

Image Magnification Using Level-Set Reconstruction

Bryan S. Morse and Duane Schwartzwald
Department of Computer Science, Brigham Young University
3361 TMCB, Provo, UT 84602
morse@cs.byu.edu

Abstract

Image magnification is a common problem in imaging applications, requiring interpolation to “read between the pixels”. Although many magnification/interpolation algorithms have been proposed in the literature, all methods must suffer to some degree the effects of imperfect reconstruction—false high-frequency content introduced by the underlying original sampling. Most often, these effects manifest themselves as jagged contours in the image. This paper presents a method for constrained smoothing of such artifacts that attempts to produce smooth reconstructions of the image’s level curves while still maintaining image fidelity. This is similar to other iterative reconstruction algorithms and to Bayesian restoration techniques, but instead of assuming a smoothness prior for the underlying intensity function it assumes smoothness of the level curves. Results show that this technique can produce images whose error properties are equivalent to the initial approximation (interpolation) used while their contour smoothness is both visually and quantitatively improved.

1. Introduction

Millions of digital images available today through the Internet and other sources are frequently downloaded and integrated into various types of media. While modern printers and displays support fine detail, images available electronically are rarely of such high resolution. This is especially true for home computing, where limited bandwidth often makes distribution of high-resolution images impractical. Typical screen-resolution images are 72 to 100 dots per inch (dpi) while even low-cost printers are 1200 dpi or higher. Although multiresolution formats have been developed for distributing images at both screen-displayable and printable resolutions, use of such formats is still relatively uncommon and requires that the high-resolution data be available. Thus, many image users are left today with low-resolution images displayed or printed on high-resolution devices—still looking like low-resolution images.

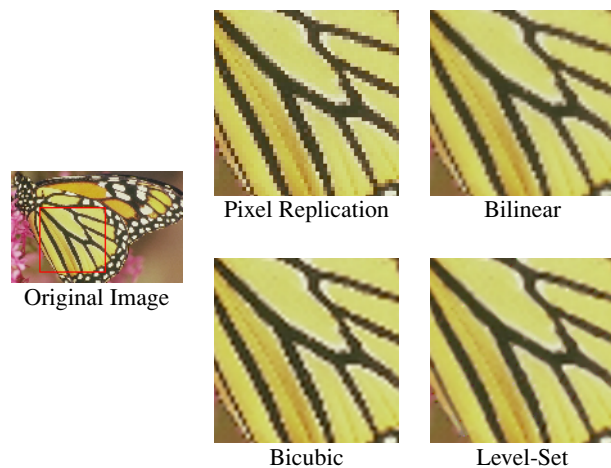


Figure 1. “Monarch” image with 3x magnification. Compare the results of pixel replication (sharpest edges, worst jaggies), bilinear interpolation (blurred edges, less jaggies), bicubic interpolation (sharper edges, worse jaggies), and level-set magnification (sharp as bicubic interpolation, smoother contours).

Standard interpolation methods treat the problem primarily as either fitting a function or filtering (or both) [9, 28]. In either case, the reconstruction is imperfect, and false high-frequency components are introduced into the interpolated image [7, 18]. As a result, they still show artifacts of the original discretization, as indeed all magnification methods must to some degree. These artifacts demonstrate themselves most commonly as alignment to the original pixels (Figure 1).¹ This causes what should be smooth contours in the image to be jagged—the well known “jaggies”.

Instead of approaching interpolation as “fitting the function”, this paper approaches it as “fitting the visual geometry”: reconstructing the geometry of the original image’s

¹Color versions of this and other images in this paper may be found in the conference proceedings CD-ROM and on the author’s web site.

level curves (spatial curves of constant intensity). Using a differential equation, this reconstruction smooths the image contours from an initial approximation *while maintaining fidelity to the original lower-resolution image*. Thus, it directly attacks one of the the most perceptible artifacts of image reconstruction and causes the reconstructed image to preserve smooth contours in the original.

2. Image Interpolation

Because interpolation attempts to approximate an intensity function or surface from sampled data, most reconstruction methods have their roots in either fitting functions to sampled data or in sampling theory.

Functional interpolation treats an intensity surface as a sampled two-dimensional function and attempts to fit this function to the samples, often using polynomials of various degree (linear, cubic, etc.) [23]. For two-dimensional images, these become bilinear and bicubic functions [8]: interpolation in x then interpolation in y or vice versa. More sophisticated interpolating functions and non-exact fitting of the data can be used (e.g., [16]), but the idea is the same—best-fit functions. While these methods can do a good job of approximating the image’s intensity surface, the metric for evaluation is typically error in intensity, not visual appeal.

Filtering approaches often outperform function-fitting approaches by recognizing the frequency-domain effects of the original sampling [9, 28]. These methods attempt to undo the spectrum replication caused by sampling by approximating the effects of a (physically unrealizable) ideal low-pass filter [6, 15, 17, 18, 21, 22]. Convolution by approximations to a sinc function provide reasonable approximations to such an ideal filter, but again, the objective is to minimize pass-through of the offending frequencies rather than considering visual properties of the resulting image.

Edge-directed interpolation algorithms [1] fit smooth subpixel edges to the image and use these to prevent cross-edge interpolation. These methods create sharper edges, and by fitting the edge contours *spatially* rather than *functionally*, they also produce smoother edges. While these methods in part address the effects of interpolation on visual contours, they raise two questions: how do you define the edges of interest, and what do you do elsewhere? Rather than trying to extract specific image curves for smooth reconstruction, the method presented here works by reconstructing smooth approximations of *all* of the image level-set contours simultaneously.

PDE-based approaches to level-set interpolation have been used previously in the literature [4, 5, 14, 19]. Caselles, *et al.* [5] showed that the only operators satisfying certain requirements, particularly closure under interpolation, are those involving second derivatives in the gradient and tangent directions. One variation of this method

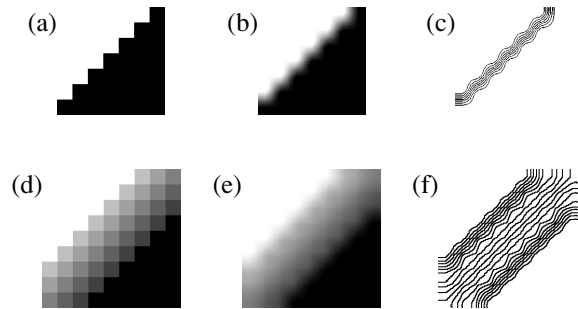


Figure 2. Isophote reconstruction errors introduced by interpolation. When a black-and-white edge (a, magnified) is bicubically interpolated (b), it shows artifacts of the original sampling. Individual level curves (c, eight levels) are jagged instead of smooth. When a similar edge with more gradual transition (d, magnified) is interpolated (e), the isophotes (f, 20 levels) are also jagged.

(minimizing curvature in the gradient direction) can be used to smoothly interpolate missing contours between known curves or points. However, we want to interpolate the contours themselves, not between them. Their second variation (minimizing curvature in the gradient-tangent direction, which we use here) has been used for disocclusion [14] or inpainting [4] with remarkably successful results. However, the solution for this equation is not unique [5], requiring an additional constraint that minimizes total variation. These approaches are also developed only for connected regions, not filling in contours from isolated points.

3. Level-Curve Interpolation

One of the most visually significant geometric properties of images is their *level curves* or *isophotes* (curves of constant intensity). These curves are what give images their perceptual contours. Although level curves don’t capture all geometric information that one might want in analyzing image content [11], reconstruction of the isophotes produces a visually convincing reconstruction of the image.

Figure 2 shows an example of the effects of interpolation on isophotes. If a simple black-and-white edge (2a) is interpolated bicubically (2b), the result shows the underlying pixel grid. The effects of this interpolation on the isophotes can be seen by examining individual level curves in the reconstruction (2c).

Performing the same operations on a blurred edge (intermediate greylevels in the transition) shows similar results. Even though each original level curve (2d) is straight, the level curves of the resulting reconstruction (2e) are not (2f).

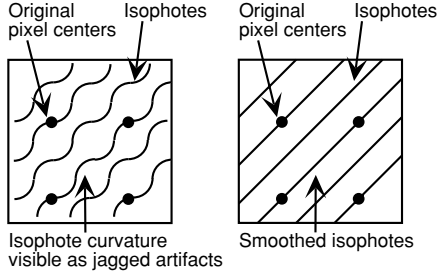


Figure 3. Constrained level set smoothing. Smoothing the level curves as much as possible while still maintaining level curve topology and the values at the known pixels produces a convincing image reconstruction.

These examples suggest an alternative approach to image reconstruction: smooth fitting of level curves based on the original image constraints (Figure 3). This can be phrased as a reconstruction/optimization problem: find the set of level curves that

- Preserve level set topology,
- Preserve intensities at known positions (pixels), and
- Are each as smooth as possible.

Notice that the problem, when stated in this way, follows the general form of a constrained optimization problem and uses the common “smoothness” prior [3]. However, it is not intensity-surface smoothness but *level-set contour smoothness* that serves as the optimization prior.

Although there is no closed-form solution for these constraints (indeed, no unique solution exists without additional constraints), this can be approached iteratively in a fashion similar to gradient-descent minimization:

1. Begin with an approximation of the magnification by using existing interpolation algorithms, and
2. Iteratively minimize isophote curvature while preserving fidelity to the original lower-resolution image.

We may use any existing interpolation method as an initial approximation, but as with all optimization methods, the better the initial approximation, the better the result. In particular, level set reconstruction smooths contours while preserving edge sharpness, so it is only as sharp as the original approximation used. We now turn to an explanation of this second part: the iterative minimization of isophote curvature constrained by the sampled image.

4. Level Set Manipulation

At first glance, manipulation of the level curves requires explicitly finding and fitting each curve, much as individual edges must be found and fitted in edge-directed approaches [1]. However, such explicit curve-fitting is not

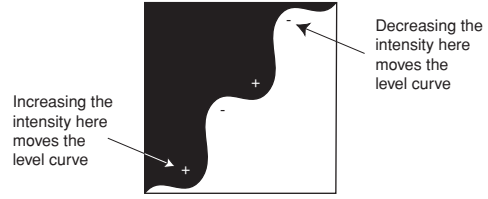


Figure 4. Changes in pixel intensities move level curves spatially according to (2).

necessary. Instead, we can directly manipulate the level curve passing through each pixel respectively by manipulating the intensities at that pixel.

Osher and Sethian [20, 25] have demonstrated *curve evolution* techniques for manipulating 1-dimensional curves in 2-dimensional domains by embedding the curve as a level curve of a function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ of two variables. Altering this function alters its level curves and thus alters the specific level curve that represents the curve of interest. The relationship between changing the value of the function ϕ and moving the curve in the direction of its normal is

$$\phi_t = F \|\nabla \phi\| \quad (1)$$

where F is the speed of movement of the curve in its normal direction and ϕ_t is the change in the embedding function ϕ with respect to time.

Research by Alvarez, Lions, and Morel [2], Sethian, Malladi, and Kimmel [11, 12, 13, 25], and others has shown that one can extend this to describe how changes in individual pixel intensities alter their local level curves. The image itself (I) assumes the role of the embedding function ϕ :

$$I_t = F \|\nabla I\| \quad (2)$$

By using the negative isophote curvature $-\kappa$ as the speed F , level curves contract at a rate proportional to their curvature—places of high curvature contract more quickly than smoother parts of the curve (Figure 4):

$$I_t = -\kappa \|\nabla I\| \quad (3)$$

This can be used to perform edge-preserving smoothing, noise removal and other image enhancement [2, 11, 12, 13, 25], and shape evolution and description [10]. We use a constrained form of this to reduce the artifacts of imperfect reconstruction. Whereas other applications of level-set smoothing attempt to enhance the original image, we are attempting to *reconstruct* the original image (warts and all).

Calculation of the isophote curvature κ similarly does not require explicit representation of the level curve. It can be calculated from local derivatives of the intensity [27]:

$$\kappa = \operatorname{div}(\nabla I / \|\nabla I\|) = -\frac{I_x^2 I_{yy} - 2I_x I_y I_{xy} + I_y^2 I_{xx}}{(I_x^2 + I_y^2)^{3/2}} \quad (4)$$

Substituting this into (3) and recognizing that $\|\nabla I\| = (I_x^2 + I_y^2)^{1/2}$, the desired flow is

$$I_t = \frac{I_x^2 I_{yy} - 2I_x I_y I_{xy} + I_y^2 I_{xx}}{I_x^2 + I_y^2} \quad (5)$$

5. Constrained Smoothing

We implement the differential equation in (5) using a difference equation based on Euler’s method [23]. In this form, the method produces the results expected of level-set smoothing: jagged edges are smoothed and thus the reconstruction artifacts are diminished. However, standard level-set smoothing does more than just smooth jagged edges—it also smooths away features of objects, ultimately shortening level curves until they disappear altogether. The result is an appealing but oversmoothed image (Figure 6a).

The method presented in this paper goes beyond simple level-set smoothing by imposing additional constraints that preserve accuracy to the original image (anchors), preserve level set topology relative to these anchors, identify and smooth jaggies rather than arbitrarily shorten all curves, and to generally avoid oversmoothing. The combined effects of these constraints are illustrated in Figure 6.

5.1. Image Anchors

Curvature flow provides two of our three goals: preservation of isophote topology and isophote length/curvature minimization, but we must add an additional constraint to preserve intensities at the original pixels:

$$I_t = \begin{cases} 0 & \text{for an original pixel location} \\ -\kappa \|\nabla I\| & \text{otherwise} \end{cases} \quad (6)$$

As the level-set contours “flow”, they are constrained by the unchanging anchor pixels (Figure 5). This is similar to constrained curve evolution as presented in [25].

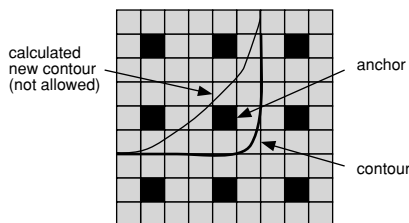


Figure 5. Anchor constraint. Original sampled pixels retain their values during the contour smoothing process. This preserves fidelity to the original image.

5.2. Explicit Topology Constraint

If the level sets are moved too quickly (too large a step size for the numerical implementation of (6)), level curves may move past their associated anchors. While methods for ensuring the stability of level-set smoothing have been shown to preserve topology [26] and [24], such methods require small step sizes that are often too slow for our intended application. One may preserve the level-set topology of the initial reconstruction, especially relative to low-resolution anchors, by introducing an explicit topology constraint² as follows:

1. Calculate the desired next-iteration value for all pixels based on (6).
2. For each pixel that is increasing, limit its value to less than the lowest next-iteration value of the greater-valued neighboring pixels.
3. Similarly, for each pixel that is decreasing, limit its value to more than the largest next-iteration value of the lesser-valued neighboring pixels.

The net effect of this is to ensure that all greater-valued neighbors stay larger and all lesser-valued neighbors stay smaller, thus preserving topology.

With this explicit topology constraint, coupled with a step-size reduction schedule, reasonable results can be calculated for color images in as few as five iterations.

5.3. Inflection Constraint

As pointed out previously in Section 2, there is no unique solution to this optimization [5]. This is because there are constraints only on the smoothness of the curves themselves, *not between the curves*. However, we don’t want to arbitrarily smooth between the curves as in [4, 14] because we want to maintain the sharpness of our original approximation. By the same token, we don’t want to artificially introduce discontinuities either. If allowed to continue constrained only by original-pixel anchors, level-curve shortening flow tends to act like an elastic band shrinking to fit a set of nails hammered into a board: the minimum-length solution is piecewise linear (Figure 6c). Indeed, if there is only a single anchor bounded by a level curve, the curve will shrink to this single point.

It must be remembered then that our goal is to smooth contours, not simply shrink them. To smooth jagged contours while preventing other forms of curve shortening, we introduce a constraint that separates jagged contours from curves without local inflections. This constraint simply requires that if a pixel is increasing or decreasing in value, at

²It must be emphasized that this part of the algorithm is *not* necessary unless one wants to use a more aggressive step size than stability requirements would normally dictate.



a) Unconstrained level-set smoothing



b) Anchor constraints only



c) Anchor and topology constraints



d) Anchor, topology, and inflection constraints

Figure 6. Effects of the constraints described in Sections 5.1–5.3. Unconstrained level-set smoothing oversmooths the image, removing jaggies but losing significant details (a). Anchors help to constrain the flow somewhat, but the curves may move past their anchors, producing results similar to unconstrained smoothing (b). Topological constraints allow the anchor pixels to serve their function, even with an aggressive step size. However, small closed contours that surround a single pixel are still allowed to collapse unconstrained. Likewise, contours between similar-valued pixels are allowed to shorten unconstrained until they produce piecewise-linear segments, giving an almost polygonal look to some of the contours (c). These problems are addressed by the inflection constraint (d).

least one of the neighboring pixels must change in the opposite direction. (See Figure 4.) Thus, jagged contours may be smoothed only by simultaneous pulling “in” on convex parts and “out” on neighboring concave parts of the curve. Already convex curves are not allowed to change.

5.4. Color Images

Level-set reconstruction can be extended to color images by applying the constrained flow to the individual color planes separately. Although this would appear to be susceptible to separation as each plane flows independently, our experience is that the constraints are sufficient to limit the

separation. In particular, the anchor constraints guarantee that any separation is sub-pixel with respect to the original image. Smoothing their level sets independently following independent interpolation appears to produce color separation no worse than the initial interpolation itself.

We have also experimented with a method based on *Beltrami flow* [11], which treats a color image not as three separate 2-dimensional manifolds but as a single 2-dimensional manifold in a 5-dimensional space (two spatial plus three color dimensions). Our results using this method were, however, not appreciably different from those produced by treatment of the individual color planes independently.

6. Results

Results of level-set smoothing can be seen in Figures 7–10. In each case, the image enhanced by constrained level-set smoothing preserves the sharpness of the original (bicubically interpolated) approximation while reducing artifacts.

6.1. Quantitative Comparisons

A standard method for evaluating interpolated images is to measure the error introduced by the interpolation as compared to a “perfect” interpolation. To measure this, we first take a higher-resolution image, reduce it by a factor f , then enlarge it by the same factor f using reconstruction, and compare the reduced-then-magnified image to the original.

Table 1 shows that the mean squared error for smoothed bicubically interpolated images is only modestly better than for bicubic interpolation alone. This is not surprising, though. The central idea of level-set reconstruction is to produce images that look better while staying true to the data. While images smoothed with constrained level-set smoothing are not necessarily “more accurate” than such images without smoothing, *their error is no worse and they are more visually appealing*.

To measure this other objective, producing images with smoother contours, we also measured the mean absolute level-set contour curvature $E\{|\kappa|\}$. This might seem like “begging the question”—measuring the property we’re explicitly trying to minimize—but it is interesting to compare the measurements of this quantity to visual impressions. For example, bilinear reconstruction consistently produces measurably smoother contours than bicubic reconstruction while introducing more error—agreeing with common wisdom that bilinear reconstruction looks “less jagged” but “more blurred”. Bicubic reconstruction, using a higher-order fit to the original points, introduces less error but also produces more jagged contours (Figure 1). In each case, bicubic reconstruction followed by level-set smoothing produces error results comparable to bicubic reconstruction but produces smoother contours. Indeed, the contour-smoothed interpolated images consistently produce contours that are significantly smoother than bicubic or bilinear reconstruction *without introducing additional intensity error*.

For comparison, we also compared our results to a 3×3 box-filtered original image. (One must compare to a low-pass filtered version of the original because of the low-pass filtering required prior to reduced sampling, unless one is trying to sharpen as well as interpolate.) Bicubic interpolation followed by contour smoothing consistently produces results that are as smooth as the low-pass filtered original.

7. Conclusions and Future Work

Level set reconstruction, by focusing on visually significant properties of interpolation artifacts, can significantly improve the results of existing methods for image magnification. The results have error characteristics comparable to the initial interpolation method but with contour smoothness comparable to the image prior to downsampling.

As with any iterative optimization technique, level set reconstruction depends heavily on the initial approximation. We have tested level set reconstruction using various initial interpolation methods, and in all cases it significantly improved the results. The effect of level set reconstruction on even better initial approximations should be explored.

Allebach [1] has noted that interpolation should consider the original image samples as area-based samples, not point samples. We are pursuing variations of anchor constraints that act as area-average, not point, anchors.

Another area of continued exploration is to see whether it is possible to achieve better results for larger magnifications by applying a smaller magnification, performing level set reconstruction, more magnification, more level set reconstruction, etc. We have tried this by comparing 4x magnification followed by level set smoothing to a process consisting of 2x magnification, smoothing, 2x magnification, and smoothing again. The results are marginally improved, but not significantly. However, it does suggest that this might be a better way to perform larger magnifications.

Acknowledgments

This research was supported in part by funding from Adobe Corporation. Our thanks go to Greg Gilley, Martin Newell, Gregg Wilensky, and Peter Ullmann for their helpful comments and support.

References

- [1] J. Allebach and P. W. Wong. Edge-directed interpolation. In *IEEE International Conference on Image Processing*, pages 707–710, 1996.
- [2] L. Alvarez, P. L. Lions, and J. M. Morel. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, 29(3):845–866, 1992.
- [3] H. C. Andrews and B. R. Hunt. *Digital Image Restoration*. Prentice-Hall, 1977.
- [4] M. Bertalmio, G. Sapiro, C. Ballester, and V. Caselles. Image inpainting. In *Computer Graphics Proceedings, Annual Conference Series*. ACM SIGGRAPH, 2000.
- [5] V. Caselles, J.-M. Morel, and C. Sbert. An axiomatic approach to image interpolation. *IEEE Transactions on Image Processing*, 7(3):376–386, 1998.



Figure 7. Comparison of the “monarch” image at 100 dpi (a), magnified to 300 dpi using bicubic interpolation (b), then smoothed using level set reconstruction (c). 100 dpi versions of the magnified images can be seen in Figure 1.

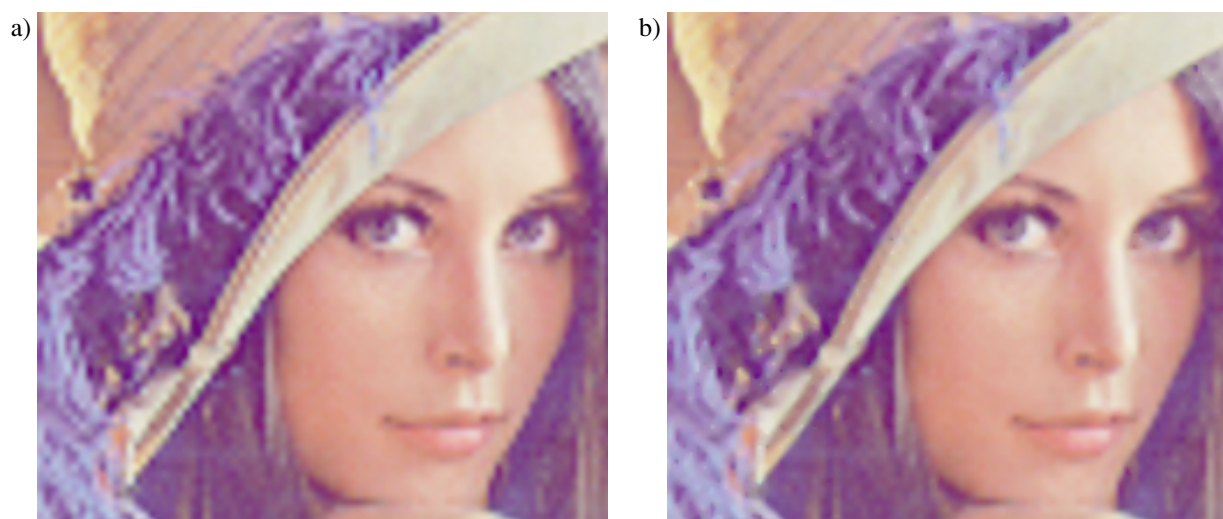


Figure 8. Comparison of the “lena” image magnified 3x using bicubic interpolation (a), then smoothed using level set reconstruction (b).

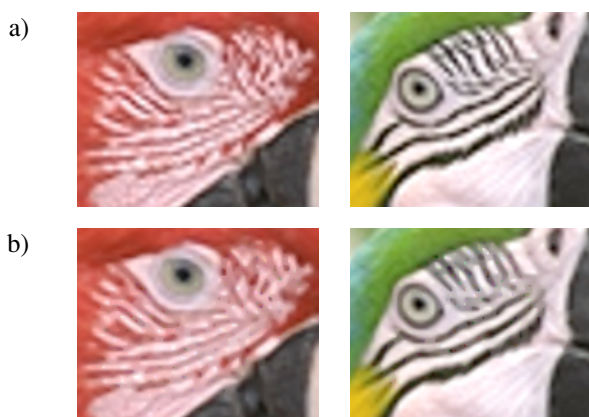


Figure 9. Portions of the “parrots” image magnified 3x using bicubic interpolation (a), then smoothed using level set reconstruction (b).

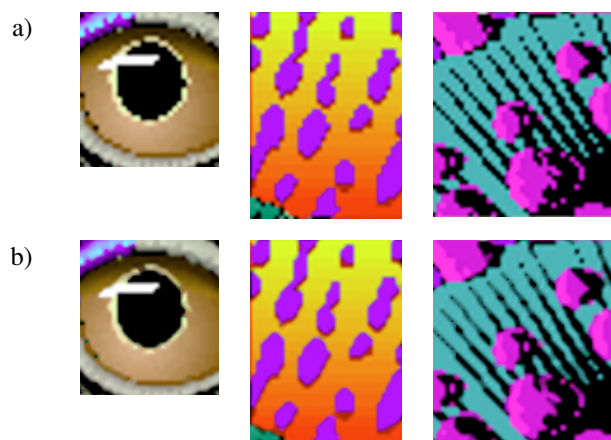


Figure 10. Portions of the “frymire” magnified 3x using bicubic interpolation (a), then smoothed using level set reconstruction (b).

| Image | Reconstruction Method | Mean Squared Error | | | Mean Contour Curvature ($E\{ \kappa \}$) | | |
|-----------|-----------------------|--------------------|---------|---------|--|-------|------|
| | | Red | Green | Blue | Red | Green | Blue |
| "Lena" | Replication | 81.93 | 116.45 | 59.67 | 2.49 | 3.36 | 2.47 |
| | Bilinear | 19.62 | 29.59 | 19.61 | 1.08 | 1.61 | 1.16 |
| | Bicubic | 13.09 | 26.85 | 19.37 | 1.09 | 1.68 | 1.20 |
| | Bicubic + Flow | 12.67 | 25.93 | 18.18 | 0.77 | 1.10 | 0.85 |
| | Original (3x3 Box) | — | — | — | 0.69 | 1.08 | 1.88 |
| "Frymire" | Replication | 1359.14 | 1487.54 | 1410.97 | 9.34 | 10.60 | 9.67 |
| | Bilinear | 573.78 | 671.29 | 611.84 | 6.67 | 7.61 | 6.90 |
| | Bicubic | 668.10 | 797.61 | 713.97 | 7.59 | 8.90 | 7.86 |
| | Bicubic + Flow | 557.87 | 720.43 | 608.26 | 4.64 | 5.58 | 4.95 |
| | Original (3x3 Box) | — | — | — | 5.00 | 5.45 | 5.12 |
| "Monarch" | Replication | 218.81 | 221.18 | 188.91 | 3.80 | 3.78 | 3.73 |
| | Bilinear | 62.32 | 62.89 | 57.98 | 2.09 | 2.16 | 2.09 |
| | Bicubic | 62.06 | 65.85 | 63.02 | 2.31 | 2.41 | 2.30 |
| | Bicubic + Flow | 51.36 | 55.08 | 51.99 | 1.49 | 1.57 | 1.53 |
| | Original (3x3 Box) | — | — | — | 1.07 | 1.23 | 1.31 |

Table 1. Mean squared error and mean contour curvature ($E\{|\kappa|\}$) for different reconstruction methods. In each case, the measurements agree with visual impression. (See Figures 1 and 7–10.)

- [6] R. L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Gr.*, 5(1):51–72, 1986.
- [7] F. C. Crow. The aliasing problem in computer-generated shaded images. *Communications of the ACM*, 20(11):799–805, November 1977.
- [8] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, second edition, 1990.
- [9] A. S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, 1995.
- [10] B. Kimia, A. Tannenbaum, and S. Zucker. Shapes, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1995.
- [11] R. Kimmel, R. Malladi, and N. Sochen. Images as embedded maps and minimal surfaces: Movies, color, and volumetric medical images. In *Computer Vision and Pattern Recognition*, 1997.
- [12] R. Malladi and J. A. Sethian. Image processing: flows under min/max curvature and mean curvature. *Graphical Models and Image Processing*, 58(2):127–141, 1996.
- [13] R. Malladi and J. A. Sethian. A unified approach to noise removal, image enhancement, and shape recovery. *IEEE Transactions on Image Processing*, 5:1554–1568, 1996.
- [14] S. Masnou and J.-M. Morel. Level lines based disocclusion. In *Fifth IEEE International Conference on Image Processing (ICIP'98)*, 1998.
- [15] N. Max. An optimal filter for image reconstruction. In J. Arvo, editor, *Graphics Gems II*, pages 101–104. Academic Press, 1991.
- [16] D. Metaxas and E. Milios. Reconstruction of a color image from nonuniformly distributed sparse and noisy data. *CVGIP: Graphical Models and Image Processing*, 54(2):103–111, Mar. 1992.
- [17] D. P. Mitchell. Generating antialiased images at low sampling densities. *Computer Graphics Proceedings, Annual Conference Series*, 21(4):65–72, 1987.
- [18] D. P. Mitchell and A. N. Netravali. Reconstruction filters in computer graphics. *Computer Graphics Proceedings, Annual Conference Series*, 22(4):221–228, 1988.
- [19] B. S. Morse and D. Schwartzwald. Isophote-based interpolation. In *Fifth IEEE International Conference on Image Processing (ICIP'98)*, 1998.
- [20] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation. *J. Comput. Phys.*, 79:12–49, 1988.
- [21] S. K. Park and R. S. Schowengerdt. Image reconstruction by parametric cubic convolution. *Computer Vision, Graphics, and Image Processing*, 23:258–272, 1983.
- [22] M. J. Pavicic. Convenient anti-aliasing filters that minimize "bumpy" sampling. In A. S. Glassner, editor, *Graphics Gems*, pages 144–146. Academic Press, 1990.
- [23] S. M. Pizer and V. L. Wallace. *To Compute Numerically*. Little, Brown, and Company, 1983.
- [24] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.
- [25] J. A. Sethian. *Level Set Methods*. Cambridge University Press, 1996.
- [26] B. M. ter Haar Romeny, editor. *Geometry-Driven Diffusion in Computer Vision*. Kluwer Academic Publishers, 1994.
- [27] B. M. ter Haar Romeny and L. Florack. A multiscale geometric model of human vision. In B. Hendee and P. N. T. Wells, editors, *Perception of Visual Information*. Springer-Verlag, Berlin, 1991.
- [28] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.