PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESCUELA DE POSGRADO

MULTI-SCALE IMAGE INPAINTING WITH LABEL SELECTION BASED ON LOCAL STATISTICS

By

Daniel Leoncio Paredes Zevallos

Thesis submitted in partial fulfillment of the requirements for the degree of
Master in Digital Signal and Image Processing
in the Graduate School of the Pontificia Universidad Católica del Perú.

Thesis Supervisor: Paul Antonio Rodriguez Valderrama

Examining commitee members:
Jorge Richard Chavez Fuentes
Francisco Fabián Cuellar Córdova

Lima
2014

Abstract

We proposed a novel inpainting method where we use a multi-scale approach to speed up the well-known Markov Random Field (MRF) based inpainting method. MRF based inpainting methods are slow when compared with other exemplar-based methods, because its computational complexity is $O(|\mathcal{L}|^2)$ ($\mathcal{L}$ feasible solutions' labels). Our multi-scale approach seeks to reduces the number of the $\mathcal{L}$ (feasible) labels by an appropiate selection of the labels using the information of the previous (low resolution) scale. For the initial label selection we use local statistics; moreover, to compensate the loss of information in low resolution levels we use features related to the original image gradient.

Our computational results show that our approach is competitive, in terms reconstruction quality, when compare to the original MRF based inpainting, as well as other exemplar-based inpaiting algorithms, while being at least one order of magnitude faster than the original MRF based inpainting and competitive with exemplar-based inpaiting.

Contents

Chapter 1

Introduction

Image inpainting, an ancient art itself [1], is a technique of modifying (reparing) an image in an undetectable form. Originally, its key objective was to fill-in the missing or damaged parts of the artistic work, and to restore its unity. In time, movies, photographs and other type of visual works have been digitized, so digital inpainting applications emerged. Some applications are scaling-up images by superresolution, compression, reconstructing old photographs, and removal of overlaid text or graphics. It is also possible to add or remove objects.

In parallel to image inpainting, there was another field that also attracted a considerable amount of research over the last years called texture synthesis (Fig. 1.1(a)). Given a small texture sample as input, we are then asked to generate an arbitrarily large output texture, which maintains the visual characteristics of the input [2]. In general, an ideal image inpainting algorithm should take in account texture synthesis in order to generated a visually plausible result.

(a) (b)

Figure 1.1: (a) Texture synthesis results [3]. (b) Classification of image inpainting methods

Image inpainting methods can be classified in diffusion-based and patch-based. In [1], holes in the images are filled by diffusing linear structure of surrounding regions along isophote direction by solving partial differential equations (PDEs). Inspired by this diffusion method, numerous models, including curvature-driven diffusion (CDD) [4] and variational approaches [5], are incorporated into the inpainting task. The variational approaches aim to solve a model based on Bayesian principle and geometric images models (curvatures) using second order PDEs. However, diffusion-based methods only works on small

gaps (i.e. scrathes), and mostly, on non-textured images.

On the other hand, patch-based approaches ([6], [7], [8]) are proposed to address inpainting problems with large missing image portions. Similar to the well studied domain texture synthesis ([9], [10], [3]), results can be generated by sampling and copying the best matching color values from known regions to missing regions at patch level. Among patch-based methods we should mention the work of Criminisi *et al.* [6], which is one of the most influential works in exemplar-based methods. The algorithm assigns priorities to the blocks that intersects the unknown region and then looks for the best match from the known region to fill the gap. The priority is determined by data and confidence terms. Confidence term is given by the amount of known data in the block while data term is derived from measuring isophote strength orthogonal to the fill front.

(a) (b)

Figure 1.2: (a) Image input. (b) Result after patch-based method from [6]

Even though, common patch-based methods generate better results than diffusion-based, they are not supported by solid mathematical fundations, and thus, inpainted images may present visual inconsistencies (see Fig. 1.2(b)). In order to overcome this limitation, global optimization models have been proposed and state-of-the-art models are based on Markov Random Fields (MRF) ([11], [12], [13], [2], [14], [15]). Among global optimization models, we should mention the work of Sun *et al.* [11], where the authors present an interactive approach to image inpainting, in which the user indicates what important structures should be completed before remaining unknown regions are filled in. The user-defined structures are modeled as a Markov Random Field and the cost function involved is minimizing using loopy Belief propagation algorithm, and the remaining region is filled using a common exemplar-based approach. However, there are other works in which no user intervention is required and still plausible results are obtained. For example, the video-completion algorithm of Wexler *et al.* [12], which could also be used for image inpainting, is based on a non-parametric optimization introduced by [9] which leads to improve texture synthesis. On the other hand, this algorithm is sensitive to initialization and may get easily trapped in a poor local minima. In order to fix this problem, another global optimization approach has been presented by Komodakis and Tziritas in [13]. The unknown region is modeled follow-

ing the Markov Random Field approach and the goal is to fill the hole such that global and local consistency should be preserved, thus, the functional in this case aims to maximize the similarities between neighbour patches and candidates and the solution is given by Belief Propagation algorithm. Although, the algorithm produces acceptable results it could take several hours to inpaint an image no greater than 512x512. Thus, authors, in [2], proposed an improvement for the belief propagation algorithm by pruning the candidates in each iteration. This modification has a relevant impact on the computation time by making the algorithm at least 10 times faster.

In this thesis, we proposed a computational efficient method for image inpainting based on a multi-scale approach. Our multi-scale approach, which is different from the prune method proposed by [2], first performs a label selection based on (simple) local statistics. Then, using a multi-scale approach, we reduce the number of possible solutions while preserving the texture information in lower levels avoiding fake local minimas. This procedure gives a substantial speedup when compared with previous methods. Preliminary results have been presented in the *2013 European Signal and Image Processing Conference* [16], and in [17] we present an application oriented to catadioptric omnidirectional images, where we combine our multi-scale image inpainting (MII) algorithm with unwrapping techniques ([18], [19]) as the MII can not be applied in the omnidirectional image directly.

Finally, let us explain the organization of the thesis. In chapter 2 we summary the state of the art for image inpainting methods. In chapter 3 we present our multi-scale inpainting method. In chapter 4 experimental results are given as well as comparisons with other algorithms and parameter dependency. At the end of the thesis we summarize the conclusions (Chapter 5) and recommendations.

Chapter 2

State-of-The-Art

In this chapter we summarily describe the Markov Random Field (MRF) model, Belief Propagation (BP) algorithm and its variation, loopy BP. We also mention the MRF model for image inpainting, as well as the energy function in this case and the optimization algorithm (Priority BP) to solve the problem.

2.1 Markov Random Fields

Markov Random Fields (MRFs) are defined as a probabilistic model over undirected graph G such that it is described by $G = (\nu, \varepsilon)$, where ν and ε denotes the nodes and edges respectively. Let us denote $P(\mathbf{X} = x)$ as the probability distribution for state of each node, where $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots)$ is a sequence of random variables. This probability distribution can be expressed in terms of an energy or cost function $E(\mathbf{x})$ using the Gibbs' distribution [20]. In (2.1) $\mathbf{Z}$ is a function that does not depend on $\mathbf{X}$.

$$P(\mathbf{x}) = \frac{1}{\mathbf{Z}} \exp(-E(\mathbf{x})) \tag{2.1}$$

The most common form of inference over the posterior MRF in vision and image-processing problems is *maximum a posteriori* (MAP) estimation [20], in other words, we need to compute the sequence of x_i that maximize $P(\mathbf{x})$ or minimize $E(\mathbf{x})$. It is also possible to write the energy function as a sum of a unary V_i and pairwise V_{ij} term described by (2.2). We are going to apply the Markov Random Field model to inpainting problem, thus, we are going to use (2.2) through the whole thesis.

$$E(\mathbf{x}) = \sum_{(i,j)\in\varepsilon} V_{ij}(x_i, x_j) + \sum_{i\in\nu} V_i(x_i) \tag{2.2}$$

2.1.1 Belief Propagation

Belief propagation (BP) is a message passing based algorithm to either find the marginal posteriors or *maximum a posteriori* on tree structured graphs. There are two versions of BP: the *sum-product* and *max-product*, as long as we focus on solving the MAP problem we use the max-product version. Note that in order to compute marginal posterior, sum-product

should be used instead.

Let us denote $m_{ij}(x_j)$ the message from node i (children) to node j (parent). The message passing in BP algorithm has two steps, in the first one messages are passed from *children* nodes to their *parents* until they reach the *root* node (Fig. 2.1). The second step is the MAP estimation.

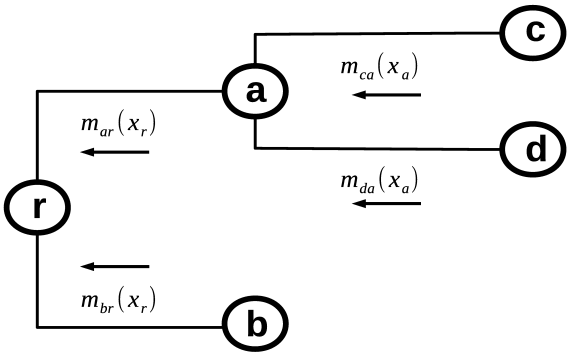


Figure 2.1: Message passing inference in a tree structured graph. First, messages are sent from nodes $\{c, d\}$ to node a, and then $\{a, b\}$ send their messages to root node r.

$$m_{ij}(x_j) = \max_{x_i} \left\{ P(x_i, x_j) \prod_{k \in N_c(i)} m_{ki}(x_i) \right\} \tag{2.3}$$

$$x_i^* = \operatorname*{argmax}_{x_i} P(x_j^*, x_i) \prod_{k \in N_c(i)} m_{ki}(x_i) \tag{2.4}$$

where $N_c(i)$ is the set of all the children of node i. Besides messages passing, estimation is from root node to children, thus, given x_j^* it is possible the find x_i^* using (2.4). Since the root node has not parents its MAP estimation (x_r^*) is just the multiplication of the messages from its children nodes; in other words it can be considered the only parent for root node is itself.

Sometimes it is preferred to convert multiplications of the elements of a given sequence to additions by applying the logarithm function, in our case we obtain (2.5) and (2.6) from (2.3) and (2.4) respectively. Recall that the energy function is defined in (2.1) and thus in (2.6) the function is minimized.

$$m_{ij}(x_j) = \min_{x_i} \left\{ \sum_{(i,j) \in \varepsilon} V_{ij}(x_i, x_j) + \sum_{i \in \nu} V_i(x_i) + \sum_{k \in N_c(i)} m_{ki}(x_i) \right\} \tag{2.5}$$

$$x_i^* = \operatorname*{argmin}_{x_i} \left\{ \sum_{(i,j) \in \varepsilon} V_{ij}(x_i, x_j^*) + \sum_{i \in \nu} V_i(x_i) + \sum_{k \in N_c(i)} m_{ki}(x_i) \right\} \tag{2.6}$$

2.1.2 Loopy Belief Propagation

We have seen that it is possible to find the optimal solution for MAP estimation with the max-product BP in a tree structured graphs. On the other hand, there are also graphs with

loops (Fig. 2.2), where belief propagation algorithm can not be guaranteed to find the global maximum. Nevertheless, a modification of max-product BP algorithm has been proved to be effective heuristic for MAP estimations in graph with loops [20].

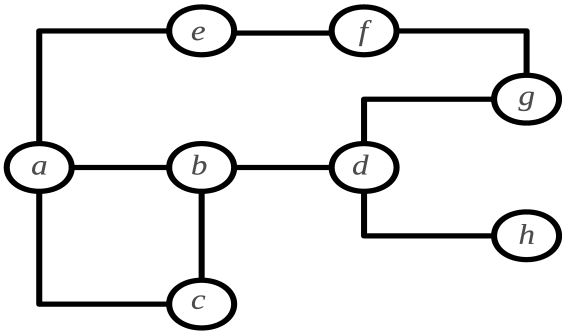
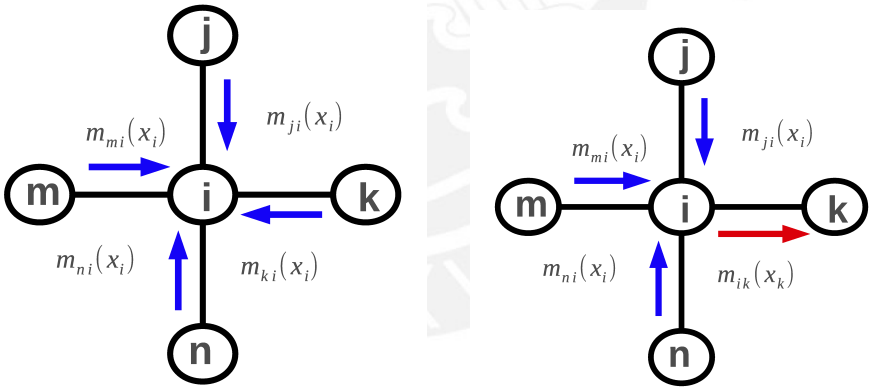


Figure 2.2: MRFs with loops. Nodes $\{a, b, c\}$ compose a loop as well as nodes $\{a, b, d, e, f, g\}$

We can find the min-marginal of $E(\mathbf{x})$ using (2.7). The min-marginal $M(x_i)$ is a function that provides information about the minimum values of the energy $E(\mathbf{x})$ under different constraints [21] and, under BP formulation, it can be also computed using messages using (2.9), where $N(j)$ refers to the neighbourhood of node j. In Fig. 2.3 is shown how message passing is perfomed.



(a) Message passing to compute (2.8) (b) Message passing to compute (2.9)

Figure 2.3: (a) When node i wants to send a message to node k, it has to collect all the messages from its neighbourhood except for k. (b) In order to compute $M(x_i)$ node i needs to gather all the messages from its neighbourhood.

$$M(x_i) = \min_{\mathbf{x}-\{x_i\}} E(\mathbf{x}) \tag{2.7}$$

$$M(x_i) = V_i(x_i) + \sum_{k\in\{N(i)\}} m_{ki}(x_i) \tag{2.8}$$

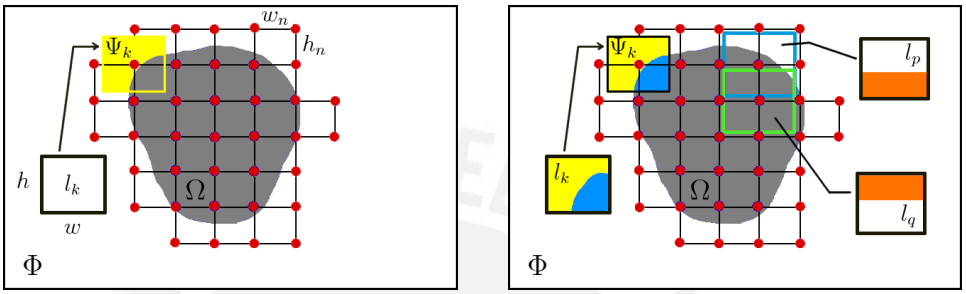
$$m_{ji}(x_i) = \min_{x_j} \left\{ V_j(x_j) + V_{ji}(x_j, x_i) + \sum_{k\in N(j)-\{i\}} m_{kj} \right\} \tag{2.9}$$

Therefore the estimation of MAP is found by minimizing function $M(x_i)$.

$$x_i^* = \underset{x_i}{\operatorname{argmin}}\, M(x_i) \tag{2.10}$$

2.2 Markov Random Field Model for image inpainting

In this section we succintly describe [13] a MRF solution to the inpainting problem on which our proposed algorithm is based. We also briefly mention some improvements to [13] as well as other works that include multi-scale approaches.



(a) MRF image model (b) MRF potentials

Figure 2.4: (a) MRF image model. Red dots are MRF nodes π_k, and black lines are edges ε. Ψ_k is the original patch of size h, w centered at node π_k and $\ell_k \in \mathcal{L}_k$ is a sample label. (b) Data cost V_k is calculated over yellow region. Pairwise potential V_{pq} is calculated over orange region

Given an input image, it is assumed that it can be divided in two regions: known or *source* region (Φ), and region to be filled or *target* (Ω). The target region is modeled as a group of nodes with a horizontal and vertical spacing of (h_n, w_n) pixels respectively. An undirected graph is constructed $G = (\nu, \varepsilon)$, where the set $\nu = \{\pi_k\}_{k=1}^N$ correspoding to all MRF nodes and ε is the set of all edges connecting adjacent MRF nodes and consist of 4-neighborhood system (see Fig. 2.4(a)). Let $\mathcal{L}_k = \{\ell_1, \ldots, \ell_n\}$ denotes the sample labels of node π_k such that $\mathcal{L}_k \in \Phi$. The idea is to find optimal labels candidate configuration $\Lambda = \{\ell_1^*, \ldots, \ell_N^*\}$; in this context, assigning optimal label ℓ_k^* to the node π_k is the same as copying the patch over the node's position. Let $V_p(\ell_p)$ denote the *data cost* such that $V_p = SSD(\Psi_p, \ell_p)$, where SSD is the sum of squared differences and Ψ_p is the original patch centered at node π_p, and ℓ_p is one of the sample labels that belongs to $\mathcal{L}_p$; and let V_{pq} denote the *pairwise potential* between neighbour nodes (π_p, π_q) such that $V_{pq} = SSD(\ell_p, \ell_q)$ in their overlap region (see figure 2.4(b)). Therefore, based on MRF model and these definitions, the energy function defined in (2.2) changes to:

$$E(\Lambda) = \sum_p V_p(\ell_p) + \sum_{p,q} V_{pq}(\ell_p, \ell_q) \tag{2.11}$$

The cost function (2.11) is minimized using the iterative algorithm described in section 2.1.2 called *max-product Belief Propagation*. During each iteration nodes their give "opinions" about other nodes by passing messages to their neighbourhood. Under this formulation, the message from node π_p to node π_q in graph G can be denoted as $m_{pq}(\ell_q)$,

which reflects how likely node π_p "believes" node π_q should be assigned label l_q. The updated message is given by (2.12), where $N(p)$ denotes the neighbourhood of node π_p.

$$m_{pq}^{t}(\ell_q) = \min_{\ell_p} \left\{ Vp(\ell_p) + Vpq(\ell_p, \ell_q) + \sum_{r \neq q, r \in N(p)} m_{rp}^{t-1}(\ell_p) \right\} \tag{2.12}$$

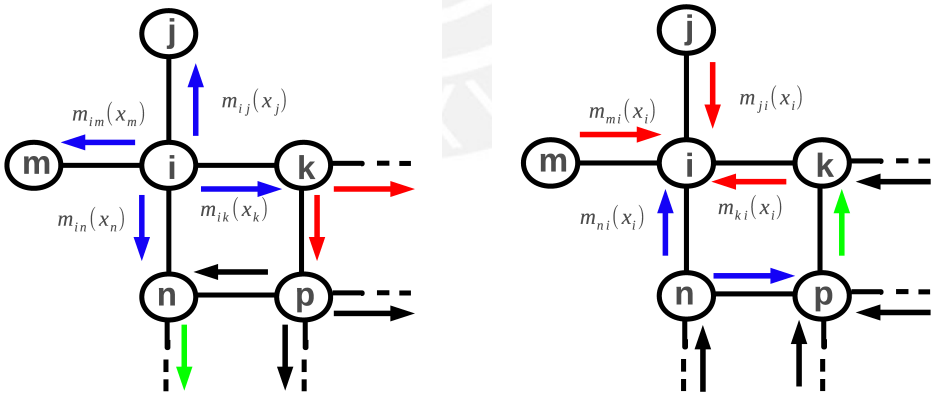
Once all messages have converged, each node π_p collects the messages from its neighbourhood and compute the min-marginal (2.10) or, under this formulation, their *belief* (2.14) which represents the probability of assigning label ℓ_p to node π_p. Finally, the optimal labels are found by (2.13):

$$\ell_p^* = \operatorname*{argmax}_{\ell_p} \{b(\ell_p)\} \tag{2.13}$$

$$b_p(\ell_p) = -V_p(\ell_p) - \sum_{r \in N(p)} m_{rp}(\ell_p) \tag{2.14}$$

2.2.1 Priority-BP

Belief propagation (BP) is a slow algorithm. If $|\mathcal{L}|$ denotes the total number of labels then the BP computational complexity is given by $O(|\mathcal{L}|^2)$. In [2] a new approach, called priority-BP, was proposed to reduce BP's complexity: at each iteration, the algorithm executes *ForwardPass* and then *BackwardPass*; the ForwardPass declare nodes, at the beginning, as uncommitted and visit them in order of highest priority (Fig. 2.5(a)), declaring as committed, prunning their lables, sending messages to their neighbour uncommitted nodes and updating their beliefs and priorities as well; in the BackwardPass nodes are visited in reverse order (Fig. 2.5(b)), declaring uncommitted, sending messages to their neighbour committed nodes, and updating belief and priorities.



(a) ForwardPass (b) BackwardPass

Figure 2.5: Message Passing in (a) FowardPass and (b) BackwardPass.Note that if we denote $P(i)$ as priority of node i, then $P(i) > P(k) > P(p) > P(n)$.

Let $b_p^{rel}(\ell_p)$ denote the relative belief of node π_p such that $b_p^{rel}(\ell_p) = b_p(\ell_p) - b_p^{max}(\ell_p)$, where $b_p^{max}(\ell_p) = \max_{\ell_p \in \mathcal{L}_p} b_p(\ell_p)$. And let b_{prune} be a predefined prune threshold such that the set of labels of node π_p is reduced, let denote as $\mathcal{L}_p^{prune} := \{\forall \ell_p \in \mathcal{L}_p : b_p^{rel}(\ell_p) \geq b_{prune}\}$.

Note that the computational complexity of priority-BP is $O(|\mathcal{L}_{max}|)$, where $\mathcal{L}_{max} = |\mathcal{L}_p^{prune}|$ and $\mathcal{L}_{max} \ll |\mathcal{L}|$. In practice a $\mathcal{L}_{min}$ and $\mathcal{L}_{max}$ are defined by the user.

2.2.2 Shortcomings of Priority-BP

The prune method proposed by [2] is still somehow inefficient because all possible labels are considered, which implies more memory and more computation time. Thus, some authors have proposed to perform a label selection as a first step. For example, in [22] it was proposed to use statistics of patch offset matching similar patches in the known region, obtaining their offsets and computing offsets histogram in order to prune labels. Nevertheless, this procedure requires a high-effienct implementation to avoid wasting computation time while computing offset matching. On the other hand, in [23] it was proposed a context-aware label selection, which limits the search for labels to the areas of interest based on contextual information using Gabor-based and color descriptors. However, Gabor filter computational complexity is $O(M^2N^2)$, where M and N are the $\max\{width, height\}$ for the filter and the image respectively.

2.3 Other multi-scale approches

Standard MRF-based image inpainting algorithms are of limited practical use when the input image is larger than 512x512. Some published works have proposed multi-scale solutions ([2], [12]), but they report lower quality results (than standard MRF-based image inpainting methods) due to fake local minimals. However, in [15] it was proposed to use also gradient information (SURF - descriptors [24]) while working with a multi-scale approach which improve the overall performance. Besides of being based on the intensity, descriptors used in [15] are based on the gradient which tend to be very practical useful while differentiating between geometric structures (edges).

As we explain in the introduction and [16], [17] we present a new method for inpainting, improving the computation time, based on a multi-scale approach, pre-selection of the labels (based on local statistics) and descriptors to avoid fake minimas.

Chapter 3

Multi-scale Image Inpainting algorithm

3.1 Proposed Algorithm

Based on the description of [2] as well as on the relationship between label selection and speed-up (Section 2.2), in this Section we propose a simple procedure based on local statistics to perform a label selection improving the computational performance of [2]. Moreover we also propose a multi-scale approach to improve the computational efficiency of our algorithm.

3.1.1 Label selection

We propose to guide the label selection using local statistics. To this end, we use [25], where a simple method was propose to estimate the noise variance (of the observed image) based only on its local variance; the local variance is computed using a window centered at pixel p; however, it is possible to get information about other features such as texture or edges if the window size is large enough.

The label selection is performed in a similar way as the *Forward Pass* in Priority-BP algorithm. Declaring nodes, at the beginning, as uncommitted and visiting them in order of highest priority, declaring as committed, selecting their lables and updating priorities. We propose a new scheme of priorities based on local variance and a confidence term that measures the known information in source patch Ψ_k (similar ideas have been used in [6]). Let $\sigma_k^2 = var\{k\}$ denote the local variance of the node π_k. Note that each σ_k^2 is a 3-dimensional vector ($\sigma_k^2 = [\sigma_{kR}^2, \sigma_{kG}^2, \sigma_{kB}^2]$) related to each color channel in a RGB image. And also let $V(k)$ and $C(k)$ denote the variance and confidence term, and $P(k)$ the priority of node π_k. They are calculated as follows:

$$C(k) = \frac{|\Psi_k \cap \Phi|}{|\Psi_k|} \tag{3.1}$$

$$V(k) = \frac{\|\sigma_k^2\|_2}{V_{max}} \tag{3.2}$$

$$P(k) = (1-\lambda)C(k) + \lambda V(k) \tag{3.3}$$

where the operator $|x|$ defines the number of elements of x, $V_{max} = \max_{\pi_k \in \nu} V(k)$ and $\lambda \in [0, 1]$. For each node π_k we calculate local variance using only known information near the node $\Psi_k \cap \Phi$ (Fig. 3.1). During the label selection process, it is necessary to check whether a node is reliable or not, so if $C(k) \geq \tau$ then $\sigma_k^2 = var\{\Psi_k \cap \Phi\}$, else we re-estimate the local variance of node π_k using (3.4) where $T = |N(k)| + 1$.

$$\sigma_k^2 = \frac{\sum_{i \in \{N(k) \cup \{k\}\}} \sigma_i^2}{T} \tag{3.4}$$

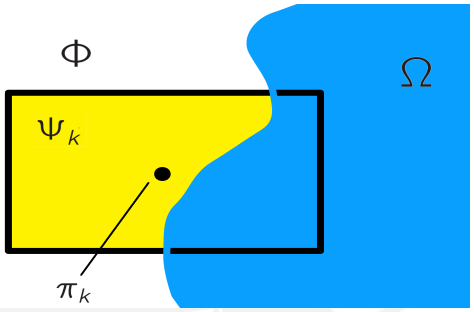


Figure 3.1: Local variance and confidence term of node π_k are calculated over the yellow region.

Let σ_{RGB}^2 denote the local variance vector of the input image such that $\sigma_{RGB}^2 := \{var\{p\} : p \in \Phi\}$. In order to perform the label selection of node π_k we first compute the distance vector d_n between σ_{RGB}^2 and σ_k^2 (3.5). The distance d_n is described by unimodal histogram (h_n), thus, it is possible to find a threshold τ_h using the unimodal thresholding algorithm described in [26], in which it focuses in find the maximum distance between the line the pass through the maximun and minimum value of h_n and the histogram itself, as show in figure 3.2. Finally, the set of labels of node π_k is denoted by $\mathcal{L}_k = \{\ell_1, \ldots, \ell_n\}$ such that ℓ_n is the patch centered at pixel $p \in \Omega^C$ which satisfy the condition $d_n \leq \tau_h$.

$$d_n = \|\sigma_{RGB}^2 - \sigma_n^2\|_2 \tag{3.5}$$

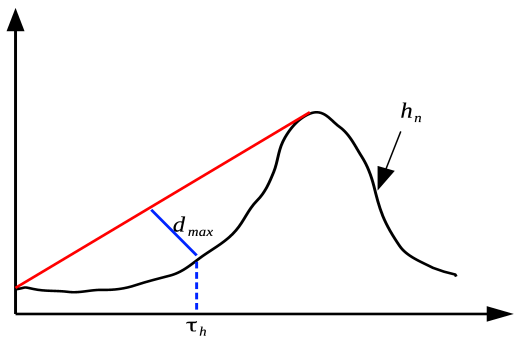


Figure 3.2: Unimodal thresholding described in [26].

3.1.2 Pixel decriptors

Following similar ideas, as in [15], we also use descriptors to avoid losing texture information of the original image while working with at lower levels; SURF-gradient descriptors tend to be very useful for differentiating between geometric structures (edges) because of their robustness and simplicity, and in [15] it has been shown that these descriptors give good results in multi-scale setting. The descriptors are defined as follows:

$$g_x(p) = \frac{1}{2^{2L}} \sum_{q \in N^0(p)} \nabla_x U(q) \tag{3.6}$$

$$g_y(p) = \frac{1}{2^{2L}} \sum_{q \in N^0(p)} \nabla_y U(q) \tag{3.7}$$

$$G_x(p) = \frac{1}{2^{2L}} \sum_{q \in N^0(p)} |\nabla_x U(q)| \tag{3.8}$$

$$G_y(p) = \frac{1}{2^{2L}} \sum_{q \in N^0(p)} |\nabla_y U(q)| \tag{3.9}$$

where $U = \frac{I_R + I_G + I_B}{3}$ denotes the image intensity of the RGB input image I, and $N^0(p)$ is the 2^Lx2^L square set of pixels of the original image that are represented by pixel p at Lth level. Descriptors $g_x(p)$ and $g_y(p)$ give us information about gradual changes of the gradient and its direction on each axis. On the other hand, $G_x(p)$ and $G_y(p)$ give us information about the "magnitude" of the gradient on each direction. Some examples are shown in Fig. 3.3.

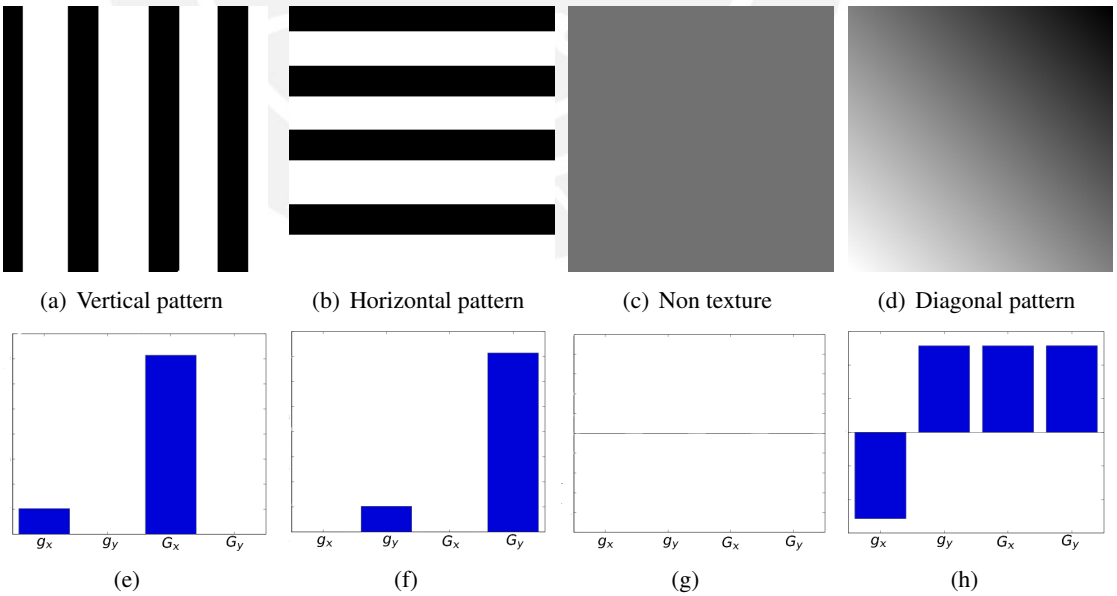


Figure 3.3: From top to bottom, the pattern and its SURF description. The SURF descriptors are presented in the following order g_x, g_y, G_x, G_y. It is easy to see that patterns (a) and (b) are described by (e) and (f). Since pattern (c) has no texture, descriptors are equally to zero. Finally, pattern (d) has components in both directions, thus, none descriptor is zero.

Algorithm 1: Multi-scale Image Inpainting

Initialization at level M;
Perform label selection;
Belief based label pruning;
Propagate the remaining labels to the upper level;
Multiresolution Pruning;
for $k \leftarrow M-1:1$ **do**
    Similarity based label pruning;
    Propagate the remaining labels to the upper level;
At level zero;
Similarity based label pruning ;
Priority-BP algorithm ;
Perform hole filling;

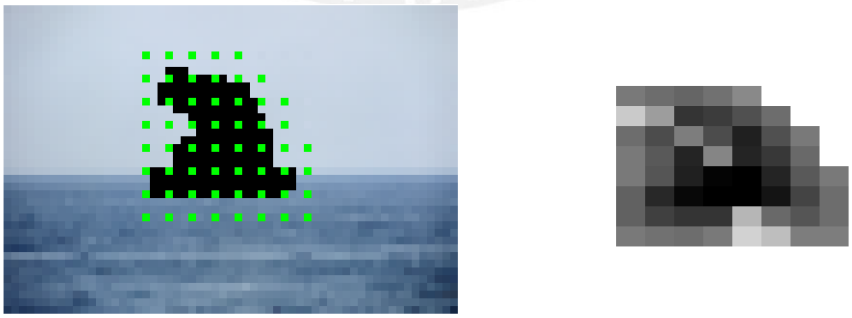
Note that at *L*th low-resolution level, each pixel p will be described by color and texture (gradient) features, as follows

$$I^L(p) = \{r(p)\,, g(p)\,, b(p)\,, g_x(p)\,, g_y(p)\,, G_x(p)\,, G_y(p)\}$$

where I^L denotes the low resolution image. The first three components are the color information, and the last four are the texture information. In our algorithm, while working at low-resolution levels we use this pixel representation to compute data cost V_p and pairwise potential V_{pq}.

3.1.3 Multi-scale image inpainting

Our image inpainting method is based on a multi-scale framework, and it is summarized in Algorithm 1. The main characteristics of our method are (i) the label selection using local statistics in step 1, (ii) the multi-scale pruning process in step 2 and (iii) a new scheme of priorities based on the confidence and variance/belief/similarity (depends on the case, the first one is used in label selection and the other two are used in pruning process) terms.



(a) Markov Random Field (b) Priority map

Figure 3.4: (a) Green nodes are all the nodes $\pi_k^L \in \nu^L$. (b) Priority map correspond to the first iteration at the lowest level, the darker the color the lower priority

Now that we are describing our approach, it is neccesary to fix some notations. Let denote Π be a set of grids such that $\Pi = \{\nu^0, \ldots, \nu^L\}$, where ν^l is the set of nodes at the

l^{th} level, and let denote π_k^l such that $\pi_k^l \in \nu^l$ and each node has a set of labels $\mathcal{L}_k^l$. As we mentioned before, we use local variance based segmentation to select labels $\mathcal{L}_k^L$ for each node π_k^L at the lowest resolution. Then, at this level we perform one iteration *ForwardPass* from priority-BP algorithm [2] in order to have a compact and high-confidence set of labels for each node which will be refined in uppers levels. In the following levels we perform the prunning process described in [23] because it is faster than belief based prunning process and it works properly while working with a reliable predefined set of labels. This process is similar as one iteration *ForwardPass* but here we stress that in [2] *similarities* rather than *beliefs* are used as a measure in the pruning process. Let $S_p(\ell_p^l)$ denote the similarity such that initially $S_p(\ell_p^l) = V_p(\ell_p^l), \forall \ell_p^l \in \mathcal{L}_k^l \wedge l \in \{L-1, L-2, \ldots, 1\}$. Once both priorities and similirities of all nodes have been initalized, *"ForwardPass"* is executed. The similarity update equation is defined by (3.10)

$$S_q(\ell_q^l) = S_q(\ell_q^l) + min\left\{V_{pq}(\ell_p^l, \ell_q^l)\right\}, \forall \ell_q^l \in \mathcal{L}_j^l, \tag{3.10}$$

where $\mathcal{L}_j^l$ is the set of labels of neighbour node π_j^l of π_k^l at l^{th} level. In either case Belief or Similarity based label pruning, it is necessary to have a priority scheme wich have influence in the quality of the image inpainting problem in general. For each node $\pi_k^l, l \in \{L, \ldots, 0\}$, we propose to combine the belief/similarity based term with the confidence term at the l^{th} level using (3.11). Note that only the confidence term will not change during each iteration of pruning process, while the other term will be updated.

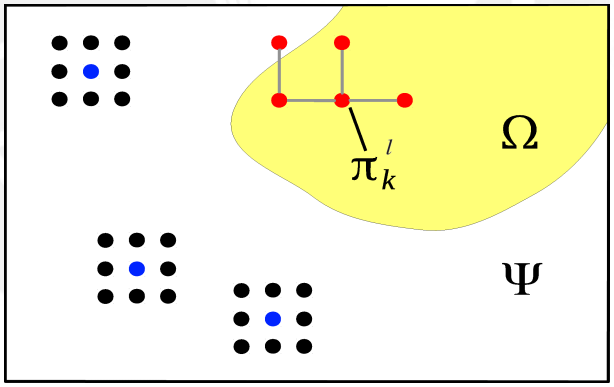


Figure 3.5: Blue points represent the upsampled set of labels $\mathcal{L}_k^{l+1}$ of node π_k^{l+1} at certain level l, and black points represent the propagation of each label $\ell_k^l \in \mathcal{L}_k^l$.

$$P^l(k) = (1-\lambda)C^l(k) + \lambda p_{b,s}^l(k) \tag{3.11}$$

$$p_b^l(k) = \frac{1}{|b_k^{rel}(\ell_k^l) \geq b_{conf}|} \tag{3.12}$$

$$p_s^l(k) = \frac{1}{|S_k(\ell_k^l) \leq T_{sim}|} \tag{3.13}$$

where $p_{b,s}^l$ is the priority using beliefs (3.12) or similarities (3.13), $\lambda \in [0,1]$ and T_{sim}, b_{conf}

are predefined thresholds. With this new scheme we ensure that nodes near source region and specially the ones that have less labels will be pruned earlier (less labels implies more node confidence about its label choice) . In figure 3.4(b) is shown the priority map of 3.4(a) and as it was expected visiting order in the missing region is inwards from the boundary.

After pruning proccess, we propagate reamining labels to upper level such that we refine label selection using a window search around propagated labels as it is shown in figure 3.5. In practice, the windows search is defined as a set of offsets $W = \{(x, y) : x, y \in \{-1, 0, 1\}\}$. Finally, on the original image we perfom Priority-BP algorithm to get the final label for each node.

Chapter 4

Experimental Results

In this chapter, we evaluate the proposed image inpainting algorithm on a variety of natural images and we compare to well-known algorithms "Region Filling and Object Removal by Exemplar-Based Image Inpainting" [6] and " Image Completion Using Efficient Belief Propagation via Priority Scheduling and Dynamic Pruning" [2]. We present results for two cases: object removal and sratches, and we also provide an error analysis. Our multi-scale approach with label selection and [6] have been implemented in MATLAB, while [2] is a C++ opensource implementation (http:// lafarren.com/image-completer/). We run all the implementations on a Intel i7-2630QM @ 2.00GHz with 6GB RAM.

4.1 Object Removal

Object removal is a real concern in image inpainting. We solve this problem using our approach and our results are compared against [6] and [2]. Settings and results are presented on table 4.1. Results show that our results are competitive, in terms reconstruction quality, when compare to the ones obtained using the algorithm described in [2].

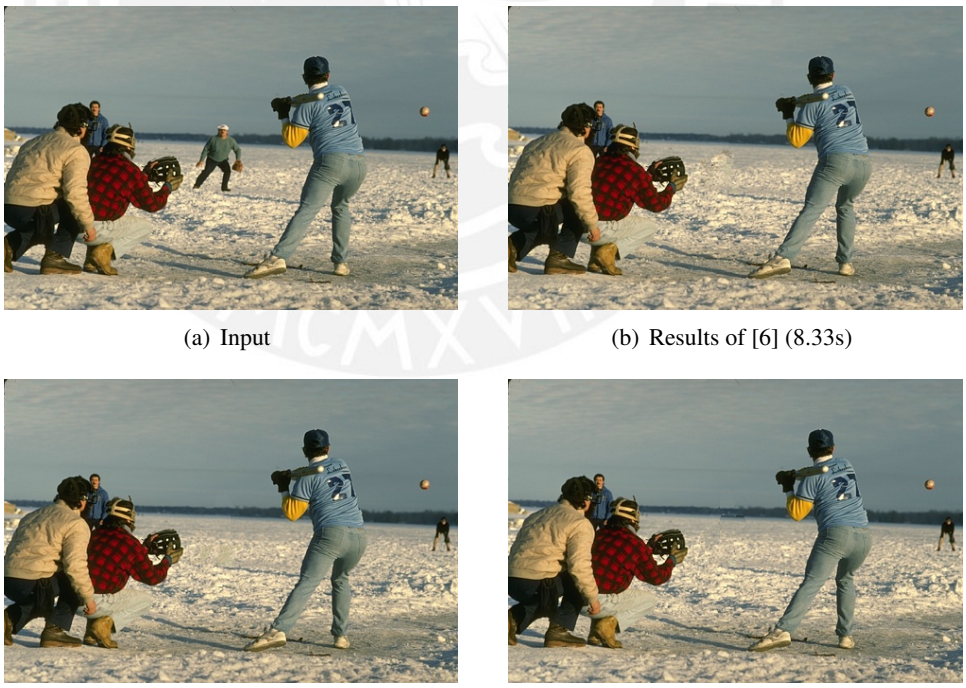
The computational results show that our method outperfoms the original Markov Random Field based inpainting method proposed by Komodakis *et al.* [2]. It is 10-15 times faster than C++ implementation of the original global optimization image inpainting (which is a significant computational improvement) and it is comparable to the greedy algorithm proposed by [6].

	Blackbird	**Baseball**	**Ruins**	**Oregon**	**Giraffes**	**Golf**
Criminisi's [6]	58.32s	8.33s	8.86s	46.54s	5.51s	31.19
Komodakis' [2]	980.63s	47.54s	48.60s	119.42s	53.46s	123.79s
Our approach	47.85s	5.75s	11.66s	9.79s	6.66s	11.56s
	Multi-Scale and label selection Settings					
Patch Radius w	2	2	2	3	2	3
Patch Radius w_v	2	2	3	4	2	4
Number of Levels	3	3	3	3	3	3

Table 4.1: Summarized computation time result for test images.

(a) Input (b) Results of [6] (58.32s) (c) Results of [2] (980.63s) (d) Ours (47.85s)

Figure 4.1: Inpainting results for ”BlackBird”. (a) Input image, results of (b) Criminisi [6], (c) Komodakis [2] and (d) our method.



(a) Input (b) Results of [6] (8.33s)

(c) Results of [2] (47.54s) (d) Ours (5.75s)

Figure 4.2: Inpainting results for ”baseball”. (a) Input image, results of (b) Criminisi [6], (c) Komodakis [2] and (d) our method.

(a) Input

(b) Results of [6] (8.86s)

(c) Results of [2] (48.60s)

(d) Ours (11.66s)

Figure 4.3: Inpainting results for "Ruins". (a) Input image, results of (b) Criminisi [6], (c) Komodakis [2] and (d) our method.

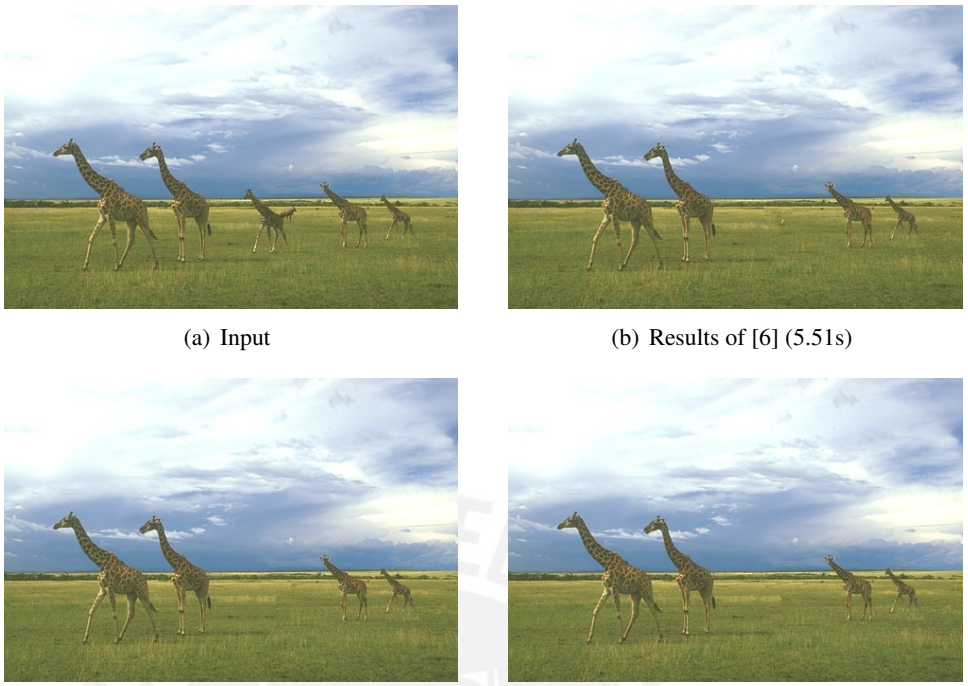
(a) Input

(b) Results of [6] (46.54s)

(c) Results of [2] (119.42s)

(d) Ours (9.79s)

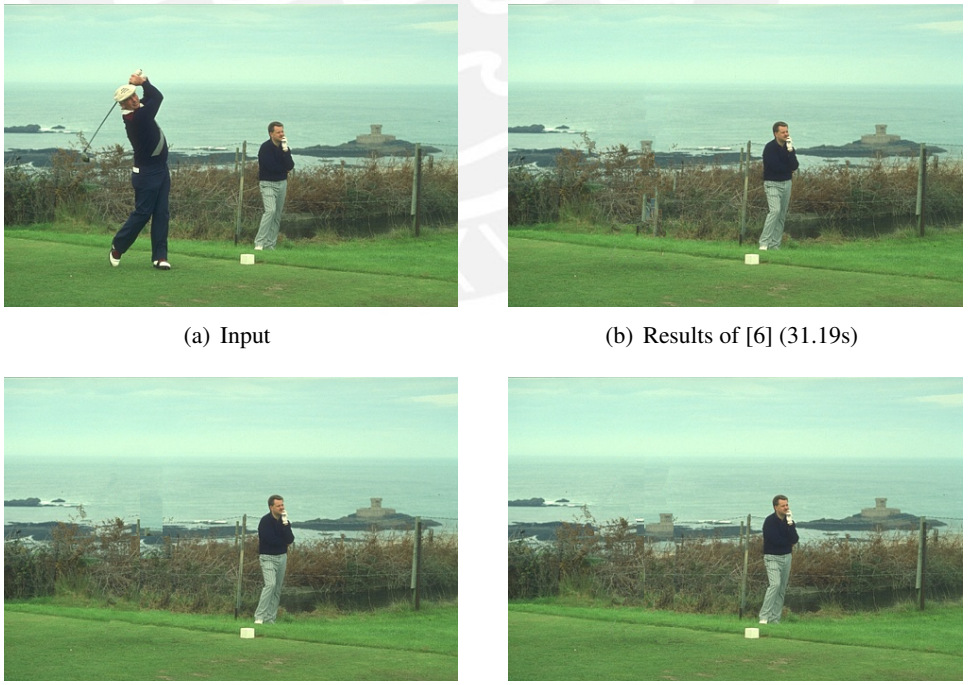
Figure 4.4: Inpainting results for "Oregon". (a) Input image, results of (b) Criminisi [6], (c) Komodakis [2] and (d) our method.



(a) Input (b) Results of [6] (5.51s)

(c) Results of [2] (53.46s) (d) Ours (6.66s)

Figure 4.5: Inpainting results for ”Giraffes”. (a) Input image, results of (b) Criminisi [6], (c) Komodakis [2] and (d) our method.



(a) Input (b) Results of [6] (31.19s)

(c) Results of [2] (123.79s) (d) Ours (11.56s)

Figure 4.6: Inpainting results for ”Golf”. (a) Input image, results of (b) Criminisi [6], (c) Komodakis [2] and (d) our method.

4.2 Parameters dependency

We analyse how the parameters related to our approach affects inpainting results. The parameters to be analysed are label selection, patch radius used for label selection w_v, number of levels, gradient-based descriptors also the influence of patch radius rate or gain ($Gain = w^l/w^{l+1}, l \in \{0, \ldots, L-1\}$).

4.2.1 Label selection

The inpainting results are improved by performing the label selection; indeed, artifacts on output images can be reduced when using proper criterion for label selection (Fig. 4.7). In addition, it can be observed that there is another advantage on performing the label selection based on local statistics, lower computational times are achieved.

(a) Input (320x213) (b) Without label selection (10.33s) (c) Complete proposed method (6.91s)

Figure 4.7: Inpainting quality and computation time are dependent on label selection

4.2.2 Patch radius for label selection

Through the filling hole process while using our algorithm it is important to setup the patch radius (w_v) or size to be used when estimating local variance. In most of the cases it is enough to use the same patchsize used to create MRF grid for label selection; however, what happens if it is not possible to have a good local variance estimated due to fact there is no enough reliable information, specially near boundaries. In these cases it is needed to increase the window search to improve the estimation, but without making changes to the original MRF grid. Thus, it is important to be able to choose, either automatically or manually, the window search to estimate the local variance.

In figure 4.8 it is observed the influence of choosing different w_v. The setting for the example was: patch radius at lowest level w^L for MRF model is 2, $gain = 2$ and $\#L = 4$. In this case choosing $w_v > w^L$ have improved the results.

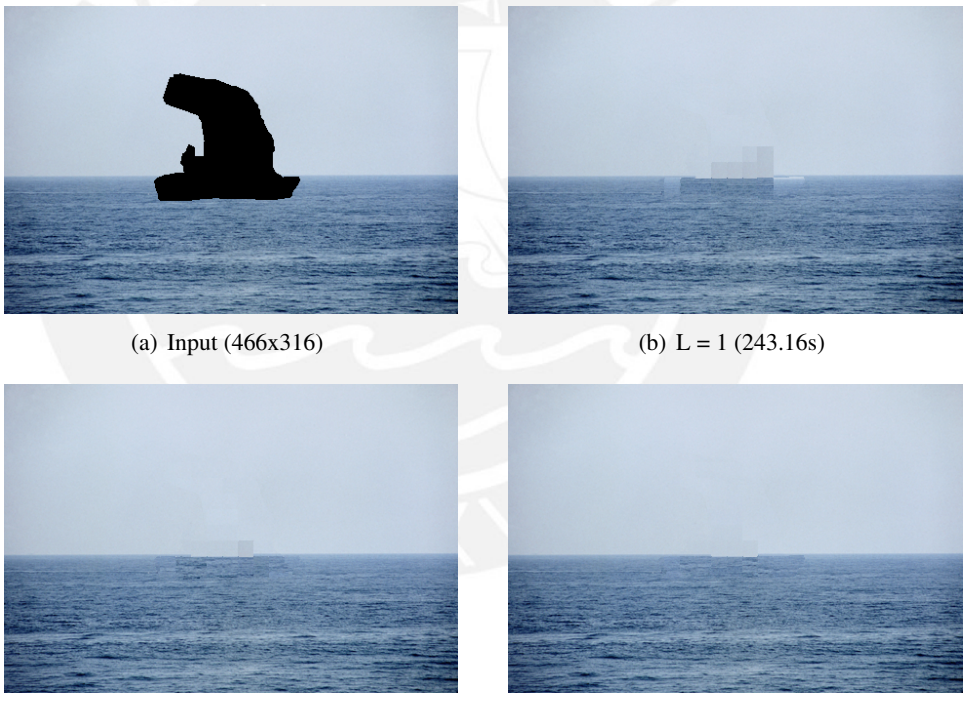
4.2.3 Number of levels

The computation time, as it is expected, is highly dependent on the number of levels and it can have some impact on the final results as well. We can observe in figure 4.9 the results by choosing $L = 1, 2, 3$. For $L = 1$ (Fig. 4.9b), result image presents some artifacts, and these artifacts disappear when using more levels. However, more levels does not necessary mean better quality, it will depends on the input image, also do not forget that increasing or

(a) Input (321x481) (b) $w_v = 2$ (20.56s) (c) $w_v = 6$ (20.59s) (d) $w_v = 9$ (20.53s)

Figure 4.8: Dependency of patch radius for label selection.

decreasing the value of this parameter will affect computation time dramatically. Number of levels is primarily limited by the gain and initial patchsize, in an extreme case the final patchsize could be so large that the output image could present artifacts or inconsistencies.



(a) Input (466x316) (b) L = 1 (243.16s)

(c) L = 2 (19.27s) (d) L = 3 (8.50s)

Figure 4.9: Computation time is highly dependent on number of levels.

4.2.4 Gradient-based descriptors

In chapter 3, we described gradient-based descriptors which allow us to use texture and structure information from original image at different levels. This descriptors have been proved to be effective in [15]. By using gradient-based descriptors, in our case SURF descriptors, it is possible to avoid fake minimas when estimating the set of labels for each node. In figure 4.10 we can observe the improvement on the results just by using SURF

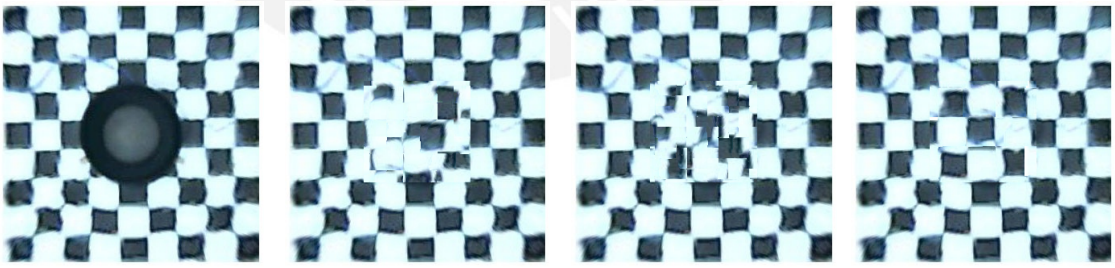
descriptors, specially in the part of shadows. However, the drawback is reflecting in the computation time (fig. 4.10), since the vector that represents each pixel has more dimensions.

(a) Input (321x481) (b) without SURF descriptors (14.94s) (c) with SURF descriptors (39.68s)

Figure 4.10: Importance of gradient-based descriptors.

4.2.5 Patch radius rate

The patch radius rate or gain is defined as the relation between patch radius at level l and its immediately lower level $l+1$, such that $Gain = w^l/w^{l+1}, l \in \{0, \ldots, L-1\}$. Gain could not be lower than 1 or greater than 2, in either case it has no sense since we are going from low resolution levels to high ones and also the rate between two images from consecutive levels is 2 for each dimension. The final result is highly dependent of this rate, as we can observe in figure 4.11, for a gain of 2 the inpainting process outperforms, in term of quality, the other two cases. Nevertheless, it does not mean that the higher the gain the better the results, the optimal value should depend on image context aware (textures and structures).



(a) Input (330x330) (b) Gain = 1.2 (12.62s) (c) Gain = 1.5 (11.90s) (d) Gain = 2.0 (12.32s)

Figure 4.11: Patch radius rate dependency.

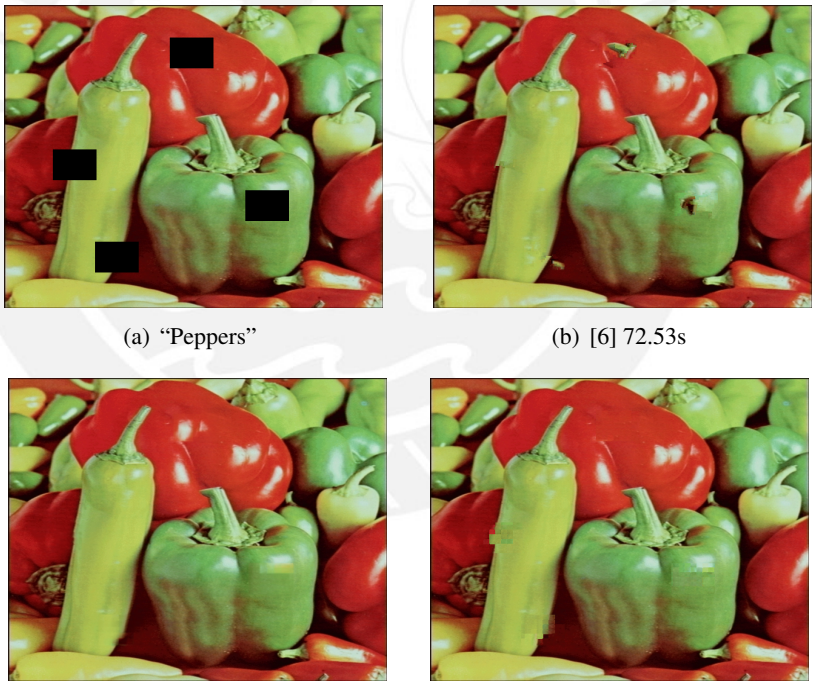
4.3 Error analysis

In this sections we focus on restoring images. For quality measure we used signal-to-noise ratio (SNR) and structural similarity indexes [27]. We compare our method against [6] and

[2] in two specific task, block recovery and scratchs. Processing times are presented below each result. As it can be observed in table 4.2, proposed method shows best performance in block recovery task. On the other hand, in scratchs the performance is similar than [2], but still better than [6].

	"Peppers"			**"PalmTree"**		
	SNR	**SSIM**	**Time**	**SNR**	**SSIM**	**Time**
[6]	12.8692dB	0.9093	72.53s	18.4224dB	0.9544	26.93s
[2]	13.1594dB	0.9184	784.14s	16.6563dB	0.9274	118.82s
Ours	15.7277dB	0.9530	63.92s	18.8355dB	0.9578	46.75s
	"Buildings"			**"Ostrich"**		
	SNR	**SSIM**	**Time**	**SNR**	**SSIM**	**Time**
[6]	27.3859dB	0.9929	16.06s	12.3052dB	0.8704	9.51s
[2]	26.6868dB	0.9916	83.61s	14.5442dB	0.9272	164.06s
Ours	18.3638dB	0.9424	24.74s	14.0931dB	0.9198	69.74s

Table 4.2: Results of Error analysis. "Peppers" and "PalmTree" were used for block recovery test, and "Buildings" and "Ostrich" for scratches recovery test. The indexes were calculated over corrupted (black) regions



(a) "Peppers" (b) [6] 72.53s

(c) [2] 784.14s (d) Ours 63.92s

Figure 4.12: Results for block recovery. (a) Input image, (b) results of [6] (SNR=12.87 dB, SSIM=0.9093), (c) results of [2] (SNR=13.16 dB, SSIM=0.9184) and (d) our results (SNR=15.73 dB , SSIM=0.9530)

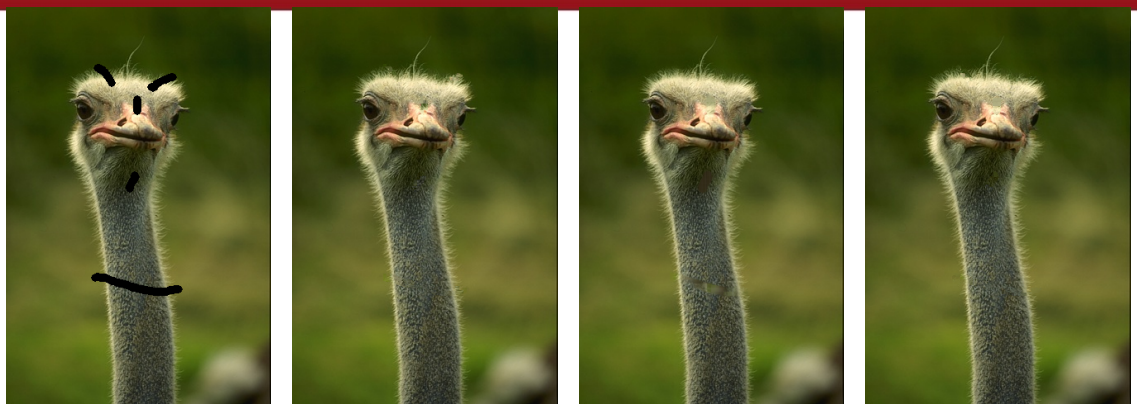
(a) "PalmTree" (b) [6] 26.93s (c) [2] 118.82s (d) Ours 46.75s

Figure 4.13: Results for block recovery. (a) Input image, (b) results of [6] (SNR=18.42 dB, SSIM=0.9544), (c) results of [2] (SNR=16.66 dB, SSIM=0.9274) and (d) our results (SNR=18.84 dB , SSIM=0.9578)

(a) "Buildings" (b) [6] 16.06s

(c) [2] 83.61s (d) Ours 24.74s

Figure 4.14: Results for scratches.(a) Input image, (b) results of [6] (SNR=27.39 dB, SSIM=0.9929), (c) results of [2] (SNR=26.69 dB, SSIM=0.9916) and (d) our results (SNR=18.37 dB , SSIM=0.9424)



(a) "Ostrich" (b) [6] 9.51s (c) [2] 164.06s (d) Ours 69.74s

Figure 4.15: Results for scratches.(a) Input image, (b) results of [6] (SNR=12.31 dB, SSIM=0.8704), (c) results of [2] (SNR=14.54 dB, SSIM=0.9272) and (d) our results (SNR=14.09 dB , SSIM=0.9198)

Chapter 5

Conclusions

A novel formulation for Markov Random Field image inpainting models have been presented. The proposed method uses a multi-scale approach to solve the inpainting problem in which we compensate the loss of information in low resolution levels by using gradient information of the original image. We also used local statistics to perform a initial label selection in the lowest level in order to reduce the number of labels per MRF node.

Our computational simulations show that the reconstruction quality of our approach is of comparable quality to results obtained by the original MRF based inpainting method, as well as to other exemplar-based inpaiting algorithms. Moreover, our approach is at least one order of magnitude faster than the original MRF based inpainting method, while at the same time is competitive with exemplar-based inpainting algorithms. Although the reconstruction quality of our method is comparable to that of exemplar-based methods, we acknowledge that it depends on several parameters. However, from our analisys, it seems possible to improve our proposed algorithm so that it could choose the optimal values automatically; this would be considered in future developments.

Recommendations

- The presented method is highly dependent on the parameters and users tend to waste time finding, subjectively, the best setting possible. Thus, it is important to develop methods or algorithms to tune parameters automatically.
- Our multi-scale image inpainting methods fails when the ”hole” is large compared to optimal patchsize (assuming that the other parameters are fixed). We suggest to use other or more features to perform label selection because local variance a simple one and during variance propagation local variance of inner nodes tends to zero. Alternatively we could develop a new method for label selection to solve variance loss.
- It could be interesting to develop a C or C++ code of our method and compare the results with the ones obtained in MATLAB.

Bibliography

[1] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *SIGGRAPH*, 2000, pp. 417–424.

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

[3] V. Kwatra, A. Schdl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Transactions on Graphics, SIGGRAPH 2003*, vol. 22, no. 3, pp. 277–286, July 2003.

[4] T. Chan and J. Shen, "Non-texture inpainting by curvature-driven diffusions (cdd)," *The Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436–449, 2001.

[5] T. Chan and J. Shen, "Variational image inpainting," *Communications on Pure and Applied Mathematics*, vol. 58, pp. 579–619, 2005.

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

[7] T. Kwok, H. Sheung, and C. Wang, "Fast query for exemplar-based image completion," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3106–3115, Dec. 2010.

[8] J. Wu and Y. Chou, "An effective content-aware image inpainting method," *Journal of Information Science and Engineering*, vol. 28, no. 4, pp. 755–770, 2012.

[9] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *In International Conference on Computer Vision*, 1999, pp. 1033–1038.

[10] P. Harrison, "A non-hierarchical procedure for re-synthesis of complex textures," in *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2001, pp. 190–197.

[11] J. Sun, L. Yuan, J. Jia, and H. Shum, "Image completion with structure propagation," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 861–868, July 2005.

[12] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007.

[13] N. Komodakis, "Image completion using global optimization," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, Washington, DC, USA, 2006, CVPR '06, pp. 442–452.

[14] N. Kawai, T. Sato, and N. Yokoya, "Image inpainting considering brightness change and spatial locality of textures and its evaluation," in *Pacific-Rim Symposium on Image and Video Technology*, 2009, pp. 271–282.

[15] Y Liu and V Caselles, "Exemplar-based image inpainting using multiscale graph cuts," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1699–1711, 2013.

[16] D. Paredes and P. Rodríguez, "Multi-scale image inpainting with label selection based on local statistics," in *21st European Signal Processing Conference (EUSIPCO)*, 2013, pp. 1–5.

[17] D. Paredes, P. Rodríguez, and N. Ragot, "Catadioptric omnidirectional image inpainting via a multi-scale approach and image unwrapping," in *Proceedings of the IEEE International Symposium on Robotic and Sensors Environments*, 2013, pp. 67–72.

[18] P. Bourgeois, P. Rodriguez, and N. Ragot, "Omnidirectional catadioptric image unwrapping via total variation regularization.," in *Proceedings of the IEEE International Symposium on Robotic and Sensors Environments*. 2011, pp. 220–225, IEEE.

[19] S. Roebert, T. Schmits, and A. Visser, "Creating a bird-eye view map using an omnidirectional camera," in *Proceedings of the 20th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC 2008)*, October 2008, pp. 233–240.

[20] A. Blake and P. Kohli, "Introduction to markov random fields," in *Markov Random Fields for Vision and Image Processing*, A. Blake, P. Kohli, and C. Rother, Eds., pp. 297–310. MIT Press, 2011.

[21] Pushmeet Kohli and Philip H. S. Torr, "Measuring uncertainty in graph cut solutions - efficiently computing min-marginal energies using dynamic graph cuts," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 30–43.

[22] K. He and J. Sun, "Statistics of patch offsets for image completion," in *European Conference on Computer Vision*, 2012, pp. 16–29.

[23] T. Ruzic, A. Pizurica, and W. Philips, "Markov random fields based image inpainting with context-aware label selection," 2012, International Conference on Image Processing, pp. 1733–1736.

[24] H. Bay, T. Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*, 2006, pp. 404–417.

[25] S. Aja-Fernández, G. Vegas-Sánchez-Ferrero, M. Martín-Fernández, and C. Alberola-López, "Automatic noise estimation in images using local statistics. additive and multiplicative cases," *Image Vision Comput.*, vol. 27, no. 6, pp. 756–770, 2009.

[26] P. Rosin, "Unimodal thresholding," *Pattern Recognition*, vol. 34, no. 11, pp. 2083–2096, 2001.

[27] Zhou Wang, Ligang Lu, and Alan C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, 2004.