q))$ to (p, q).

The offsets are the labels associated to each pixel. In our case the set of labels is a finite set $\mathcal{L} \subseteq \{q - p : q \in \Omega^c, p \in \Omega\}$. Given a labeling $m : \Omega \to \mathcal{L}$ and a particular label $\alpha \in \mathcal{L}$, we say that the labeling $\tilde{m}$ is within an α-expansion of m if $\tilde{m}(p) = m(p)$ each time $\tilde{m}(p) \neq \alpha$. As we shall see below, the problem of finding the minimum of the energy in the class of

all α-expansions of m can be formulated as the minimization of an energy functional depending on a finite set of binary variables. Then the algorithm realizes cycles through the set of labels $\alpha \in \mathcal{L}$, finding each time the lowest energy α-expansion move from the current labeling. It adopts $\tilde{m}$ as the new current labeling if $\tilde{m}$ has a lower energy than the current one. The algorithm ends when there is no more α-expansion move with lower energy, for any label α.

To minimize the energy (7) we construct a graph whose vertices correspond to pixels in Ω. The labeling is identified with the offset function m that takes a finite number of values $\mathcal{L}$. If m is the current labeling and $\alpha \in \mathcal{L}$, we associate a binary value $X(p)$ to each vertex p so that $X(p) = 1$ if p adopts label α and $X(p) = 0$ if it keeps the old label $m(p)$. We then write the energy of a labeling $\tilde{m}(p)$ within an α-expansion of m as a function depending on a finite set of binary variables.

Let m be a current labeling, and let us restrict the energy $\mathcal{E}$ to the α-expansions of m. Then the energy is defined on the binary function $X : \Omega \to \{0, 1\}$ as

$$\mathcal{E}(X) = \sum_{p\in\Omega} \widetilde{e}_d(p; X(p)) + \lambda \sum_{(p,q)\in E} \widetilde{e}_s(p, q; X(p), X(q)),$$

where $\widetilde{e}_d(p; X(p)) = e_d(p; (1 - X(p))m(p) + X(p)\alpha)$ and $\widetilde{e}_s(p, q; X(p), X(q)) = e_s(p, q; (1 - X(p))m(p) + X(p)\alpha, (1-X(q))m(q)+X(q)\alpha)$. For simplicity, let us write $\widetilde{e}_s(X(p), X(q))$ instead of $\widetilde{e}_s(p, q; X(p), X(q))$. Notice that $\widetilde{e}_s$ satisfies the regularity assumption [37]

$$\widetilde{e}_s(0, 0) + \widetilde{e}_s(1, 1) \leq \widetilde{e}_s(0, 1) + \widetilde{e}_s(1, 0). \tag{8}$$

Indeed

$$\widetilde{e}_s(0, 0) = |u(p + m(p)) - u(p + m(q))|,$$
$$\widetilde{e}_s(1, 1) = |u(p + \alpha) - u(p + \alpha)| = 0,$$
$$\widetilde{e}_s(0, 1) = |u(p + m(p)) - u(p + \alpha)|,$$
$$\widetilde{e}_s(1, 0) = |u(p + \alpha) - u(p + m(q))|,$$

and (8) follows from the triangular inequality for the Euclidean norm in $\mathbb{R}^3$. Thus, we can use the α-expansion algorithm as described in [37]. As proved in [38], if $e_s(p, q; \cdot, \cdot)$ is a metric, then the α expansion algorithm provides a solution whose energy is bounded by $c\mathcal{E}^*$, where $\mathcal{E}^*$ is the global minimum of the original energy functional and c is a universal constant that depends on the energy. But, notice that for each $(p, q) \in E$ $e_s(p, q; \cdot, \cdot)$ is symmetric and satisfies the triangle inequality, although it may happen that $e_s(p, q; \alpha, \beta) = 0$ with $\alpha \neq \beta$. Thus, we cannot directly apply the convergence result to the energy (7) based on $e_s(p, q; \cdot, \cdot)$. But we can apply it if we use instead

$$e_s^{TTV}(p, q; \alpha, \beta) = e_s(p, q; \alpha, \beta) + \epsilon \min(|\alpha - \beta|, k_2), \tag{9}$$

where $\epsilon, k_2 > 0$, $\alpha, \beta \in \mathcal{L}$ (in our experiments, $\epsilon = 20$, $k_2 = 5$). In this case, $e_s^{TTV}(p, q; \alpha, \beta) \neq 0$ if $\alpha \neq \beta$, it is symmetric in α, β, and, when restricted to α-expansions, it satisfies (8). This is sufficient for the previously mentioned results to hold [38]. As we have experimentally observed, using either e_s or e_s^{TTV} we obtain the same type of results.

Finally, let us observe that one can interpret the proposed energy in terms of a MRF [44].

Remark 3.1: As we mentioned above, in practice we take $\mathcal{N}_p$ as the 4-neighborhood of p. This is mainly motivated by computational reasons, since the number of edges of the graph is proportional to the size of this neighborhood. Thus, our choice represents a significant reduction of computational time (around 40% with respect to taking $\mathcal{N}_p$ as the 8-neighborhood of p), without compromising the quality of the results.

Remark 3.2: Thanks to the choice of the offset m as our optimization variable, the function $e_s(p, q; \cdot, \cdot)$ is symmetric (hence $e_s^{TTV}(p, q; \alpha, \beta)$ is a metric) and the convergence result mentioned above holds. The symmetry property does not hold if we use the correspondence map T as optimization variable.

B. Optimization of $\mathcal{E}$ using graph cuts at several scales

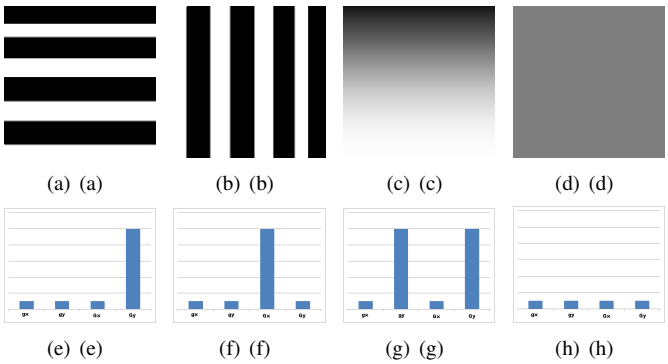
To further enforce global consistency and to speed up convergence, we can solve the energy optimization problem (1) using the graph cuts algorithm in a multiscale Gaussian pyramid. Each pyramid level decreases image resolution to half in each spatial dimension. The optimization starts at the coarsest level and the solution is propagated to finer levels for further refinement. This is a common strategy in the context for image inpainting (e.g. [27], [28], [43], [41]).

To fix notations we use L levels of resolution, the level k being associated to the sampling grid $\Pi_k := \{0, \ldots, 2^{N-k} - 1\}^2$, $k = 0$ and $k = L - 1$ corresponding to the highest and lowest resolution levels, respectively. Let m_k denote the offset map at level k.

At lowest resolution, the offset map could be found by searching in all positions of the hole's complement. But, in most cases, at low resolution the pixels surrounding the inpainting domain and the pixels in the hole have a similar pattern and the search range can be constrained by imposing a bound on the length of the vector $m_{L-1}(p)$. In practice we impose that the length of $m_{L-1}(p)$ is smaller than γw_h, where $\gamma > 0$ is a constant value and w_h is the maximum distance between the pixels in the hole and Ω^c (at that level of resolution). In our experiment, we take $\gamma = 2$.

After the offset map m_k at level k is obtained, we extend it to the previous level of resolution $k - 1$. For that we define $\overline{m}_{k-1}(2p) = 2m_k(p)$ for each $p \in \Pi_k$, and we interpolate it on $\Pi_{k-1} \setminus \Pi_k$ using a nearest neighbor interpolation. Then we refine the offset map at level $k - 1$ by minimizing the energy (1) at that level using as a set of labels $\mathcal{L}$ a neighborhood of $(0, 0)$ in $\mathbb{Z}^2$. The final offset map at level $k - 1$ is $m_{k-1}(p) \in \overline{m}_{k-1}(p) + \mathcal{L}$. To speed up computations we used $\mathcal{L} = \{(1, 0), (0, 1), (0, 0), (-1, 0), (0, -1)\}$. Due to the multiscale strategy, this refinement is sufficient in most cases, and increasing the neighborhood size does not improve the results. In our experiments we have tested this, up to a neighborhood of size 21×21.

The reasons for using a multiscale graph cuts algorithm are two-fold: the first one is obviously to reduce the computational cost, the graph labeling at the original image resolution being computationally intensive because of the large number of labels and pixels in Ω. The second is that the smoothness



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

(e) (e) (f) (f) (g) (g) (h) (h)

Fig. 2. Feature vector representation.

term, that uses at each scale the 4 nearest neighbors, can capture more spatial information at the lowest resolution level. This helps to avoid getting stuck into bad local minima. For the same reason, also the multi-scale strategy is used in the PatchMatch method [42].

In our experiments, the image size at the lowest resolution is around 80×80, and we use 2 to 5 levels for most images. During the image downscaling from the original size to the smallest one, most of the image information, specially the structure and texture of the regions, is lost. Thus, it is easy to find wrong offset labels for pixels at lowest resolution, and is difficult to correct this at higher resolution using a small search range.

In order to avoid this problem, we keep more structure and texture information at lowest resolution. For that, when working in the lowest resolution grid Π_{L-1}, we use a feature vector with 7 components, namely

$$U(p) = \{(u_i^{L-1}(p))_{i=1}^3, g_x(p), g_y(p), G_x(p), G_y(p)\},$$

where u^{L-1} denotes the image at lowest resolution, $p \in \Pi_{L-1}$, and the last four items are the gradient-related features

$$g_x(p) = \frac{1}{2^{2L}} \sum_{q \in \mathcal{V}_p} \nabla_x I(q), \quad g_y(p) = \frac{1}{2^{2L}} \sum_{q \in \mathcal{V}_p} \nabla_y I(q),$$

$$G_x(p) = \frac{1}{2^{2L}} \sum_{q \in \mathcal{V}_p} |\nabla_x I(q)|, \quad G_y(p) = \frac{1}{2^{2L}} \sum_{q \in \mathcal{V}_p} |\nabla_y I(q)|.$$

$I(q) = \frac{\sum_{i=1}^3 u_i(q)}{3}$ denotes the image intensity, and $\mathcal{V}_p$ is the $2^L \times 2^L$ square set of pixels of the highest resolution grid Π_0 that are represented by pixel $p \in \Pi_{L-1}$. We have denoted $\nabla_x I(q) = I(i+1, j) - I(j)$, $\nabla_y I(q) = I(i, j+1) - I(j)$, $q = (i, j) \in \Pi_0$ (these values being zero if q is on the boundary of the image domain, corresponding to a mirror symmetry of the original image). Thus, the gradient features at lowest resolution are just the average of gradient information computed at the highest resolution level. This type of features is also used in SURF (Speeded Up Robust Features) descriptor [46].

When working at the lowest resolution level, we redefine the energy $\mathcal{E}$ by replacing d_u in (4) and (6) by

$$\begin{aligned} d_U(p, p')^2 = &\textstyle\sum_{i=1}^3 (u_i^{L-1}(p) - u_i^{L-1}(p'))^2 \\ &+ (g_x(p) - g_x(p'))^2 + (g_y(p) - g_y(p'))^2 \\ &+ (G_x(p) - G_x(p'))^2 + (G_y(p) - G_y(p'))^2, \end{aligned} \tag{10}$$

where $p, p' \in \Pi_{L-1}$. The use of additional features at lower resolution levels is the main difference with other methods that also use a multiscale strategy, e.g. [27], [28], [43], [41].

Figure 2 shows the properties of the feature space representation for four distinctively different image-intensity patterns within a sub-region. Due to the smoothness of the downscaled image, the three different patterns shown in Figures 2(a), 2(b), and 2(c) will result in a pattern similar to the one in Figure 2(d) when they are represented at lowest resolution, the structure and texture information being lost. Notice that one cannot distinguish these four patterns using only information at the lowest resolution level, and the graph cuts algorithm easily finds a wrong offset map. However, the situation improves if we represent the pixel at lowest resolution using the proposed feature space, where the last four feature entries represent the nature of the underlying intensity pattern at the original image resolution. In Figure 2(a), which contains a square wave pattern in the y direction, the value of G_y is high, but all others are relatively low. If the intensity increases gradually in the y direction as in Figure 2(c), both values g_y and G_y are high. The proposed feature vector representation permits to eliminate the ambiguity represented in Figure 2 and improve the accuracy of the offset map.

As we have said before, the proposed features are similar to SURF descriptors [46]. Being based on the gradient, they tend to be more useful for differentiating between geometric structures (based on edges). But, although one could devise textures for which they are not discriminative, they represent also an improvement in this case, and we can use their performance in shape recognition tasks in favor of this assertion, acknowledging also the limitations (e.g. ambiguities) that they have in this context. As shown by our experiments, the proposed features, which combine color components and gradients from the original image, result into a distinctive representation for different structures and textures. Other feature vectors based on Gabor functions or wavelets could also be used.

In practice, we first build-up four gradient images $\nabla_x I, \nabla_y I, |\nabla_x I|, |\nabla_y I|$ at the original image size, and then downscale them to lowest resolution. Then we compute the values of g_x, g_y, G_x, G_y at lowest resolution.

C. On regularization terms

Robust regularization techniques, e.g. total variation [45], are frequently employed for ill-posed inverse problems in image processing. As we have already mentioned, the smoothness term can be thought of as a regularization term (by the intermediate of the image u). Since we are using graph cuts as optimization procedure, we can easily add other regularization terms, as in (9). We can also consider the penalization of large offsets.

We assume that if two patches in Ω^c have a similar distance to the target patch centered at $p \in \Omega$, the patch with the shortest offset has a higher probability to fit the target patch. As in [43], based on this assumption, large offsets are penalized using the cost

$$E_r = \sum_{p\in\Omega, p+m(p)\in\Omega^c} \frac{\beta_1}{1+e^{-k_2 m(p)^2}}, \tag{11}$$

where $\beta_1, k_2 > 0$ (in our experiments we have taken $\beta_1 = 200$ and $k_2 = 0.25$). Notice that this is a unary term that is added to the energy $\mathcal{E}$.

Using the penalization of large offsets and the regularization e_s^{TTV} in (9) we can obtain improvements for some images, as shown in Figure 3. However, in most of the tested images, we can achieve similar results without using both regularization terms.

IV. Experiments

In this section, we evaluate the proposed image inpainting algorithm on a variety of natural images. We start by describing the parameter setup used in our experiments. Then we display the results obtained with our algorithm in several cases of test, namely in scratch/text removal, error concealment, and object removal. We also compare our algorithm with some state-of-art inpainting methods.

A. Implementation Details

Let us specify the parameter choices that are needed to implement the proposed algorithm.

1) The Patch Size in the Data Term: The size of the patch Ψ in the data term helps to capture the local image characteristics around the boundary of the hole, and get a good continuation of the image structure and texture. We use the following practical rule: we fix the size of the patch for images at lowest resolution and we increase linearly the size when doubling the resolution in each dimension. In practice, the lowest resolution image size we consider is around 80×80 for which we use a patch of size 7×7. For a real number a, let us denote by $[a]$ is the greatest integer $\leq a$. Thus, given an image of size $M \times N$ (larger than 80×80) a good practical rule is to use patches of size $w_p \times w_p$, where $w_p = 7 + 2\left[\frac{3}{2}k\right]$, and k satisfies $4^{-k}M \times N = 80 \times 80$. We have used the expression $2\left[\frac{3}{2}k\right]$ instead of $3k$ to have patches of odd dimensions.

As an example, when the image size is 1000×1000, the patch size is 17×17. Figure 4 illustrates how the choice of patch size affects the quality of the inpainting result. Smaller patch sizes allow more matching possibilities without capturing the image structure, and this does not always permit to obtain a good reconstruction. Up to a certain limit, a bigger patch size can capture the texture characteristics better, however with less matching possibilities. From the experiments we conclude that our adaptive patch size rule permits to obtain satisfactory results.

2) The Number of Multi-scale Levels: We set up the size of the image at lowest resolution, say $a \times b$. The number of levels of the multi-scale algorithm is set using the following rule: we consider one level of resolution for images of size $a \times b$ and we add an additional level when doubling the resolution in each dimension. As we said above, in practice, we take the size of the lowest resolution images as 80×80. Thus, given an image of size $M \times N$ (larger than 80×80) the number of levels is set to $L = [k] + 1$, where $[k]$ is given as above by $4^{-k}M \times N = 80 \times 80$.

As an illustration, when the size of the original image is 800×600, as in Figure 5a, we have $L = 4$. In the first row of Figure 5 we show the results obtained for different number of levels, the processing times are also reported within parenthesis. As one can see, the computation time decreases significantly with the number of levels. When using 5 levels, as in Figure 5(d), we have a lower computational complexity, although some block artifacts appear in the inpainted image. For comparison, we display in Figure 5(e) the result obtained using Komodakis et al.'s method [27]. Considering the computational cost and the quality of the results obtained, in the experiments that follow the number of levels is adaptively set to $L = [k] + 1$.

In the second row of Figure 5 we show the results obtained using different number of resolution levels for a smaller image, of size 200×200. In this case, the value of L given by the above rule is $L = 2$.

3) Search Range: For computational efficiency, we can also set a restricted search range around the hole, specifying it by a bound on the maximum offset. The bound is given by γw_h, where γ is a constant value, and w_h is the maximum distance between the pixels in the hole and its complement, that is $w_h = \max_{p\in\Omega} d(p, \Omega^c)$. We show in Figure 6 how the search range affects the inpainting results. We see that the best results are obtained for $\gamma = 2$ in Figure 6(c). When the search range is too small, e.g. $\gamma = 1$ in Figure 6(b), one does not capture enough similar patterns. The result obtained when we search in the full hole's complement, displayed in Figure 6(d), is a little worse than the one in Figure 6(c). We observe that, in Figure 6(c), appropriate pixels that exist near the inpainting region are preferentially selected for matching. Let us also point that, a larger search range will increase the computation cost significantly. Thus, the strategy of restricting the search range may be useful if the available computation time imposes us this choice.

B. Experimental results

1) Benefits of Using the Proposed Feature Representation: In order to justify the improvement of inpainting performance due to the proposed feature representation, in Figure 7 we compare the results obtained when we consider three different types of features to represent the pixel information. In the first one, see Figure 7(b), pixels (at each scale) are represented using only the color vector. In the second, see Figure 7(c), they are described at each scale using the color and gradient vectors. In the third, see Figure 7(d), we use our feature representation. In this case, the recovered texture and structure are more plausible. This may be explained since the feature representation helps to eliminate the ambiguities at low resolution, and improves the accuracy of the offset map.

2) Scratch and Text Removal: Let us evaluate the performance of the proposed inpainting algorithm for scratch

(a) Image with mask (b) No regularization (c) With regularization (9) (d) With regularization (11) (e) With both regularizations

Fig. 3. Comparison of different regularization terms.

(a) Image with mask (b) $w_p = 3$ (c) $w_p = 5$ (d) $w_p = 11$ (e) Adaptive

Fig. 4. The influence of the patch size on the results.

(a) Image with mask (b) With 3 levels (828s) (c) With 4 levels (84s) (d) With 5 levels (39s) (e) Komodakis' (998s)

(f) Image with mask (g) With 1 level (83s) (h) With 2 levels (28s) (i) With 3 levels (12s) (j) Komodakis' (37s)

Fig. 5. The influence of the number of levels of the multiscale algorithm.

(a) Image with mask (b) $\gamma = 1$ (13s) (c) $\gamma = 2$ (31s) (d) Full image (93s)

Fig. 6. Influence of the search range.

(a) Image with mask (b) Color (c) Color+Gradient (d) Feature representation

Fig. 7. Influence of the Feature Representation.

and text removal by comparing it with the geometry-based inpainting algorithm in [13] and the exemplar-based one in [19]. The original images (which can be found in http://r0k.us/graphics/kodak/) and the corresponding results are shown in Figure 8. The first column shows the images with the scratch or text mask, the second, third and fourth columns display the results obtained using the methods in [13], [19], and ours, respectively. For qualitative comparison, the peak signal-to-noise ratio (PSNR) (indeed the average of the PSNRs of the color channels (R, G, B)), and the processing time are also presented within parenthesis.

As one can see in Figure 8, Bornemann et al.'s method [13] introduces some blur, but since the inpainting area is very thin, visually, the result is comparable with that of Criminisi et al.'s method [19]. The result obtained with the proposed method improves the above methods on both objective and subjective criteria.

3) Block Recovery for Spatial Error Concealment: Error concealment is widely used as a post-processing step to alleviate the negative effect of erroneous blocks. In this experiment, we consider the case of blocks of size 16x16 which are of common use for image and video encoding. Figure 9 shows the results obtained using the methods in [13], [19], and ours. For qualitative comparison, the PSNR and the processing time are also presented within parenthesis. As it can be seen, the proposed method shows better performance for block recovery than those in [13] and [19].

4) Object Removal: We applied the proposed algorithm to inpaint the missing region after object removal and we display some results comparing it with other algorithms. Some of the images used (see Figures 10 and 11) were taken from the image benchmark [43] available at http://yokoya.naist.jp/research/inpainting/. Some others (see Figures 12 and 13) were taken from [35], available at http://graphics.cs.cmu.edu/projects/scene-completion/. In Figure 10, we compare our algorithm with Bornemann et al. [13], Criminisi et al. [19], Bugeau et al. [33], Komodakis et al. [27], (using the implementation of the code in http://lafarren.com/image-completer/), Pritch et al. [41], and Barnes et al. [42] methods. These images are challenging because the inpainting area is large and contains both structure and texture information. The geometry-based method [13] introduces blurring artifacts. Although the exemplar-based methods [19], [33], [27], [41], [42] do not suffer from blurring, they do not recover the structure information perfectly, may be because the edges arriving to the hole's boundary are not very sharp. The proposed method outperforms the others, giving more plausible results. Figure 11 shows some other examples. As these results show, edges and textures are well recovered using the proposed method.

Figures 12 and 13 show some results obtained using images taken from [35]. They are larger than those in the two previous experiments and the holes are also larger. Figure 12 shows the comparison with three other methods [19], [27], [35]. As one can see, our method is able to recover the large area of the grassland and trees. Hays et al. method [35] also reconstructed a plausible image with different visual effects, the reason being that their method uses also other images to copy from. Other results that permit to compare the proposed method with that of Hays et al. [35] are displayed in Figure 13. The method in [35] performs the image completion using the information from a huge database of photographs gathered from the Web. We achieve a comparable result using a single image.

5) Computational Cost: We have implemented the proposed method using MATLAB programming language on Intel 2.67 GHz CPU, without optimization. The method is very efficient especially for large images. For instance, our method takes 56 seconds for Figure 8, compared with the 724 seconds

(a) Image with mask (b) Bornemann's[13] (39.32dB,27s) (c) Criminisi's[19] (39.30dB,724s) (d) Ours (40.47dB, 56s)

(e) Image with mask (f) Bornemann's[13] (34.45dB,31s) (g) Criminisi's[19] (33.77dB,807s) (h) Ours (35.18dB,74s)

Fig. 8. Results for scratch and text inpainting.

(a) Image with mask (b) Bornemann's[13] (37.65dB,34s) (c) Criminisi's[19] (36.27dB,374s) (d) Ours (39.29dB,71s)

Fig. 9. Results for block recovery.

needed by Criminisi et al.'s method [19]. For Figure 12, Criminisi et al.'s [19] and Komodakis et al.'s [27] methods take 524 and 429 seconds to fill-in the holes, respectively. Our method only needs 78 seconds.

V. Conclusions

A novel formulation of exemplar-based inpainting is proposed based in the minimization of an energy functional written in terms of the offset map. The proposed energy function jointly considers the continuity of the reconstructed image on the boundary of the inpainting domain and a visually coherent reconstruction inside the hole. We used a multiscale graph cuts algorithm to efficiently solve the energy optimization problem. Moreover, the use of a feature vector representation permits to compensate the loss of information at low resolution levels. Our experiments show the good performance of the proposed algorithm when compared with other recent algorithms.

Acknowledgment

This work was partially funded by the EU project "2020 3D Media: Spatial Sound and Vision" under FP7-ICT. Y. Liu acknowledges partial support from the Torres Quevedo Program from the Ministry of Science and Innovation in Spain (MICINN), co-funded by the European Social Fund (ESF). V. Caselles also acknowledges partial support by MICINN project, reference MTM2009-08171, by GRC reference 2009 SGR 773 and by "ICREA Acadèmia" prize for excellence in research funded both by the Generalitat de Catalunya.

References

[1] Z. Tauber, Z-N Li, M. Drew, "Review and preview: disocclusion by inpainting for image-based rendering," *IEEE T-SMC-Part C,* vol.37,no.4, pp.527-540, July 2007.

[2] S. Zinger, L. Do, P. H. N. de With. "Free-viewpoint depth image based rendering," *J. Visual Communication and Image Representation*, pp.533-541, 2010.

[3] D. Liu, X. Sun, F. Wu, S. Li, and Y-Q. Zhang, "Image compression with edge-based inpainting," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 17, no. 10, pp. 1273-1287, Oct. 2007.

[4] Z. Xiong, X. Sun, and F. Wu, "Block-based image compression with parameter-assistant inpainting," *IEEE Trans. Image Processing*, vol. 19, no. 6, pp. 1651-1657, 2010.

[5] Y-Q. Liu, J. Wang and H-H. Zhang, "Adaptive Patch-based Inpainting for Image Block Recovery," in *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)*, pp.52-59, 2010.

[6] S. Masnou, "Disocclusion: a variational approach using level lines," *IEEE Trans. on Image Processing*, 11(2), 68-76,2002.

[7] M. Nitzberg, D. Mumford, and T. Shiota, "Filtering, segmentation, and depth," *Lecture notes in computer science*, vol. 662, Springer, 1993.

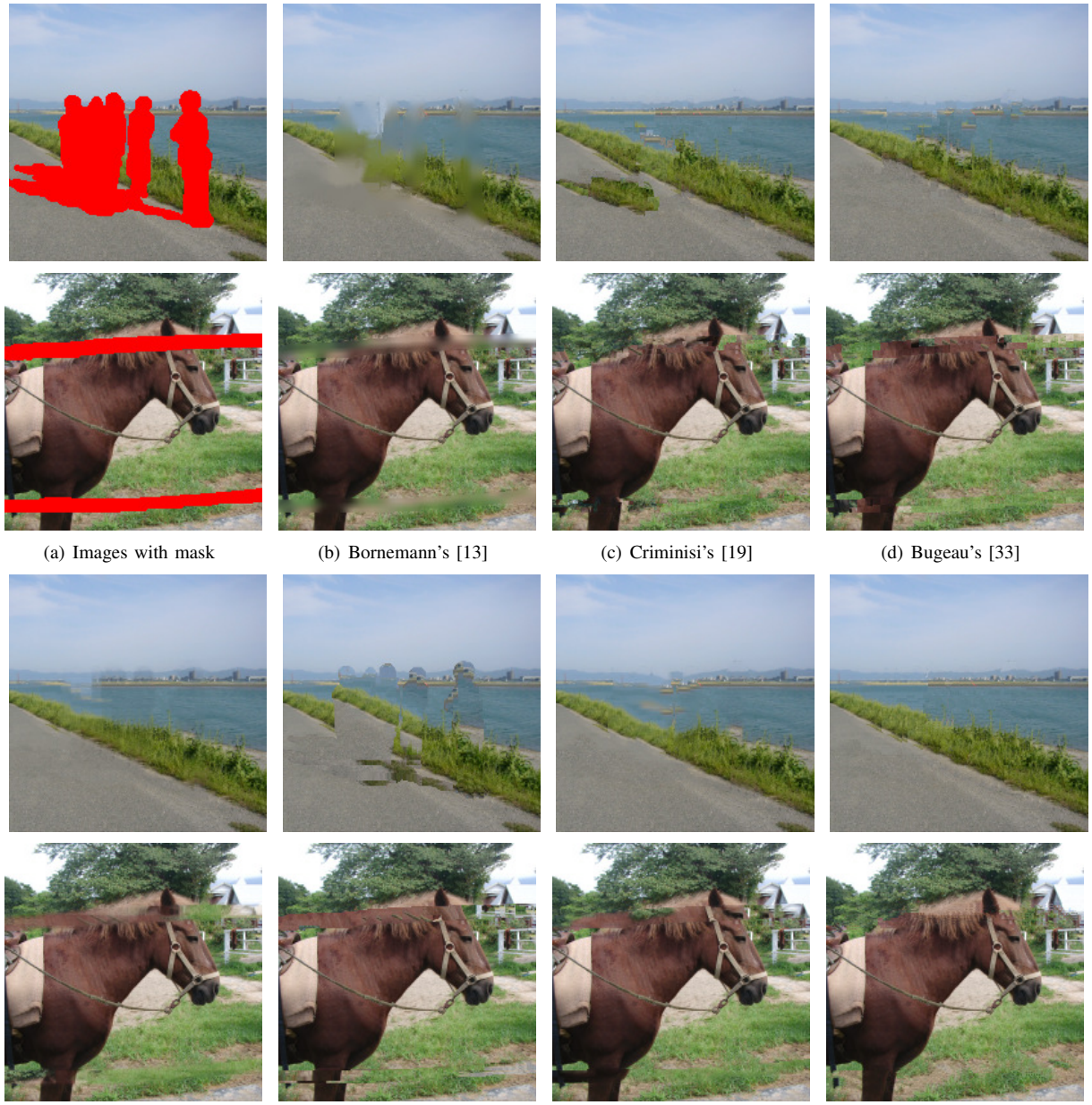
[8] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. "Image inpainting," in *ACM SIGGRAPH*, 2000.

[9] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Processing*, vol.10, no.8, pp.1200-1211, 2001.

[10] T. F. Chan and J. Shen. "Non-texture inpainting by curvature driven diffusion," *J. Visual Comm. Image Rep.*, vol.12,no.4, pp.436-449,2001.

[11] T. F. Chan, S. H. Kang, and J. Shen. "Euler's elastica and curvature based inpaintings," *SIAM J. App. Math.*, vol.63, no.2, pp.564-592, 2002.

[12] D. Tschumperle. "Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's," *International Journal of Computer Vision*, vol. 68, no.1,pp.65-82, 2006.



(a) Images with mask (b) Bornemann's [13] (c) Criminisi's [19] (d) Bugeau's [33]

(e) Komodakis' [27] (f) Pritch' [41] (g) Barnes' [42] (h) Ours

Fig. 10. Results for object removal.

[13] F. Bornemann and T. Marz, "Fast image inpainting based on coherence transport," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 3, pp. 259-278, 2007.

[14] A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics," in *Proc. International Conference on Computer Vision*, pp. 305-313,2003.

[15] P. Perez, M. Gangnet, and A. Blake, "PatchWorks: Example-based region tiling for image editing," *Technical report, Microsoft Research*, 2004.

[16] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol.82, no.2, pp.205-229, 2009.

[17] A. Efros and T. Leung. "Texture synthesis by non-parametric sampling," in *Proc. ICCV*, 1999.

[18] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, pp. 882-889, 2003.

[19] A. Criminisi, P. Pérez, and K. Toyama. "Region filling and object removal by exemplar-based inpainting," *IEEE Trans. on Image Processing*, vol.13,no.9, pp.1200–1212, 2004.

[20] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," *ACM Trans. on Graphics. Special issue: Proc. of ACM SIGGRAPH*, vol. 22, no. 3, pp. 303–12, 2003.

[21] Z. Xu and J. Sun. "Image inpainting by patch propagation using patch sparsity," *IEEE Transactions on Image Processing*, vol.19, no.5, pp.1153-1165,2010.

[22] J. Sun, L. Yuan, J. Jia, and H. Shum. "Image completion with structure propagation," in *Proc. SIGGRAPH*, 2005.

[23] J. Jia and C. Tang. "Inference of Segmented Color and Texture Description by Tensor Voting," *IEEE Trans. Pat. Anal. and Mach. Intell.* vol. 26, no.6, pp.771-786,2004.

[24] P. Arias, V. Caselles, and G. Sapiro, "A variational framework for non-local image inpainting", *Proceedings EMMCVPR, Lecture Notes in Computer Science*, pp. 345–58. Springer, 2009.

[25] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, "A Variational Framework for Exemplar-Based Image Inpainting", *International Journal of Computer Vision*, pp.319-347, 2011.

[26] P. Arias, V. Caselles, and G. Facciolo, "Analysis of a Variational Framework for Exemplar-Based Image Inpainting", *Multiscale Modeling*

Fig. 11. Results for object removal. The first and third rows display the original images with their holes, the second and forth rows show the results obtained using the proposed method.

(a) Image with mask (b) Criminisi's[19] (524s) (c) Komodakis'[27] (429s) (d) Hays'[35] (e) Ours (78s)

Fig. 12. Results for object removal using images from [35].

(a) Image with mask (b) Hays' [35] (c) Ours

Fig. 13. Results for object removal using images from [35].

and Simulation, vol.10, no.2, 2012.

[27] N. Komodakis and G. Tziritas. "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *IEEE Trans. Image Processing*, 16(11):2649-2661, 2007.

[28] Y. Wexler, E. Shechtman, and M. Irani. "Space-time completion of video," *IEEE Trans. Pat. Anal. and Mach. Intell.*, vol.29, no.3, pp.463-476, 2007.

[29] Y. Yang, Y. Zhu, and Q. Peng. "Image completion using structural priority belief propagation," in *Proc. ACM Int. Conf. on Multimedia*, pp.717-720, 2009.

[30] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," in *Proc. SIGGRAPH*, 2005, pp.795-802.

[31] L. Demanet, B. Song, and T. Chan. "Image inpainting by correspondence maps: a deterministic approach," in *Proc. VLSM*, 2003.

[32] J. Aujol, S. Ladjal, and S. Masnou, "Exemplar-Based Inpainting from a Variational Point of View", *SIAM J. Math. Analysis*, pp.1246-1285,2010.

[33] A. Bugeau, M. Bertalmo, V. Caselles, and G. Sapiro, "A Comprehensive Framework for Image Inpainting", *IEEE Transactions on Image Processing*, 2010, pp.2634-2645.

[34] G. Peyre. "Manifold models for signals and images," *Comp.Vis. and Im. Unders.* Vol.113, no.2, pp.249-260, 2009.

[35] J. Hays and A. A. Efros, "Scene Completion Using Millions of Photographs," *ACM Transactions on Graphics*, Vo1. 51, Issue 10, pp. 87-94, 2007.

[36] O. Whyte, J. Sivic, and A. Zisserman. "Get out of my picture! internet-based inpainting," *In Proc. British Machine Vision Conference (BMVC)*, 2009.

[37] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimizedvia Graph Cuts?," *IEEE Trans. Pat. Anal. and Mach. Intell.*, vol. 26, no. 2, pp. 147-159, 2004.

[38] Y. Boykov and V. Kolmogorov. "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pat. Anal. and Mach. Intell.*, vol.23, no.11, pp.1–18, 2003.

[39] Y. Boykov and V. Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pat. Anal. and Mach. Intell.*, vol.26, no.9, pp.1124–1137, 2004.

[40] A. Agarwala, M. Dontcheva, M. Agrawala, S.M. Drucker, A. Colburn, B. Curless, D. Salesin, and M.F. Cohen, "Interactive digital photomontage", *ACM Trans. Graph.*, vol.23,no.3,pp.294-302, 2004.

[41] Y. Pritch, E. Kav-Venaki, and S. Peleg, "Shift-Map Image Editing," in *Proc. ICCV*, Kyoto, Sept. 2009, pp. 151-158.

[42] C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," in *Proc. SIGGRAPH*, pp. 1-11, 2009.

[43] N. Kawai, T. Sato, N. Yokoya, "Image Inpainting Considering Brightness Change and Spatial Locality of Textures and Its Evaluation," In *Proc. PSIVT* 2009, LNCS 5414, pp.271-282.

[44] P. Brémaud, "Markov chains: Gibbs fields, Monte Carlo simulation, and queues," Springer Verlag 1999.

[45] L.I. Rudin, S. Osher, E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, 1992.

[46] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool , "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346–359, 2008