MERL – A MITSUBISHI ELECTRIC RESEARCH LABORATORY
http://www.merl.com

Learning low-level vision

William T. Freeman and Egon C. Pasztor
MERL, Mitsubishi Electric Research Laboratory
201 Broadway; Cambridge, MA 02139
freeman@merl.com, pasztor@merl.com

TR-99-12 July 1999

Abstract

We show a learning-based method for low-level vision problems–estimating scenes from images. We generate a synthetic world of scenes and their corresponding rendered images. We model that world with a Markov network, learning the network parameters from the examples. Bayesian belief propagation allows us to efficiently find a local maximum of the posterior probability for the scene, given the image. We call this approach VISTA–Vision by Image/Scene TrAining.

We apply VISTA to the "super-resolution" problem (estimating high frequency details from a low-resolution image), showing good results. For the motion estimation problem, we show figure/ground discrimination, solution of the aperture problem, and filling-in arising from application of the same probabilistic machinery.

To appear in: IEEE International Conference on Computer Vision, Corfu, Greece, 1999.

1. First printing, TR99-12, March, 1999
2. revised version, July, 1999.

Learning low-level vision

William T. Freeman and Egon C. Pasztor
MERL, a Mitsubishi Electric Res. Lab.
201 Broadway, Cambridge, MA 02139
freeman, pasztor@merl.com

Abstract

We show a learning-based method for low-level vision problems–estimating scenes from images. We generate a synthetic world of scenes and their corresponding rendered images. We model that world with a Markov network, learning the network parameters from the examples. Bayesian belief propagation allows us to efficiently find a local maximum of the posterior probability for the scene, given the image. We call this approach VISTA–Vision by Image/Scene TrAining.

We apply VISTA to the "super-resolution" problem (estimating high frequency details from a low-resolution image), showing good results. For the motion estimation problem, we show figure/ground discrimination, solution of the aperture problem, and filling-in arising from application of the same probabilistic machinery.

1 Introduction

We seek machinery for learning low-level vision problems, such as motion analysis, inferring shape and albedo from a photograph, or extrapolating image detail. For these problems, given *image* data, we want to estimate an underlying *scene*. The scene quantities to be estimated might be projected object velocities, surface shapes and reflectance patterns, or missing high frequency details.

Low-level vision problems are typically underconstrained, so Bayesian [3, 23, 38] and regularization techniques [31] are fundamental. There has been much work and progress (for example, [23, 25, 15]), but difficulties remain in working with complex, real images. Typically, prior probabilities or constraints are made-up, rather than learned. A general machinery for a learning-based solution to low-level vision problems would have many applications.

A recent research theme has been to learn the statistics of natural images. Researchers have related those statistics to properties of the human visual system [28, 2, 36], or have used statistical methods with biologically plausible image representations to analyse and synthesize realistic image textures [14, 8, 42, 36]. These methods may help us understand the early stages of representation and processing, but unfortunately, they don't address how a visual system might *interpret* images, i.e., estimate the underlying scene.

We want to combine the two research themes of scene estimation and statistical learning. We study the statistical properties of a synthetically generated, *labelled* world of images with scenes, to learn how to infer scenes from images. Our prior probabilities can then be rich ones, learned from the training data.

Several researchers have applied related learning approaches to low-level vision problems, but restricted themselves to linear models [21, 16], too weak for many applications. Our approach is similar in spirit to relaxation labelling [33, 22], but our Bayesian propagation algorithm is more efficient and we utilize large sets of labelled training data.

We interpret images by modeling the relationship between local regions of images and scenes, and between neighboring local scene regions. The former allows initial scene estimates; the later allows the estimates to propagate. We train from image/scene pairs and apply the Bayesian machinery of graphical models [29, 5, 20]. We were inspired by the work of Weiss [39], who pointed out the speed advantage of Bayesian methods over conventional relaxation methods for propagating local measurement information. For a related approach, but with heuristically derived propagation rules, see [34].

We call our approach VISTA, Vision by Image/Scene TrAining. It is a general machinery that may apply to various problems. We illustrate it for estimating missing image details, and estimating motion.

2 Markov network

For given image data, y, we seek to estimate the underlying scene, x (we omit the vector symbols for notational simplicity). We first calculate the posterior probability, $P(x|y) = cP(x, y)$ For this analysis,

we ignore the normalization, $c = \frac{1}{P(y)}$, a constant over x. Under two common loss functions [3], the best scene estimate, $\hat{x}$, is the mean (minimum mean squared error, MMSE) or the mode (maximum a posteriori, MAP) of the posterior probability.

In general, $\hat{x}$ can be difficult to compute [23] without approximations. We make the Markov assumption: we divide both the image and scene into patches, and assign one node of a Markov network [13, 29, 20] to each patch. Given the variables at intervening nodes, two nodes of a Markov network are statistically independent. We connect each scene patch to its corresponding image patch, and to its nearest neighbors, Fig. 1. Solving a Markov network involves a *learning* phase, where the parameters of the network connections are learned from training data, and an *inference* phase, when the scene corresponding to particular image data is estimated.

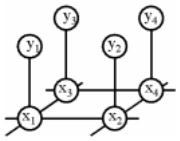


Figure 1: Markov network for vision problems. Observations, y, have underlying scene explanations, x.

For networks without loops, the Markov assumption leads to simple "message-passing" rules for computing the MAP and MMSE estimates [29, 40, 20]. Writing those estimates for x_j by marginalizing (MMSE) or taking the max (MAP) over the other variables gives:

$$\hat{x_j}_{MMSE} = \int_{x_j} x_j dx_j \int_{\text{all } x_i,\, i \neq j} P(x,y)dx \quad (1)$$

$$\hat{x_j}_{MAP} = \underset{[x_j]}{\text{argmax}} \; \underset{[\text{ all } x_i,\, i \neq j \,]}{\text{max}} \; P(x,y) \quad (2)$$

For a Markov random field, the joint probability over the scenes x and images y can be written as [4, 13, 12]:

$$P(x,y) = \prod_{\text{neighboring } i,j} \Psi(x_i, x_j) \prod_k \Phi(x_k, y_k), \quad (3)$$

where we have introduced pairwise compatibility functions, Ψ and Φ, described below. The factorized structure of Eq. (3) allows the marginalization and maximization operations of Eqs. (1) and (2) to pass through to the compatibility function factors with the appropriate arguments. For a network without loops, the resulting expression can be computed using repeated, local computations [29, 40, 20], summarized below: the MMSE estimate at node j is

$$\hat{x_j}_{MMSE} = \int_{x_j} x_j \Phi(x_j, y_j) \prod_k L_{kj} dx_j, \quad (4)$$

where k runs over all scene node neighbors of node j. We calculate L_{kj} from:

$$L_{kj} = \int_{x_k} \Psi(x_k, x_j) \Phi(x_k, y_k) \prod_{l \neq j} \tilde{L}_{lk} dx_k, \quad (5)$$

where $\tilde{L}_{lk}$ is L_{lk} from the previous iteration. The initial $\tilde{L}_{lk}$'s are 1. After at most one iteration per x_i of Eq. (1), Eq. (4) and (5) give Eq. (1). The MAP estimate equation, Eq. (2), yields analogous formulae, with the integral of Eq. (5) replaced by max_{x_k}, and $\int_{x_j} x_j$ of Eq. (4) replaced by $argmax_{x_j}$. For linear topologies, these propagation rules are equivalent to well-known Bayesian inference methods, such as the Kalman filter and the forward-backward algorithm for Hidden Markov Models [29, 26, 39, 20, 11].

Finding the posterior probability distribution for a grid-structured Markov network with loops is computationally expensive and a variety of approximations have been proposed [13, 12, 20]. Strong empirical results in "Turbo codes" [24, 27] and recent theoretical work [40, 41] provide support for a very simple approximation: applying the propagation rules derived above *even in a network with loops*. Table 1 summarizes results from [41]: (1) for Gaussian processes, the MMSE propagation scheme will converge only to the true posterior means. (2) Even for non-Gaussian processes, if the MAP propagation scheme converges, it finds at least a local maximum of the true posterior probability.

2.1 Learning the compatibility functions

One can measure the marginal probabilities relating local scenes, x_i, and images, y_i, as well as neighboring local scenes, x_i and x_j. Iterated Proportional Fitting (e.g., [37, 18]) is a scheme to iteratively modify the compatibility functions until the empirically measured marginal statistics agree with those predicted by the model, Eq. (3). For the problems presented here, we found good results by using the marginal statistics measured from the training data, without modifications by iterated proportional fitting. Based on a factorization described in [10, 9], for a message from scene nodes j to k, we used $\Psi(x_j, x_k) = \frac{P(x_j, x_k)}{P(x_k)}$ and $\Phi(x_j, y_j) = P(y_j | x_j)$. We fit the probabilities with mixtures of Gaussians.

An alternate method, which we find gives comparable results, not shown here, is to use scene and image

Belief propagation algorithm	*Network topology*	
	no loops	arbitrary topology
MMSE rules	MMSE, correct posterior marginal probs.	For Gaussians, correct means, wrong covs.
MAP rules	MAP	Local max. of posterior, even for non-Gaussians

Table 1: Summary of results from [41], assuming convergence of belief propagation.

patches with spatially overlap their neighbors. We assume a Gaussian noise penalty on the multiple observations of the same pixels in the overlap region, yielding $\Psi(x_k, x_j) = \exp^{-(d_k - d_j)^2/2\sigma^2}$, where d_k and d_j are the corresponding values of the scenes described at nodes k and j in their region of common support, and σ is a penalty parameter.

2.2 Probability Representation

Inspired by the success of [17, 8], we use a sample-based representation for inference. We describe the posterior probability as a set of weights on scenes observed in the training set. Given an image to analyze, for each node we collect a set of 10 or 20 "scene candidates" from the training data which have image data closely matching the local observation. We evaluate the posterior probability only at those scene values. The propagation algorithms, Eq. (5) and (4) then are discrete matrix calculations. This simplification focuses the computation on only those scenes which render to the observed image data.

3 Super-resolution

For the super-resolution problem, the input *image* is a low-resolution image. The *scene* to be estimated is a higher resolution image. A good solution to this problem would allow pixel-based images to be handled in a relatively resolution-independent manner. Applications could include enlargment of digital or film photographs, upconversion of video from NTSC format to HDTV, or image compression.

At first, the task may seem impossible—the high resolution data is not there. However, we can see edges in the low-resolution image that we know should remain sharp at the next resolution level. Furthermore, based on the successes of recent texture synthesis methods [14, 8, 42, 36], we might expect to handle textured areas well, too.

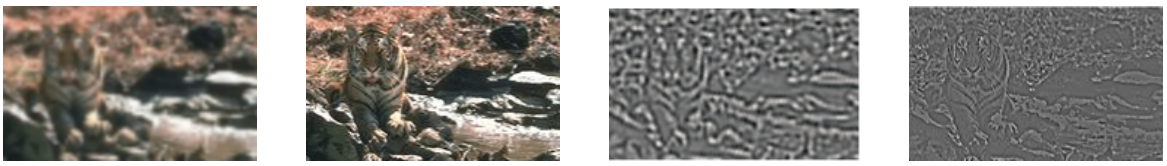
Others [35] have used a Bayesian method, making-up the prior probability. In contrast, the Markov network learns the relationship between sharp and blurred images from large amounts of training data, and achieves better results. Among the non-Bayesian methods, fractal image representation [32] (Fig. 8c) only gathers training data from the one image, while selecting the nearest neighbor from training data [30] misses important spatial consistency constraints (Fig. 4a).

We apply VISTA to this problem as follows. By blurring and downsampling sharp images, we construct a training set of blurred and sharp image pairs. We linearly interpolate each blurred image back up to the original resolution, to form an input *image*. The *scene* to be estimated is the high frequency detail missing from the blurred image, Fig. 2a, b. We then take two image processing steps to ease the modeling burden: (1) we bandpass filter the blurred image, because we believe the lowest frequencies won't predict the highest ones; (2) we normalize both the bandpass and highpassed images by the local contrast [19] of the bandpassed image, because we believe their relationship is independent of local contrast, Fig. 2c, d. We undo this normalization after scene inference.

We extracted center-aligned 7x7 and 3x3 pixel patches, Fig. 3, from the training images and scenes. Applying Principal Components Analysis (PCA) [6] to the training set, we summarized each 3-color patch of image or scene by a 9-d vector. From 40,000 image/scene pair samples, we fit 15 cluster Gaussian mixtures to the marginalized probabilities, assuming spatial translation invariance. For efficiency, we pruned frequently occurring image/scene pairs from the training set.

Given a new image, not in the training set, from which to infer the high frequency scene, we found the 10 training samples closest to the image data at each node (patch). The 10 corresponding scenes are the candidates for that node. We evaluated $\Psi(x_j, x_k)$ at 100 values (10 x_j by 10 x_k points) to form a compatibility matrix for messages from neighbor nodes j to k. We propagated the probabilities by Eq. (5).

To process Fig. 5a, we used a training set of 80 images from two Corel database categories: African grazing animals, and urban skylines. Figure 4a shows the nearest neighbor solution, at each node using the scene corresponding to the closest image sample in the training set. Many different scene patches can explain each image patch, and the nearest neighbor solution is very choppy. Figures 4b, c, d show the first 3 iterations of MAP belief propagation. The spatial consistency imposed by the belief propagation finds plausible and



(a) input (b) desired output (c) image (d) scene

Figure 2: We want to estimate (b) from (a). The original image, (b) is blurred, subsampled, then interpolated back up to the original resolution to form (a). The missing high frequency detail, (b) minus (a), is the "scene" to be estimated, (d) (this is the first level of a Laplacian pyramid [7]). The low frequencies of (a) are removed to form the input bandpassed "image". We contrast normalize the image and scene by the local contrast of the input bandpassed image, yielding (c) and (d).

Figure 3: Training data samples for super-resolution problem. The large squares are the *image* data (mid-frequency data). The small squares above them are the corresponding *scene* data (high-frequency data).

consistent high frequencies for the tiger image from the candidate scenes. Figure 5 shows the result of applying this method recursively to zoom two octaves. The algorithm keeps edges sharp and invents plausible textures. Standard cubic spline interpolation, blurrier, is shown for comparison.

Figure 6 explores the algorithm behavior under different training sets. The estimated images properly reflect the structure of the training worlds for noise, rectangles, and generic images. Figure 8 depicts in close-up the interpolation for image (a) using an ideal training set of images taken at the same place and same time (but not of the same subject) (d), and a generic training set of images (e) (Fig. 7 shows the training sets). Both estimates look more similar to the true high resolution result (f) than either cubic spline interpolation (b) or zooming by a fractal image compression algorithm (c). Edges are again kept sharp, while plausible texture is synthesized in the hair.

4 Motion Estimation

To show the breadth of the VISTA technique, we apply it to the problem of motion estimation. The *scene* data to be estimated are the projected velocities of moving objects. The *image* data are two successive image frames. Because we felt long-range interactions were important, we built Gaussian pyramids (e.g., [19]) of both image and scene data, connecting patches to nearest neighbors in both scale and position.

Luettgen et al. [26] applied a related message-passing scheme in a multi-resolution quad-tree network to estimate motion, using Gaussian probabilities. While the network did not contain loops, its structure generated artifacts along quad-tree boundaries, artificial statistical boundaries of the model.

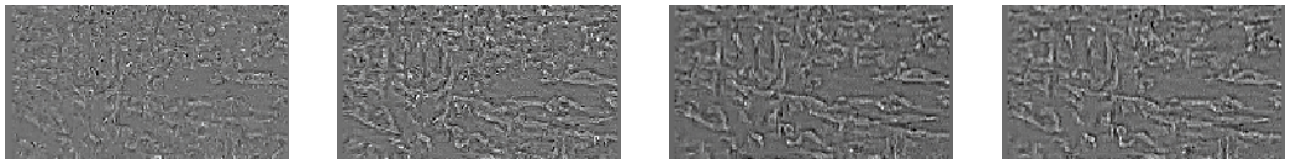
To show the algorithm working on simple test cases, we generated a synthetic world of moving blobs, of random intensities and shapes. We wrote a tree-structured vector quantizer, to code 4 by 4 pixel by 2 frame blocks of image data for each pyramid level into one of 300 codes for each level, and likewise for scene patches.

During training, we presented approximately 200,000 examples of irregularly shaped moving blobs of a contrast with the background randomized to one of 4 values. For this vector quantized representation, we used co-occurance histograms to measure the compatibility functions, see [10].

Figure 10 shows six iterations of the inference algorithm (Eqs. 4 and 5) as it converges to a good estimate for the underlying scene velocities. The same machinery we applied to super-resolution leads to, for this problem, figure/ground segmentation, aperture problem constraint propagation, and filling-in (see caption). The resulting inferred velocities are correct within the accuracy of the vector quantized representation.

5 Summary

We described an approach we call VISTA–Vision by Image/Scene TrAining. One specifies prior probabilities on scenes by generating typical examples, creating a synthetic world of scenes and rendered images. We break the images and scenes into a Markov network, and learn the parameters of the network from



(a) Nearest neighbor (b) belief prop., iter. 0 (c) belief prop., iter. 1 (d) belief prop., iter. 3

Figure 4: (a) Nearest neighbor solution. The choppiness indicates that many feasible high resolution scenes correspond to a given low resolution image patch. (b), (c), (d): iterations 0, 1, and 3 of Bayesian belief propagation. The initial guess is not the same as the nearest neighbor solution because of mixture model fitting to $P(y|x)$. Underlying the most probable guess shown are 9 other scene candidates at each node. 3 iterations of Bayesian belief propagation yields a probable guess for the high resolution scene, consistent with the observed low resolution data, and spatially consistent across scene nodes.

(a) 85 x 51 input (b) cubic spline (c) belief propagation

Figure 5: (a) 85 x 51 resolution input. (b) cubic spline interpolation in Adobe Photoshop to 340x204. (c) belief propagation zoom to 340x204, zooming up one octave twice.

the training data. To find the best scene explanation given new image data, we apply belief propagation in the Markov network, an approach supported by experimental and theoretical studies.

The intuitions of this paper–propagate local estimates to find a best, global solution–have a long tradition in computational vision [1, 33, 15, 31]. The power of the VISTA approach lies in the large training database, allowing rich prior probabilities and rendering models, and the belief propagation, allowing efficient scene inference.

Applied to super-resolution, VISTA gives results that we believe are the state of the art. Applied to motion estimation, the same method resolves the aperture problem and appropriately fills-in motion over a figure. The technique may apply to related vision problems as well, such as line drawing interpretation, or distinguishing shading from reflectance.

Acknowledgements We thank Y. Weiss, E. Adelson, A. Blake, J. Tenenbaum, and P. Viola for helpful discussions. Thanks to O. Carmichael and J. Haddon for verifying the method of overlapping patches for computing compatibility functions.

References

[1] H. G. Barrow and J. M. Tenenbaum. Computational vision. *Proc. IEEE*, 69(5):572–595, 1981.

[2] A. J. Bell and T. J. Senjowski. The independent components of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

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

[4] J. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. Royal Statist. Soc. B*, 36:192–326, 1974.

[5] T. Binford, T. Levitt, and W. Mann. Bayesian inferernce inmodel-based machine vision. In J. F. Lemmer and L. M. Kanal, editors, *Uncertainty in artificial intelligence*. Elsevier Science, 1988.

[6] C. M. Bishop. *Neural networks for pattern recognition*. Oxford, 1995.

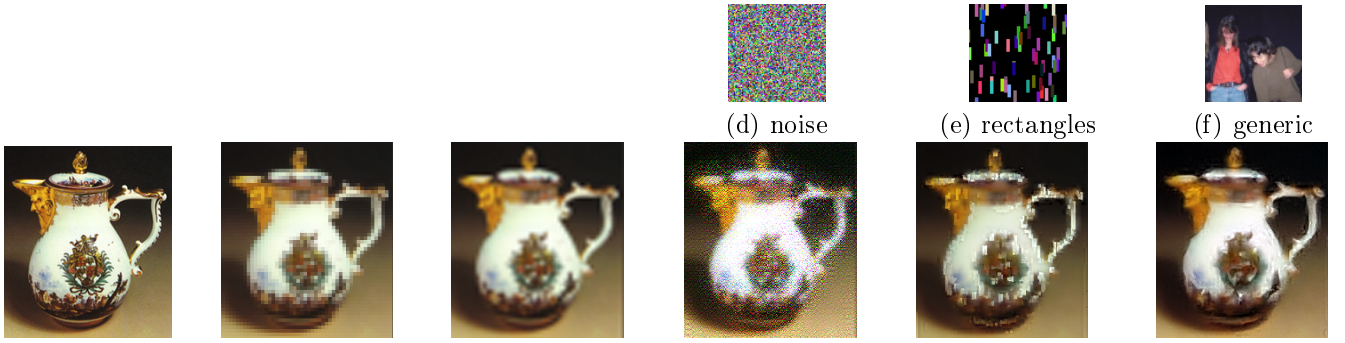
[7] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Comm.*, 31(4):532–540, 1983.

[8] J. S. DeBonet and P. Viola. Texture recognition using a non-parametric multi-scale statistical model. In *Proc. IEEE Computer Vision and Pattern Recognition*, 1998.

[9] W. T. Freeman and E. Pasztor. Markov networks for low-level vision. Technical report, MERL, a Mitsubishi Electric Research Lab., 1999. http://www.merl.com/reports/TR99-08/.

[10] W. T. Freeman and E. C. Pasztor. Learning to estimate scenes from images. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Adv. Neural Information Processing Systems*, volume 11, Cambridge, MA, 1999. MIT Press. See also http://www.merl.com/reports/TR99-05/.

[11] B. J. Frey. *Graphical Models for Machine Learning and Digital Communication*. MIT Press, 1998.



(a) Actual (b) Input (c) Cubic spline (d) noise (e) rectangles (f) generic (g) Train: noise (h) Train: rects (i) Train: generic

Figure 6: Effect of different training sets. (a) was blurred, and subsampled by 4 in each dimension to yield the low-resolution input, (b). Cubic spline interpolation to full resolution in Adobe Photoshop loses the sharp edges, (c). We recursively zoomed (b) up two factors of two using the Markov network trained on 10 images from 3 different "worlds": (d) random noise, (e) colored rectangles, and (f) a generic collection of photographs. The estimated high resolution images, (g), (h), and (i), respectively, reflect the statistics of each training world.

images from "picnic" training set images from "generic" training set

Figure 7: Sample images from the 10 images in the "picnic" and "generic" training sets. Sharp and blurred versions of these images were used to create the training data for Fig. 8d and e.

[12] D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRF's: surface reconstruction. *IEEE Pattern Analysis and Machine Intelligence*, 13(5):401–412, May 1991.

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

[14] 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.

[15] B. K. P. Horn. *Robot vision*. MIT Press, 1986.

[16] A. C. Hurlbert and T. A. Poggio. Synthesizing a color algorithm from examples. *Science*, 239:482–485, 1988.

[17] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. on Computer Vision*, pages 343–356, 1996.

[18] T. Jaakola. Machine learning seminar notes, 1999. http://www.ai.mit.edu/people/tommi/class/udest.ps.

[19] B. Jahne. *Digital Image Processing*. Springer-Verlag, 1991.

[20] M. I. Jordan, editor. *Learning in graphical models*. MIT Press, 1998.

[21] D. Kersten, A. J. O'Toole, M. E. Sereno, D. C. Knill, and J. A. Anderson. Associative learning of scene parameters from images. *Applied Optics*, 26(23):4999–5006, 1987.

[22] J. Kittler and J. Illingworth. Relaxation labelling algorithms–a review. *Image and Vision Computing*, (11):206–216, 1985.

[23] D. Knill and W. Richards, editors. *Perception as Bayesian inference*. Cambridge Univ. Press, 1996.

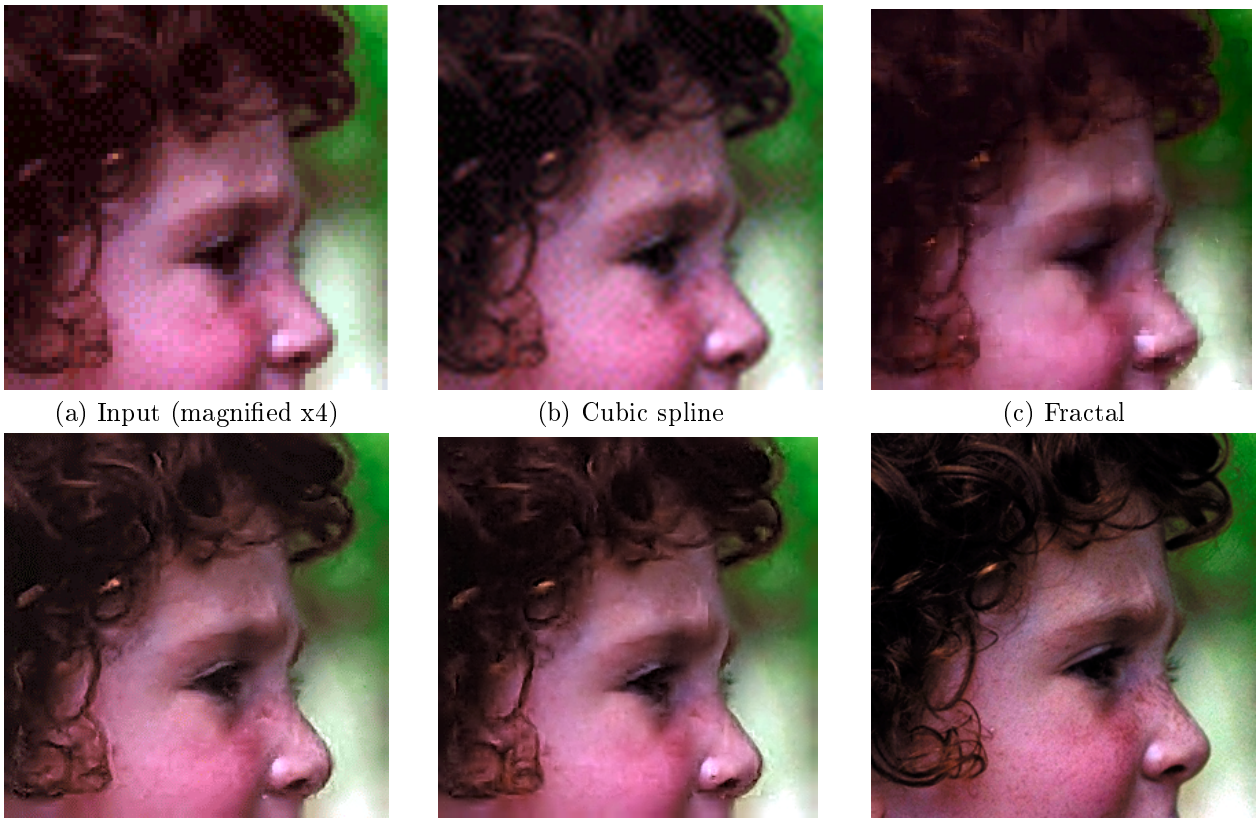
[24] F. R. Kschischang and B. J. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Journal on Selected Areas in Communication*, 16(2):219–230, 1998.

[25] M. S. Landy and J. A. Movshon, editors. *Computational Models of Visual Processing*. MIT Press, Cambridge, MA, 1991.

[26] M. R. Luettgen, W. C. Karl, and A. S. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Trans. Image Processing*, 3(1):41–64, 1994.

[27] R. McEliece, D. MackKay, and J. Cheng. Turbo decoding as as an instance of Pearl's 'belief propagation' algorithm. *IEEE Journal on Selected Areas in Communication*, 16(2):140–152, 1998.

[28] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a



(a) Input (magnified x4) (b) Cubic spline (c) Fractal

(d) "Picnic" training set (e) "Generic" training set (f) Actual full-resolution

Figure 8: (a) Low-resolution input image. (b) Cubic spline 400% zoom in Adobe Photoshop. (c) Zooming luminance by public domain fractal image compression routine [32], set for maximum image fidelity (chrominance components were zoomed by cubic spline, to avoid color artifacts). Both (c) and (d) are blurry, or have serious artifacts. (d) Markov network reconstruction using a training set of 10 images taken at the same picnic, none of this person. This is the best possible fair training set for this image. (e) Markov network reconstrution using a training set of *generic* photographs, none at this picnic or of this person, and fewer than 50% of people. The two Markov network results show good synthesis of hair and eye details, with few artifacts, but (d) looks slightly better (see brow furrow). Edges and textures seem sharp and plausible. (f) is the true full-resolution image.

sparse code for natural images. *Nature*, 381:607–609, 1996.

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

[30] A. Pentland and B. Horowitz. A practical approach to fractal-based image compression. In A. B. Watson, editor, *Digital images and human vision*. MIT Press, 1993.

[31] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(26):314–139, 1985.

[32] M. Polvere. Mars v. 1.0, a quadtree based fractal image coder/decoder, 1998. http://inls.ucsd.edu/y/Fractals/.

[33] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Systems, Man, Cybern.*, 6(6):420–433, 1976.

[34] E. Saund. Perceptual organization of occluding contours generated by opaque surfaces. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Ft. Collins, CO, 1999.

[35] R. R. Schultz and R. L. Stevenson. A Bayesian approach to image expansion for improved definition. *IEEE Trans. Image Processing*, 3(3):233–242, 1994.

[36] E. P. Simoncelli. Statistical models for images: Compression, restoration and synthesis. In *31st Asilomar Conf. on Sig., Sys. and Computers*, Pacific Grove, CA, 1997.

[37] P. Smyth, D. Heckerman, and M. I. Jordan. Probabilistic independence networks for hidden Markov

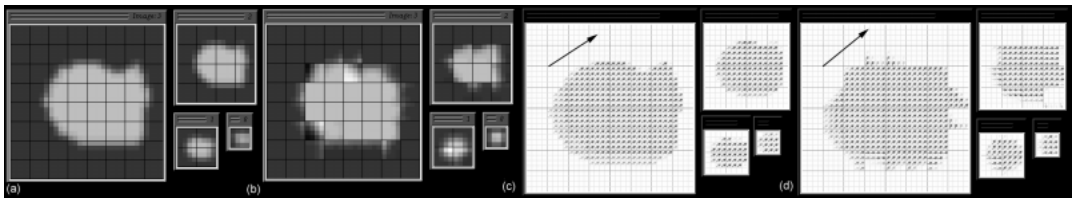


Figure 9: (a) First of two frames of image data (in Gaussian pyramid), and (b) vector quantized. (c) The optical flow scene information, and (d) vector quantized. Large arrow added to show small vectors' orientation.

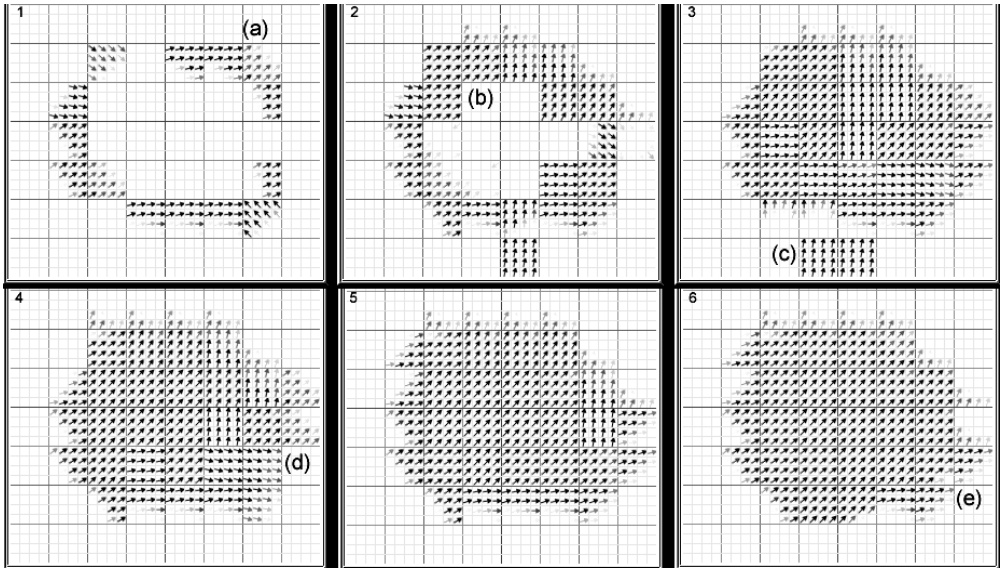


Figure 10: The most probable scene code for Fig. 9b at first 6 iterations of Bayesian belief propagation. (a) Note initial motion estimates occur only at edges. Due to the "aperture problem", initial estimates do not agree. (b) Filling-in of motion estimate occurs. Cues for figure/ground determination may include edge curvature, and information from lower resolution levels. Both are included implicitly in the learned probabilities. (c) Figure/ground still undetermined in this region of low edge curvature. (d) Velocities have filled-in, but do not yet all agree. (e) Velocities have filled-in, and agree with each other and with the correct velocity direction, shown in Fig. 9.

probability models. *Neural Computation*, 9(2):227–270, 1997.

[38] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-level Vision*. Kluwer Academic Publishers, Boston, 1989.

[39] Y. Weiss. Interpreting images by propagating Bayesian beliefs. In *Adv. in Neural Information Processing Systems*, volume 9, pages 908–915, 1997.

[40] Y. Weiss. Belief propagation and revision in networks with loops. Technical Report 1616, AI Lab Memo, MIT, Cambridge, MA 02139, 1998.

[41] Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. Technical Report UCB.CSD-99-1046, Berkeley Computer Science Dept., 1999. www.cs.berkeley.edu/~ yweiss/gaussTR.ps.gz.

[42] S. C. Zhu and D. Mumford. Prior learning and Gibbs reaction-diffusion. *IEEE Pattern Analysis and Machine Intelligence*, 19(11), 1997.