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

Example-Based Super-Resolution

William T. Freeman, Thouis R. Jones, and Egon C. Pasztor*
MERL, Mitsubishi Electric Research Labs.
201 Broadway
Cambridge, MA 02139

TR-2001-30 August 2001

Abstract

Image-based models for computer graphics lack resolution independence: they cannot be zoomed much beyond the pixel resolution they were sampled at without a degradation of quality. Interpolating images usually results in a blurring of edges and image details. We describe image interpolation algorithms which use a database of training images to create plausible high-frequency details in zoomed images. Image pre-processing steps allow the use of image detail from regions of the training images which may look quite different from the image to be processed. These methods preserve fine details, such as edges, generate believable textures, and can give good results even after zooming multiple octaves.

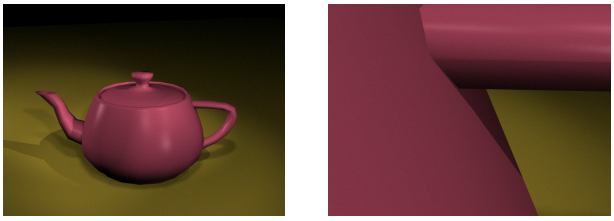
1. First printing, TR2001-30, August, 2001.

* Egon Pasztor's present address:
MIT Media Lab
20 Ames St.
Cambridge, MA 02139

Example-based Super-resolution

William T. Freeman, Thouis R. Jones, Egon C. Pasztor

Mitsubishi Electric Research Laboratories (MERL)
201 Broadway
Cambridge, MA 02139



(a) (b)

Figure 1: An object modelled by traditional polygon techniques may lack some of the richness of real-world objects, but behaves properly under zooming, (b).

Abstract

Image-based models for computer graphics lack resolution independence: they cannot be zoomed much beyond the pixel resolution they were sampled at without a degradation of quality. Interpolating images usually results in a blurring of edges and image details. We describe image interpolation algorithms which use a database of training images to create plausible high-frequency details in zoomed images. Image pre-processing steps allow the use of image detail from regions of the training images which may look quite different from the image to be processed. These methods preserve fine details, such as edges, generate believable textures, and can give good results even after zooming multiple octaves.

1 Introduction

As shown in Fig. 1, polygon-based representations of 3-dimensional objects offer resolution independence over a wide range of scales. Object boundaries remain sharp as one zooms in on the object until very close range, where faceting appears due to finite polygon size.

Constructing polygon models for complex, real-world objects can be difficult. Image-based rendering (IBR) is a complementary approach for representing and rendering objects, using cameras to obtain rich models directly from real-world data. Unfortunately, these representations no longer have resolution independence. When we zoom into a bitmapped image, we get a blurred image. Figure 2 shows the problem for an IBR "version" of teapot image, rich with real-world detail. We know the teapot's features should remain sharp as we zoom in on them, yet standard pixel interpolation methods, such as pixel replication (b, c) and cubic spline interpolation (d, e), introduce artifacts or blurring of edges. For images zoomed 3 octaves, such as these, sharpening the interpolated result has little useful effect (f, g).

A method to achieve higher resolution views of pixel-based image representations, which we will call super-resolution, would have some of the best of both worlds, complex models and resolution independence. In addition, many other applications in graphics or image processing could benefit from such pixel resolution independence, such as texture mapping, enlarging consumer photographs, and converting NTSC video content to HDTV. We don't expect perfect resolution independence—even the polygon representation doesn't have that—but increasing the resolution independence of pixel-based representations is an important task for image-based rendering. Our example-based super-resolution algorithm yields Fig. 2 (h, i).

2 Related approaches

Figure 3 shows several complementary ways to increase the apparent resolution of an image: (a) sharpening, (b) aggregation from multiple frames, and (c) single-frame super-resolution. We feel each should be used wherever possible. Sharpening amplifies details that are present in the image. Integrating resolution information over multiple frames is sometimes called super-resolution. For the purposes of this paper, we will always mean single-frame super-resolution.

Super-resolution relates to image interpolation—how should one interpolate between the digital samples of a photograph? Researchers have long studied this problem, although only recently using machine learning or sampling approaches, which offer much power.

Cubic spline interpolation [9] is a very common image interpolation function, but suffers from blurring of edges and image details. Recent attempts to improve on cubic spline interpolation [12, 16, 3] have met with limited success. Schreiber and collaborators [12] proposed a sharpened Gaussian interpolator function to minimize information spillover between pixels and optimize flatness in smooth areas. Schultz and Stevenson [13] have used a Bayesian method for super-resolution, but hypothesized the prior probability.

These analytic approaches often suffer from percieved loss of detail in textured regions. A proprietary, undisclosed algorithm, Altamira Genuine Fractals 2.0 [1] (an Adobe Photoshop plug-in), does as well as any of the non-training-based methods, but still suffers from blur in regions of texture and at fine lines.

2.1 Example-based approaches

One would expect that the richness of real-world images would be difficult to capture analytically. This motivates a learning-based approach: in a training set, learn the fine details that correspond to different image regions seen at a low-resolution; then use those learned relationships to predict fine details in other images. For the past several years [5, 6], we have been exploring this approach for enlarging images.

To motivate why this approach should work at all, note that a collection of image pixels are special signals which have much less variability than would a corresponding set of completely random variables. Researchers have studied these regularities to account for the early processing stages of the mammalian visual systems [4, 15]. We exploit these regularities in our algorithms, as well: we use small pieces of one image, modified for generalization by the

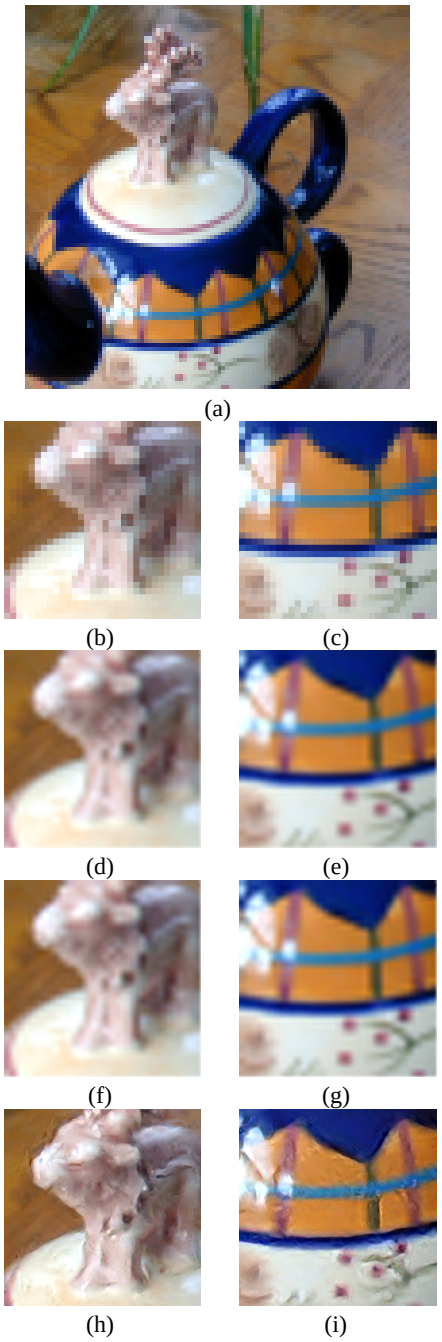


Figure 2: (a) An image (100x100) of a real-world teapot shows a richness of texture, but yields a blocky or blurred image when zoomed in by a factor of 8 in each dimension by (b, c) pixel replication or (d, e) cubic spline interpolation. (Images (b) through (i) were 32x32 pixel original sub-images, zoomed by 8 to 256x256 images). Sharpening the cubic spline interpolation may not help to increase the perceptual sharpness (f, g, using "sharpen more" in Adobe Photoshop). (h, i) show the results of our one-pass super-resolution algorithm, maintaining edge and line sharpness, and plausible texture details.

appropriate pre-processing, to create plausible image information in a second image. Without very specific training data, it is not reasonable to expect to generate the *correct* high-resolution information. We aim for the more attainable goal of generating *visually plausible* image details, such as sharp edges, and plausible looking texture.

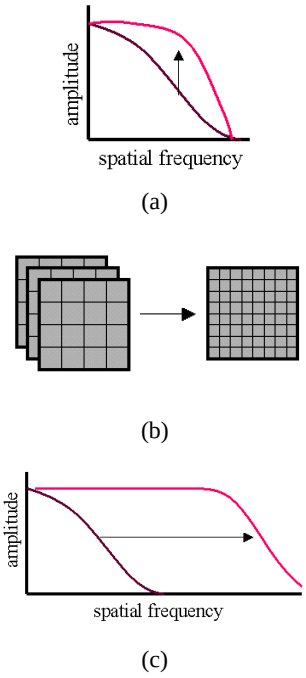


Figure 3: Different complementary approaches to increase the perceptual resolution of an image. (a) Shows schematically the change in the spatial frequency amplitude spectrum of an image associated with *image sharpening.* Existing high frequencies in the image are amplified. This is often useful to do, provided noise isn't amplified, but not the subject of super-resolution. (b) Extracting a single high-resolution frame from a sequence of low-resolution video images is useful, and is also sometimes referred to as super-resolution. (c) The super-resolution goal of this paper is to estimate missing high-resolution detail that is not present in the original image, and which cannot be made visible by simple sharpening.

3 Training set generation

To generate our training set, we start from a collection of high resolution images, and degrade each of them in a manner corresponding to the degradation we plan to undo in the images we later process. Typically, we blur and subsample them to create a low-resolution image of $\frac{1}{4}$ the number of original pixels.

We apply an initial analytic interpolation, such as cubic spline, to the low-resolution image. We only need to store the differences between a cubic spline interpolation of the image, and the true, high resolution image. Figure 4 (a) and (c) show low and high resolution versions of an image; (b) is the initial up-interpolation (bilinear was used for this example).

We want to store the high-resolution patch corresponding to every possible low-resolution image patch; these patches are typically 5x5 and 7x7 pixels, respectively. Even restricting ourselves to plausible image information, this is a huge amount of information to store, so we need to pre-process the images to remove variablitity and make the training sets as generally applicable as possible.

We believe that the highest resolution components of the low-resolution image (b) are most important in predicting the extra details present in (c). We filter out the lowest-frequency components of (b), so that we don't have to store example patches for all possible lowest frequency component values. We also believe that the relationship between high and low-resolution image patches is essentially independent of local image contrast, and we don't want to have to store examples of that relationship for all possible values of the local image contrast. The resulting bandpass filtered and con-

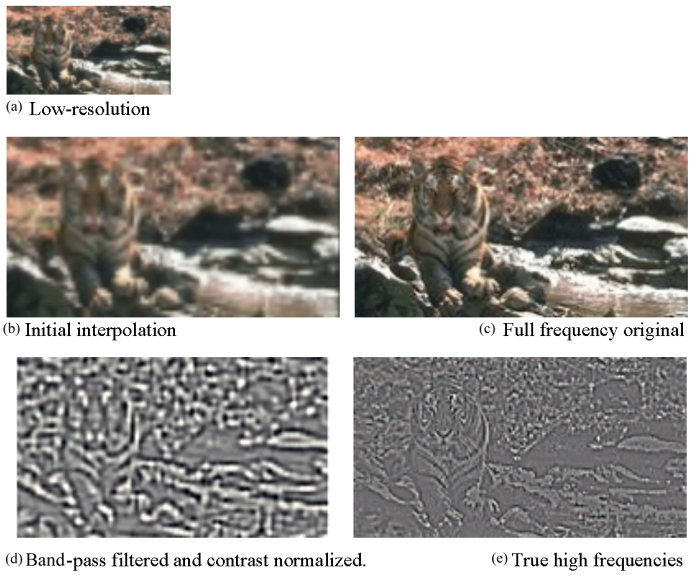


Figure 4: Image pre-processing steps for training images. We start from a low-resolution image, (a), and its corresopnding high-resolution source, (c). We form an initial interpolation, (b), of the low-resolution image to the higher pixel sampling resolution. In the training set, we store corresponding pairs of patches from (f) and (e), which are the bandpass or highpass filtered and contrast normalized versions of (b) and (c), respectively. This processing allows the same patch pair examples to apply in different image contrasts and low-frequency offsets.

trast normalized image pairs used for training are shown in Fig. 4 (d) and (e). We undo the contrast normalization step upon reconstruction of the high-resolution image.

3.1 Markov network algorithm

If local image information alone were sufficient to predict the missing high resolution details, we should be able to use the training set patches by themselves for super-resolution. For a given input image to enlarge, we would apply the pre-processing steps, break the image into patches, and look-up the missing high resolution detail. Unfortunately, that approach doesn't work, as illustrated in Fig. 5 (a); the high resolution detail image looks like oatmeal. The local patch alone is not sufficient to estimate plausible looking high resolution detail.

Fig. 5 (b) illustrates why. For a given low-resolution input patch, we searched a typical training database of about 100,000 patches to find the 16 closest examples to the input patch, shown in the second line of Fig. 5 (b). Each of these looks fairly similar to the input patch. The bottom row shows the high resolution detail corresponding to each of these training examples; each of those looks quite distinct from the other. This illustrates that local patch information alone is not sufficient for super-resolution; spatial neighborhood effects must be taken into account.

We have modelled the spatial relationships between patches using a Markov network, for which well-known uses in image processing include [7]. In Fig. 6, the circles represent network nodes, and the lines indicate statistical dependencies between nodes. We let the low-resolution image patches be observation nodes, y. We select the 16 or so closest examples to each input patch as the different states of the hidden nodes, x, that we seek to estimate. For this network, the probability of any given high-resolution patch choice for each node is the product of all sets of *compatibility matrices* ψ relating the possible states of each pair of neighboring hidden nodes, and

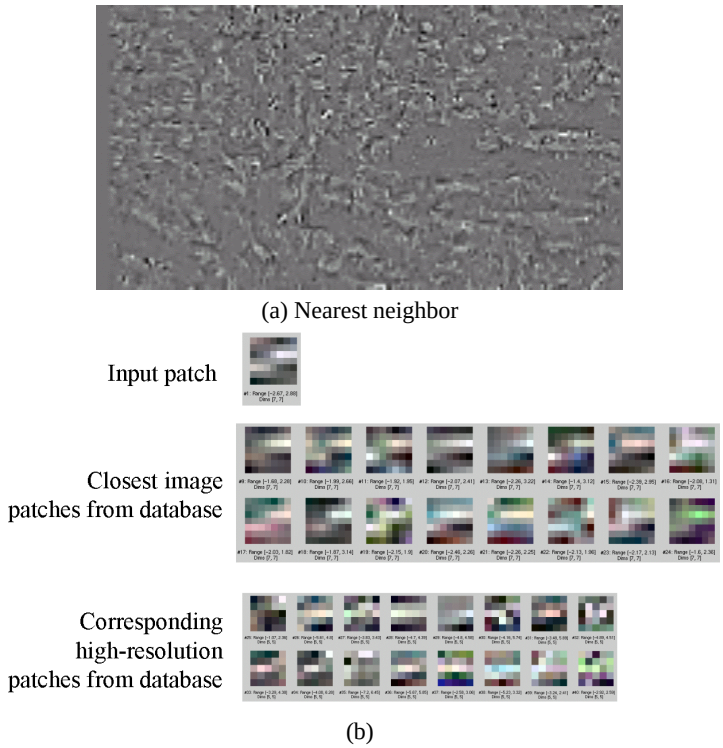


Figure 5: (a) Estimated high frequencies for tiger image (Fig. 4 (e) are the true high frequencies) formed by substituting the high frequencies of the closest training patch to Fig. 4 (d). The lack of a recognizable image indicates that a nearest neighbor algorithm is not sufficient; spatial context must also be used. (b) An input patch, and similar low-resolution (middle rows) and paired high-resolution (bottom rows) patches. For many of these similar low-resolution patches, the high-resolution patches are quite different, reinforcing the lesson from (a) above.

matrices ϕ relating each observation to the underlying hidden states.

To specify the ψ functions of the Markov network, we use a simple trick. We sample the nodes of the input image so that the high-resolution patches overlap with each other by one or more pixels. In the region of overlap, the pixel values of compatible neighboring patches should agree. We measure d_{ij}^{ab}, the sum of squared differences between patch candidates i and j at nodes a and b. The compatibility matrix between nodes a and b is then $\phi_{ab}(i,j) = \exp(-\frac{(d_{ij}^{ab})^2}{2\sigma})$, where σ is a noise parameter. We use a similar quadratic penalty on differences between the observed low-resolution image patch, and the candidate low-resolution patch found from the training set, to specify the Markov network compatibility function, ψ.

The optimal high-resolution patches at each node is that collection which maximizes the probabilty of the Markov network. Finding the exact solution can be computationally intractible, but we have found good results using the approximate solution obtained by running a fast, iterative algorithm called Belief Propagation. Typically, 3 or 4 iterations of the algorithm are sufficient, see Fig. 7.

3.2 Single-pass algorithm

The fact that belief propagation converged to a solution of the Markov network so quickly led us to believe the problem was not a difficult one. We found a simple one-pass algorithm which gives results that are nearly as good as the iterative solution to the Markov network.

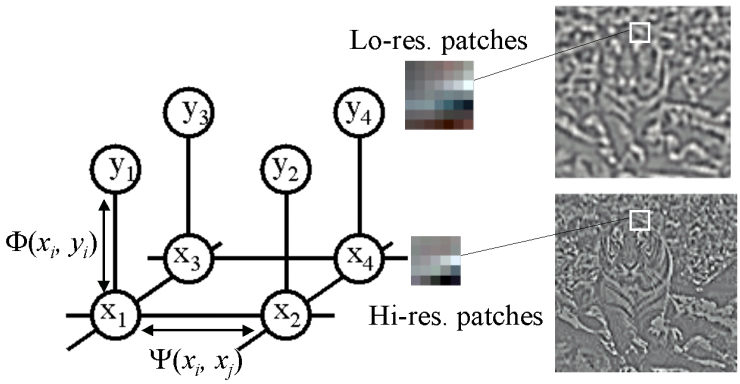


Figure 6: Markov network model for super-resolution problem.

In our algorithm, we only compute high resolution patch compatibilities for neighboring patches that are already fixed, typically the patches above and to the left, in raster-scan order processing. If we pre-structure the training data properly (Figure 11), matching the local low-resolution image data as well as selecting the compatible high resolution patch candidate can all be done in a single operation: finding the nearest neighbor to a given input vector in the training set. The simplification avoids steps of finding the compatibility matrices and the iterative belief propagation algorithm, with negligible reduction in image quality.

Figure 8 shows an image zoomed with super-resolution, along with the same image zoomed with cubic spline and the true high resolution image. At the bottom are the images from the training set used in the super-resolution zoom. The center section shows the details of a few patches in the zoomed image and their corresponding best matches in the training set. The top and bottom rows show the image content of the patches in the super-resolution image and the training set. The second and fifth rows show the low resolution, contrast normalized patches. The third row shows the high resolution content of the true high resolution image, and the fourth shows the high resolution patch chosen by the super-resolution algorithm. Although not perfect, the matches between the true and estimated high resolution patches are reasonably good. Note that the algorithm is able to make use of training patch examples from source image regions that look very different than the regions where they are inserted into the zoomed image. For example, the orange bordered patch corresponds to a shadow boundary on wood in the training image (of 3 girls), but is applied to zoom up a green plant occlusion boundary. The bandpass filtering and contrast normalization allows for this re-use, which makes the training set more powerful.

4 Single-pass algorithm details

In the simplest terms, one-pass super-resolution generates the missing high frequency content of a zoomed image as a sequence of predictions from local image information. The input image is subdivided into low-frequency patches which are traversed in raster scan order. At each step, a high-frequency patch is selected from the training set based on the local low-frequency details as well as adjacent, previously determined high-frequency patches.

In the algorithms described below, (nonpredictive) scaling up of images is performed via cubic spline interpolation, and scaling down by convolving with a [0.25 0.5 0.25] blurring filter and subsampling on the even indices. ([6] used linear interpolation for the upsampling, which puts more interpolation burden on the rest of the algorithm).

4.1 Prediction

Given the highest frequencies in an input image, the super-resolution algorithm predicts the next octave up, i.e. the frequencies missing from an image zoomed with cubic interpolation. The output of the algorithm is the sum of its input and the high frequency predictions (Figure 9).

The high frequencies are predicted for NxN pixel patches at a time, in raster-scan order. Each prediction is based on two competing requirements. First, the high frequency patch should come from a location in the training image that has a similar low-frequency appearance. Second, the high-frequency prediction should agree at the edges of the patch with its neighbors, to prevent discontinuities.

The first requirement can be fulfilled by extracting a low-frequency patch (MxM, not necessarily the same size as the high frequency patch) from the image we are zooming and searching for a match in the training set made up of pairs of low- and high-frequency patches. To meet the second criterion, we overlap predicted patches at their borders (Figure 10). When searching the training set, the high-frequency data previously predicted is also used in selecting the best pair. A user-controlled weighting factor α is used to adjust the relative importance of the low frequency patch versus the overlap with high frequency patches.

The super-resolution algorithm operates under the assumption that the predictive relationship between low and high frequency patches is independent of contrast, and we therefore normalize patch pairs by the average absolute value of the low frequency patch, across the color channels (plus some small ϵ to avoid overflow).

The pixels in low-frequency patch and the high-frequency overlap are concatenated to form a search vector. The training set is also stored as a set of vectors, so searching for a match can be accomplished by finding the nearest neighbor in the training set. When a match is found, the contrast normalization is reversed on the high-frequency patch, and it is added to the output image (Figure 11).

4.1.1 Search algorithm

We search for matches using an L_2 norm. Due to the high dimension of the search space, finding the absolute best match would be computationally prohibitive. Instead, we use a tree-based, approximate nearest neighbor search. The tree is built by recursively splitting the training set in the direction of higher variation. At each step we divide the set of tiles in half, to maintain a balanced tree.

We use "best-first" search of the tree to find a good match. This allows for a speed-quality tradeoff: by searching more branches of the tree we can find a better match. Since best-first search is unlikely to give the true best match without searching most or all of the tree, we improve the best-first match by a greedy walk in the graph connecting approximate nearest neighbors in the training set. This improves the match with negligible cost. In all examples in this paper, we connect each patch pair to its 32 approximate nearest neighbors, computed by a method similar to [10].

4.2 Training

Training sets for the super-resolution algorithm are built from bandpass and highpass pairs taken from a set of training images. Spatially corresponding MxM low-frequency and NxN high-frequency patches are taken from image pairs at a set of sampling locations (usually, M=7, N=5, and samples are taken at every pixel).

Patch pairs are contrast normalized as described above. The search vector for a patch pair is created by the concatenation of the low-frequency patch and the region that will be overlapped in the high-

frequency patch during the prediction phase, adjusted by the consistency weighting factor α (Figure 11).

We used the same set of training images for all the super-resolution examples in this paper, shown in Figure 12. They were taken with a Nikon Coolpix 950 digital camera at 640x480 resolution and highest quality compression settings.

4.3 Parameter settings

Attention to parameter settings can improve image quality. For both levels of zooming, we used 5x5 pixel high-resolution patches (N=5) with 7x7 pixel low-resolution patches (M=7). The overlap between adjacent high-resolution patches was 1. These settings do well to capture small details. Larger patch sizes can be used for less accentuation of small details.

For a more conservative estimate of the higher resolution detail, the algorithm can be performed 4 times at staggered offsets relative to the patch sampling grid. This gives 4 independent estimates of the high frequencies, which can then be averaged together, smoothing some image details.

The parameter α controls the tradeoff between matching the low-resolution patch data and finding a high-resolution patch that is compatible with its neighbors. The value $\alpha = 0.1\frac{M^2}{2N-1}$ gave good quality results in our experiments. The fraction adjusts for the relative areas of low-frequency patches and overlapped high-frequency pixels.

5 Results

Figures 13 and 14 shows our algorithm applied to a brick wall and a man's face. The training set was taken from the images in Figure 12. The resulting zooms are significantly sharper than those from cubic spline interpolation, preserving sharp edges and image details.

Figure 15 shows an example where our low-level training set alone is not enough to distinguish JPEG compression noise from correct image data; the algorithm interprets the artifacts as image data and enhances them. Extensions of specialized high-level models such as [2] could be needed.

In a simple exploration of the relation of the zoomed image to the training images (see [6] for others), we enlarged the image of Fig. 8 using a pathological training set of images of text. Nonetheless, the algorithm does its best to explain the observed low-resolution image in its vocabulary of text examples, resulting in a zoom with high resolution detail formed out of concatenated characters, Fig. 16.

5.1 Discussion

Recent work by Hertzmann *et al* [8] has also used a training-based method to perform super-resolution, in the context of analogies between images. Our method differs in that it operates on tiles rather than per-pixel, providing a performance benefit. It also normalizes the training set according to contrast, and assumes that the highest frequency details in an image can be predicted using only the next lower octave. These two generalizations allow us to zoom a wider class of images using a single, generic training set, rather than being restricted to operating on images that are very similar to the training image.

A training-based approach was used by [11], but no attempt was made to enforce the spatial consistency constraints necessary for good image quality.

If well-known objects are sparsely sampled in the image, an image extrapolation based on local image evidence alone will not produce the new details that the viewer expects. Very small face images are suseptable to this problem. To address these properly, higher-level reasoning would have to be added to the algorithm. Baker and Kanade [2] have recently explored super-resolution algorithms tuned to a particular class of images, such as faces or text.

In the zoomed-up images, low-contrast details next to high contrast edges may be lost, due to the contrast normalization fixing on the level of the high contrast edge. Independent contrast normalization for different image orientations, each zoomed separately, might address this problem. However, it is not clear that a one-pass implementation would suffice for that modification.

Finally, the algorithm works best when the resolution or noise degradations of the data match those of the images to which it is applied.

Numerically, the root-mean-squared error from the true high frequencies tend to be approximately the same as for the original cubic spline interpolation. Unfortunately, this metric has only a loose correlation with percieved image quality [12]. Typical processing times for the single-pass algorithm are two seconds to enlarge a 100x100 image up to 200x200 pixels.

We have focussed on the case of enlarging single images. The case of enlarging moving images is different in two respects: (1) there is more input data; multiple observations of the same pixel could be used for super-resolution. (2) Care must be taken to ensure coherence across subsequent frames so that the made-up image details do not scintillate in the moving image.

6 Conclusions

There is a surprising regularity across images, such that a training set made from the images of Fig. 12 can be used to invent missing details in many others (all those in this paper). While a training set tuned to the images to be processed of course works best, a training set of generic images can handle a very broad class of inputs.

We have built on the training-based super-resolution algorithm of [6], and introduced a faster, simpler, and, we believe, better algorithm for one-pass super-resolution. The algorithm requires only a nearest-neighbor search in the training set for a vector derived from each patch of local image data. This one-pass super-resolution algorithm is a step toward achieving resolution independence in image-based representations.

These algorithms are an instance of a general training-based approach that may be useful for image processing or graphics applications (see [6, 14]). Training sets can be built to help enlarge images, remove noise, estimate 3-d surface shapes, and attack other imaging applications.

7 Acknowledgements

We thank Ted Adelson, Owen Carmichael, and John Haddon for helpful discussions.

References

[1] Altamira Genuine Fractals 2.0, 2000. www.altamira.com.

[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] F. Fekri, R. M. Mersereau, and R. W. Schafer. A generalized interpolative VQ method for jointly optimal quantization and interpolation of images. In *Proc. ICASSP*, volume 5, pages 2657–2660, 1998.

[4] D. J. Field. What is the goal of sensory coding. *Neural Computation*, 6:559–601, 1994.

[5] 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/.

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

[7] 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.

[8] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *ACM SIGGRAPH*, 2001. In *Computer Graphics* Proceedings, Annual Conference Series.

[9] R. Keys. Bicubic interpolation. *IEEE Trans. Acoust. Speech, Signal Processing*, 29:1153–1160, 1981.

[10] S. A. Nene and S. K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Pattern Analysis and Machine Intelligence*, 19(9):989–1003, September 1997.

[11] 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.

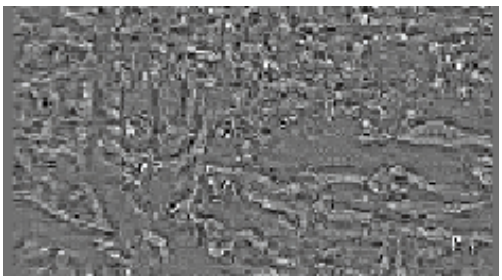
[12] W. F. Schreiber. *Fundamentals of electronic imaging systems*. Springer-Verlag, 1986.

[13] 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.

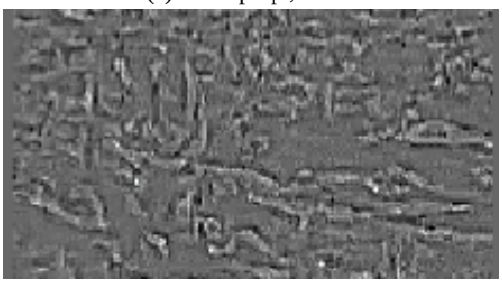
[14] H. Sidenbladh and M. J. Black. Learning image statistics for Bayesian tracking. In *Intl. Conf. on Computer Vision*, 2001.

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

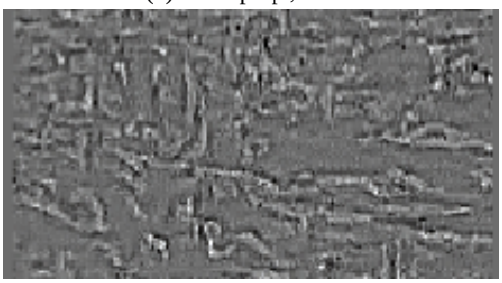
[16] S. Thurnhofer and S. Mitra. Edge-enhanced image zooming. *Optical Engineering*, 35(7):1862–1870, July 1996.



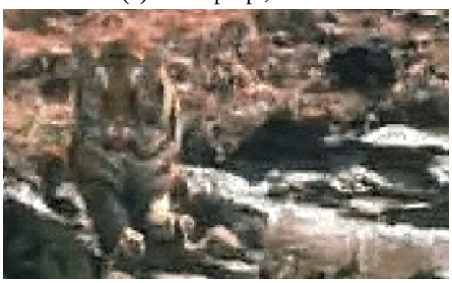
(a) belief prop., iter. 0



(b) belief prop., iter. 1



(c) belief prop., iter. 3



(d) belief propagation

Figure 7: Belief propagation solution to Markov network for super-resolution. (a), (b), and (c) are the estimated high frequencies after 0, 1, and 3 iterations of belief propagation. (d) is the estimated full-resolution image. (The inverse of the contrast normalization used in Fig. 4 (d) was applied to Fig. 7 (c). The result was added to Fig. 4 (b) to obtain Fig. 7 (d)). The training set for this image was two categories of the Corel database, including other tigers, but not this image [6].

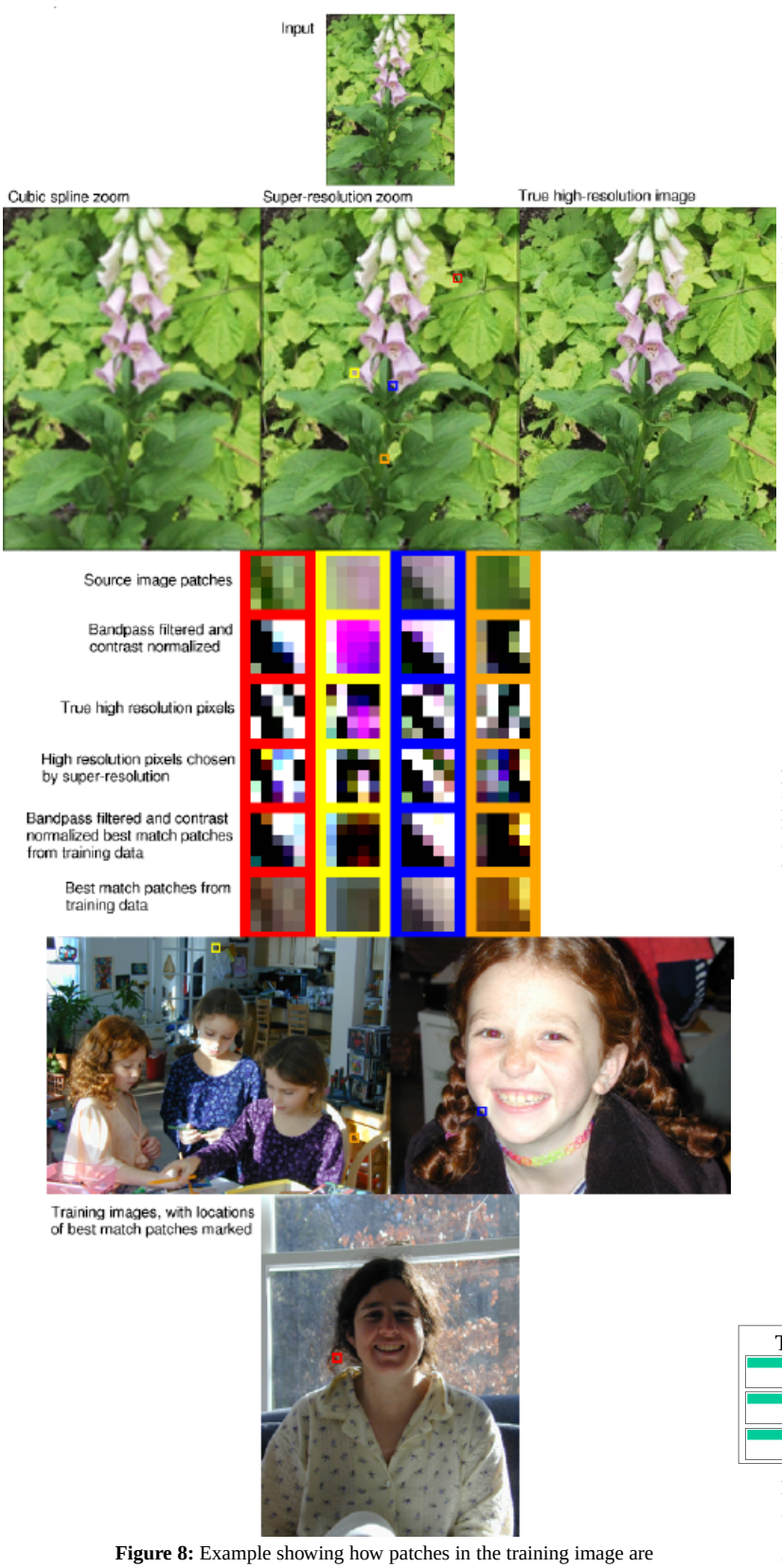


Figure 8: Example showing how patches in the training image are used to create detail in the test image; see text.

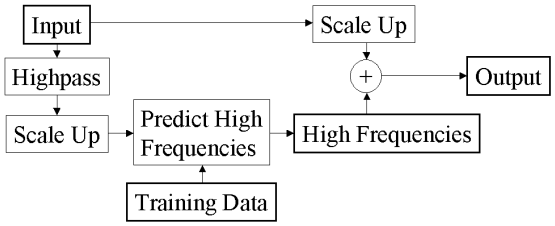


Figure 9: High level view of run time processing in our super-resolution algorithm.

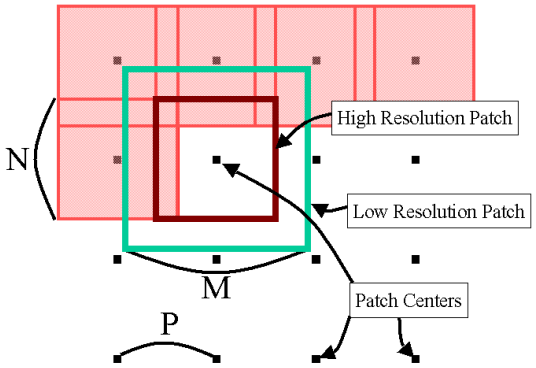


Figure 10: Patch centers, low-resolution and high-resolution patches, and high-resolution patch overlap. The shaded high-resolution patches have already been processed, and the shaded area overlapped by the current (unshaded) patch are used to enforce spatial consistency in the high-resolution details.

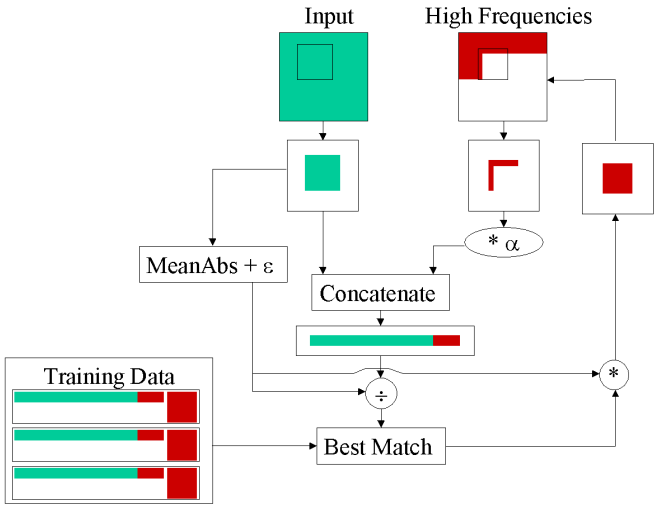


Figure 11: Block diagram showing raster-order per-patch processing. At each step, local low- and high-frequency details (shown in green and red, respectively) are used to search the training set for a new high-frequency patch, which is added to the high-frequency image.

Figure 12: Training images used for the examples of this paper, unless otherwise stated. Patches were sampled at 1 pixel offsets over each of these images and over their synthetically generated low-resolution counterparts (after pre-processing steps). These six 200x200 images yielded a training set of slightly over 200,000 high and low resolution image patch pairs.

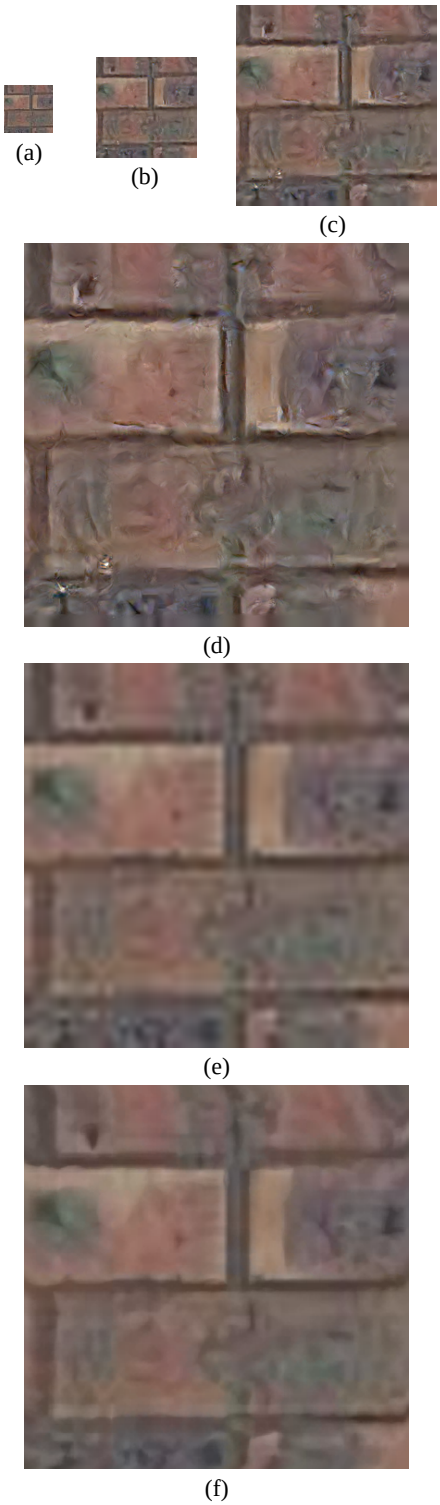


Figure 13: Super-resolution can be used for zooming into texture-mapped surfaces. (a) original texture in 52x52 pixel bitmap. super-resolution zoomed by 2 (b), by 4 (c) and by 8 (d) times in each dimension. Same level of zooming, using (e) cubic spline interpolation, and (f) Altamira Genuine Fractals proprietary software. Our super-resolution algorithm invents sharp, plausible image details, using examples previously observed in the training database.

(a)

(b)

(c)

Figure 14: (a) Original image. (b) cubic spline interplation (c) Super-resolution interpolation

(a)

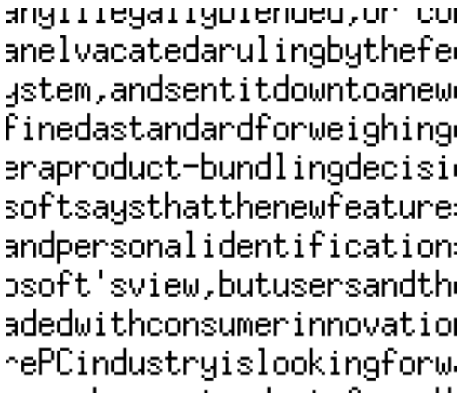
(b)

(c)

(d)

(e)

Figure 15: Failure example. (a) original image. (b) and (d): cubic spline interpolation by factor of 4 in each dimension. Note JPEG compression artifacts made visible. (c) and (e): one pass super-resolution interpolation. Without high-level information, the algorithm treats the JPEG noise as signal, and amplifies it.



(a)

(b)

(c)

Figure 16: Super-resolution example using pathological training set composed entirely of text in one font; (a) is an example image from the training set. (b) zoomed image, and (c) close-up. Note that the algorithm does as best as it can to invent plausible detail for this image, forming contours by concatenated letters.