TO APPEAR IN THE PROCEEDINGS OF IEEE CVPR 2006

Image Completion Using Global Optimization

Nikos Komodakis and Georgios Tziritas
Computer Science Department, University of Crete
{komod,tziritas}@csd.uoc.gr

Abstract

A new exemplar-based framework unifying image completion, texture synthesis and image inpainting is presented in this work. Contrary to existing greedy techniques, these tasks are posed in the form of a discrete global optimization problem with a well defined objective function. For solving this problem a novel optimization scheme, called Priority-BP, is proposed which carries two very important extensions over standard belief propagation (BP): "priority-based message scheduling" and "dynamic label pruning". These two extensions work in cooperation to deal with the intolerable computational cost of BP caused by the huge number of existing labels. Moreover, both extensions are generic and can therefore be applied to any MRF energy function as well. The effectiveness of our method is demonstrated on a wide variety of image completion examples.

1. Introduction

Based only on the observed part of an incomplete image, the goal of image completion is to fill its (possibly large) missing part so that a visually plausible outcome is obtained. Compared to statistical-based [11] or PDE-based methods [1] (which are mostly suitable for texture synthesis or inpainting respectively), *exemplar-based* techniques have been the most successful in dealing with this problem up to now. Here one tries to fill the unknown region simply by copying pixels or source patches from the observed part of the image [9, 4]. Starting with the seminal work in [4], these methods have been mainly used for the purpose of texture synthesis while, recently, a few authors have tried to extend them to image completion as well. In this case, however, a major drawback of these approaches stems from their greedy way of filling the image, which can often lead to visual inconsistencies. Some methods try to alleviate this problem by taking a more global approach and using an EM-like scheme for optimizing the completion process [8, 13]. EM, however, is known to be particularly sensitive to initialization and can be trapped to poor local minima. Other methods require assistance from the user instead. *E.g.* Jian Sun *et al*. [12] require the user to specify the curves on which the most salient missing structures reside, while Drori *et al*. [3] use "points of interest". Also, a few approaches rely on having a segmentation of the input image [7]. But natural images segmentation is an extremely difficult task and no general method for reliably solving it currently exists. Finally, recent exemplar-based methods also place emphasis on the order by which the image synthesis proceeds, usually using a confidence map for this purpose[2, 3]. However, two are the main shortcomings of such techniques. First, the confidence map is computed based on heuristics and ad hoc principles that may not apply in the general case and second, once an observed patch has been assigned to a missing block of pixels, that block cannot change its assigned patch thereafter. This last fact reveals the greediness of these techniques, which may again lead to visual inconsistencies. To overcome all these limitations, a new exemplar-based approach for image completion is proposed which makes the following contributions:

1) Contrary to greedy synthesis methods, we pose image completion as a discrete global optimization problem with a well defined objective function. **2)** Our formulation applies not only to image completion but also to texture synthesis and image inpainting, thus providing a unified framework for all of these tasks. **3)** No user intervention is required by our method, which manages to avoid greedy patch assignments by maintaining (throughout its execution) many candidate source patches for each block of missing pixels. **4)** To this end, a novel optimization scheme is proposed, the "Priority-BP" algorithm, which carries 2 major improvements over standard belief propagation: *"label pruning"* and *"priority-based message scheduling"*. Together, they bring a dramatic reduction in the overall computational cost of BP which would otherwise be intolerable due to the huge number of existing labels. We should finally note that both extensions are generic and can be used for the optimization of any MRF (*i.e.* a wide class of problems in vision).

2. Image completion as a discrete global optimization problem

Given an input image $\mathcal{I}_0$ as well as a *target region* $\mathcal{T}$ and a *source region* $\mathcal{S}$ (where $\mathcal{S}$ is always a subset of $\mathcal{I}_0 - \mathcal{T}$), the goal of image completion is to fill $\mathcal{T}$ in a visually plausible way simply by copying patches from $\mathcal{S}$. We propose to turn this into a discrete optimization problem with a well defined objective function. To this end, we propose the use of the following discrete Markov Random Field (MRF):

The labels $\mathcal{L}$ of the MRF will consist of all $w \times h$ patches from the source region $\mathcal{S}$[1]. For defining the nodes of the MRF, an image lattice will be used with an horizontal and vertical spacing of gap_x and gap_y pixels respectively. The MRF nodes $\{n_i\}_{i=1}^N$ will be all lattice points whose $w \times h$ neighborhood intersects the target region, while the edges $\mathcal{E}$ of the MRF will make up a 4-neighborhood system on that lattice (see Figure 1(a)).

The single node potential $V_i(l)$ (called *label cost* hereafter), for placing patch l over node n_i, will encode how well that patch agrees with the source region around n_i and will equal the following sum of squared differences (SSD):

$$V_i(l) = \sum_{p\in[-\frac{w}{2}\ \frac{w}{2}]\times[-\frac{h}{2}\ \frac{h}{2}]} \mathcal{M}(n_i+p)\big(\mathcal{I}_0(n_i+p)-\mathcal{I}_0(l+p)\big)^2, \quad (1)$$

where $\mathcal{M}(\cdot)$ is a binary mask non zero only in region $\mathcal{S}$ (obviously, due to this definition, the label costs of interior nodes, *i.e.* nodes whose $w \times h$ neighborhood does not intersect $\mathcal{S}$, will be all zero). In a similar fashion, the pairwise potential $V_{ij}(l, l')$, due to placing patches l, l' over neighbors n_i, n_j, will measure how well these patches agree at the resulting region of overlap and will again be given by the SSD over that region (see Figure 1(b)). Note that gap_x and gap_y are set so that such a region of overlap always exists.

Based on this formulation, our goal will then be to assign a label $\hat{l}_i \in \mathcal{L}$ to each node n_i so that the total energy $E(\{\hat{l}_i\})$ of the MRF is minimized where:

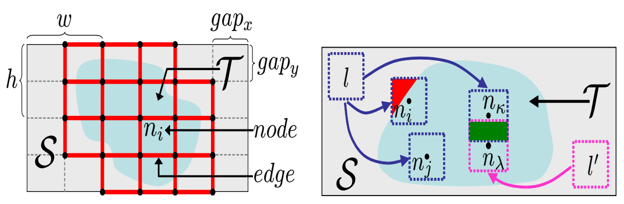
$$E(\{\hat{l}_i\}) = \sum_{i=1}^{N} V_i(\hat{l}_i) + \sum_{(i,j)\in\mathcal{E}} V_{ij}(\hat{l}_i, \hat{l}_j) \quad (2)$$

Intuitively, any algorithm optimizing this energy is roughly solving a huge jigsaw puzzle where source patches are the puzzle pieces while region $\mathcal{T}$ represents the puzzle itself. One important advantage of our formulation is that it also provides a unified framework for texture synthesis and image inpainting. *E.g.* to handle texture synthesis (where one wants to extend an input texture T_0 to a larger region T_1) one suffices to set $\mathcal{S} = T_0$ and $\mathcal{T} = T_1 - T_0$. Moreover, our framework allows the use of (what we call) *"completion by energy refinement"* techniques, one example of which we will see later.

3. Priority-BP

Furthermore, an additional advantage would be that we can now hopefully apply belief propagation (*i.e.* a state-of-the-art optimization method) to our energy function. Unfortunately, however, this was not feasible. The reason was the intolerable computational cost of BP caused by the huge number of existing labels. Motivated by this fact, one other major contribution of this work is the proposal of a novel MRF optimization scheme, called Priority-BP, that can deal exactly with this type of problems and carries two significant extensions over standard BP: one of them, called *dynamic label pruning*, is based on the key idea of drastically reducing the number of labels. However, instead of this happening beforehand (which will almost surely lead to throwing away useful labels), pruning takes place on the fly (*i.e.* while BP is running) with a (possibly) different number of labels kept for each node. The important thing to note is that only the beliefs calculated by BP are used for that purpose. Furthermore, the second extension, called *priority-based message scheduling*, makes use of label pruning and allows us to always send cheap messages between the nodes of the graphical model. Moreover, it considerably improves BP's convergence, thus accelerating completion even further.

[1]Hereafter each label (*i.e.* patch) will be represented by its center pixel



(a) MRF nodes and edges (b) MRF potentials

Fig. 1. **(a)** An MRF with $w = 2gap_x,\ h = 2gap_y$ **(b)** For boundary node n_i its label cost $V_i(l)$ will be an SSD over the red region, while for nodes n_κ, n_λ their potential $V_{\kappa\lambda}(l, l')$ will be an SSD over the green region. Interior node n_j has zero label costs.

The significance of our contribution also grows due to the fact that (as we shall see) Priority-BP is a generic algorithm applicable to any MRF energy function. It therefore resolves what is currently considered one of the main limitations of BP: its inefficiency to handle problems with a huge number of labels. In fact, this issue has been a highly active research topic over the last years. Until now, however, the techniques that have been proposed were valid only for restricted classes of MRFs [5]. Not only that but our priority-based message scheduling scheme can be used (independently of label pruning) as a general method for accelerating the convergence of BP.

3.1. Priority-based message scheduling

BP is an iterative algorithm, which works by propagating local messages along the nodes of an MRF [10, 6]. Messages sent from node n_i to node n_j form a set $\{m_{ij}(l)\}_{l\in\mathcal{L}}$, where element $m_{ij}(l)$ indicates how likely node n_i thinks that node n_j should be assigned label l. Furthermore, messages are updated (*i.e.* sent) until convergence as follows[2]:

$$m_{ij}(l) = \min_{l_i \in \mathcal{L}} \{V_i(l_i) + V_{ij}(l_i, l) + \sum_{k:k \neq j,(k,i) \in \mathcal{E}} m_{ki}(l_i)\} \quad (3)$$

After convergence, a set of beliefs $\{b_i(l)\}_{l \in \mathcal{L}}$ is computed for each node, where belief $b_i(l)$ is defined as follows:

$$b_i(l) = -V_i(l) - \sum_{k:(k,i) \in \mathcal{E}} m_{ki}(l) \quad (4)$$

$b_i(l)$ approximates the max-marginal of the posterior at node n_i and is therefore roughly related to how likely label l is for that node. Based on this fact, a node is then assigned the label of maximum belief, i.e. $\hat{l}_i = \arg\max_{l \in \mathcal{L}} b_i(l)$. It is known that, for tree structured graphs, BP always gives the optimal solution, while, for graphs with loops, it can only guarantee to find a local optimum.

In this form, however, BP is impractical for problems with a large number of labels like ours. In particular, if $|\mathcal{L}|$ is the total number of labels (which, in our case, can be many many thousands) then just the basic operation of updating the messages from one node n_i to another node n_j takes $O(|\mathcal{L}|^2)$ time. In fact the situation is much more worse for us. The huge number of labels also implies that for any pair of adjacent nodes n_i, n_j their matrix of pairwise potentials $V_{ij}(\cdot,\cdot)$ is so large that cannot fit into memory and therefore be precomputed. That matrix therefore must be reestimated every time node n_i sends its messages to node n_j, meaning that $|\mathcal{L}|^2$ SSD calculations (between image patches) are needed for each such update.

To deal with this issue we will try to reduce the number of labels by exploiting the beliefs calculated by BP. However not all nodes have beliefs which are adequate for this purpose in our case. To see that, it suffices to observe that the label costs at all interior nodes are all equal to zero. This in turn implies that the beliefs at an interior node will initially be all equal as well, meaning that the node is "unconfident" about which labels to prefer. No label pruning may therefore take place and so any message originating from that node will be very expensive to calculate, *i.e.* it will take $O(|\mathcal{L}|)$ time. On the contrary, if we had a node whose labels could be pruned (and assuming that the maximum number of labels after pruning is L_{max} with $L_{max} \ll |\mathcal{L}|$) then any message from that node would take only $O(L_{max})$ time.

Based on this observation, we therefore propose to use a specific message scheduling scheme whose goal will be twofold. On on hand, it will make label pruning possible and favor the circulation of cheap messages. On the other hand, it will speed up BP's convergence. This issue of BP message scheduling, although known to be crucial for the success of BP, it has been largely overlooked until now. Also, to the best of the authors' knowledge it is the first time that message scheduling is used in this manner for general graphical models. Roughly, our message scheduling scheme will be based on the notion of priorities that are assigned to the nodes of the MRF. Any such priority will represent a node's confidence about which labels to prefer and will be dynamically updated throughout the algorithm's execution. Our message scheduling will then obey the following simple principle:

Message-scheduling principle. *The node most confident about its labels should be the first one (i.e. it has the highest priority) to transmit outgoing messages to its neighbors.*

There are two reasons why one may want to do this. The first is that the more confident a node is, the more label pruning it can tolerate (before sending its outgoing messages) and therefore the cheaper these messages will be. The second reason is that we also help other nodes become more amenable to pruning this way. Intuitively, this happens because the more confident a node is, the more informative its messages are going to be, meaning that these messages can help the neighbors of that node to increase their own confidence and thus become more tolerable to pruning as well. Furthermore, by first propagating the most informative messages around the graphical model we also help BP to converge much faster. This has been verified experimentally as well. *E.g.* Priority-BP never needed more than a small fixed number of iterations to converge for all of our image completion examples.

Algorithm 1 Priority-BP

```
assign priorities to nodes and declare them uncommitted
for $k = 1$ to $K$ do {$K$ is the number of iterations}
  execute *ForwardPass* and then *BackwardPass*
assign to each node $n_i$ its label $\hat{l}_i$ that maximizes $b_i(\cdot)$

*ForwardPass:*
for time = 1 to $N$ do {$N$ is the number of nodes}
  $n_i$ = "uncommitted" node of highest priority
  apply "label pruning" to node $n_i$
  forwardOrder[time] = $n_i$;  $n_i$→committed = true;
  for any "uncommitted" neighbor $n_j$ of node $n_i$ do
    send all messages $m_{ij}(\cdot)$ from node $n_i$ to node $n_j$
    update beliefs $b_j(\cdot)$ as well as priority of node $n_j$

*BackwardPass:*
for time = $N$ to 1 do
  $n_i$ = forwardOrder[time];  $n_i$→committed = false;
  for any "committed" neighbor $n_j$ of node $n_i$ do
    send all messages $m_{ij}(\cdot)$ from node $n_i$ to node $n_j$
    update beliefs $b_j(\cdot)$ as well as priority of node $n_j$
```

A pseudocode description of Priority-BP is contained in algorithm 1. Each iteration of Priority-BP is divided into a forward and a backward pass. The actual message scheduling mechanism as well as label pruning takes place during

[2]We work in the $-log$ domain so we use the min-sum version of BP

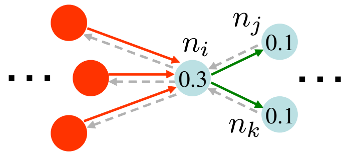


Fig. 2. Message scheduling during the forward pass: currently only red nodes have been committed and only messages on red edges have been transmitted. Among uncommitted nodes (*i.e.* blue nodes) the one with the highest priority (*i.e.* node n_i) will be committed next and will also send messages only along the green edges (*i.e.* only to its uncommitted neighbors n_j, n_k). Messages along dashed edges will be transmitted during the backward pass. Priorities are indicated by the numbers inside uncommitted nodes.

the forward pass. This is also where one half of the messages gets transmitted (*i.e.* each MRF edge is traversed in only one of the 2 directions). To this end, all nodes are visited in order of priority. Each time we visit a node, say n_i, we mark it as "committed" meaning that we must not visit him again during the current forward pass. We also prune its labels and then allow him to transmit its "cheap" (due to pruning) messages to all of its neighbors apart from the committed ones (as these have already sent a message to n_i during the current pass). The priorities of all neighbors that received a new message are then updated and the process continues with the next uncommitted (*i.e.* unvisited) node of highest priority until no more uncommitted nodes exist.

The role of the backward pass is then just to ensure that the other half of the messages gets transmitted as well. To this end, we do not make use of priorities but simply visit the nodes in reverse order (with respect to the order of the forward pass) just transmitting the remaining unsent messages from each node. For this reason no label pruning takes place during this pass. We do update node priorities, though, so that they are available during the next forward pass.

Also, as we shall see, a node's priority depends only on the current beliefs at that node. One big advantage out of this is that keeping the node priorities up-to-date can be done very efficiently in this case since only priorities for nodes with newly received messages need to be updated. The message scheduling mechanism is further illustrated in Figure 2.

3.2. Assigning priorities to nodes

It is obvious that our definition of priority will play a very crucial role for the success of the algorithm. As already mentioned, priority must relate to how confident a node is about the labels that should be assigned to him with the more confident nodes having higher priority. An important thing to note in our case is that the confidence of a node will depend solely on information that will be extracted by the BP algorithm itself. This makes our algorithm generic (*i.e.* applicable to any MRF energy function) and therefore appropriate for a very wide class of problems.

In particular our definition of confidence (and therefore priority as well) for node n_i will depend only on the current set of beliefs $\{b_i(l)\}_{l\in\mathcal{L}}$ that have been estimated by the BP algorithm for that node. Based on the observation that belief $b_i(l)$ is roughly related to how likely label l is for node n_i, one way to measure the confidence of this node is simply by counting the number of likely labels *e.g.* those whose belief exceed a certain threshold b_{conf}. The intuition for this is that the greater this number the more labels with high probability exist for that node and therefore the less confident that node turns out to be about which specific label to choose. And vice versa, if this number is small then node n_i needs to choose its label only among a small set of likely labels. Of course only relative beliefs $b_i^{rel}(l) = b_i(l) - b_i^{max}$ (where $b_i^{max} = \max_{l\in\mathcal{L}} b_i(l)$) matter in this case and so by defining the set $\mathbb{CS}(n_i) = |\{l \in \mathcal{L} : b_i^{rel}(l) \geq b_{conf}\}|$ (which we will call the *confusion set* of node n_i hereafter) the priority of n_i is then inversely related to the cardinality of that set:

$$\text{priority}(n_i) = \frac{1}{|\mathbb{CS}(n_i)|} \tag{5}$$

This definition of priority also justifies why during either the forward or the backward pass we were allowed to update priorities only for nodes that had just received new incoming messages: the reason is that the beliefs (and therefore the priority) of a node may change only if at least one incoming message to that node changes as well (this is true due to the way beliefs are defined, *i.e.* $b_i(l) = -V_i(l) - \sum_{k:(k,i)\in\mathcal{E}} m_{ki}(l)$). Although we tested other definitions of priority as well (e.g. by using an entropy-like measure on beliefs) the above criterion for quantifying confidence gave the best results in practice by far.

3.3. Applying Priority-BP to image completion

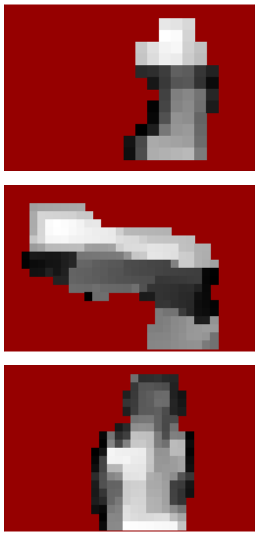
We pause here for a moment (postponing the description of label pruning to the next section) in order to stress the advantages of applying our algorithm to image completion while also showing related results.

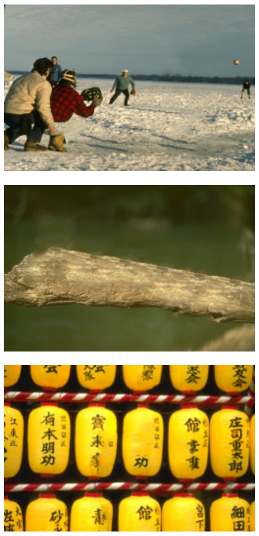
First of all we should mention that although confidence has already been used for guiding image completion in other works as well [2, 3], our use of confidence differs (with respect to these approaches) in two very important aspects. The first is that we use confidence in order to decide upon the order of BP message passing and not for greedily deciding which patch to fill next. These are two completely different things: the former is part of a principled global optimization procedure, whereas the latter just results in patches that cannot change their appearance after they have been filled.

The second aspect is that in all of the previous approaches the definition of confidence was mostly based either on heuristics or on ad hoc principles that were simply

(a) original image

(b) masked image


(c) visiting order during first forward pass


(d) Priority-BP result

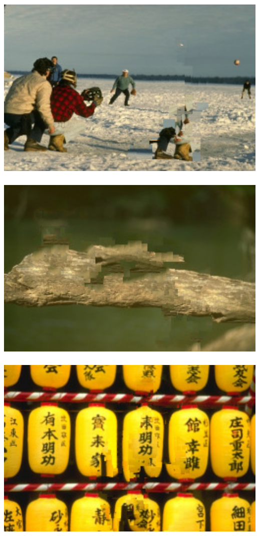
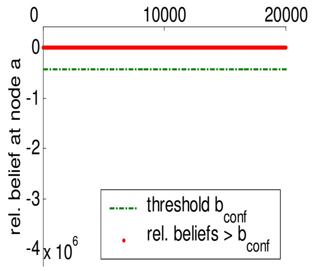
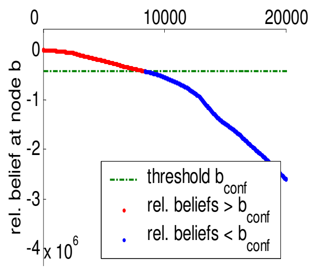

(e) result of [2]

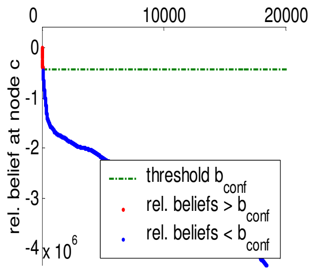
Fig. 3. In column (c) darker patches correspond to nodes that are visited earlier during message scheduling at the first forward pass



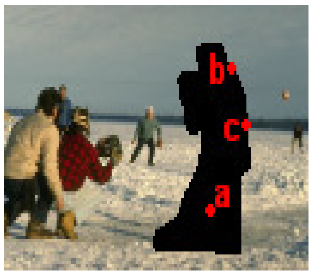
(a) node a has minimum priority



(b) node b has low priority



(c) node c has high priority



(d) the MRF nodes a, b and c

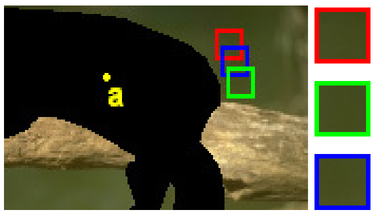
Fig. 4. The plots in (a), (b) and (c) show the sorted relative beliefs for the MRF nodes a, b and c in figure (d) at the start of Priority-BP. Relative beliefs plotted in red (*i.e.* $\geq b_{conf}$) correspond to labels in the confusion set. The size of this set determines the node's priority.

making use of application-specific knowledge about the image completion process. On the contrary, as we saw, our definition of confidence is generic and therefore applicable to any kind of images. Moreover, this way our method is placed on firm theoretical grounds.

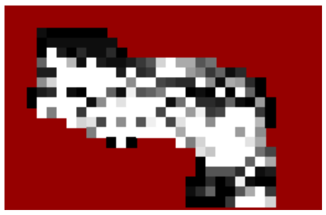
Three examples of applying Priority-BP to image completion are shown in Figure 3. As can be seen, the algorithm has managed to fill the missing regions in a visually plausible way. The third column in that figure shows the visiting order of the nodes during the first forward pass (based on our definition of priority). The darker a patch is in these images, the earlier the corresponding node was visited. Notice how the algorithm learns by itself how to propagate first the messages of the nodes containing salient structure, where the notion of saliency depends on each specific case. *E.g.* the nodes that are considered salient for the first example of Figure 3 are those lying along the horizon boundary. On the contrary, for the second example of that figure, the algorithm prefers to propagate information along the MRF edges at the interior of the wooden trunk first. *The remarkable thing is that in both cases such information was not explicitly encoded but was, instead, inferred by the algorithm.*

This is in contrast to the state-of-the-art method in [2] where the authors had to hardwire isophote-related information into the definition of priority (*i.e.* a measure which is not always reliably extracted or even appropriate *e.g.* in images with texture). The corresponding results produced by that method are shown in the last column of Figure 3. In these cases only one label (*i.e.* patch) is greedily assigned to each missing block of pixels and so any errors made early cannot be later backtracked, thus leading to the observed visual inconsistencies. On the contrary, due to our global optimization approach, any errors that are made during the very first iterations can be very well corrected later since our algorithm always maintain not one but many possible labels for each MRF node. A characteristic case for this is the third example in Figure 3 where, unless one employs a global optimization scheme, it is not easy to infer the missing structure.

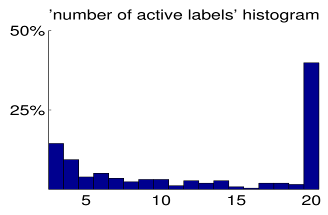
Also, the plots in Figures 4(a), 4(b), 4(c) illustrate our definition of priority in equation (5). They display the largest 20000 relative beliefs (sorted in descending order)



(a)



(b)



(c)

(d)

Fig. 5. **(a)** Although the red, green and blue patches correspond to distinct labels, they are very similar and so only one has to be an active label for a node. **(b)** A map with the number of active labels per node (for the 2^{nd} example of Figure 3). Darker patches correspond to nodes with fewer labels. As can be seen, interior nodes often require more labels. **(c)** The corresponding histogram showing the percentage of nodes using a certain number (in the range $L_{min} = 3$ to $L_{max} = 20$) of active labels. **(d)** The active labels for node a in Fig. (a).

that are observed at the very beginning of the algorithm for each of the MRF nodes a, b, c in Figure 4(d) respectively. Relative beliefs plotted in red correspond to labels in the confusion set. Node a, being an interior node, has initially all the labels in its confusion set (since their relative beliefs are all zero) and is therefore of lowest priority. Node b still has too many labels in its confusion set due to the uniform appearance of the source region around that node. On the contrary node c is one of the nodes to be visited early during the first forward pass since only very few labels belong to its confusion set. Indeed, even at the very beginning, we can easily exclude (*i.e.* prune) many source patches from being labels of that node without the risk of throwing away useful labels. This is why Priority-BP prefers to visit him early.

3.4. Label pruning

The main idea of "label pruning" is that, as we are visiting the nodes of the MRF during the forward pass (in the order induced by their priorities), we dynamically reduce the number of possible labels for each node by discarding labels that are unlikely to be assigned to that node. In particular, after committing a node, say n_i, all labels having a very low relative belief at n_i, say less than b_{prune}, are not considered as candidate labels for n_i thereafter. The remaining labels are called the *"active labels"* for that node. An additional advantage we gain this way is that after all MRF nodes have pruned their labels at least once (e.g. at the end of the first forward pass), then we can precompute the reduced matrices of pairwise potentials (which can now fit into memory) and thus greatly enhance the speed of our algorithm. The important thing to note is that "label pruning" relies only on information carried by the Priority-BP algorithm itself as well. This keeps our method generic and therefore applicable to any energy function. A key observation, however, relates to the fact that label pruning is a technique not meant to be used on its own. Its use is allowed only in conjunction with our priority-based message scheduling scheme of visiting most confident nodes first (*i.e.* nodes for which label pruning is safe and does not throw away useful labels). This is exactly the reason why label pruning does not take place during the backward pass.

In practice we apply label pruning only to nodes whose number of active labels exceeds a user specified number L_{max}. To this end, when we are about to commit a node we traverse its labels in order of belief (from high to low) and each such label is declared active until either no more labels with relative belief greater than b_{prune} exist or the maximum number of active labels L_{max} has been reached. In the case of image completion, however, it turns out that we also have to apply an additional filtering procedure as part of label pruning. The problem is that otherwise we may end up having too many active labels which are similar to each other, thus wasting part of the L_{max} labels we are allowed to use. This issue is further illustrated in Figure 5(a). To this end, as we traverse the sorted labels, we declare a label as active only if it is not similar to any of the already active labels (where similarity is measured by calculating the SSD between image patches), otherwise we skip that label and go to the next one. Alternatively, we apply a clustering procedure to the patches of all labels beforehand (*e.g.* cluster them into textons) and then never use more than one label from each cluster while traversing the sorted labels. Finally, we should note that for all nodes a (user-specified) minimum number of active labels L_{min} is always kept.

The net result of label pruning is thus to obtain a compact and diverse set of active labels for each MRF node (all of them having reasonably good beliefs). *E.g.* Figure 5(b) displays the number of active labels used by each of the nodes in the second example of Figure 3. The darker a patch is in that figure, the fewer are the active labels of the corresponding node. As it was expected, interior nodes often require more active labels to use. The corresponding histogram showing the percentage of nodes that use a certain number of active labels is displayed in Figure 5(c). Notice that more than half of the MRF nodes do not use the maximum number of active labels (which was $L_{max} = 20$ in this case). Also, Fig. 5(d) displays the active labels that have been selected by the algorithm for node a in Fig. 5(a).

4. Extensions & further results

Completion via energy refinement: One advantage of posing image completion as an optimization problem is that

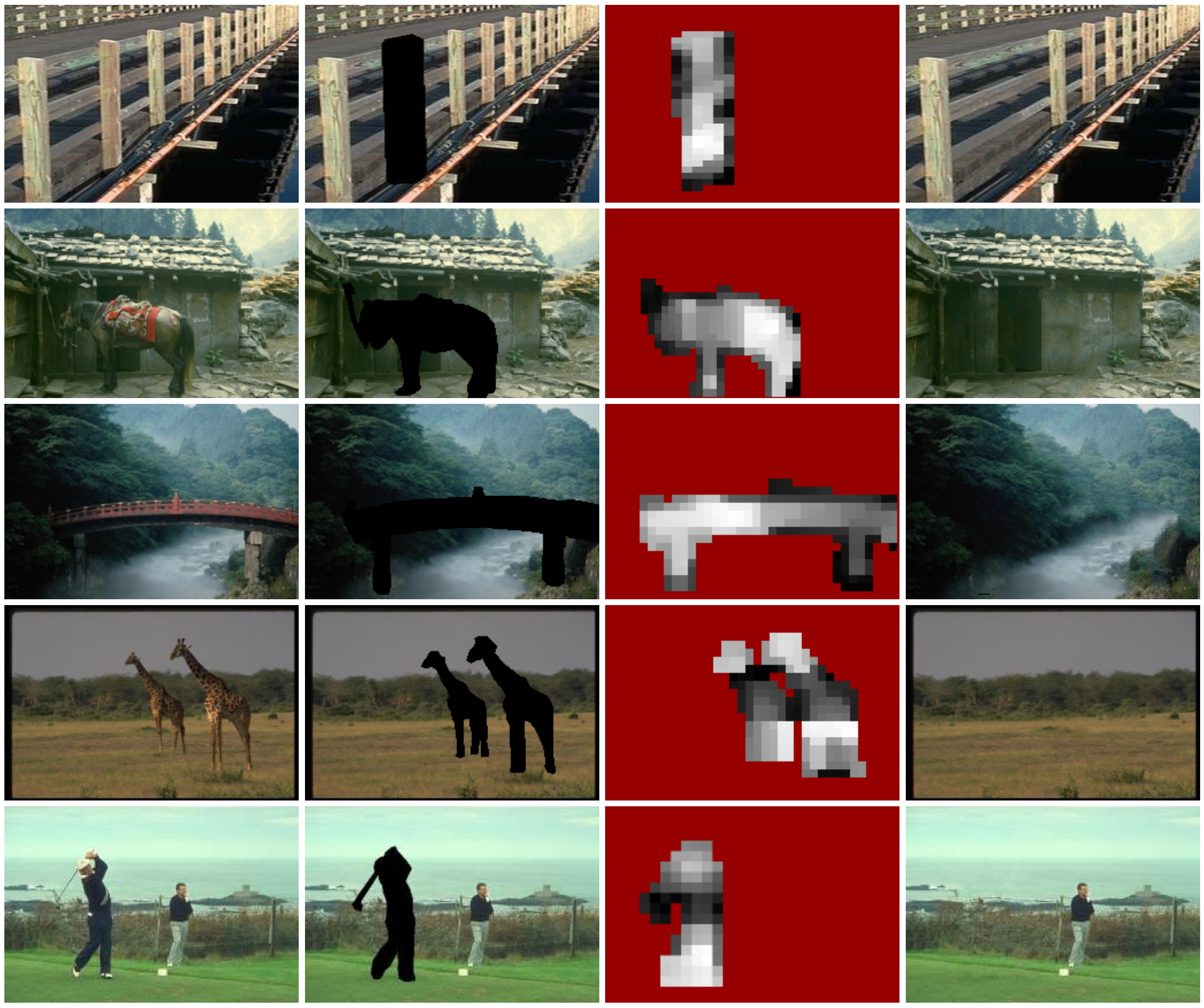
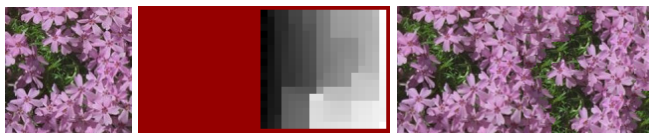


Fig. 6. Image completion. From left to right: original images, masked images, visiting order at 1^{st} forward pass, Priority-BP results

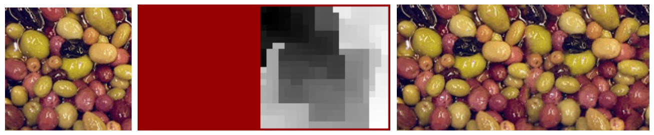
one can now refine completion simply by refining the energy function (*i.e.* adding more terms to it). *E.g.*, to favor spatial coherence during image completion (*i.e.* to fill the target region with large chunks of the source region), one simply needs to add the following *"incoherence penalty terms"* V^0_{ij} to our energy function: $V^0_{ij}(l_i, l_j) = w_0$ if $l_i - l_j \neq n_i - n_j$ while in all other cases $V^0_{ij}(l_i, l_j) = 0$. These terms simply penalize (with a weight w_0) the assignment of non-adjacent patches (with centers l_i, l_j) to adjacent nodes n_i, n_j and have proved useful in texture synthesis problems (*e.g.* see Figure 7(b)). Thanks to the ability of Priority-BP to handle effectively any energy function, we intend to explore the utility (with respect to image completion) of many other refinement terms in the future. We believe that this will also be an easy and effective way of applying prior knowledge or imposing user specified constraints on the image completion process.

Pyramid-based image completion: Another important advantage of our method is that it can also be used in multi-scale image completion, where a Gaussian pyramid of images $\mathcal{I}_k, \mathcal{I}_{k-1}, \ldots, \mathcal{I}_0$ is given as input. For this, we begin by applying Priority-BP to the image at the coarsest scale $\mathcal{I}_k$. The output of this procedure is then up-sampled and the result, say $\mathcal{I}'_k$, is used for guiding the completion of the image $\mathcal{I}_{k-1}$ at the next finer scale. To this end, the only part of our algorithm that needs to be modified is that of how label costs are computed. In particular, the mask $\mathcal{M}$ in equation (1) will now be non zero everywhere and so not only pixels from the source region of $\mathcal{I}_{k-1}$ are taken into account but also pixels from the unknown target region, where now the values for these pixels are borrowed from the approximation image $\mathcal{I}'_k$. The rest of the algorithm remains the same and this process is repeated until we reach the image at the finest scale $\mathcal{I}_0$. An advantage we gain this way is that features at multiple scales can be captured.

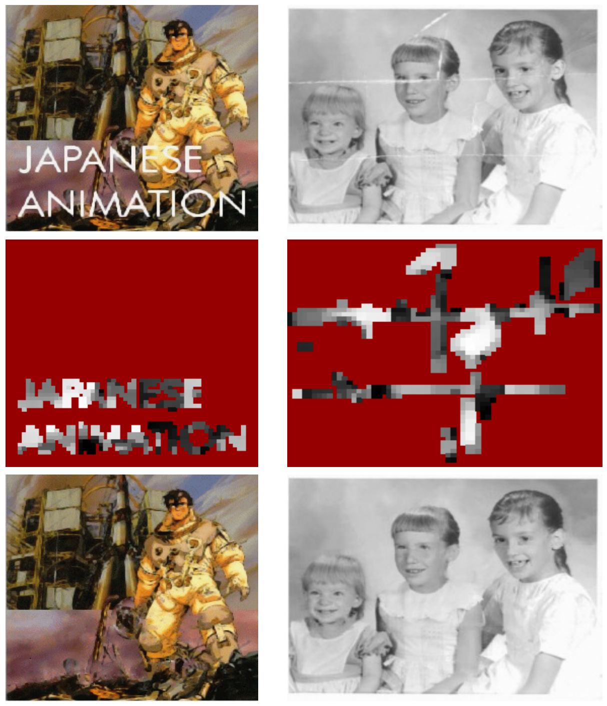
Figure 6 contains further results on image completion. These results along with those in Figure 3 demonstrate the effectiveness of our method. As can be seen, Priority-BP



(a) texture synthesis



(b) texture synthesis with “incoherence penalty terms”



(c) text removal and image inpainting

Fig. 7. Priority-BP results (order at 1^{st} forward pass is also shown)

was able to handle the completion of smooth regions, textured areas, areas with structure as well as any combinations of the above. Besides image completion, we also show results on texture synthesis, text removal as well as image inpainting (Figure 7). Our method had no problem of handling these tasks as well. In Figure 7(b) we demonstrate an example of using the “incoherence penalty terms” in texture synthesis. As one can observe, the output texture does contain large chunks of the input texture as intended. Also in the last example of Figure 6 we show the final result of a pyramid-based image completion. In this case the input image was 481×321 and a 2-level pyramid has been used. Also, for all examples we show the visiting order of the nodes during the first forward pass. In our tests the patch size ranged between 7×7 and 27×27. The running time on a 2.4GHz CPU varied from a few seconds up to 2 minutes for 256×170 images, while the maximum number of labels L_{max} was set between 10 and 50 (depending on the input's difficulty). For all of the examples, the belief thresholds were set equal to $b_{conf} = -\mathrm{SSD}_0$, $b_{prune} = -2 \cdot \mathrm{SSD}_0$, where SSD_0 represents a predefined mediocre SSD score between $w \times h$ patches. For the composition of the final patches, these are usually blended with weights that are proportional to the confidence of the corresponding nodes. More elaborate schemes, like feathering or multi-resolution splining, have been also tried in some cases. Finally, to accelerate the SSD calculations required by the algorithm, the common method of moving to the frequency domain and using the fast Fourier transform has been applied.

5. Conclusions

A novel approach which treats image completion, texture synthesis and image inpainting in a unified manner has been presented. To avoid visually inconsistent results due to greedy patch assignments, we pose all of these tasks in the form of a discrete labeling problem with a well defined objective function. To solve that problem, a novel global optimization scheme, Priority-BP, has been proposed that carries two very important extensions over standard BP: priority-based message scheduling and label pruning. Our algorithm does not rely on image-specific prior knowledge and can be applied to any kind of images. Furthermore, it is generic (*i.e.* applicable to any MRF energy) and thus copes with one of the main limitations of BP: its inefficiency to handle problems with a huge number of labels. Finally, a wide variety of examples have verified its effectiveness.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH*, 2000. 1

[2] A. Criminisi, P. Pérez, and K. Toyama. Object removal by exemplar-based inpainting. In *CVPR*, 2003. 1, 4, 5

[3] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. In *SIGGRAPH*, 2003. 1, 4

[4] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999. 1

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *CVPR*, 2004. 2

[6] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *IJCV*, 40(1):25–47, 2000. 2

[7] J. Jia and C.-K. Tang. Image repairing: Robust image synthesis by adaptive nd tensor voting. In *CVPR*, 2003. 1

[8] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra. Texture optimization for example-based synthesis. In *SIGGRAPH*, 2005. 1

[9] V. Kwatra and *et al.* Graphcut textures: Image and video synthesis using graph cuts. In *SIGGRAPH*, 2003. 1

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988. 2

[11] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1), 2000. 1

[12] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. In *SIGGRAPH*, 2005. 1

[13] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *CVPR*, pages 120–127, 2004. 1