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

Daniel Paredes, Paul Rodriguez

Electrical Department
Pontificia Universidad Católica del Perú
Lima, Perú

ABSTRACT

In this paper, 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 reduce the number of the $\mathcal{L}$ (feasible) labels by an appropriate 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.

Index Terms— inpainting, multi-scale, local statistics, Markov Random Field

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, such as scaling-up images by superresolution, reconstructing old photographs, and removal of overlaid text or graphics.

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. Inspired by this diffusion method, numerous models, including variational approaches [2] and curvature-driven diffusion (CDD) [3], are incorporated into inpainting task. However, diffusion-based methods can only fill small gaps (i.e. scrathes), mostly, on non-textured images.

On the other hand, patch-based approaches ([4], [5]) are proposed to address inpainting problems with large missing image portions. Similar to well studied domain texture synthesis ([6], [7]), their results can be generated by sampling and copying the best matching color values from known regions to missing regions at patch level.

Even though, common patch-based methods generate better results than diffusion-based, they are greedy methods, and thus, inpainted images may present visual inconsistencies. To overcome this limitation, global optimization models have been proposed and the state-of-the-art models are based on Markov Random Fields (MRF) ([8], [9], [10], [11]).

In this paper, 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 [12], 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, thus fake local minima are avoided. This procedure gives a substantial speedup when compared with previous methods.

The paper is organized as follows. In Section 2 we briefly review the well-known Markov Random Field (MRF) based inpainting method as well as some of its improvements. Then our proposed method is described in Section 3. Experiments and computational results are presented in Section 4. Finally, we give our concluding remarks in Section 5.

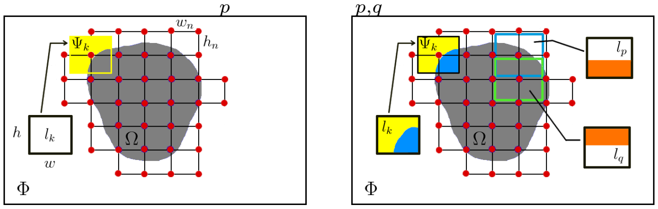
2. MARKOV RANDOM FIELD (MRF) MODEL AND GLOBAL OPTIMIZATION PROBLEM

In this section we summarily describe [10] a MRF solution to the inpainting problem on which our proposed algorithm is based. We also briefly mention some improvements to [10] as well as other works that also include multi-scale approaches.

2.1. Markov Random Field Model for image inpainting

Given an input image, it is assumed that it could 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. ε is the set of all edges connecting adjacent MRF nodes and consist of 4-neighborhood system (see Figure 1a). Let $\mathcal{L}_k = \{l_1, \ldots, l_n\}$ denote the sample labels of node π_k such that $\mathcal{L}_k \in \Phi$. The idea is to find optimal labels candidate configuration $\Lambda = \{l_1^*, \ldots, l_N^*\}$; in this context, assigning optimal labels l_k^* to a node π_k is the same as copying the patch over the node's position. Let $V_p(l_p)$ denote the *data cost* such that $V_p = SSD(\Psi_p, l_p)$, where SSD is the sum of squared differences and Ψ_p is the original patch centered at node π_p, and l_p is a sample label of node π_p; and let V_{pq} denote the *pairwise potential* between neighbour nodes (π_p, π_q) such that $V_{pq} = SSD(l_p, l_q)$ in their overlap region (see figure 1b). Therefore, based on MRF model and these definitions, the energy function was defined as follow:

$$E(\Lambda) = \sum_p V_p(l_p) + \sum_{p,q} V_{pq}(l_p, l_q) \quad (1)$$



(a) MRF image model

(b) MRF potentials

Fig. 1: (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 sample label is $l_k \in \mathcal{L}_k$. (b) Data cost V_k is calculated over yellow region. Pairwise potential V_{pq} is calculated over orange region

The cost function (1) is minimized using an iterative algorithm called *max-product Belief Propagation* [8]. Let denote $N(p)$ be the neighbourhood of node p. During each iteration nodes give "opinions" by passing messages to their neighbourhood. In this formulation, the message from node π_p to node π_q in graph G can be denoted as $m_{pq}(l_q)$, which reflects how likely node π_p "believes" node π_q should be assigned label l_q. The updated message is given by

$$m_{pq}^t(l_q) = min\left\{ Vp(l_p) + Vpq(l_p, l_q) \ + \sum_{r \neq q, r \in N(p)} m_{rp}^{t-1}(l_p) \right\} \quad (2)$$

Once all messages have converged, each node π_p collect the messages from its neighbourhood and compute their *belief* (4) which represents the probability of assigning label l_p to node π_p. Finally, the optimal labels are found by (3):

$$l_p^* = argmax\{b(l_p)\} \quad (3)$$

$$b(l_p) = -V_p(l_p) - \sum_{r \in N(p)} m_{rp}(l_p) \quad (4)$$

2.2. 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 [12] 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, declaring as committed, prunning their lables, sending messages to their neighbour uncommitted nodes and updating their beliefs and priorities as well; the BackwardPass nodes are visited in reverse order, declaring uncommitted, sending messages to their neighbour committed nodes, and updating belief and priorities. The computational complexity of priority-BP is $O(|\mathcal{L}_{max}|), \mathcal{L}_{max} \ll |\mathcal{L}|$, where $\mathcal{L}_{max}$ is defined by the user.

2.2.1. Shortcomings of Priority-BP

The prune method proposed by [12] is still somewhat 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 [13] it was proposed to use statistics of patch offset matching similar patches in the known region and obtaining their offsets.On the other hand, in [14] 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.

2.3. Other multi-scale approches

Standard MRF-based image inpainting algorithms are of limited practical use when the input image is bigger than 512x512. Some published works have proposed multi-scale solutions ([12], [9]), but they report lower quality results (than standard MRF-based image inpainting methods) because of fake local minimals. However, in [11] it was proposed to use the gradient information as aditional descriptor (SURF - descriptors [15]) while working with a multi-scale approach which improve the overall performance.

3. PROPOSED ALGORITHM

Based on the description of [12] as well as on the relation between label selection and speed-up (Section 2.2), in this Section we proposed a simple procedure 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 for label selection and pruning process.

3.1. Label selection

We proposed to guide the label selection using local statistics. To that end, we use [16], where a simple method was proposed 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 big 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 proposed 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 [4]). Let $\sigma_k^2 = var\{k\}$ denote the local variance of the pixel $k \in \Omega^C$. Note that each σ_k^2 is a 3-dimensional vector ($\sigma_k^2 = [\sigma_{k_R}^2, \sigma_{k_G}^2, \sigma_{k_B}^2]$). 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:

$$C(k) = \frac{|\Psi_k \cap \Phi|}{|\ \Psi_k\ |}, \qquad V(k) = \frac{\|\sigma_k^2\|_2}{V_{max}} \tag{5}$$

$$P(k) = (1-\lambda)C(k) + \lambda V(k) \tag{6}$$

where the operator $|x|$ defines the number of elements of x, $V_{max} = \max_{\pi_k} V(k)$ and $\lambda \in [0,1]$. For each node π_k we calculate local variance using only known information near the node $\Psi_k \cap \Phi$. During the label selection process 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-estimated the local variance of node π_k using (7), where $T = |N(k)| + 1$.

$$\sigma_k^2 = \frac{\sum_{i=q\in\{N(k)\cup\{k\}\}} \sigma_i^2}{T}, \qquad d_k = \|\sigma_{RGB}^2 - \sigma_k^2\|_2. \tag{7}$$

Let σ_{RGB}^2 denote the local variance vector of the input image such that $\sigma_{RGB}^2 := \{var\{p\} : p \in \Psi\}$. In order to perform the label selection of node π_k we first compute the distance vector d_n between local variance image σ_{RGB}^2 and σ_k^2 (7). 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 [17]. Finally, the set of labels of node π_k is denoted by $\mathcal{L}_k = \{l_1, \ldots, l_n\}$ such that l_n is the patch centered at pixel $p \in \Omega^C$ which satisfy the condition $d_n \leq \tau_h$.

3.2. Pixel decriptors

Following similar ideas, as in [11], we use SURF-gradient descriptors for the texture information of high-resolution 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 [11] it has been shown that these descriptors give good results in multi-scale setting. When we are at *L*th low-resolution level, each pixel p will be described by color and texture features, as follows

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

where U^L denotes the low resolution image. The first three components are the color information, and the last four are the texture information, in [11] it was described how to calculate the texture components. In our algorithm, we use this pixel representation to compute data cost V_p and pairwise potential V_{pq} at low-resolution levels. In practice, when working on the original image we only use color descriptors due to gradient-based descriptors are zero.

Algorithm 1: Multi-scale Image Inpainting

---

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

3.3. Multi-scale image inpainting

As mentioned before, we use local variance based segmentation to select labels $\mathcal{L}_k$ for each lowest resolution node π_k. Then, at this level we perform one iteration *ForwardPass* from priority-BP algorithm [12] in order to have a compact and high-confidence set of labels for each node. In the upper levels, the pruning process follows the one described in [14], in a similar fashion as one iteration *ForwardPass*, nothing that in [12] *similarities* rather than *beliefs* are use as a measure in the pruning process. Let $S_p(l_p)$ denote the similarity such that initially $S_p(l_p) = V_p(l_p), \forall l_p \in \mathcal{L}_k$. Once both priorities and similiraties of all nodes have been initalized, *"ForwardPass"* is executed. The similarity update equation is defined by (8)

$$S_q(l_q) = S_q(l_q) + min\left\{V_{pq}(l_p, l_q)\right\}, \forall l_q \in \mathcal{L}_j, \tag{8}$$

where $\mathcal{L}_j$ is the set of labels of neighbour node π_j of π_k. 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 π_k, we proposed to combine the belief/similarity based term with the confidence term using (9):

$$P(k) = \quad (1-\lambda)C(k) + \lambda p_{b,s}(k), \tag{9}$$

where $p_{b,s}$ is the priority using beliefs [12] or similarities [14]. With this new scheme we ensure that the nodes near the missing region boundaries are the most confident about its labels. Note that only the confidence term will not change during each iteration of pruning process, while the other term will be

(a) Inputs (b) results of [4] (c) results of [12] (d) proposed (WLS) (e) proposed (LS)

Fig. 2: Inpainting results for "Giraffe" (320x213), "Ship" (466x316) and "Elephant" (339x225) via [4], [12], our method without label selection (WLS) and with label selection (LS).

updated. After pruning proccess in all low resolution level, we propagate the remaining labels to the upper levels such that we have a better label selection using a window search around the propagated label. Finally, in the original image we perfom the Priority-BP algorithm to get the final label for each node.

4. EXPERIMENTS AND RESULTS

The proposed image inpainting method has been tested in a number of natural images. Number of levels, patch radius for MRF model (w) and label selection (w_v) are defined by the user. The test images presented (see Fig. 2), are (from top to bottom) "Giraffe" (320x213), "Ship" (466x316) and "Elephant" (339x225). In our experiments the patch radius in each level increase by the factor of two for "Giraffe" and "Elephant", and in "Ship" the factor is 1.8 . We used the same w for all the methods, $w = 16$. For our multi-scale approach we used three levels, while w_v were $w_v = 3$, $w_v = 4$ and $w_v = 5$ respectively, and the maximum number of labels per node were $l_{max} = 3$, $l_{max} = 4$ and $l_{max} = 5$. For belief propagation algorithm, the settings in [12] and our method were the same (six iterations). In Fig. 2 we show that our method produce similar results as [12] in term of reconstruction quality.

In table 1 we show the computation time for [4] and [12] and our proposed method on images from Figure 2. Our method and [4] have been implemented in MATLAB (in the first one we have interfaced MATLAB code with SIMD instructions to speed-up), while [12] is a C++ opensource implementation (http:// lafarren.com/image-completer/). We run all the programs on a Intel i7-2630QM @ 2.00GHz with 6GB RAM. The computational results show that our method is 10-15 times faster than C++ implementation of the original global optimization image inpainting (which is a significant computational improvement) and in cases is even faster than the greedy algorithm proposed by [4]. And in terms of quality, our results are similar to [12] in both cases, with and without label selection. However, we can see in table 1 that is possible to get a speed-up around 1.4 if we perfom the label selection.

Additionally, we performed experiments to measure the performace in the case of block and scratch recovery. Fig. 3 shows the results obtained using the methods in [4], [12], and ours. For qualitative comparison, the SNR is presented where our proposed method shows best performance in block recovery task. On the other hand, in scratchs the performance is similar than [12], but still better than [4].

	Computation Time			
Image	[4]	[12]	Proposed WLS	Proposed LS
"Giraffe"	7.51s	60.40s	10.33s	6.91s
"Ship"	18.33s	153.63s	13.74s	10.75s
"Elephant"	7.24s	65.86s	6.36s	4.30s

Table 1: Comparison of computation time for different images via Criminisi's [4], Komodakis's [12], our method without label selection (WLS) and our final method (LS).

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, compensating the loss of information in low resolution levels by using gradient information of the original image. We also proposed to use local statistics (lowest level's initial label selection) in order to reduce the number of labels per MRF node.

Our computational simulations show that the reconstruction quality of our approach is, subjectively, of comparable quality to results obtained by the original MRF based inpainting method, as well as to other exemplar-based inpaiting algo-

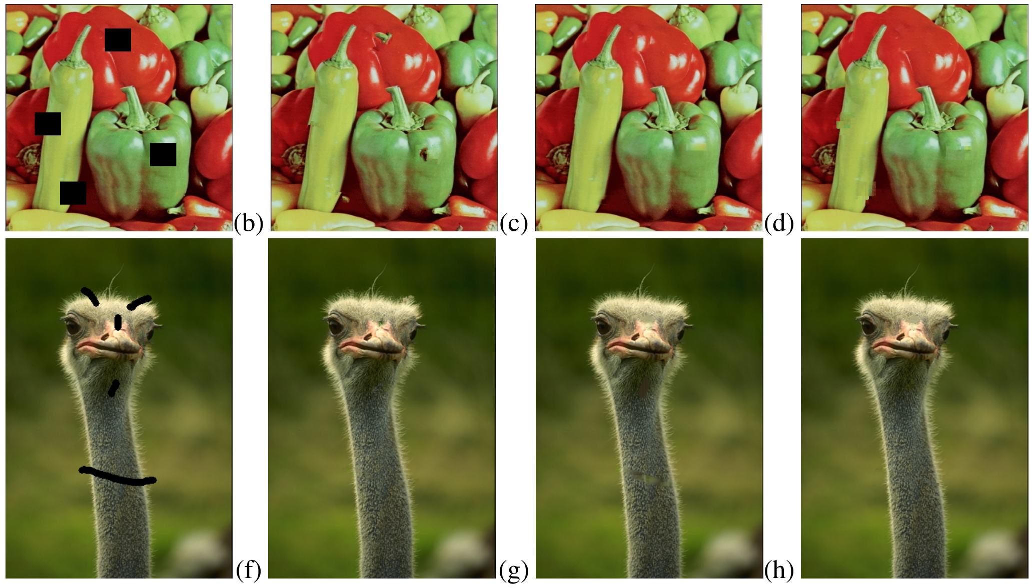


Fig. 3: Results for block recovery and scratch: (a) "Peppers", (b) [4] 26.454 dB, (c) [12] 26.735 dB, (d) ours 29.312 dB, (e) "Ostrich", (f) [4] 27.322 dB, (g) [12] 29.289 dB and (h) ours 29.110 dB.

rithms. 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 inpaiting algorithms.

6. REFERENCES

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

[2] T. Chan and J. Shen, "Variational image inpainting," *Comm. Pure Applied Math*, vol. 58, pp. 579–619, 2005.

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

[4] A. Criminisi, P. Pérez, 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, 2004.

[5] J. Wu and Y. Chou, "An effective content-aware image inpainting method," *J. Inf. Sci. Eng.*, vol. 28, no. 4, pp. 755–770, 2012.

[6] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Int. Conf. on Comp. Vis.*, 1999, pp. 1033–1038.

[7] P. Harrison, "A non-hierarchical procedure for re-synthesis of complex textures," in *WSCG Computer Graphics*, 2001, pp. 190–197.

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

[9] 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.

[10] N. Komodakis, "Image completion using global optimization," in *Computer Vision and Pattern Recognition*, Washington, DC, USA, 2006, CVPR '06, pp. 442–452.

[11] Y. Liu and V. Caselles, "Exemplar-based image inpainting using multiscale graph cuts.," *IEEE Trans. on Image Proc.*, vol. , no. , pp. , 2012.

[12] 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, 2007.

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

[14] T. Ruzic, A. Pizurica, and W. Philips, "Markov random fields based image inpainting with context-aware label selection," in *ICIP*, 2012, pp. 1733–1736.

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

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

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