Markov Random Fields for Super-resolution and Texture Synthesis

Bill Freeman and Ce Liu

February, 2010

1 Introduction

Suppose we want to digitally enlarge a photograph. The input is a single, low-resolution image, and the desired output is an estimate of the high-resolution version of that image. This problem can be phrased as one of "image interpolation": we seek to interpolate the pixel values between our observed samples. Image interpolation is sometimes called super-resolution, since we are estimating data at a resolution beyond that of the image samples. In contrast with multi-image super-resolution methods, where a high-resolution image is inferred from a video sequence, we are interested in estimating high-resolution images from a single low-resolution example [10].

There are many analytic methods for image interpolation, including pixel replication, linear and cubic spline interpolation [22], and sharpened Gaussian interpolation [23]. When we interpolate in resolution by a large amount, such as a factor of four or more in each dimension, these analytic methods typically suffer from a blurred appearance. Following a simple rule, they tend to make conservative, smooth guesses for image appearance.

We can address this problem with two techniques. The first is to use an example-based representation to handle the many special cases we expect. We describe the pre-processing and representaton issues for our example-based representation below. Second, we use a graphical model framework to reason about global structure. The super-resolution problem has a structure similar to other low-level vision tasks: we accumulate local evidence (which may be ambiguous) and propagating it across space. A Markov random field is an appropriate structure for this: local evidence terms can be modeled by unary potentials $\psi_i(x_i)$ at a node i with states x_i. Spatial propagation occurs through pairwise potentials, $\phi_{ij}(x_i, x_j)$, between nodes i and j, or through higher order potentials. The joint probability then has the factorized form,

$$P_{\vec{x}}(\vec{x}) = \frac{1}{Z} \prod_i \psi_i(x_i) \prod_{(ij) \in E} \phi_{ij}(x_i, x_j), \tag{1}$$

where E is the set of edges in the MRF denoted by the neighboring nodes, i and j, and Z is a normalization constant such that the probabilities sum to one [17]. The local statistical relationships allow information to propagate long distances over an image.

1.1 Image pre-filtering

To develop the super-resolution algorithm, we first specify the desired model of subsampling and image degradation that we seek to undo. For the examples in this paper, we assume we low-pass filter the desired high-resolution image, then subsample by a factor of four in each dimension, to obtain the observed low-resolution image. The low-pass filter is a 7x7 pixel Gaussian filter, normalized to have unit sum, of standard deviation 1 pixel. We start from a high-resolution image, and blur it and subsample to generate the corresponding low-resolution image. We apply this model to a set of training images, to generate some number of paired examples of high-resolution and low-resolution image patch pairs.

It is convenient to handle the high- and low-resolution images at the same sampling rate–the same number of pixels. After creating the low-resolution image, we perform an initial interpolation up to the sampling rate of the full-resolution image. Usually this is done with cubic spline interpolation, to create what we will call the "upsampled low-resolution image".

We want to exploit whatever invariances we can to let the training data generalize beyond the training examples. We use two heuristics to try to extend the reach of the examples. First, we don't believe that all spatial frequencies of the low-resolution are needed to predict the missing high-frequency image components, and we don't want

to have to store a different example patch for each possible value of the low-frequency components of the low-resolution patch. So we apply a low-pass filter to the upsampled low-resolution image in order to divide it into two spatial frequency bands. We call the output of the low-pass filter the "low-band", L; the upsampled low-resolution image minus the low-band image gives what we'll call the "mid-band", M. The difference between the upsampled low-resolution image and the original image is the "high-band", H.

A second operation to increase the scope of the examples is contrast normalization. We assume that the relationship of the mid-band, M, to high-band, H, data is independent of the local contrast level. So we normalize the contrast of the mid- and high-band images in the following way:

$$[\hat{M}, \hat{H}] = \frac{[M, H]}{\mathrm{std}(M) + \delta} \tag{2}$$

where $\mathrm{std}(\cdot)$ is standard deviation operator, and δ is a small value which sets the local contrast level below which we do not adjust the contrast. Typically, $\delta = 0.0001$ for images that range over zero to one.

1.2 Representation of the unknown state

We have a choice about what we estimate at the nodes of the MRF. If the variable to be estimated at each node is a single pixel, then the dimensionality of the unknown state at a node is low, which is good. However, it may not be feasible to draw valid conclusions about single pixel states from only performing computations between pairs of pixels. That may place undo burden on the MRF inference. We could remove that burden if a large patch of estimated pixels is assigned to one node, but then the state dimensionality at a node may be unmanagably high.

To address this, we work with entire image patches at each node, to provide sufficient local evidence, but use other means to constrain the state dimensionality at a node. First, we restrict the solution patch to be one of some number of exemplars, typically image examples from some training set. In addition, we take advantage of local image evidence to further constrain the choice of exemplars to be from some smaller set of candidates from the training set. The result is an unknown state dimension of 20 to 40 states per node.

Figure 2 illustrates this representation. The top row shows an input patch from the (bandpassed, contrast normalized) low-resolution input image. The next two rows show the 30 nearest-neighbor examples from a database of 658,788 image patches, extracted from 41 images. The low-res patches are of dimension 25×25, and the high-res patches are of dimension 9×9. The bottom two rows of Fig. 2 show the corresponding high-resolution image patches for each of those 30 nearest neighbors. Note that the mid-band images look approximately the same as each other and as the input patch, while the high-resolution patches look considerably different from each other. This tells us that the local information from the patch by itself is not sufficient to determine the missing high resolution information, and we must use some other source of information to resolve the ambiguity. The state representation is then an index into a collection of exemplars, telling which of the unknown high resolution image patches is the correct one, illustrated in Fig. 3. The resulting MRF is shown in Fig. 1.

1.3 MRF parameterization

We can define a local evidence term and pairwise potentials of the Markov random field if we make assumptions about the probability of encountering a training set exemplar in the test image. We assume any of our image exemplars can appear in the input image with equal probability. We account for differences between the input and training set patches as independent, identically distributed Gaussian noise added to every pixel. Then the local evidence for a node being in sample state x_i depends on the amount of noise needed to translate from the low-resolution patch corresponding to state x_i to the observed mid-band image patch, $\vec{p}$. If we denote the band-passed, contrast normalized mid-band training patch associated with state x_i as $\vec{M}(x_i)$ then

$$\psi_i(x_i) = \exp |\vec{p} - \vec{M}(x_i)|^2 / (2\sigma^2) \tag{3}$$

where we write 2-d image patches as rasterized vectors.

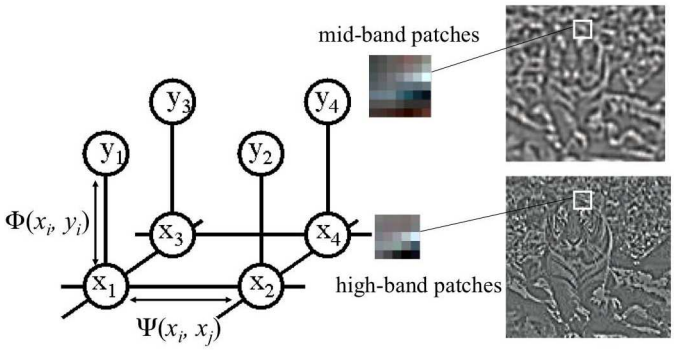

Figure 1: Patch-based MRF for low-level vision. The observations y_i are patches from the mid-band image data. The states to be estimated are indices into a dataset of high-band patches.

To construct the compatibility term, $\phi_{ij}(x_i, x_j)$, we assume we have overlapping high-band patches that should agree with their neighbors in their regions of overlap, see Fig. 4. Any disagreements is again attributed to a Gaussian noise process. If we denote the band-passed, contrast normalized high-band training patch associated with state x_i as $\vec{H}(x_i)$, and introduce an operator O_{ij} that extracts as a rasterized vector the pixels of the overlap region between patches i and j (with the ordering compatible for neighboring patches), then we have

$$\phi_{ij}(x_i, x_j) = \exp |O_{ij}(H(x_i)) - O_{ji}(H(x_j))|^2/(2\sigma^2), \tag{4}$$

In the examples we show below, we used a mid-band and high-band patch size of 9x9 pixels, and used a patch overlap region of size 3 pixels.

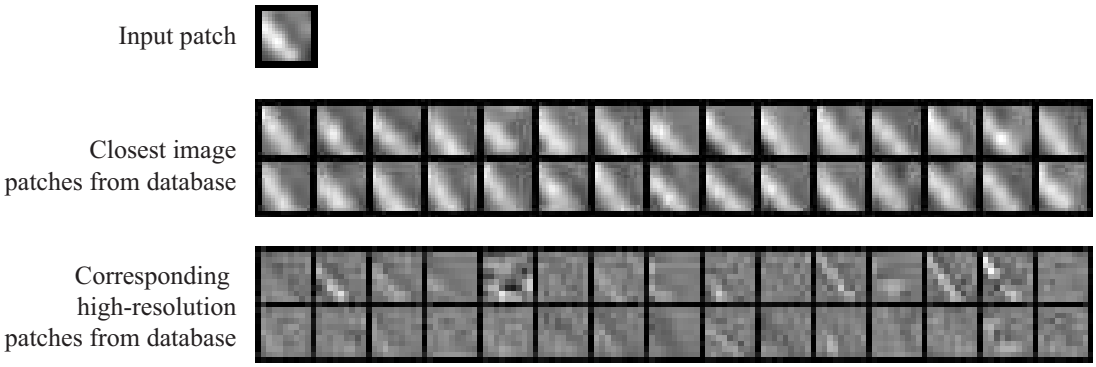

Figure 2: top: input patch (mid-band bandpass filtered, contrast normalized). We seek to find the high-resolution patch associated with this. Middle: Nearest neighbors from database to the input patch. The found patches match this reasonably well. Bottom: The corresponding high-resolution patches associated with each of the retrieved mid-band bandpass patches. These show more variability than the mid-band patches, indicating that more information than simply the local image matches is needed to select the proper high-resolution image estimate. Since the resolution requirements for the color components are lower than for luminance, we use an example-based approach for the luminance, and interpolate the color information by a conventional cubic spline interpolation.

1.4 Loopy belief propagation

We have set-up the Markov Random Field such that each possible selection of states at each node corresponds to a high-resolution image interpretation of the input low-resolution image. The MRF probability, the product of all the local evidence and pairwise potentials in the MRF, assigns a probability to each possible selection of states

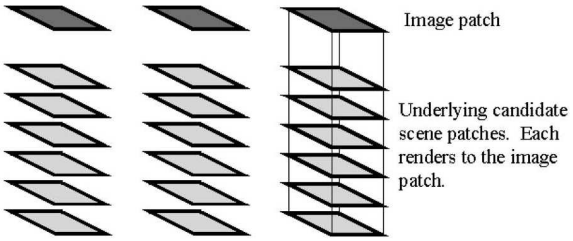

Figure 3: The state to be estimated at each node. Using the local evidence, at each node, we have a small collection of image candidates, selected from our database. We use the belief propagation to select between the candidates, based on compatibility information.

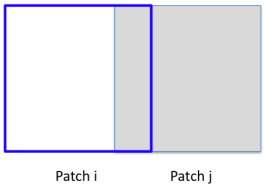

Figure 4: The patch-patch compatibility function is computed from the sum of squared pixel differences in the overlap region.

according to Eq. (1). Each configuration of states specifies an estimated high-band image, and we seek the high-band image that is most favored by the MRF we have specified. This is the task of finding a point estimate from a posterior probability distribution.

In Bayesian decision theory [3] the optimal point estimate depends on the loss function used–the penalty for guessing wrong. With a penalty proportional to the square of the error, the best estimate is the mean of the posterior. However, if all deviations from the true value are equally penalized, then the best estimate is the maximum of the posterior. Using Belief Propagation [20], both estimates can be calculated exactly for an MRF that is a tree.

We consider first the case of the posterior mean, which requires marginalizing the posterior over the states of all other nodes. For a network without loops, the sums over node states for the marginalization can be distributed efficiently over the network in a message-passing algorithm. We define a set of messages, $m_{ij}(x_j)$ along each direction of each edge; the messages can be initialized to random values between zero and one. The messages are functions of the states of the node receiving the message. A message from node i to node j is updated according to,

$$m_{ij}(x_j) \leftarrow \sum_{x_j} \phi(x_i, x_j)\phi_j(x_j) \prod_{k \in \eta(j)_1} m_{kj}(x_j) \tag{5}$$

For the case of a tree network, these updates occur until the messages no longer change. Then the marginal probability at each node is the product of all the incoming messages and the local potential:

$$p_{x_i}(x_i) = \phi_i(x_i) \prod_{j \in \eta(i)} m_{ji}(x_i) \tag{6}$$

When the Markov network forms a tree, belief propagation is simply an efficient redistribution of the sums involved in marginalization, and iterations of Eq. (5) yield the exact marginals by Eq. (6).

Interestingly, for a network with loops, it is often still useful to apply the same update and marginal probability equations, although in that case, the marginal probabilities are only an approximation. The message updates are run until convergence, or for a fixed number of iterations (here, we used 30 iterations). Fixed points of these iterative update rules correspond to stationary points of a well-known approximation used in statistical physics, the

Bethe approximation [26]. Good empirical results have been obtained with that approximation [12, 10], and we use it here.

For our approximation to the MMSE estimate, we take the mean (weighted by the marginals from Eq. (6)) of the candidate patches at a node. It is also possible to approximate the MAP estimate by substituting the summation operator of Eq. (5) with "max", then selecting the patch maximizing the resulting "max-marginal" given in Eq. (6). These solutions are often sharper, but with more artifacts, than the MMSE estimate.

To piece together the final image, we undo the contrast normalization of each patch, average neighboring patches in regions where they overlap, add-in the low and mid-band images, and add-in the analytically interpolated chrominance information. Figure 5 summarizes the steps in the algorithm, and Fig. 6 shows other results. The perceived sharpness is significantly improved, and the belief propagation iterations significantly reduce the artifacts that would result from estimating the high-resolution image based on local image information alone. (Figure 7 provides enlargments of cropped regions from those two figures.) The code used to generate the images in Sect. 1.4 is available for download at http://people.csail.mit.edu/billf/.

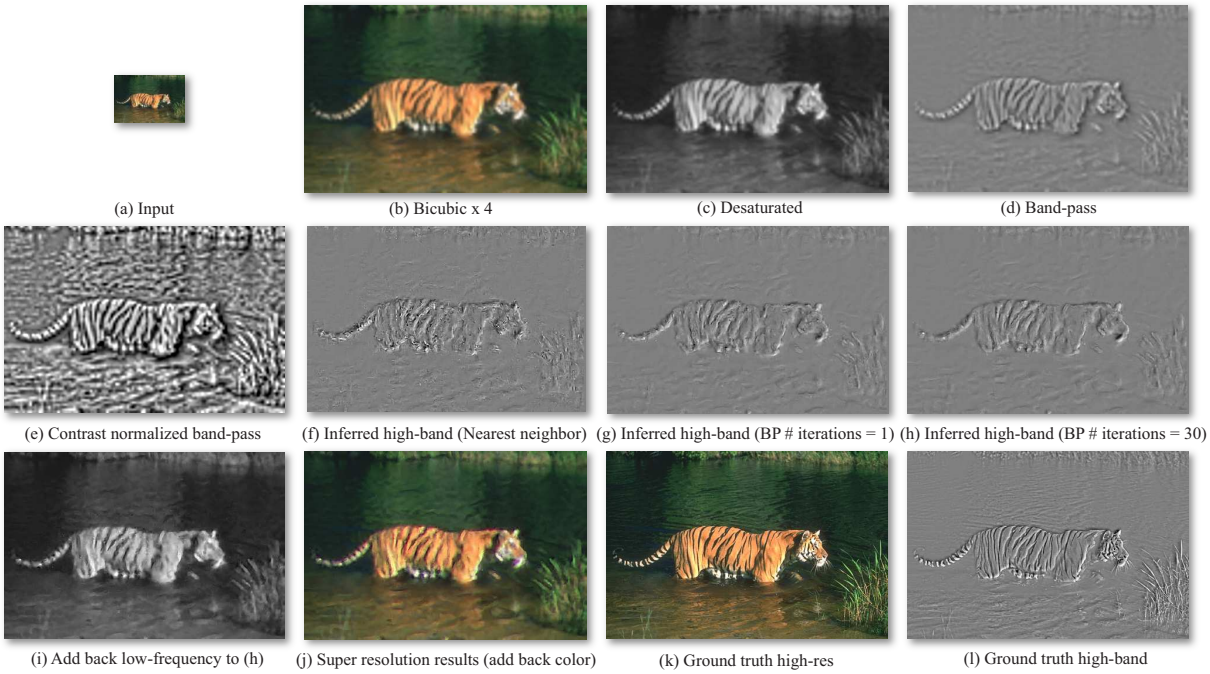

Figure 5: Images showing the example-based super-resolution processing. (a) input image, of resolution 120×80. (b) Cubic spline interpolation up to a factor of four higher resolution in each dimension. (c) We extract the luminance component for example-based processing (and use cubic spline interpolation for the chrominance components). (d) A high-pass filtering of this image gives us the mid-band output, shown here. (e) Display of the contrast normalized mid-band. The contrast normalization extends the utility of the training database samples beyond the contrast value of each particular training example. (f) the high frequencies corresponding to the nearest neighbor of each local low-frequency patch. (g) After 1 iteration of belief propagation, much of the choppy high frequency details of (f) are removed. (h) converged high resolution estimates. (i) Image (c) added to image (h)–the estimated high frequencies added back to the mid and low-frequencies. (j) Color components added back in. (k) comparison with ground truth. (l) true high frequency components.

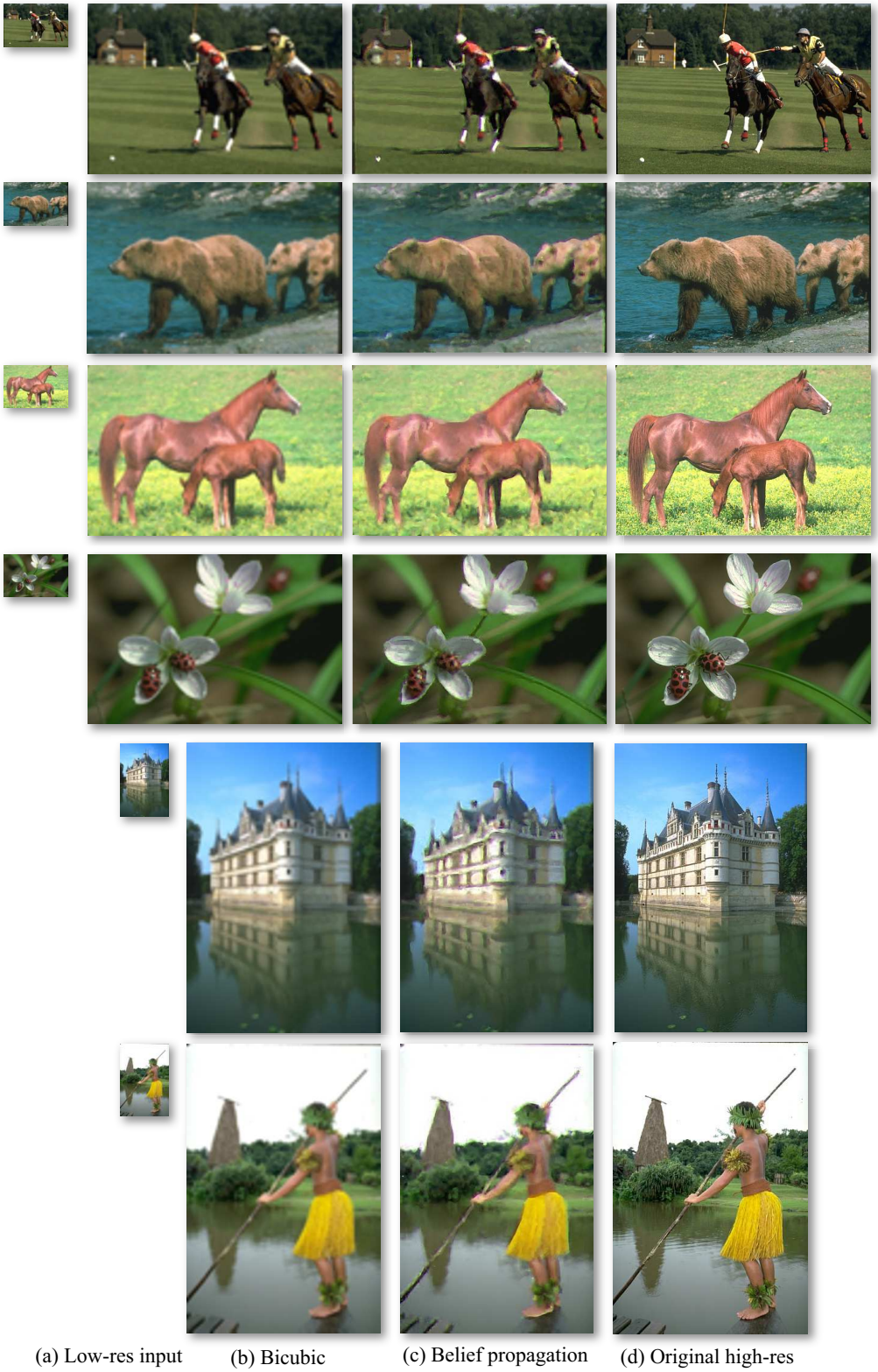

Figure 6: Other example-based super-resolution outputs. (a) Input low-res images. (b) Bicubic interpolation (x4 resolution increase). (c) Belief propagation output. (d) The true high-resolution images.

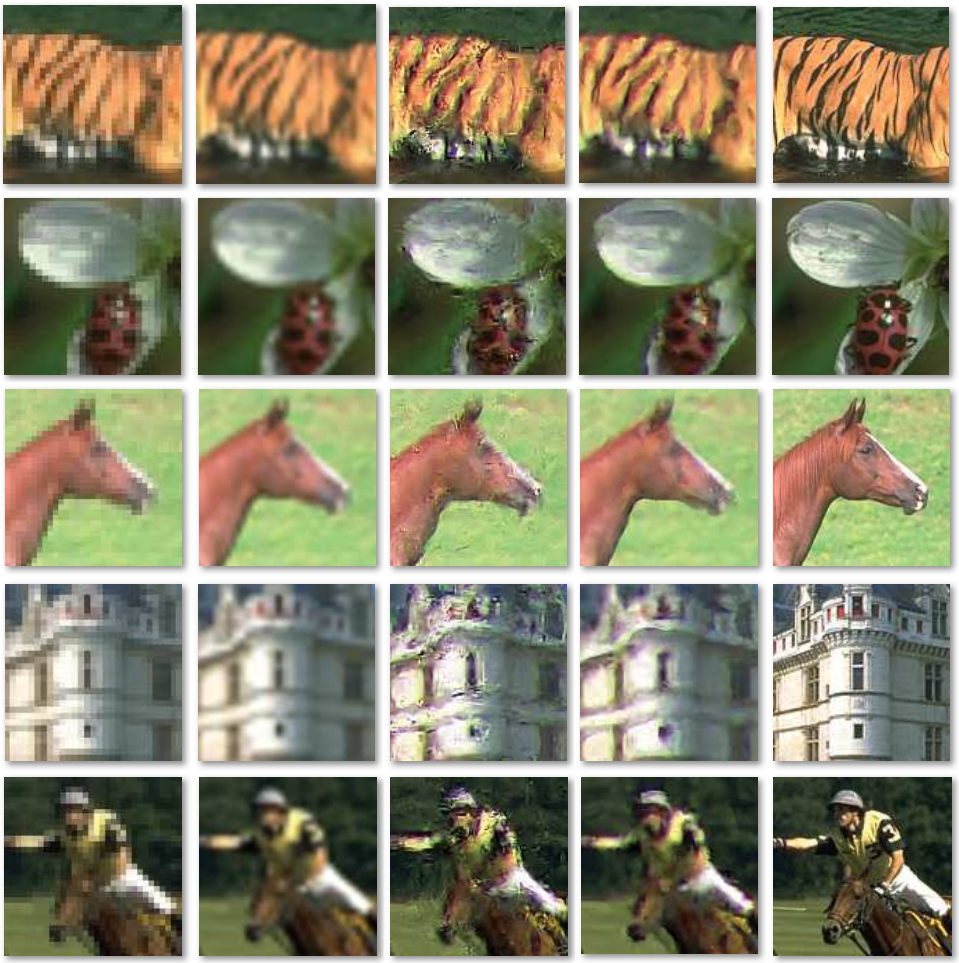

Figure 7: The closeups of Figure 5 and 6. (a) Input low-res images. (b) Bicubic interpolation (x4 resolution increase). (c) Nearest neighbor output. (d) Belief propagation output. (c) The true high-resolution images.

1.5 Texture synthesis

This same example-based Markov Random Field machinery may be applied to other low-level vision tasks, as well [10]. Another application involving image patches in Markov Random Fields is texture synthesis. Here, the input is a small sample of a texture to be sythesized. The output is a larger portion of that texture, having the same appearance but not made from simply repeating the input texture.

Non-parametric texture methods have revolutionized texture synthesis. Notable examples include Heeger and Bergen [15], De Bonet [5], Efros and Leung [7]. However, these methods can be slow. To speed them up, and address some image quality issues, Efros and Freeman [6] developed a non-parametric patch-based method; a related method was developed independently by Liang et al [18]. This is another example of the patch-based, non-parametric Markov random field machinery described above for the super-resolution problem.

For texture synthesis, the idea is to draw patch samples from random positions within the source texture, then piece them together seamlessly. Figure 8, from [6], tells the story. In (a), random samples are drawn from the source texture, and place in the synthesized texture. With random selection, the boundaries between adjascent texture blocks are quite visible. (b) shows instead texture synthesis with overlapping patches selected from the input texture to match the left and top borders of the texture region that has been synthesized so far. The border artifacts are greatly suppressed, yet some are still visible. (c) Shows the result of adding an additional step to the processing of (b): we select an optimal ragged boundary using "image quilting", described shortly below.

There is an MRF implied by the model above, with the same ψ_{ij} compatibility term between neighboring patches as we had for the super-resolution problem. For this texture synthesis problem, there is no local evidence term. [1]. This makes solution of the problem using belief propagation to be nearly impossible, since there is not small list of candidate patches available at each node. The state dimension cannot be reduced to a managable level.

As an alternative, we adopt a greedy algorithm, described in detail in [6], that only approximates the optimal assignment of training patch to MRF node. We process the image in a raster scan fashion, top-to-bottom in rows, left-to-right within each row. Except at the image boundaries, we always have two borders with patches filled-in for any patch we seek to select. To add a patch, we randomly select a patch from the source texture from the top 5 matches to the top and left boundaries values. This algorithm can be thought of as a particularly simple, approximate method to find the patch assignments that maximize the MRF of Eq. (1), where the pair-wise compatibilities $\phi_{ij}(x_i, x_j)$ are as for super-resolution, but there are no local evidence terms, $\phi_i(x_i)$. Figure 9 shows nine examples of textures synthesized from input examples, shown in the smaller images to the left of each synthesis example. Note that the examples exhibit the perceptual appearance of the smaller patches, but are synthesized in a realistic non-repeating pattern.

1.5.1 Image quilting–dynamic programming

Now we return to the goal of finding the optimal ragged boundary between two patches. We seek the optimal tear to minimize the visibility of artifacts caused by differences between the neighboring patches. We describe the algorithm for finding the optimal tear in a vertical region and the extension to a horizontal tear is obvious. Let the difference between two adjascent patches in the region of overlap be $d(i, j)$, where i and j horizontal and vertical pixel coordinates. For each row, we seek the column $q(j)$ of an optimal path of tearing between the two patches. This optimal path should follow a contour of small difference values between the two patches. We seek to minimize

$$\hat{q} = \text{argmin}_{q(j)} \sum_{j}^{K} d(q(j), j)^2 \tag{7}$$

under the constraint that the tear forms a continuous line, $|q(j) - q(j-1)| \leq 1$.

[1] For a related problem, texture transfer [6], we can have local evidence constraints.

This optimal path problem has a well-known solution through dynamic programming [4], which has been exploited in various vision and graphics applications [24, 6]. This is equivalent to finding the maximum posterior probability through max-product belief propagation. We summarize the algorithm:

```
Initialization:
p(i,1)) = d( i, 1)

for j = 2:N
  p(i,j) =  p(i, j-1) + min_k d(k,j)
end
```

, where the values considered for the minimization over k are i, and $i \pm 1$. Using an auxiliary set of pointers indicating the optimal value of the $\min_k$ operation at each iteration, the path $q(i)$ can be found from the values of $p(i, j)$. This method has also been used to hide image seams in "seam carving", [1].

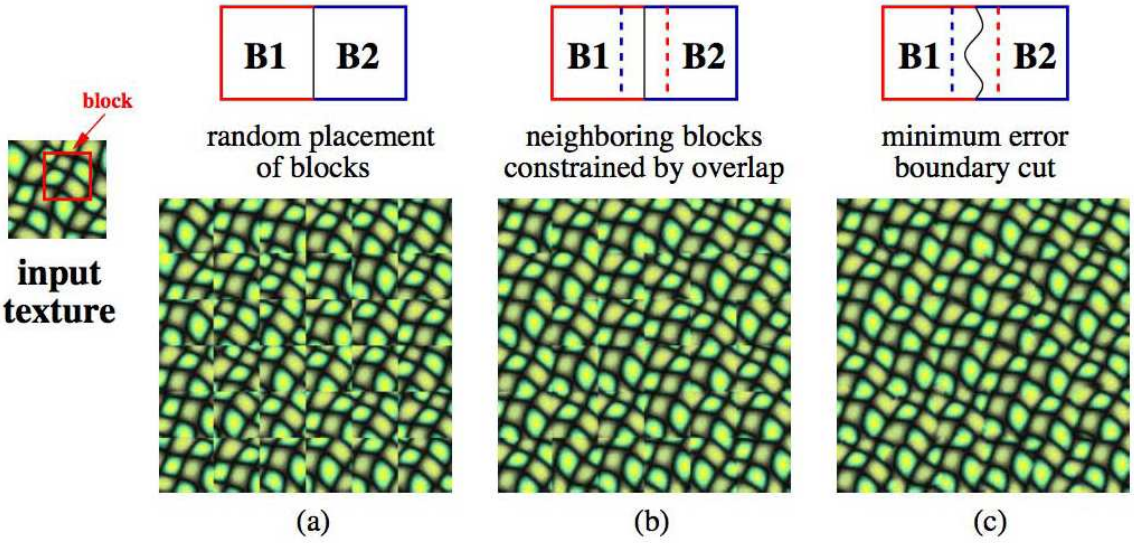

Figure 8: Patch samples of an input texture can be composited to form a larger texture in a number of different ways. (a) A random placement of texture samples gives strong patch boundary artifacts. (b) We can select only patches that match well with neighbors in an overlap region, but there are still some boundary artifacts in the composite image. (c) Selecting the best seam through the boundary region of neighboring patches removes most artifacts. Figure reprinted from [6].

2 Selected related applications by others

Markov random fields have been used extensively in image processing and computer vision. Geman and Geman brought Markov random fields to the attention of the vision community, and showed how to use MRF's as image priors in restoration applications, [13]. Poggio, Gamble and Little used MRF's in a framework unifying different computer vision modules, [21].

The example-based approach has been built on by others. This method has been used in combination with a resolution enhancement model specific to faces [2] to achieve excellent results in hallucinating details of faces [19]. Huang and Ma have proposed finding a linear combination of the candidate patches to fit the input data, then applying the same regression to the output patches, simulating a better fit to the input [25]. (A related approach was also used in [11]).

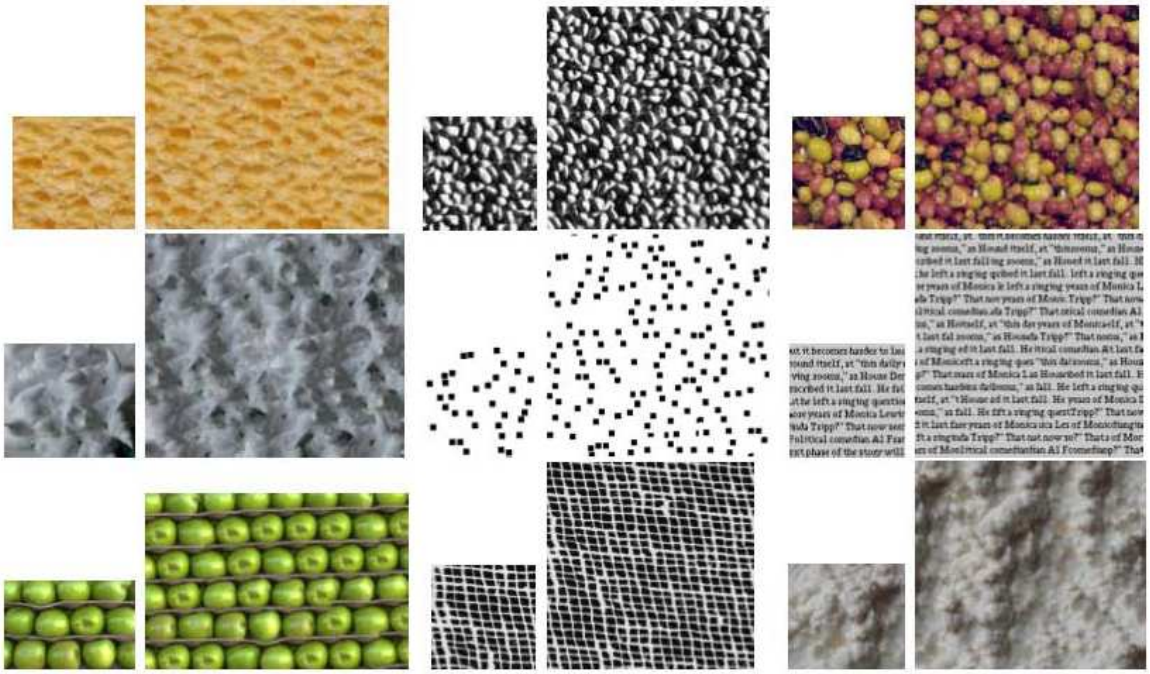

Figure 9: A collection of source (small image) and corresponding synthesized textures made using the patch-based image quilting method. Figure reprinted from [6].

Optimal seams for image transitions were found in a 2-d framework, using graph cuts in Kwatra et al [16]. Example-based image priors were used for image-based rendering in the work of Fitzgibbon, Wexler, and Zisserman, [9]. Fattal used edge models for image upsampling [8]. Glasner et al also used an example-based approach for super-resolution, relying on self-similarity within a single image [14].

References

[1] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Computer Graphics (SIGGRAPH '07)*, 26(3), 2007.

[2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2000.

[3] J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer, 1985.

[4] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2000.

[5] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proc. SIGGRAPH 97*, pages 361–368, 1997. In *Computer Graphics*, Annual Conference Series.

[6] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH*, 2001. In *Computer Graphics* Proceedings, Annual Conference Series.

[7] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Intl. Conf. on Comp. Vision*, 1999.

[8] R. Fattal. Upsampling via imposed edges statistics. In *ACM SIGGRAPH*, 2007. In *Computer Graphics* Proceedings, Annual Conference Series.

[9] A.W. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. In *Proc. International Conference on Computer Vision*, 2003.

[10] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Intl. J. Computer Vision*, 40(1):25–47, 2000. See http://www.merl.com/reports/TR2000-05/.

[11] W. T. Freeman, J. B. Tenenbaum, and E. C. Pasztor. Learning style translation for the lines of a drawing. *ACM Transactions on Graphics (TOG)*, 22:33 – 46, 2003.

[12] B. J. Frey, R. Koetter, and N. Petrovic. Very loopy belief propagation for unwrapping phase images. In *Adv. in Neural Info. Proc. Systems*, volume 13. MIT Press, 2001. http://www.psi.toronto.edu/pubs/2001/sppu-nips01.pdf.

[13] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[14] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *Proc. International Conference on Computer Vision*, 2009.

[15] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *ACM SIGGRAPH*, pages 229–236, 1995. In *Computer Graphics* Proceedings, Annual Conference Series.

[16] V. Kwatra, A. Schdl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. In *ACM SIGGRAPH*, 2003. In *Computer Graphics* Proceedings, Annual Conference Series.

[17] S. Z. Li. *Markov random field modeling in image analysis*. Springer, 2009.

[18] L. Liang, C. Liu, Y. Xu, B. Guo, and H. Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 2001. in press.

[19] C. Liu, H. Y. Shum, and W. T. Freeman. Face hallucination: theory and practice. *International Journal of Computer Vision*, 75(1):115–134, 2007.

[20] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[21] T Poggio, E B Gamble, and J J Little. Parallel integration of vision modules. *Science*, 242:436–440, 1989.

[22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge Univ. Press, 1992.

[23] W. F. Schreiber and D. E. Troxel. Transformation between continuous and discrete representations of images: A perceptual approach. *PAMI*, 7(2):176–178, 1985.

[24] A. Shashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. In *Proc. International Conference on Computer Vision*, pages 321–327, 1988.

[25] J. Yang, J. Wright, T. Huang, and Y. Ma. image super-resolution as sparse representation of raw image patches. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2008.

[26] J. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *International Joint Conference on Artificial Intelligence (IJCAI 2001)*, 2001. Distinguished Papers Track.