Context-aware patch-based image inpainting using Markov random field modelling

Tijana Ružić and Aleksandra Pižurica *Member, IEEE*

Abstract—In this paper, we first introduce a general approach for context-aware patch-based image inpainting, where textural descriptors are used to guide and accelerate the search for well-matching (candidate) patches. A novel top-down splitting procedure divides the image into variable size blocks according to their context, constraining thereby the search for candidate patches to non-local image regions with matching context. This approach can be employed to improve the speed and performance of virtually any (patch-based) inpainting method. We apply this approach to the so-called global image inpainting with the Markov random field (MRF) prior, where MRF encodes a priori knowledge about consistency of neighbouring image patches. We solve the resulting optimization problem with an efficient low-complexity inference method. Experimental results demonstrate the potential of the proposed approach in inpainting applications like scratch, text and object removal. Improvement and significant acceleration of a related global MRF-based inpainting method is also evident.

Index Terms—inpainting, patch-based, Gabor filtering, texture features, context-aware

I. INTRODUCTION

Image inpainting, or image completion, is an image processing task of filling in the missing region in an image in a visually plausible way. Applications include image restoration (e.g., scratch or text removal), image coding and transmission (recovery of missing blocks), photo-editing (object removal), virtual restoration of digitized paintings (crack removal), etc. In literature, two categories of image inpainting approaches can be distinguished: diffusion- and patch-based.

Diffusion-based methods [1]–[4] fill in the missing region (the "hole") by smoothly propagating image content from the boundary to the interior of the missing region. The problem of propagating linear structures, e.g., object lines and boundaries, that are interrupted by the hole, is then often formulated in terms of solving partial differential equations. Although these approaches yield good results when inpainting long thin regions, they experience difficulties in replicating texture, which

is largely due to their local nature (using only information in the immediate surrounding of the missing pixel).

Patch-based methods fill in the missing region patch-by-patch by searching for well-matching replacement patches (i.e., candidate patches) in the undamaged part of the image and copying them to corresponding locations. While these approaches share some ideas with patch-based texture synthesis [5], [6], they focus additionally on structure propagation either by defining the filling order [7]–[11], using human intervention [12] or decomposing the image into structure and texture components [13]–[15]. Compared to diffusion-based methods, patch-based methods typically produce better results, especially when inpainting larger holes.

Patch-based methods can be categorized into "greedy" [7], [9], [16]–[20], multiple candidate [10], [11], [21]–[23] and global [8], [12], [24]–[28]. The "greedy" ones choose only one best match for each patch to be filled, called the target patch, based on its known pixels. This is achieved in an iterative process that gradually completes the missing region. Multiple candidate methods infer the missing region using weighted average [11], [21] or a sparse combination [10] of multiple candidate patches at each location. Finally, global methods define inpainting as a global optimization problem. Multiple candidates are here typically called *labels*. A label is chosen for each position so that the whole set of labels (at all positions) minimizes a global optimization function. Many global methods [8], [24], [26]–[28] model global image context with a Markov random field (MRF). In [8], the objective function is optimized with a smart algorithm based on belief propagation, called *priority* belief propagation (p-BP), which discards unnecessary labels by visiting them in some meaningful order. Although such label pruning significantly reduces the number of labels, the method is still very complex and runs into difficulties when applied on large images. Another global method based on spatial coherency was proposed in [25].

A crucial aspect of patch-based methods is the search for candidate patches. Solutions to avoid the time consuming exhaustive search include confining the search to a local window [17], directional search [9], [11], search along user-specified curves [12], and utilizing already existing segmentation of the image [15], [20]. PatchMatch [29], a fast patch search method, is employed together with the global method from [25] within Adobe Photoshop's CS5 *Content Aware Fill*. A very recent, advanced method from [30] limits the candidate set by analysing the statistics of patch offsets, but then it treats inpainting as a photomontage problem, where shifted images are combined according to these offsets to yield the inpainted

T. Ružić and A. Pižurica are with the Department of Telecommunications and Information Processing (TELIN-IPI-iMinds), Ghent University, 9000 Ghent, Belgium (e-mail: tijana.ruzic@telin.ugent.be; aleksandra.pizurica@telin.ugent.be).

image. The authors remarked that this method may fail when the desired offsets do not form a dominant statistics, i.e., there is insufficient number of similar patches.

In this paper, we propose a novel *context-aware* global MRF-based inpainting method. The main idea is to employ contextual (textural) descriptors to guide and improve the inpainting process. Two most important contributions are (i) a novel context-aware patch selection strategy and (ii) an efficient inference method for global MRF-based inpainting. The first contribution is not limited to global inpainting and it aims at improving and accelerating the search for candidate patches in patch-based methods in general. Our framework is general also in the sense that it allows the use of arbitrary contextual descriptors, e.g., those used in image retrieval [31], scene classification [32], etc. In our practical method, we choose to use normalized texton histograms computed from Gabor filter responses as contextual descriptors. Similar texton histograms (computed from responses of different filters) were previously used for image segmentation [33], [34], texture classification [35] and image retrieval [36]. However, to our knowledge, the use of textons or texton histograms for image inpainting has never been explored before. We shall consider two different strategies for dividing the image into regions based on the context: a simple division into fixed-size square non-overlapping blocks and a more sophisticated division into blocks of adaptive sizes. For the latter strategy, we propose a novel *top-down splitting procedure*, which is also based on contextual descriptors.

The second main contribution of this paper is specific to MRF-based inpainting. We introduce a novel optimization approach, which builds upon our recent inference method [37] to make it suitable for MRF-based inpainting with huge number of labels. Compared to the related method from [8], this approach is faster and consumes less memory, allowing processing of larger images. Comparative results with other related inpainting methods demonstrate potentials of the proposed method for scratch or text removal and object removal.

Some preliminary ideas and parts of this work were reported in conference papers [19], [28]. In this paper, those preliminary ideas evolved into a solid framework resulting from more elaborate analysis and validation and from improvements on various aspects of contextual descriptors, block division strategy and optimization approach.

The paper is organized as follows. The proposed general context-aware approach is described in Section II, where Section II-A explains the proposed context-aware patch selection, Section II-B introduces a novel algorithm for context-based image division into adaptive-size blocks, Section II-C discusses alternative choices for context representation, while Section II-D deals with the choice of contextual descriptors. Section III presents the complete inpainting method, where the proposed context-aware approach is employed together with an efficient inference method and with novelties in priority definition and label pruning. Section IV describes the experiments and the results. The paper is concluded with Section V.
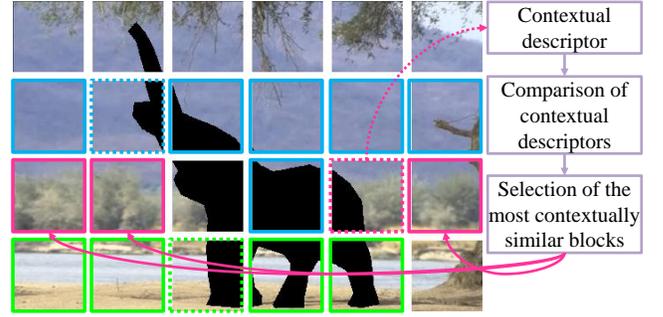

Fig. 1: An illustration of the proposed context-aware approach for inpainting.

II. CONTEXT-AWARE APPROACH FOR INPAINTING

In this section, we introduce a general context-aware approach, which can be used with any inpainting algorithm. The main idea is to guide the search for patches to the areas of interest based on contextual features. Fig. 1 illustrates this concept: contextual descriptors are assigned to image blocks, which can be of fixed size (like in Fig. 1) or adaptive. For the missing region within a given block, well-matching candidate patches will be found in the contextually similar blocks. The benefit is twofold: the search for well-matching patches is accelerated and the inpainting result is improved.

A. Context-aware patch selection

Let the input image I be defined on a lattice S. Pixel positions on this lattice are represented by a single index $p \in S$, assuming raster scan ordering. Let $\Omega \subset S$ denote the region to be filled (*target* region), and $\Phi \subset S$ denote the known part of the image (*source* region), where $\Omega \cup \Phi = S$. Suppose we divide the image into $M \times N$ square non-overlapping blocks, like in Fig. 1 (an extension to adaptive blocks is described in Section II-B). We denote by B_l an image block centred at the position l. The central positions of all the blocks form a set λ, which is determined, together with the block sizes, by the particular block division scheme.

The idea of our context-aware approach is to constrain the source region for target patches from a block B_l to a region $\Phi^{(l)} \subset \Phi$ with the context well matching that of B_l. We assign to each block B_l a contextual descriptor $\mathbf{c}^{(l)}$, which, in general, is some feature vector that characterizes spatial content and textures within the block. Let us define a *measure of contextual dissimilarity* $\bar{H}^{(l,m)}$ as

$$\bar{H}^{(l,m)} = d(\mathbf{c}^{(l)}, \mathbf{c}^{(m)}), \qquad (1)$$

where $d(\mathbf{c}^{(l)}, \mathbf{c}^{(m)})$ is some distance measure between contextual descriptors $\mathbf{c}^{(l)}$ and $\mathbf{c}^{(m)}$. The more similar context of the blocks B_l and B_m, the lower $\bar{H}^{(l,m)}$. Let $\Sigma^{(l)}$ denote the set of positions of the blocks that are contextually similar to B_l. In general, we can write

$$\Sigma^{(l)} = \left\{ m | \bar{H}^{(l,m)} \leq \tau \wedge m \in \lambda \right\}, \qquad (2)$$

Algorithm 1 Context-aware patch selection

 1: **for all** B_l such that $l \in \lambda$ and $B_l \cap \Omega \neq \emptyset$ **do**
 2:     set $\Phi^{(l)} = \emptyset$
 3:     **if** B_l is reliable **then**
 4:         compute $\bar{H}^{(l,m)}, \forall m \in \lambda$ (Eq. (1))
 5:         define new source region $\Phi^{(l)}$ (Eqs. (2) and (3))
 6:     **else**
 7:         **for all** neighbouring blocks B_n **do**
 8:             repeat steps 2-5
 9:             add $\Phi^{(n)}$ to $\Phi^{(l)}$
10:         **end for**
11:     **end if**
12: **end for**

---

where τ is some block similarity threshold. The *constrained source region* $\Phi^{(l)}$ is then a union of *known* parts of blocks indexed in $\Sigma^{(l)}$:

$$\Phi^{(l)} = \cup_{m \in \Sigma^{(l)}} (B_m \cap \Phi). \quad (3)$$

Note that the current block itself is always a part of $\Phi^{(l)}$. Some examples of block matching for fixed-size blocks are shown in Figs. 1 and 2(a). Current blocks are denoted by a dashed-line border, and their contextually similar blocks by a solid-line border of matching color.

In practice, some blocks may be dominated by missing pixels (e.g., two central blocks in the third row of Fig. 1). We consider a block with less than half known pixels, as *unreliable* and we do not rely on its contextual descriptor, but we rather determine its constrained source region based on the neighbouring blocks.[1] The proposed context-aware approach is summarized as pseudo-code in Algorithm 1. It applies also to the adaptive blocks, just that the set λ is determined by the adaptive block division scheme (described next).

B. Division into blocks of adaptive sizes

In most natural images some image areas call for finer division than the others (see the example in Fig. 2). Moreover, the optimal size of blocks can differ from one image to another. We introduce a simple *top-down splitting procedure* that automatically divides the image into blocks of adaptive sizes depending on the "homogeneity" of their texture.

Fig. 3 illustrates the proposed splitting procedure for one image block B_l. We need to favour that the splits in horizontal and vertical directions alternate through levels in order to prevent splitting along one direction only. Therefore, we assign each block a *directional flag* $\delta^{(l)} \in \{h, v\}$. This flag determines the direction, horizontal (h) or vertical (v), along which the evaluation of the block's homogeneity will have the priority. Let $B_{l_{1d}}$ and $B_{l_{2d}}$ denote two sub-blocks of B_l along direction d (see Fig. 4). We measure the inhomogeneity of the block B_l along direction d as the contextual dissimilarity from Eq. (1):

---

[1]In practical implementation, the constrained source region of an *unreliable* block consists of the known part of the block itself, the constrained source regions of all of its neighbouring reliable blocks and the known parts of its neighbouring unreliable blocks themselves.
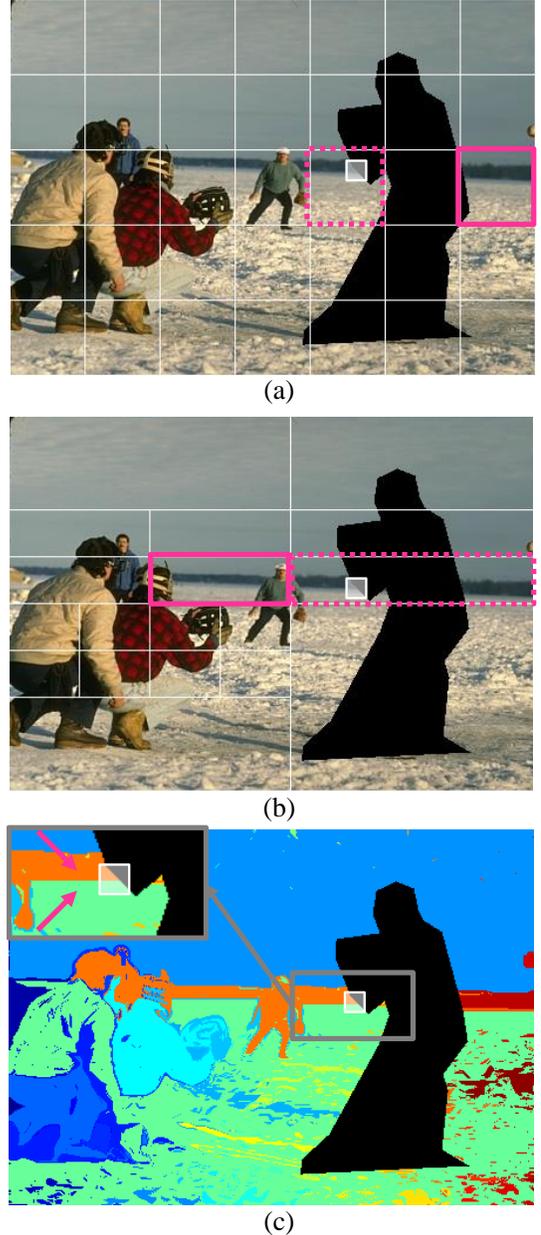

(a)

(b)

(c)

Fig. 2: (a) and (b) Image division into 5×7 blocks of fixed size and into blocks of adaptive sizes, respectively. The constrained source region for the current target patch (marked in white) consists of the current block (in dashed-line pink rectangle) and its contextually similar block(s) (in solid-line pink rectangle). (c) Result of the graph-based image segmentation technique [38]. The constrained source region for the current target patch will consist of the union of orange and green segments (indicated by pink arrows in the zoomed-in part).

$$\bar{H}_d^{(l)} = \bar{H}^{(l_{1d}, l_{2d})}, \quad d = h, v. \quad (4)$$

Splitting along direction d is allowed only if $\bar{H}_d^{(l)}$ exceeds a given block similarity threshold τ. $\bar{H}_h^{(l)}$ and $\bar{H}_v^{(l)}$ are evaluated sequentially, in the order that depends on the directional flag $\delta^{(l)}$. If $\delta^{(l)} = h$, we first evaluate $\bar{H}_h^{(l)}$, and then *only* if $\bar{H}_h^{(l)} \leq$
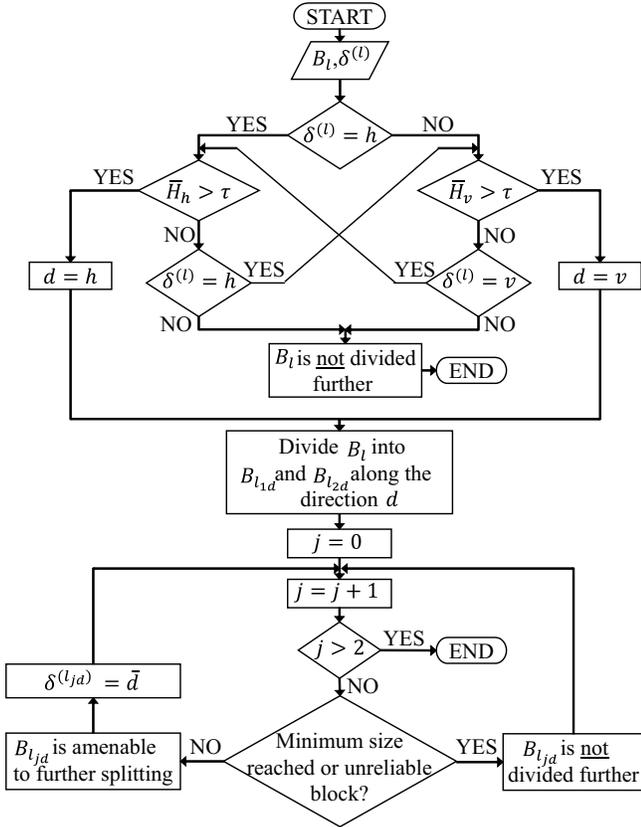
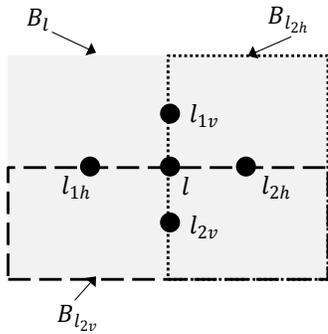Fig. 3: Block diagram of the proposed top-down splitting procedure (see text for notations).

Fig. 4: A division of a block into sub-blocks.

τ, we evaluate $\bar{H}_v^{(l)}$. The order is reversed when $\delta^{(l)} = v$. Hence, the block can be split along one direction only.

If neither $\bar{H}_h^{(l)}$ and $\bar{H}_v^{(l)}$ exceeds τ, B_l is not split any further. Otherwise, it is split along the direction d into two new sub-blocks $B_{l_{jd}}$, $j = 1, 2$, both of which can be declared as amenable to further splitting or not, depending on their size and reliability (fraction of known pixels). The directional flag is always set to the direction opposite of the splitting direction of the parent block. Hence, in Fig. 3 $\delta^{(l_{jd})} = \bar{d}$, where $\bar{d}$ is the complement of d.

We initialize the block splitting procedure by dividing the image coarsely into four (approximately) equal blocks and assigning them directional flags based on their longer dimension. The output is the block set $\{B_l | l \in \lambda\}$, representing an adaptive image division as illustrated in Fig. 2(b). This figure also shows an example of block matching result for adaptive-size blocks obtained by Algorithm 1.

C. Discussion on context representation

In this paper, we assign contextual descriptors to image blocks of fixed or adaptive size. An alternative could be to divide the image into regions using some image segmentation technique or user input, as explored in [12], [15], [20]. Then, for the missing region within a given segment (region), well-matching candidate patches can be found within that segment. A difficulty with this approach is that the segment boundaries coincide with image structures (see Fig. 2(c)), whose correct propagation inside the missing region is crucial for the quality of the inpainting result. For example, the target patch marked in white in Fig. 2(c) contains a boundary between two segments (see the zoomed-in part). The search for its candidate patches should thus be confined to the union of these two segments (all green and orange areas), which excludes the well-matching patches that can be found on the opposite side of the missing region, because they belong to a different segment. Solution to this problem is to first connect the curves representing the segment boundaries, which leads to better inpainting results [20].

Our proposed block-based approach (see Figs. 2(a) and (b)) represents a very simple, yet effective alternative to using image segmentation for context representation. Image segmentation results depend on the chosen segmentation method and its parameters, which may influence the inpainting quality and thus, make a fair comparison difficult. Moreover, in our block-based approach, there is no need for curve connection because a block and its contextually similar blocks can span over multiple segments (see the marked blocks in Figs. 2(a) and (b)). Therefore, for a target patch containing image structure (e.g., the white patch in Fig. 2), well-matching patches can be found in contextually similar blocks, which ensures correct structure propagation inside the missing region.

D. Choice of contextual descriptors

So far, we considered a general formulation of contextual descriptors as some characterization of spatial content and textures within blocks. There are many ways to extract texture features, e.g., computing co-occurrence matrices [39], using local binary patterns [40], estimating parameters of MRF models [41], multi-channel filtering [31], [33], etc.

For our problem, multi-channel filtering is well suited, both in terms of performance and relatively simple implementation. Let G_n denote one filter from the bank of linear spatial filters at various orientations and scales, where $n = 1, \ldots, N_f$ and N_f is the total number of filters. After convolving the image I with such bank of filters, each pixel p is assigned an N_f-dimensional vector of filter outputs, $\mathbf{F}(p) = (F_1(p), \ldots, F_{N_f}(p))$, where $F_n(p) = (I * G_n)(p)$. This vector characterizes the image patch centred at that pixel.

Often, dimensionality reduction is applied to the resulting vector. For example, in [32], filter outputs were averaged

within square non-overlapping blocks to obtain coarse description of textures called a *gist*, which was employed for various computer vision tasks [32], [42], [43], and also in our previous inpainting work in [19], [28].

We observed, however, that an alternative approach, using the so-called texton histograms similar to those from [33]–[36], yields similar or even slightly better results in our setting, while requiring less parameters. Therefore, in our practical method, we implement contextual descriptors as texton histograms defined next.

Let G_n, $n = 1, \ldots, N_f$, now be specified as a bank of complex Gabor filters, and suppose we apply K-means clustering to the magnitudes of complex responses, $|\mathbf{F}(p)|$. Each of the K cluster centres represents a *texton* [33] and each pixel p is assigned to one of the K textons. $T(p)$ will denote this pixel-to-texton mapping, which takes one of the K possible values, for K textons, i.e., $T(p) = n$, $n = 1, \ldots, K$. Now we can concretize the contextual descriptor $\mathbf{c}^{(l)}$ of the block B_l as a vector of bin counts:

$$c_n^{(l)} = \frac{1}{|B_l \cap \Phi|} \sum_{p \in (B_l \cap \Phi)} \xi[T(p) = n], \quad n = 1, \ldots, K, \quad (5)$$

where $| \cdot |$ denotes the cardinality of the set and ξ is the indicator function (returning one if its argument is true and zero otherwise). The contextual dissimilarity $\bar{H}^{(l,m)}$ (Eq. (1)) can now be expressed as any distance between the histograms $\mathbf{c}^{(l)}$ and $\mathbf{c}^{(m)}$. Here, we employ the common χ^2-test:

$$\bar{H}^{(l,m)} = \chi^2(\mathbf{c}^{(l)}, \mathbf{c}^{(m)}) = \frac{1}{2} \sum_{n=1}^{K} \frac{(c_n^{(l)} - c_n^{(m)})^2}{c_n^{(l)} + c_n^{(m)}}. \quad (6)$$

Texton histograms as described above will be used in all the results in this paper. Note, however, that our general framework as described in Sections II-A and II-B can be used with other contextual descriptors as well. Optimizing the choice of contextual descriptors is out of the scope of this paper.

III. CONTEXT-AWARE MRF-BASED INPAINTING

Now we employ the proposed context-aware approach within a novel context-aware MRF-based inpainting algorithm. After constraining the search for candidate patches to the regions of well matching context, the number of labels is still too big and most of the existing inference methods will be inefficient. We propose a novel optimization approach suitable for global inpainting problem with large number of labels.

A. Notations and definitions

Let the patches be square image blocks of size $W \times W$, where $W = 2w + 1$.[2] We will treat the patches from the source region, which are the candidate patches for the target region, as labels of an MRF. By assuming an MRF model for image inpainting, as proposed in [8], spatial consistency among the candidate patches as well as their agreement with the undamaged image parts is imposed.

[2] An extension to rectangular patches of arbitrary size is straightforward.
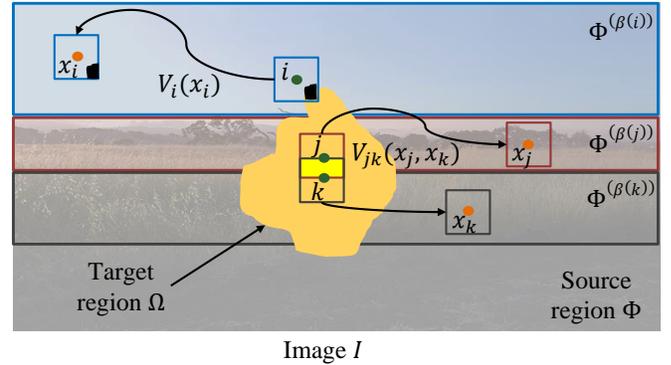

Fig. 5: An illustration of MRF notations. The green circles represent MRF nodes $i, j, k \in \nu$, and the orange circles represent the central positions x_i, x_j and x_k of their corresponding labels, which are the whole patches of pixel values centred at these positions. The labels of each node are chosen from its corresponding constrained source region (e.g., for the node i, the labels are chosen from $\Phi^{(\beta(i))}$). The black areas in the patches centred at i and x_i mark the locations of missing pixels at the node i. The label cost $V_i(x_i)$ is computed over the non-black areas of these patches. The pairwise potential $V_{jk}(x_j, x_k)$ is computed over the light yellow region.

Let a discrete lattice $\mathcal{L}$ consist of points, which are w pixels apart in horizontal or vertical direction on the image lattice S. Let $\mathcal{G} = (\nu, \varepsilon)$ denote an MRF with the set of nodes ν and the set of edges ε. The MRF is imposed over the target region Ω, meaning that ν consists of all lattice $\mathcal{L}$ points whose $W \times W$ neighbourhood intersects Ω and edges ε consist of all first-order neighbours $\langle i, j \rangle$ on the lattice $\mathcal{L}$, where $i, j \in \nu$ denote MRF nodes.

In [8], the labels Λ of the MRF were all possible patches that are completely inside the source region Φ. In our context-aware approach, for each node i, the set of labels $\Lambda_i \subset \Lambda$ will depend on the context around i. Let $\beta(i)$ be a function that returns the central position of the block to which i belongs. Hence, the block $B_{\beta(i)}$ contains the node i. The labels Λ_i are all possible patches that are completely inside the constrained source region $\Phi^{(\beta(i))}$ defined in Eq. (3), with $l = \beta(i)$. For the sake of compactness, each image patch, and thus each MRF label, will be referred to by its central position, thus the label set Λ_i is actually the label *position* set. The assignment of a label to the node i amounts to copying the patch centred at $x_i \in \Lambda_i$ to the position of the node i in the image. Fig. 5 illustrates the notations introduced above.

Finding an optimal combination of the candidate patches for the target region will result in a problem of minimizing the MRF energy

$$E(\mathbf{x}) = \sum_{i \in \nu} V_i(x_i) + \sum_{\langle i,j \rangle \in \varepsilon} V_{ij}(x_i, x_j), \quad (7)$$

where $V_i(x_i)$ is the *label cost* and $V_{ij}(x_i, x_j)$ is the *pairwise potential*. The label cost measures the agreement of a node with its labels, thus it is defined as the distance measure between the values of the known pixels in the $W \times W$

neighbourhood of the node i and the corresponding pixel values of the label centred at x_i (see Fig. 5). By using the common sum of squared differences (SSD) as a distance measure, the label cost $V_i(x_i)$ can be expressed as

$$V_i(x_i) = \sum_{di \in [-w,w] \times [-w,w]} M(i+di)\big(I(i+di) - I(x_i+di)\big)^2,$$ (8)

where M is a binary image, whose pixel values are one in the source region and zero in the target region Ω.[3] Note that if the $W \times W$ neighbourhood of a node is completely inside Ω, the label cost is zero. These nodes are called interior nodes (e.g., nodes j and k in Fig. 5). Finally, the pairwise potential $V_{ij}(x_i, x_j)$ is similarly defined as the SSD between labels centred at x_i and x_j in their nodes' overlap region (see Fig. 5).

The problem of optimizing the energy in Eq. (7) could be solved using loopy belief propagation (LBP) [44], where the solution is found by communicating *messages* between the nodes. However, applying LBP (or any standard inference method) directly may be prohibitive due to the huge number of labels of each node. Different optimization solutions have been proposed to deal more efficiently with such problems. In [24], coarse-to-fine LBP strategy was suggested. The idea is to first cluster the labels and then perform LBP at the first level on the cluster centers, resulting in the choice of one cluster per node, while at the second level, LBP is performed on the labels belonging to the chosen cluster. In [8], an improved version of BP called priority BP (p-BP) was introduced. In particular, a specific priority message scheduling and label pruning are applied. Priority is assigned to each node as inversely proportional to the number of labels whose relative belief is higher than some threshold. This means that the nodes with more confidence about their labels will have higher priority and therefore, will be visited first (in practice the ones lying on image structures and having more known pixels). Even with label pruning (discarding unlikely labels), the method of [8] is still in practice very slow, especially for bigger images. We introduce next a new faster inference method for this problem.

B. Efficient energy optimization

We now propose a computationally and memory efficient inference method for the patch-based inpainting problem in Eq. (7). The major differences with respect to p-BP [8] are the following. Firstly, instead of having a fixed label position set Λ, we consider a context-aware label position set $\Lambda_i \subset \Lambda$ for each node i. Therefore, by applying context-aware label selection, we limit the number of labels. Secondly, we introduce new formulations of priority scheduling and label pruning, resulting in faster and more memory efficient computation. Finally, we employ a different message passing inference algorithm to obtain the final inpainting result.

We divide the optimization process into three steps: initialization (computing priorities of nodes), label pruning (based on

---

[3]For multi-channel images, e.g., color images in this paper, SSD is computed as the sum of SSDs computed per each channel.

Algorithm 2 Efficient energy minimization

1: initialization:
2: **for** $i = 1$ **to** $|\nu|$ **do** $\{|\nu|$ is the total number of nodes$\}$
3:     compute $V_i(x_i)$ (Eq. (8))
4:     compute priority $P(i)$ (Eq. (11))
5:     set $v_i = 0$ {indicates whether the node is unvisited $(v_i = 0)$ or visited $(v_i = 1)$}
6: **end for**
7: label pruning:
8: compute $V_i^W(x_i)$, $\forall i \in \nu$ (Eq. (12))
9: **for** $t = 1$ **to** $|\nu|$ **do**
10:     $\hat{i} = \arg\max_{i:v_i=0} P(i)$
11:     apply label pruning: choose $L \ll |\Lambda_{\hat{i}}|$ labels $x_{\hat{i}}$ that yield L smallest $V_{\hat{i}}^W(x_{\hat{i}})$ and discard the rest
12:     **for** any neighbour j of $\hat{i}$ such that $v_j = 0$ **do**
13:         update $V_j(x_j)$ (Eq. (13)), $V_j^W(x_j)$ and $P(j)$
14:     **end for**
15:     set $v_{\hat{i}} = 1$
16: **end for**
17: inference: $\hat{\mathbf{x}} = \arg\min E(\mathbf{x})$

---

nodes' priorities) and the actual inference. A pseudo-code of the proposed energy optimization is given under Algorithm 2.

1) Initialization: This step assigns priorities to all MRF nodes, which determine their visiting order in the next phase (label pruning). Like in p-BP, we shall assign higher priority to nodes which are more confident about their labels. Since in our case the number of labels $|\Lambda_i|$ for each node i can be different, we define the priority in terms of the *relative* number of confident labels RNC_i as

$$P(i) = (RNC_i)^{-1}.$$ (9)

Our idea is to determine RNC_i without the need to compute beliefs, but rather based on the label cost $V_i(x_i)$ defined in Eq. (8). To this end, let us define the *relative* label cost as

$$V_i^{rel}(x_i) = V_i(x_i) - \min_{q \in \Lambda_i} V_i(q).$$ (10)

Now we define RNC_i and the corresponding priority $P(i)$ as

$$P(i) = (RNC_i)^{-1} = \Big(\frac{1}{|\Lambda_i|} \sum_{q \in \Lambda_i} \big(\tau_R - V_i^{rel}(q)\big)_+ \Big)^{-1}$$ (11)

where $(\zeta)_+ = 1$ if $\zeta > 0$ and zero otherwise, and τ_R is the threshold for the relative data cost, under which the assignment of a label to a node is considered as confident (practical computation of this parameter is explained in Section IV). In words, Eq. (11) says that the priority of the node i is directly proportional to the number of labels in the set Λ_i and inversely proportional to the number of labels from that set whose label cost is lower than the threshold τ_R.

2) Label pruning: This step reduces the number of labels at each node i to a relatively small number $L \ll |\Lambda_i|$ of the "best" candidate labels. To decide which labels are the best candidates, we need a suitable distance measure. This distance measure needs to take into account:

- data fidelity, as the agreement between the undamaged image part at the node i and the corresponding part of the label centred at x_i,
- contextual similarity between the regions (blocks) $B_{\beta(i)}$ and $B_{\beta(x_i)}$ that contain i and x_i, respectively.

One such possible label-pruning distance measure is contextually weighted label cost that we define as

$$V_i^W(x_i) = \left(1 - e^{-\bar{H}^{(\beta(x_i),\beta(i))}-\tau}\right)V_i(x_i), \qquad (12)$$

where $\bar{H}^{(\beta(x_i),\beta(i))}$ is the contextual dissimilarity defined in Eq. (1) and τ is the block similarity threshold from Eq. (2). Note that the weighting factor $1 - e^{-\bar{H}^{(\beta(x_i),\beta(i))}-\tau}$ becomes very small when the blocks $B_{\beta(x_i)}$ and $B_{\beta(i)}$ are contextually similar and tends to one when the contextual dissimilarity is very large. The constant τ in the exponent prevents that the weighting factor becomes zero when $\bar{H}^{(\beta(x_i),\beta(y_i))} = 0$ and enables in this way that the labels centred at x_i coming from the contextually ideally matching region can still be ordered based on their label cost $V_i(x_i)$.

After computing the label-pruning distance measure $V_i^W(x_i)$ for each node, the nodes are visited in the order of their priority (Eq. (11)) keeping L labels with the smallest $V_i^W(x_i)$ and discarding the rest. When one node chooses its labels, this information can be propagated to its neighbouring nodes. In this way, those neighbours have more information based on which they can perform label pruning, while also enforcing the agreement of labels of neighbouring nodes. As mentioned earlier, the label cost $V_i(x_i)$ of interior nodes and consequently, initial value of $V_i^W(x_i)$, is zero. Therefore, the only available information at the interior nodes is the one coming from the neighbours. We propagate the neighbouring information by updating the label cost at neighbours j of the current node $\hat{i}$ as

$$V_j^{(t+1)}(x_j) = V_j^{(t)}(x_j) + \min_{x_i} V_{\hat{i},j}(x_{\hat{i}}, x_j). \qquad (13)$$

The current node $\hat{i}$ is the node with the highest priority (see step 10 in Algorithm 2) and t denotes the update step. This updated measure can now be used directly in Eq. (12) to update $V_j^W(x_j)$. Such update definition is motivated by the update of beliefs within the global framework in p-BP, but we do not require the computation of messages. Note that each node is visited only *once* during label pruning, thus once chosen set of L labels per node remains fixed throughout the rest of the energy optimization algorithm. Therefore, the update is only necessary for *unvisited* neighbouring nodes, because their labels have not been pruned yet (see Algorithm 2).

3) Inference: After all the labels have been pruned, we can turn to minimizing the energy in Eq. (7). We employ here neighbourhood-consensus message passing (NCMP) [37], which is much faster than LBP while offering similar performance in this type of problems. In this method, one joint message, which is a function of beliefs, is sent from the whole neighbourhood ∂i to the central node i:

$$m_{\partial i \to i}(x_i) = \exp\left(-\sum_{j \in \partial i}\sum_{x_j} b_j(x_j)V_{ij}(x_i, x_j)\right). \qquad (14)$$

The node's belief $b_j(x_j)$ is updated as follows:

$$b_j(x_j) = \alpha\Theta_j(x_j)m_{\partial j \to j}(x_j), \qquad (15)$$

where $\Theta_j(x_j) = \exp\left(-V_j(x_j)\right)$ and α is a normalization factor so that beliefs sum up to one. We start the algorithm from the initial mask formed by maximum likelihood estimation, $\hat{x}_i = \arg\max_{x_i}\Theta_i(x_i)$, and then we set belief initially to the value that favours the label of that node in the initial mask. After initialization, algorithm runs iteratively until some stopping criterion is satisfied (e.g., specified number of iterations is reached or the difference between two successive results becomes negligible). At that point, for each node we choose the label that maximizes the node's belief from Eq. (15). The chosen labels are copied to their nodes' positions (in parallel), resulting in filling of the *whole* missing region. Since neighbouring labels overlap (see Fig. 5), a minimum error boundary cut [5] or a similar stitching method can be employed to find the seam along which the transition between two neighbouring patches (labels) is the least visible.

IV. EXPERIMENTS AND RESULTS

We evaluate the proposed method in applications of scratch and text removal, and image editing (object removal). The reference methods for comparison are chosen from all three categories: "greedy", multiple candidate and global. For all the analysed methods we show the best inpainting result, by optimizing the patch size (where possible). Furthermore, for our method, if not stated otherwise, we use $N_f = 18$ filters (over 3 scales and 6 orientations), $K = 16$ textons, block similarity threshold $\tau = 0.15$, $L = 10$ chosen labels and 10 iterations of the inference algorithm. Threshold for priority, τ_R, is computed as a median value of SSDs between each pair of patches in the source region, as suggested in p-BP, just that in our case this source region is constrained and it differs from one block to another. For all the results, we used the division into blocks of adaptive sizes obtained by the top-down splitting procedure. This procedure was conducted until the block size reached 1/4 of the image size for images in Section IV-A and 1/8 of the image size for images in Section IV-B, because the former images contain a close-up of the object (see Fig. 6), thus finer division would not be beneficial.

A. Experiments and comparisons for scratch and text removal

For the task of scratch and text removal, we use the dataset of four images from [10] (top row of Fig. 6), where the ground truth is available. The reference methods include the "greedy" approach from [7][4], commercial software Content Aware Fill of Adobe PhotoShop, based on [25], [29], multiple candidate sparsity-based method (MCS) [10][5], and the global p-BP method [8][6], which is mostly related to ours. Peak signal-to-noise ratio (PSNR) values indicated in Fig. 6 are computed only in the missing region, with pixel values in the range

---

[4]MatLab software from http://www.cc.gatech.edu/~sooraj/inpainting/.

[5]Test images and results were received from the authors.

[6]We use our own implementation in MatLab with $L_{min} = 3$, $L_{max=10}$ and 10 iterations of the p-BP algorithm.
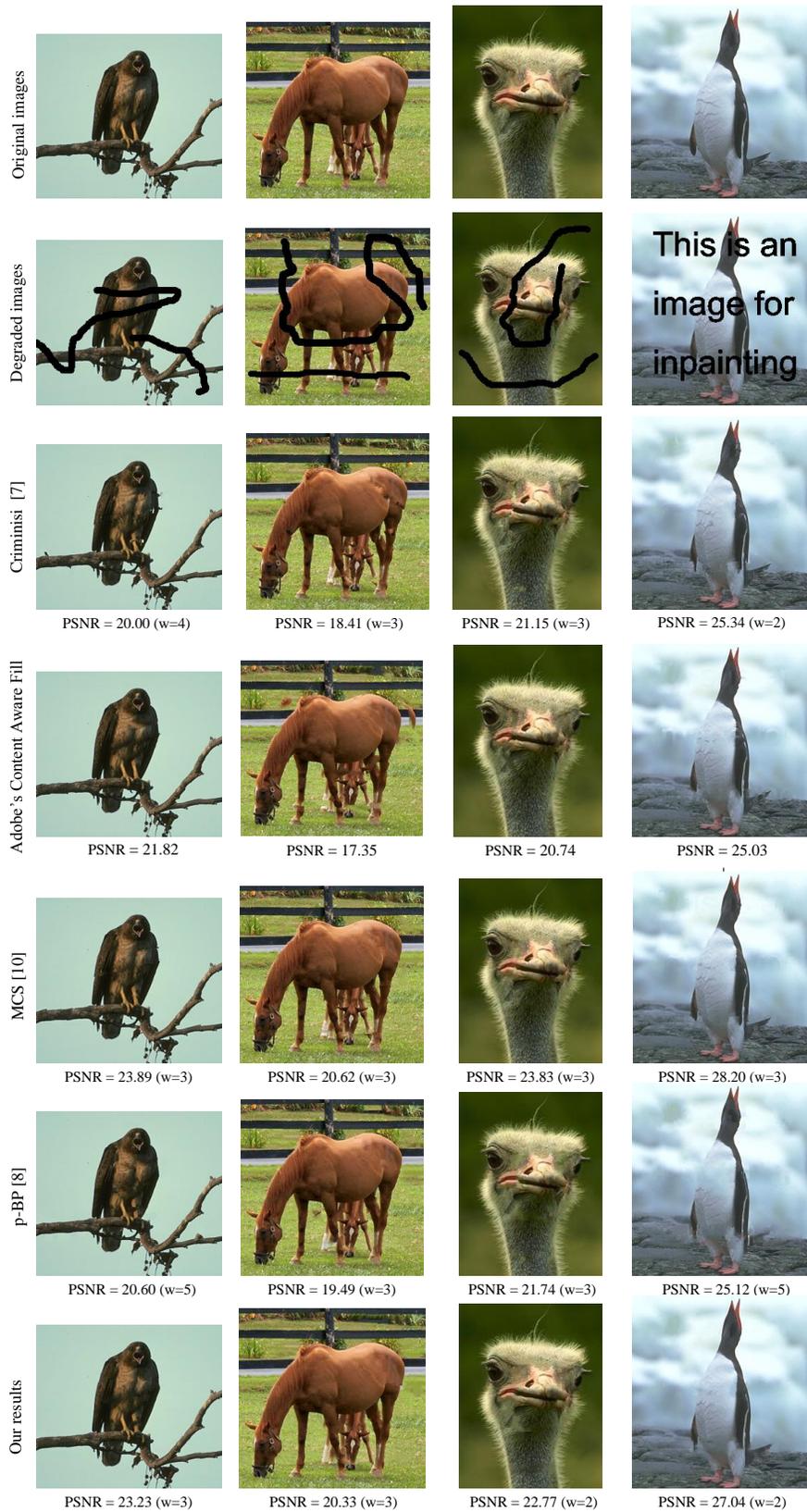
Fig. 6: Comparison of different inpainting methods for scratch and text removal (see text for details).
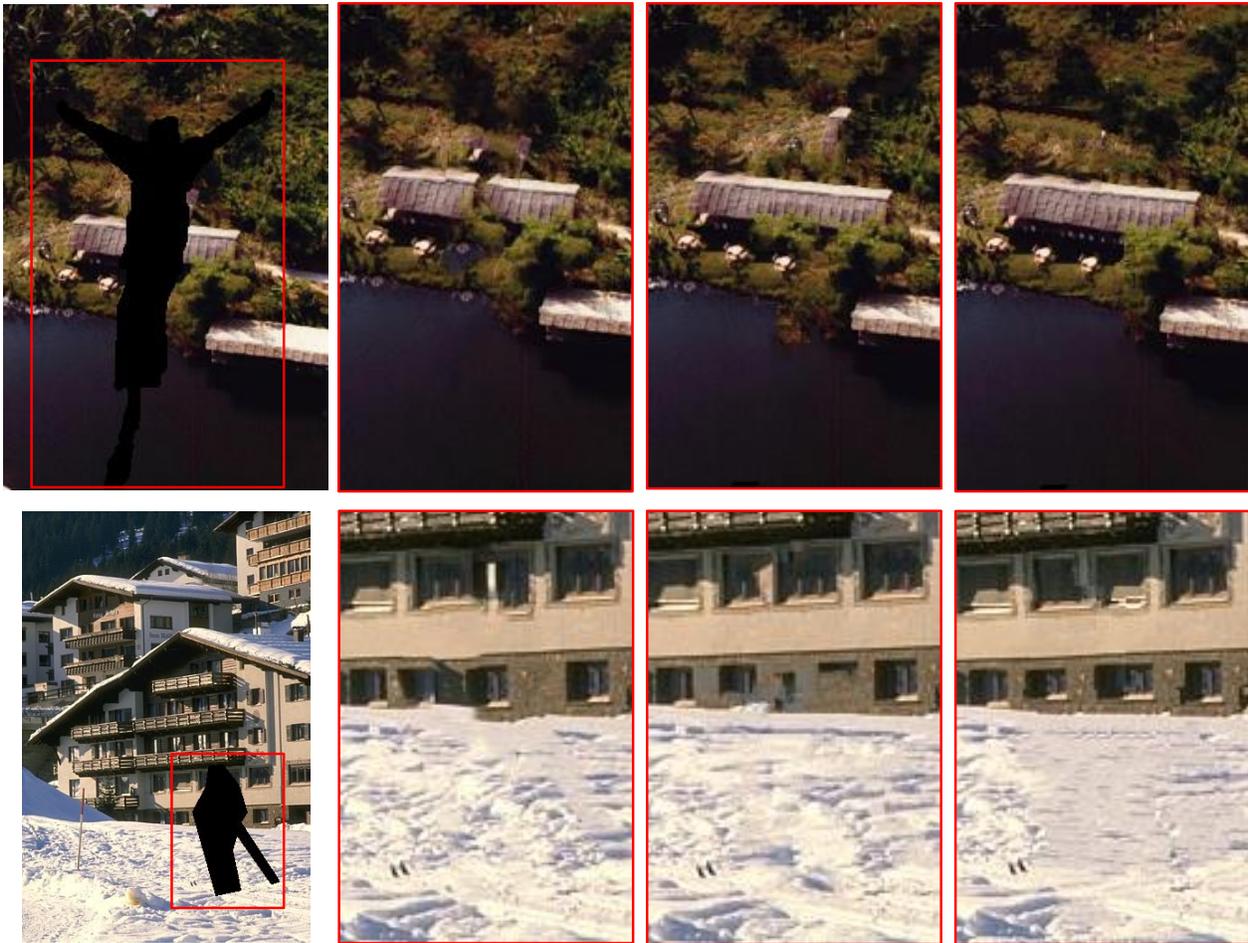
Fig. 7: Comparison of different inpainting methods for "bungee" (top) and "ski resort" image (bottom). From left to right: input image with missing region in black, result of Content Aware Fill, result of p-BP [8] with $w = 5$ ("bungee") and $w = 4$ ("ski resort"), and the result of the proposed method with $w = 6$ for both images. Area in the red rectangle in the original image is shown enlarged in the results.

[0,1]. We varied the parameter w that determines the patch size from 2 to 6 and chose the one with the highest PSNR for each method (shown in the brackets). Only Content Aware Fill does not require explicit specification of the patch size.

The results in Fig. 6 demonstrate that the proposed method gives visually pleasing result, with almost no disturbing artefacts. Compared to [7], [8] and Content Aware Fill, our method yields the best result for all images, both quantitatively (in terms of PSNR) and qualitatively. Compared to the MRF-based p-BP [8], the increase in PSNR ranges from 0.6 to 2.8dB. Our PSNR values are lower than those of MCS [10] (PSNR difference ranging from 0.5 to 1.3dB). This can be partly due to the fact that [10] is ideally suited for this type of problems (thin missing regions), while our method is generally formulated to cope with larger "holes". Nevertheless, this example shows that our method can also deal with scratch/text removal and achieve comparable, and in many cases better results than related and state-of-the-art methods.

B. Experiments and comparisons for object removal

In this subsection, we deal with a more demanding task of object removal, which requires large missing regions to be inpainted. The bigger the missing region is, the more ambiguity there is on how to fill it in.

Fig. 7 compares the results of the proposed method with Content Aware Fill and p-BP [8] on two different images. Our method seems to be more successful in preserving image structure and produces the most visually pleasing result. For example, in the "bungee" image (top), see artefacts on the building's roof and the grass area below it in the result of Content Aware Fill and the artefact in the green area above the building and the water/land border in the result of p-BP [8]. For the "ski resort" image (bottom), our method does better job in continuing the border between house and snow and inpainting the windows of the house, especially the windows in the bottom row.

The results in Figs. 8 and 9 also show improvements over p-BP [8]. Additionally, Fig. 8 shows clear advantage over the "greedy" method from [7], which is also the case for all other test images (not shown due to the lack of space). Compared to

Fig. 8: Comparison of different inpainting methods for "baseball" image. From left to right and top to bottom: input image with missing region in black, result of [7] with $w = 4$, result of Content Aware Fill, result of [23], result of p-BP [8] with $w = 3$, and the result of the proposed method with $w = 7$.

TABLE I: Comparison of computation times.

Image (patch size)	p-BP [8]	Proposed method
"bungee" ($w = 6$)	253.88s	113.76s
"ski resort" ($w = 6$)	871.93s	742.44s
"baseball" ($w = 7$)	1295.95s	499.5s
"wall" ($w = 3$)	103.33s	18.54s
"lake" ($w = 3$)	138.76s	45.14s
"office" ($w = 2$)	214.23s	58.42s

TABLE II: Computation times per each phase of the algorithms for image in Fig. 8 for $w = 7$.

Phase	p-BP [8]	Proposed method
threshold computation	144.44s	73.35s
initialization	20.29s	7.88s
label pruning	1126.45s	400.76s
inference	2.67s	0.82s
overhead computations	2.1s	16.69s

TABLE III: Comparison of time and space complexity, where $\kappa = \max_i |\Lambda_i|$ and $\kappa < |\Lambda|$.

Phase	Time		Space	
	p-BP [8]	Proposed	p-BP [8]	Proposed
initialization	$\mathcal{O}(|\Lambda|)$	$\mathcal{O}(\kappa)$	$\mathcal{O}(|\Lambda|)$	$\mathcal{O}(\kappa)$
label pruning	$\mathcal{O}(L|\Lambda|)$	$\mathcal{O}(L\kappa)$	$\mathcal{O}(6|\Lambda|)$	$\mathcal{O}(2\kappa)$
inference	$\mathcal{O}(L^2)$	$\mathcal{O}(L^2)$	$\mathcal{O}(L^2 + 6L)$	$\mathcal{O}(L^2 + 3L)$

Content Aware Fill, our results are better (see images in the second and third column of Fig. 9) or comparable. Finally, we also show the results of two multiple candidate state-of-the-art methods[7]: very recent super-resolution-based method from [23] in Fig. 8 and method from [11] in Fig. 9. Our result is comparable to that of [23], although our method preserves better the border between snow and sky. Compared to [11], our method gives superior results on all images in Fig. 9.

C. Complexity analysis

Table I shows the computation times of the p-BP [8] and the proposed method, using our own MatLab implementation of both methods on Intel i5-2520M 2.5 GHz CPU with 6GB RAM, on several test images from Figs. 7, 8 and 9. For the sake of fair comparison, we tested the algorithms for the same patch size (adapted to the image), as indicated in the first column of the table. The proposed method is obviously much faster (for some images up to 6 times).

Most of the computation time for both methods is spent on label pruning, as illustrated in Table II. Therefore, acceleration of our method is largely due to the use of contextual information, which yields a smaller (constrained) label set,

[7]Results are available on the website of the authors.

and hence there is less work for pruning. Initialization is also much accelerated due to the same reason. Finally, our inference method is also faster than p-BP, for about 3-4 times on the "baseball" image (Fig. 8), with the same number of iterations (10) and the same number of pruned labels ($L = 10$). Overhead computations include patch stitching and in the proposed method, texton computation, adaptive division into blocks and block matching. Note also in Table II that significant amount of time is needed for the threshold computation. These computations are for the most images still much faster than label pruning, except for the "ski resort" image (bottom of Fig. 7), which is also the reason for less significant speed-up over p-BP on this image.

The computation times presented above are mostly influenced by the number of required SSD computations, which depends on the number of labels of each node. SSD computations are used for the computation of label cost and
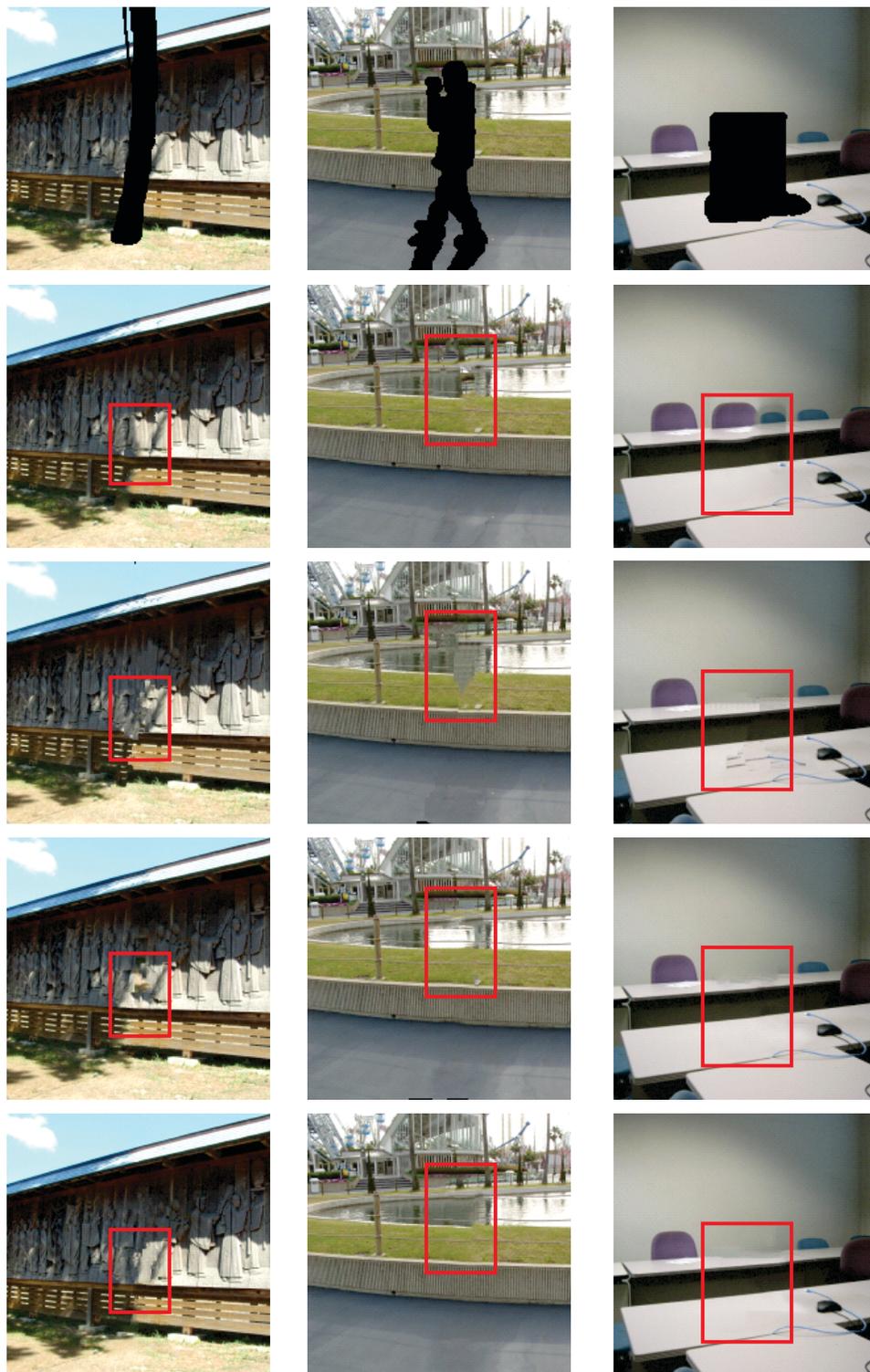
Fig. 9: Comparison of different inpainting methods on images "wall" (left), "lake" (middle) and "office" (right) from [45]. From top to bottom: input image with missing region in black, result of Content Aware Fill, result of [11], result of p-BP [8] with $w = 3$ for all three images, and the result of the proposed method with $w = 3$, $w = 3$ and $w = 2$ for the left, middle and right image, respectively.
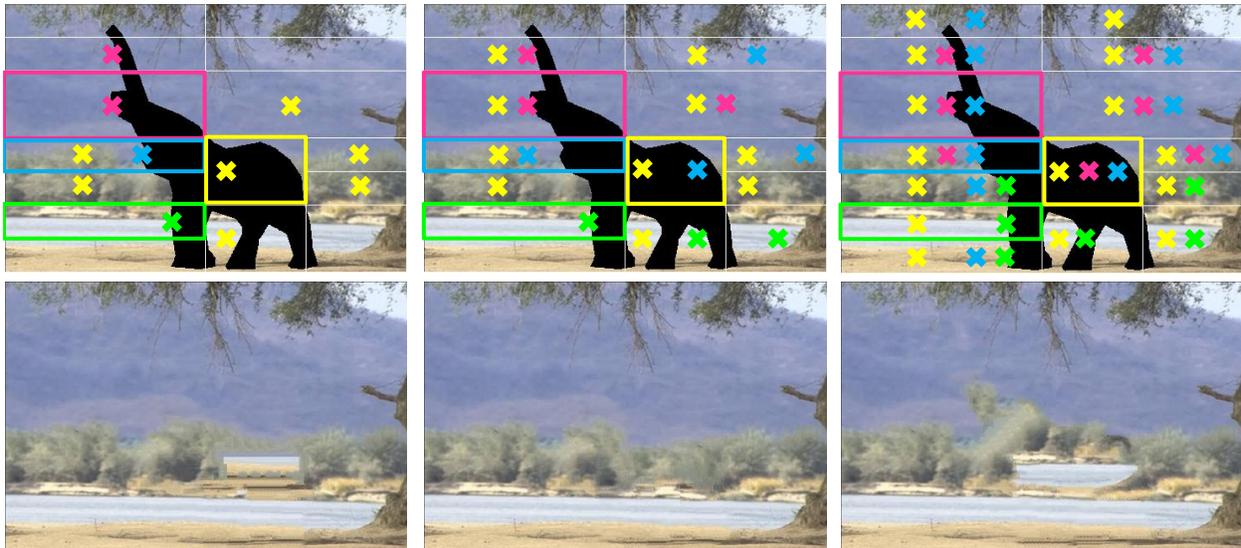
Fig. 10: Effect of block selection on the inpainting result. Top: An illustration of three different block selection strategies (see text for details), corresponding to different values of τ (0.05, 0.15 and 0.5, from left to right), all starting from the same block division. Bottom: The corresponding inpainting results.

pairwise potential (see Section III-A). Table III shows time and space complexity comparison between p-BP and the proposed algorithm for each phase in terms of the maximum number of labels. Time complexity refers to the number of computations performed by the algorithm: label cost computation in initialization (per node), pairwise potential computation in label pruning (per edge), and pre-computation of pairwise potential in inference (per edge). Space complexity refers to memory requirements. In our analysis, the most memory is needed to store label costs, beliefs and/or messages, each being a vector of dimensionality equal to the number of labels per node. Specifically, we need to store label costs in initialization of both algorithms. Label pruning of p-BP requires storing four messages (one from each neighbour), belief and label cost (total of 6 vectors), compared to two label costs in label pruning of our method. Finally, inference in both methods requires storing pairwise potential, label cost and belief, plus four messages in p-BP compared to only one message in NCMP (used for inference in our algorithm). This analysis confirms that the main reason for the reduced complexity (and thus the computation time) of our algorithm is the reduced number of labels per node due to the context-aware approach ($\kappa < |\Lambda|$, where κ is the maximum number of labels per node across all nodes). Furthermore, our algorithm requires less variables to be computed and stored.

D. Effect of the parameters

A crucial aspect in our approach is the block selection, which is determined by the block similarity threshold τ (Eq. (2)). Fig. 10 illustrates the selected blocks (top) and the corresponding inpainting results (bottom) for the three values of τ: 0.05, 0.15 and 0.5 (for the purpose of illustration, the block division and all the other parameters are kept fixed). The current (query) blocks are marked with rectangles of different colors, while their block matches (including the current blocks

themselves) are marked with a cross of the matching color. Note that the yellow block is unreliable (see Section II-A), thus its block matches are the block matches of neighbouring blocks. We can see that for small τ, most of the blocks are contextually similar only to themselves, which limits severely the search for good candidate patches and yields a poor inpainting result (left of Fig. 10). A too high τ yields too many block matches, thus the source region is insufficiently constrained. This not only produces worse result (right of Fig. 10), but also decreases computational efficiency. The best inpainting result was obtained with some intermediate threshold ($\tau = 0.15$ for the middle image in Fig. 10), which allows the selection of sufficient but limited number of blocks with well-matching context. Changing the value of τ around this value has little influence on the block selection.

We found by experimental evaluation over multiple images and block sizes, that $\tau = 0.15$ is a good choice for a block similarity threshold when using $N_f = 18$ filters (over 3 scales and 6 orientations) and $K = 16$ textons (as we did for all the images). We also made experiments with $N_f = 24$ filters (over 3 scales and 8 orientations) and $K = 32$ textons. The optimal value of τ was slightly higher in this case (0.2) and we were able to obtain somewhat finer contextual division for most images. However, the resulting slight differences did not justify the increase in complexity.

Another parameter is L, the number of labels kept after pruning. We found $L = 10$ to be a good trade-off between algorithm's computational complexity and quality of the result. Optimal patch size, like in most other patch-based inpainting methods, depends on the resolution and the content of the image, and it is therefore tuned for each image.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced a novel MRF-based inpainting method that uses context-aware approach to reduce the

number of possible labels per MRF node and choose them in such a way that they better fit the surrounding context. Context is represented within blocks of fixed or adaptive sizes using contextual descriptors in the form of normalized texton histograms. Additionally, to divide the image into blocks of adaptive size, a novel top-down splitting procedure was introduced. We also proposed a simple and efficient way to perform optimization by first pruning the labels of each node to some small number, using both the agreement of the node with its labels and the contextual similarity between regions to which the node and the label belong. Results demonstrated the benefits of such an approach in comparison with the state-of-the-art methods and improved speed in comparison with the related MRF-based method.

Acknowledgements

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *SIGGRAPH '00*, New Orleans, USA, 2000, pp. 417–424.

[2] T. F. Chan and J. Shen, "Non-texture inpainting by curvature-driven diffusions (cdd)," *J. of Vision Comm. and Image Represent.*, vol. 12, no. 4, pp. 436–449, 2001.

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

[4] D. Tschumperlé, "Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's," *Int. J. of Comp. Vision*, vol. 68, no. 1, pp. 65–82, 2006.

[5] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *SIGGRAPH '01*, 2001, pp. 341–346.

[6] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *SIGGRAPH '03*, 2003, pp. 277–286.

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

[8] N. Komodakis and G. Tziritas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *IEEE Trans. on Image Proc.*, vol. 16, no. 11, pp. 2649–2661, Nov. 2007.

[9] C.-W. Fang and J.-J. J. Lien, "Rapid image completion system using multiresolution patch-based directional and nondirectional approaches," *IEEE Trans. on Image Proc.*, vol. 18, pp. 2769–2779, Dec 2009.

[10] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. on Image Proc.*, vol. 19, no. 15, pp. 1153–1165, May 2010.

[11] O. Le Meur, J. Gautier, and C. Guillemot, "Examplar-based inpainting based on local geometry," in *ICIP '11*, 2011, pp. 3462–3465.

[12] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," *ACM Trans. on Graph.*, vol. 24, pp. 861–868, July 2005.

[13] M. Bertalmio, L. A. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. on Image Proc.*, vol. 12, no. 8, pp. 882–889, Aug 2003.

[14] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," in *SIGGRAPH '03*, 2003, pp. 303–312.

[15] J. Jia and C.-K. Tang, "Image repairing: robust image synthesis by adaptive nd tensor voting," in *CVPR '03*, 2003, pp. 643–650.

[16] A. Bugeau, M. Bertalmio, V. Caselles, and G. Sapiro, "A comprehensive framework for image inpainting," *IEEE Trans. on Image Proc.*, vol. 19, no. 10, pp. 2634–2645, Oct. 2010.

[17] Anupam, P. Goyal, and S. Diwakar, "Fast and enhanced algorithm for exemplar based image inpainting," in *PSIVT '10*, 2010, pp. 325–330.

[18] T. Ružić, B. Cornelis, L. Platiša, A. Pižurica, A. Dooms, W. Philips, M. Martens, M. De Mey, and I. Daubechies, "Virtual restoration of the Ghent altarpiece using crack detection and inpainting," in *ACIVS '11*, 2011, pp. 417–428.

[19] T. Ružić and A. Pižurica, "Texture and color descriptors as a tool for context-aware patch-based image inpainting," in *SPIE Electronic Imaging '12*, vol. 8295, Feb. 2012, pp. 82 951P–1–82 951P–11.

[20] J. Lee, D.-K. Lee, and R.-H. Park, "Robust exemplar-based inpainting algorithm using region segmentation," *IEEE Trans. on Cons. Elec.*, vol. 58, no. 2, pp. 553–561, May 2012.

[21] A. Wong and J. Orchard, "A nonlocal-means approach to exemplar-based inpainting," in *ICIP '08*, 2008, pp. 2600–2603.

[22] V. V. Voronin, V. I. Marchuk, and K. O. Egiazarian, "Images reconstruction using modified exemplar based method," in *SPIE Electronic Imaging '11*, vol. 7870, 2011, pp. 78 700N–1–78 700N–11.

[23] O. L. Meur and C. Guillemot, "Super-resolution-based inpainting," in *ECCV '12*, 2012, pp. 554–567.

[24] T. Huang, S. Chen, J. Liu, and X. Tang, "Image inpainting by global structure and texture propagation," in *Proc. of ACM Int. Conf. on Multimedia*, 2007, pp. 517–520.

[25] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. on Pattern Anal. and Mach. Intel.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

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

[27] Y. Pritch, E. Kav-Venaki, and S. Peleg, "Shift-map image editing," in *ICCV '09*, 2009, pp. 151–158.

[28] T. Ružić, A. Pižurica, and W. Philips, "Markov random field based image inpainting with context-aware label selection," in *ICIP '12*, 2012, pp. 1733–1736.

[29] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," in *SIGGRAPH '09*, 2009, pp. 24:1–24:11.

[30] K. He and J. Sun, "Statistics of patch offsets for image completion," in *ECCV '12 (2)*, 2012, pp. 16–29.

[31] J. Puzicha, T. Hofmann, and J. Buhmann, "Non-parametric similarity measures for unsupervised texture segmentation and image retrieval," in *CVPR '97*, 1997, pp. 267–272.

[32] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *Int. J. of Comp. Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[33] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Int. J. of Comp. Vision*, vol. 43, no. 1, pp. 7–27, June 2001.

[34] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. on Pattern Anal. and Mach. Intel.*, vol. 33, no. 5, pp. 898–916, May 2011.

[35] O. G. Cula and K. J. Dana, "3d texture recognition using bidirectional feature histograms," *Int. J. of Comp. Vision*, vol. 59, no. 1, pp. 33–60, Aug. 2004.

[36] C. Schmid, "Constructing models for content-based image retrieval," in *CVPR '01*, vol. 2, Dec. 2001, pp. 39–45.

[37] T. Ružić, A. Pižurica, and W. Philips, "Neighbourhood-consensus message passing as a framework for generalized iterated conditional expectations," *Pattern Rec. Let.*, vol. 33, pp. 309–318, Feb. 2012.

[38] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. of Comp. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[39] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, Nov 1973.

[40] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Rec.*, vol. 29, no. 1, pp. 51–59, Jan 1996.

[41] G. Poggi, G. Scarpa, and J. Zerubia, "Supervised segmentation of remote sensing images based on a tree-structured mrf model," *IEEE Trans. on Geosc. and Remote Sensing*, vol. 43, no. 8, pp. 1901–1911, Aug. 2005.

[42] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *Comm. of ACM*, vol. 51, no. 10, pp. 87–94, 2008.

[43] A. Torralba, K. P. Murphy, and W. T. Freeman, "Using the forest to see the trees: exploiting context for visual object detection and localization," *Comm. of ACM*, vol. 53, no. 3, pp. 107–114, 2010.

[44] J. S. Yedidia and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 736–744, Feb. 2001.

[45] N. Kawai, T. Sato, and N. Yokoya, "Image inpainting considering brightness change and spatial locality of textures and its evaluation," in *PSIVT '09*, 2008, pp. 271–282.

Tijana Ružić received her M.Sc. degree in electrical and computer engineering from the Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia, in 2008 and the Ph.D. degree in engineering from Ghent University, Ghent, Belgium, in 2014. She is currently with the Image Processing and Interpretation group, Department of Telecommunications and Information Processing, Ghent University, as a Post-Doctoral Researcher.

Her main research interests are patch representations and Markov random field models in image restoration.

Aleksandra Pižurica received the Diploma Degree in electrical engineering from the University of Novi Sad, Novi Sad, Serbia, in 1994, the M.Sc. degree in telecommunications from the University of Belgrade, Belgrade, Serbia, in 1997, and the Ph.D. degree in engineering from Ghent University, Ghent, Belgium, in 2002, where she has been a Professor in Statistical Image Modelling since 2009. She currently serves as an Associate Editor of the IEEE Transactions on Image Processing.

Aleksandra Pižurica has authored and co-authored more than 200 publications in international journals, conferences and book chapters. Her research interests are statistical image modelling with applications to image and video restoration, efficient representations of multidimensional signals and hierarchical statistical models of visual perception.