

Enhancement and Recovery in Atomic Force Microscopy Images

Alex Chen, Andrea L. Bertozzi, Paul D. Ashby, Pascal Getreuer, Yifei Lou

¹ A. Chen UCLA, Department of Mathematics, Los Angeles, CA, achen81@stanfordalumni.org

² A. L. Bertozzi UCLA, Department of Mathematics, Los Angeles, CA, bertozzi@math.ucla.edu

³ P. D. Ashby Lawrence Berkeley National Laboratory, Molecular Foundry, Berkeley, CA, pdashby@lbl.gov

⁴ P. Getreuer CMLA, ENS Cachan, France, getreuer@gmail.com

⁵ Y. Lou Georgia Institute of Technology, School of Electrical and Computer Engineering, Atlanta, GA, louyifei@gmail.com

Summary. Atomic force microscopy (AFM) images have become increasingly useful in the study of biological, chemical and physical processes at the atomic level. The acquisition of AFM images takes more time than the acquisition of most optical images, so that the avoidance of unnecessary scanning becomes important. Details that are unclear from a scan may be enhanced using various image processing techniques. This chapter reviews various interpolation and inpainting methods and considers them in the specific application of AFM images. Lower-resolution AFM data is simulated by subsampling the number of scan lines in an image, and reconstruction methods are used to recreate an image on the original domain.

The methods considered are classified in the categories of linear interpolation, nonlinear interpolation, and inpainting. These techniques are evaluated based on qualitative and quantitative measures, showing the extent to which scans times can be reduced while preserving the essence of the original features. A further application is in the removal of streaks, which can occur due to scanning errors and post-processing corrections. Identified streaks are removed, and the resulting unknown region is filled using inpainting.

1.1 Introduction

The atomic force microscope (AFM) is an extremely high magnification microscope [5]. It achieves its high resolution by moving an atomically sharp probe over surfaces and recording the highly localized interaction force. Isolating specific interaction forces such as electrostatic, magnetic, specific chemical interactions, van der Waals attraction, and Pauli repulsion enable the AFM to measure many surface properties in addition to topography [20, 12, 22, 2, 29, 17]. The AFM is also able to measure surfaces in any environment from liquids, to corrosive gases and vacuum. The high resolution, versatility, and broad information content make AFM a frequent choice for nanoscience imaging.

The current standard method of AFM data collection is the raster scan. The probe starts by traveling along the “fast scan direction,” or in the $+x$ direction. As it reaches the end of the scan region, it takes a small step in the “slow scan direction,” or $+y$ direction, and scans in the $-x$ direction until it retraces the x displacement. Another small step in the $+y$ direction is taken and the scanner moves in the $+x$ direction to initiate another scan line. Continuing in this manner, an image is formed. The backward (retrace) scans are often displayed independently from the forward (trace) scans due to errors in position from scanner nonlinearities and hysteresis. A feedback mechanism maintains the probe-sample interaction at constant force to ensure that the probe is not damaged by contact with the sample.

Because the AFM is a local probe it must collect data serially to construct an image over time which can be a significant disadvantage. The sample and probe are massive objects that are difficult to accelerate requiring relatively slow scan velocities otherwise the feedback mechanism that holds the interaction force constant may not be able to compensate quickly enough, causing erroneous readings or damage to the probe. This problem is even

more pronounced when the sample has sharp gradients. Another problem is thermal drift, the tendency for the probe and sample to move relative to each other due to temperature variations in the probe, sample, and substrate [19]. Since an image is formed point-by-point, the topography data may be skewed or distorted. As a result, it is challenging for AFM to record dynamic processes.

Figure 1.1 shows images of a chemical reaction that occurs faster than the AFM scan time. Shorter scan times are required to better capture sample dynamics. Before oxidation, the surface is atomically flat with a few step edges where the sample changes height by one atomic layer. After oxidation for the same region the material at the step edges has been reacted leading to roughening, erosion, and migration. Imaging the surface during oxidation does not sufficiently resolve the oxidation edges, since data is collected at varying times.

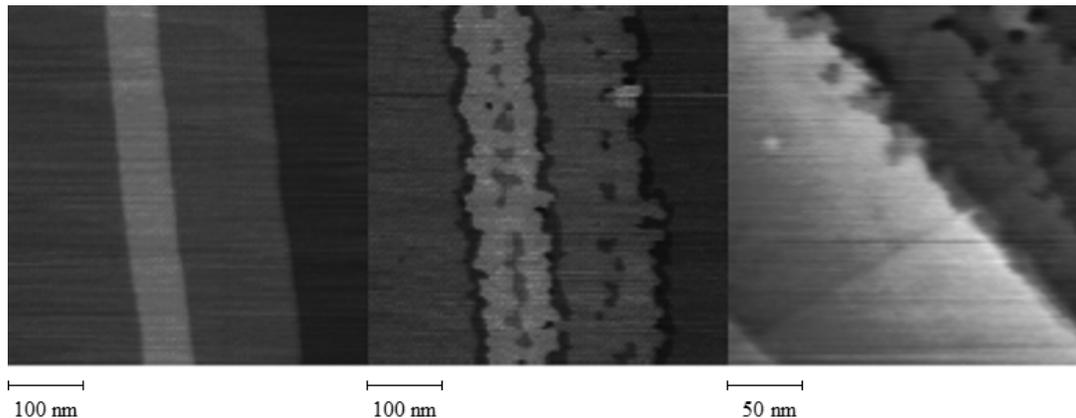


Fig. 1.1. AFM scans showing morphological changes to a potassium bromide surface when oxidized by ozone. Left: Before oxidation. Middle: After oxidation. Right: Image of another surface collected during oxidation in an attempt to observe the chemical reaction. However, the reaction happens during a single image. The time and location of specific oxidation events is unavailable.

An active area of research to improve the temporal resolution of AFM includes building lighter and stronger scanners that can operate at higher frequencies while maintaining a safe probe-sample interaction force. The best-performing instruments can record images at video rate [1, 18]. However, this approach often has significant sample size and environmental limitations compromising the versatility of AFM. Methods that increase the instrument's temporal resolution while maintaining versatility are needed.

Recording fewer scan lines per image can make AFM image collection faster. Alternatively, tracking only the boundaries of important features can drastically reduce image times. The important question regarding these methods is whether such scans still resolve the areas of interest sufficiently.

There are two classes of image processing techniques, interpolation and inpainting, used to fill in missing data. Though there is significant overlap in the methods and approaches, we generally take interpolation to denote methods based on local averaging ideas and let inpainting refer to methods that detect important image features in a known region and seek to continue these into the unknown region.

One of the major applications of interpolation is to increase the resolution of an image by interpolating intermediate values between known data points. Interpolation is thus well-suited to the problem of converting coarse raster scans to higher resolution images.

This chapter considers various image reconstruction methods and the degree to which AFM images can be enhanced. The focus is on using interpolation and inpainting techniques on AFM images that have been obtained with fewer scan lines (subsampling along the slow scan direction). This is in contrast to the typical applications of interpolation, in which both axes are usually subsampled by the same factor, and inpainting, in which one typically has large connected known regions and unknown regions.

The rest of the chapter is organized as follows. Section 1.2 introduces the reconstruction problem and relevant terminology. Sections 1.3 and 1.4 review some commonly used interpolation algorithms and assesses their strengths and weaknesses. Inpainting techniques, which look at the reconstruction from another perspective and are more readily generalizable, are addressed in section 1.5. Several reconstructions on AFM images are presented in each of these sections.

1.2 Description of the Reconstruction Problem

The scenes underlying images are often taken to lie in continuous space. When images of these scenes are captured, they are sampled at a certain rate and thus mapped to discrete space. This sampling rate is directly related to the image resolution. If the sampling rate is increased (upsampling), the image resolution is increased. Similarly, downsampling decreases image resolution. Interpolation can thus be reinterpreted as the inverse problem of recreating a higher resolution version of a given scene.

Let the given discrete image be denoted as $\bar{I}_{m,n}$ for $(m,n) \in \mathbf{Z}^2 \cap \Omega$. The problem is to reconstruct an image $I(x,y)$ with $(x,y) \in \Omega$. Such an image $I(x,y)$ should be close to \bar{I} at the points $(m,n) \in \mathbf{Z}^2 \cap \Omega$ and continue the structural features present in \bar{I} .

1.2.1 Adapting Reconstruction Methods to AFM

The fastest AFM scans that still distinguish image features are the most preferable. Image reconstruction methods can then be used to obtain an approximation of a higher resolution version. There are, however, further considerations in adapting these methods to AFM problems.

In general, the maximum sampling rate in space for data points is proportionally related to the speed of the tip, while the tip speed is limited by considerations such as feature height and scanning pattern. For a fixed tip speed, there is no benefit to sampling at a rate lower than the maximum rate since the tip still must traverse the same area [8]. This consideration fixes the number of points in each scan line for a specific velocity. However, decreasing the number of scan lines can bring practical benefits, as long as the important features are still being detected.

In terms of the reconstruction problem, this means that interpolating by different factors in the two dimensions of the image is of great interest. The usual treatment of the interpolation problem, however, is that both dimensions are scaled by the same factor. Most interpolation algorithms are valid when the scaling in the two dimensions is not equal, but in the AFM reconstruction problem, care must be taken to ensure that reconstructions consider features, such as interrupted edges, correctly.

A further distinction between the AFM application and many other applications is in image acquisition. For each pixel, optical systems use for example a charge-coupled device to measure incoming light over a small photoactive region. Since lenses and filters also introduce blur, each sample represents a weighted average over a small area in continuous space. In acquiring an AFM image, however, the tip obtains a sample by visiting a point instead of averaging over a region. While this sample still represents a convolution between the tip and the surface, it is concentrated over a smaller area. In other words, optical images aggregate over the entire domain while AFM images capture concentrated point samples.

This subtle difference in image acquisition can result in the loss of image features from subsampling. Typically, a lower resolution image can be obtained from a higher resolution version by some averaging of the original data [21]. Such averaging results in the greatest retention of information. This method is also analogous to the taking of a lower resolution camera image because the amount of light from a neighborhood of pixels is averaged. There is, however, significant information loss when the lower resolution image is formed from simply discarding certain lines of data, as in AFM subsampling. Fig. 1.2 shows the contrast between an image of annealed gold and the same image obtained by subsampling lines. In the latter image, the intensities along some edges vary due to the loss of information around the edge pixels, resulting in pixelation.

This method of sampling also makes AFM imaging more susceptible to *aliasing*. Aliasing is an effect where an oscillating pattern appears to change frequency after sampling. It occurs when sampling a pattern that is finer than what is representable with the image resolution, and manifests in the sampled data as artificial oscillations

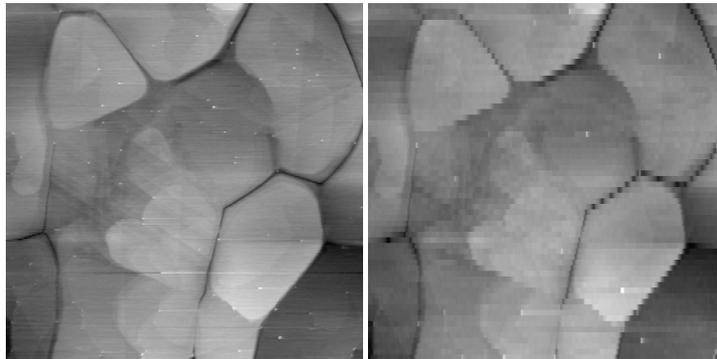


Fig. 1.2. Subsampling the “Annealed Gold” image in the slow scan (vertical) direction. Left: Original image. Right: Subsampling by a factor of 4 along the vertical axis. Nearest neighbor interpolation to preserve the aspect ratio.

and Moiré patterns. The blurring in optical systems cancels out (“anti-aliases”) most high frequency oscillations so that aliasing is limited. AFM imaging does not have as much blurring, so aliasing is a problem.

1.2.2 Desirable Traits of Reconstruction Methods

Image reconstruction methods are typically evaluated based on several criteria. Generally, preserving edge sharpness allows objects to be distinguished clearly from each other and from the background. At the same time, it is important to filter noise in order to remove random features that may obscure the image and make it difficult to evaluate. Unfortunately, the two goals are often antithetical since both edges and noise are typically defined by high gradients. In contrast to the randomness of noise, however, edges often can be identified as continuous contours traversing high gradient regions. A good reconstruction algorithm keeps edges sharp while smoothing noise.

Another consideration is the “connectivity principle,” [7] which states that edge curves should be connected through the unknown region whenever possible. This principle is based on human perception and experience, as well as the particular prevalence of long, thin objects in nature and in man-made applications. Particular examples include road inpainting [3] and the identification of bar codes [11, 10]. Reconstruction algorithms have historically had particular problems connecting such slim objects through an unknown region. Inpainting methods relying on the evolution of a fourth-order PDE [25] are more likely to satisfy the connectivity principle due to their penalization of high curvature edges.

A related problem is the shape of such an edge connection. Often, edges are connected by the shortest path [6], which can result in unrealistic kinks. The same penalization on high curvature edges discussed for the connectivity principle also fixes the over-reliance on shortest path connections. These considerations are well-studied for the typical inpainting problem, in which there are large unknown regions interrupting mostly known data.

The connectivity principle and staircasing pose particular challenges in the AFM inpainting problem. Since the known data is relatively more disconnected than in usual inpainting applications, it is unlikely to expect edge connections with the same degree of effectiveness. Indeed, experiments in section 1.7 show that this is the case.

1.3 Linear Interpolation

Linear interpolation methods average values of nearby pixels to calculate values at intermediate points. Mathematically, the reconstructed image is the result of the input image convolved with a given kernel. Linear interpolation

methods are linear in the sense that the relative weights on neighboring points in the average do not depend on their respective intensities, i.e. the convolution kernel is not dependent on the image intensity values. Thus, these methods are fast and easy to calculate. At the same time, since they do not take any edge information into account, there is always a trade-off between artificial ripples and staircasing along diagonal edges.

1.3.1 Nearest Neighbor

Nearest neighbor interpolation is one of the simplest methods of interpolation. In this method, the value of the nearest known point is copied directly without regard to any other point. That is, if $(x, y) \in \Omega$, then $I(x, y) = I(m, n)$ where $(m, n) = \operatorname{argmin}_{(i, j) \in \mathbf{Z}^2 \cap \Omega} \operatorname{dist}((i, j), (x, y))$.

Since nearest neighbor interpolation copies data points $\bar{I}_{m, n}$ without alteration, any noise will also be copied. Similarly, edges are thickened, giving the image a blocky, pixelated appearance. Nearest neighbor interpolation is also useful in comparing images on varying domain sizes without altering the underlying quality of the image.

1.3.2 Bilinear and Bicubic Interpolation

Taking averages is one of the primary methods used to eliminate noise, one of the primary problems with nearest neighbor interpolation. Polynomial interpolation methods fill in intermediate points by taking a weighted average, with the weighting depending on their distances to nearby points. Bilinear and bicubic interpolation are examples of these methods.

Bilinear interpolation is a combination of two linear interpolation steps along the x-coordinate, then along the y-coordinate. First, linear interpolation is performed along the x-coordinate for each fixed y-coordinate, followed by linear interpolation along the y-coordinate.

If $x_1 = \lfloor x \rfloor$, $y_1 = \lfloor y \rfloor$, $c = x - x_1$, $d = y - y_1$, where $\lfloor \cdot \rfloor$ denotes the floor function, then bilinear interpolation is given by

$$I(x, y) = (1 - c)(1 - d)I_{x_1, y_1} + (1 - c)dI_{x_1, y_1 + 1} + c(1 - d)I_{x_1 + 1, y_1} + cdI_{x_1 + 1, y_1 + 1}.$$

An undesirable property of nearest neighbor interpolation is that the result is artificially discontinuous between pixels. Bilinear and other higher order polynomial methods construct the interpolant from continuous piecewise polynomials, so the result is always continuous. This has an effect of smoothing the image, which improves interpolation of smooth regions and directional features.

Bicubic interpolation is similarly a two-dimensional version of cubic interpolation. The values in each square $(x, y) \in [m, m + 1] \times [n, n + 1]$, are approximated by a polynomial that has at most cubic terms in both x and y .

The resulting polynomial $I(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i, j} x^i y^j$ can be calculated by using the values \bar{I} , $\frac{d\bar{I}}{dx}$, $\frac{d\bar{I}}{dy}$, and $\frac{d^2\bar{I}}{dx dy}$ at the corners, with the derivatives being calculated numerically. Bicubic interpolation results in an even smoother reconstruction than bilinear interpolation. Since bicubic interpolation is also computationally efficient, it is often used for resizing applications [26].

1.3.3 Lanczos

Interpolation with a sinc kernel, also known as Whittaker–Shannon interpolation, has the remarkable property that the interpolation is exact when the underlying signal is bandlimited. Applied to image interpolation, this results in an extremely smooth image. Sinc interpolation also avoids the staircasing that can occur at diagonal edges. Unfortunately, since edges are the result of sharp changes, application of the sinc filter across them results in significant ripple artifacts as the edges are fitted to lower frequencies. The Lanczos filter is a windowed version of the sinc filter. The windowing of the sinc function allows for the higher frequency changes characteristic of edges. It thus provides a compromise between staircasing and ripple effects.

1.4 Nonlinear Interpolation

Nonlinear interpolation algorithms attempt to solve the problems of staircasing and rippling by taking an adaptive approach. Any image of practical interest has some structure, which can be used in the reconstruction. Thus, instead of an unbiased average of intensity values, averaging is based on the detection of edges. These methods generally attempt to average along edges to preserve edge sharpness. As with linear interpolation, noise is smoothed since these points are not identified as edge points.

1.4.1 Contour Stencils

Interpolation by contour stencils was introduced [14, 13] as a nonlinear method to detect the orientation of edges. Edges are detected in the input image by comparing each image patch to each element in a set of “contour stencils,” which predict the location and direction of edges. The stencil which provides the best match to the image patch is selected. The reconstruction step then follows by interpolating according to the selected stencil.

The predicted orientation of edges follows from measuring the total variation along a curve,

$$\|u\|_{TV(C)} = \int_0^T \left| \frac{\partial}{\partial t} u(\gamma(t)) \right| dt, \gamma: [0, T] \rightarrow C,$$

so that a small value for $\|u\|_{TV(C)}$ suggests that an edge lies along C .

In order to make the problem computationally efficient, a subset of possible image contours is used, the set Σ of “contour stencils.” An example set of contour stencils is shown in Fig. 1.3. This set of contour stencils can distinguish between eight different orientations for edges.

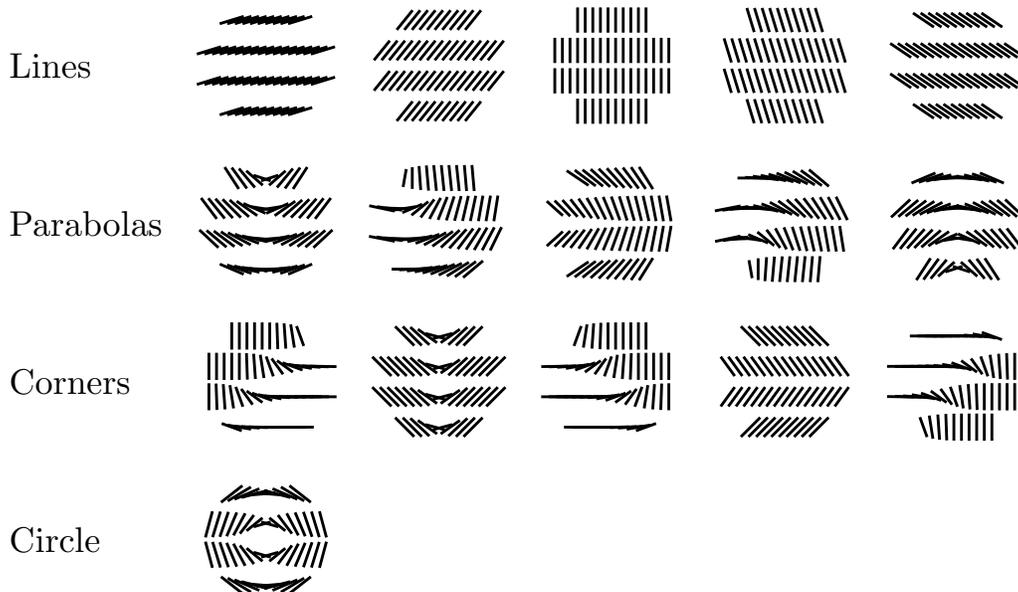


Fig. 1.3. Several contour stencils for a rectangular grid with pixel aspect ratio 4:1. The lines depict the orientation measured over each cell of the neighborhood. The stencil set comprises lines at 32 orientations, 16 parabolas, 8 corners, and a circle.

Once the contour orientations at every pixel have been estimated, the interpolant is constructed as a linear combination of oriented Gaussians. In this way, the interpolated image is encouraged to have the same contour orientations as those detected in the input image.

1.4.2 Prioritized Line Interpolation

This section introduces Prioritized Line Interpolation (PLI), which is designed to connect edges that have been broken by the subsampling process. The idea is to assume that edges are locally linear in space and are locally nearly constant. Then, starting with the highest gradient points, which are more likely to be edge points, the algorithm searches a neighborhood for possible edge connections. Once a suitable location is found, unknown points along the connection are filled in by linear interpolation.

The PLI algorithm is as follows:

1. Each point in the interior of the image is placed in a priority queue $\{A_i\}_{i=1}^N$, where N is the number of interior pixels, in decreasing order of a function based on the discrete gradient; that is, the function

$$f(u_i, j) = |u_{i+1, j} - u_{i, j}| + |u_{i-1, j} - u_{i, j}| + |u_{i, j+1} - u_{i, j}| + |u_{i, j-1} - u_{i, j}|.$$

2. For a neighborhood N_i of the point A_i (a square neighborhood centered at A_i with radius r , for simplicity), the algorithm considers any known points $\{B_{i_j}\}$ with gray value within a certain threshold t_1 .
3. For each B_{i_j} , the unknown points lying between B_{i_j} and A_i are determined by linear interpolation, as long as the sum of the absolute differences from the known points do not exceed a threshold t_2 .
4. Repeat steps 2 and 3 until a significant portion of the image is filled.
5. Any remaining unknown points are filled with some interpolation or inpainting algorithm.

PLI is able to reconstruct edges and other large-scale features well. There are significant small-scale artifacts, however, making the algorithm less suitable for reconstructing small and thin features.

In the examples that follow, $r = 10$, $t_1 = 5$, and $t_2 = 5$.

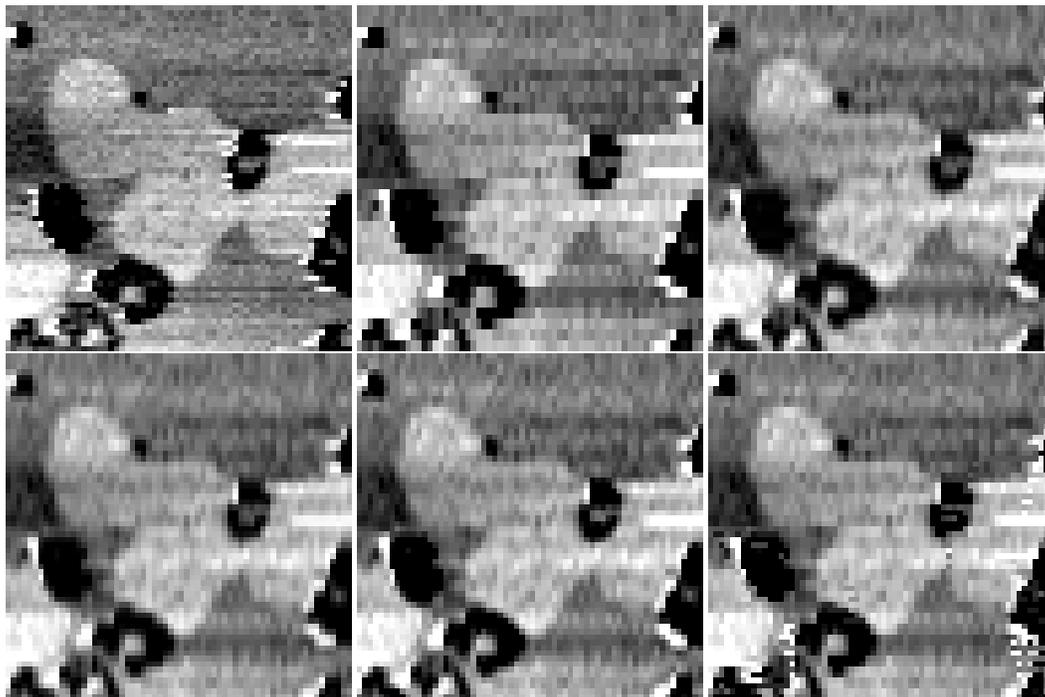


Fig. 1.4. Interpolation reconstructions from a section of an image of lipid bilayer domains on mica, subsampled by a factor of 2 on the vertical axis. Left to Right, Top to Bottom: Original image; Nearest neighbor interpolation; Lanczos-3 interpolation; Bicubic interpolation; Contour stencil interpolation; PLI.

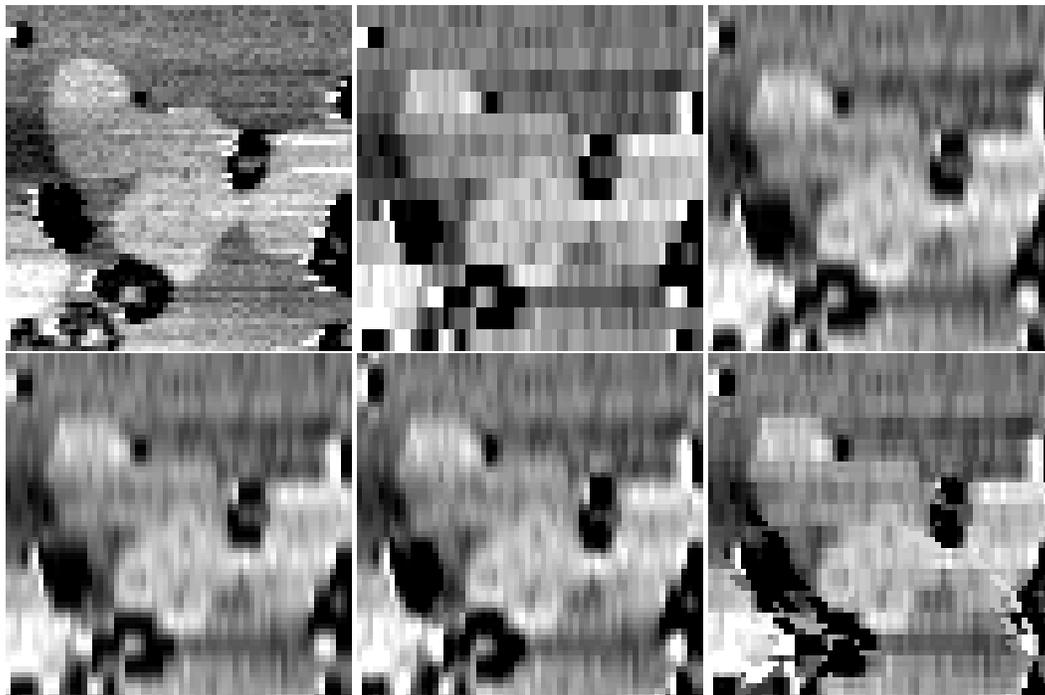


Fig. 1.5. Interpolation reconstructions from a section of an image of lipid bilayer domains, subsampled by a factor of 4 on the vertical axis. Left to Right, Top to Bottom: Original image; Nearest neighbor interpolation; Lanczos-3 interpolation; Bicubic interpolation; Contour stencil interpolation; PLI.

1.5 Variational Inpainting

The principle underlying inpainting is similar to that of nonlinear interpolation algorithms such as contour stencils. These algorithms generally detect features in the known region and continue them into the unknown region, while preserving properties such as edge continuity and curvature.

A major advantage of inpainting over interpolation methods is that they are more readily generalizable to reconstructing information in general unknown regions. With the extra information available from subsampling on only one axis instead of both, inpainting algorithms can reconstruct many features in an image more accurately. This is especially relevant to AFM applications, in which the sampling rate can be increased in the fast scan direction much more readily in the slow scan direction. Thus, the fast scan direction typically has sufficient resolution, while the slow scan direction requires enhancement. Inpainting methods, however, are usually more computationally expensive than interpolation methods.

Variational image inpainting methods define energy functionals that seek to recreate plausible images given the known data. These energies generally have the following structure:

$$E(u) = R(u) + \lambda(x,y)F(d(f,u)),$$

where $R(u)$ is a “regularization” term that penalizes unlikely image features such as high gradients (relatively less common than smooth changes), and $\lambda(x,y)F(d(f,u))$ is a data “fidelity” term that penalizes deviation from known data, as measured by the distance function $d(\cdot, \cdot)$. The weight $\lambda(x,y)$ is generally chosen to have a constant weight λ in the known region Ω_1 and 0 in the unknown region:

$$\lambda(\mathbf{x}) = \begin{cases} \lambda, & \text{if } (x,y) \in \Omega_1, \\ 0, & \text{if } (x,y) \in \Omega \setminus \Omega_1, \end{cases} \quad (1.1)$$

Minimization of the energy gives an image with the desired properties. One of the simplest models for variational inpainting with this structure is the H^1 (diffusion) model:

$$E(u) = \frac{1}{2} \int_{\Omega} |\nabla u|^2 dx dy + \frac{\lambda}{2} \int_{\Omega_1} (f - u)^2 dx dy.$$

The regularization term of the H^1 energy indeed penalizes the high gradients characteristic of noise. Unfortunately, edges are also excessively smoothed due to the squared penalty on gradients. The gradient descent equation also shows this fact:

$$u_t = \Delta u + \lambda(f - u),$$

which indicates that the propagation of information is by isotropic diffusion. Generally, features are reconstructed reasonably well, but since the diffusion is completely unbiased, significant blurring results.

A significant improvement is the total variation (TV) model of Rudin, Osher, and Fatemi [24], originally for image denoising. The TV inpainting energy is

$$E(u) = \int_{\Omega} |\nabla u| dx dy + \lambda \int_{\Omega} (f - u)^2 dx dy.$$

As in the H^1 model, large gradients are penalized, so that the model seeks smooth continuations of the data while removing noise. However, the lack of a square on the regularization term prevents excessive penalization. There are various methods to minimize this energy. Gradient descent has typically been used as a simple and straightforward method to find a minimizer. More recently, the Split Bregman method [15] and graph cuts [9] have made minimization more efficient.

1.5.1 Fourth-order Inpainting Methods

The TV model significantly improves edge definition. There are several other factors, however, that are desirable in an inpainting model. The connectivity principle as introduced in section 1.2.2. In the TV model, the connectivity principle is, in particular, often violated when connecting broken edges. If an unknown region separates two long, thin objects flowing toward each other, it is logical to assume that they should be connected through the unknown region. Yet the added amount of total variation needed to connect the objects may be high, so the TV model would keep the two objects separate.

One solution is to add a penalty on edge contour curvature, since a long, thin object which ends abruptly certainly has high curvature at its terminus. The curvature term adds to the complexity of the energy and the corresponding gradient descent. In fact, since curvature depends on second order derivatives, the corresponding gradient descent equation is a fourth order PDE.

A related problem is the shape of such an edge connection. Since the regularization term of the TV inpainting model depends only on the total variation, straight edge connections are preferred over curved edge connections, as the former would contain fewer pixels at high gradient locations.

In fourth-order inpainting methods, boundary conditions on the evolved function and its gradient need to be specified. The boundary condition on the function itself tends to promote continuity of edges near the boundary of the inpainting region. The second boundary condition promotes continuity in the gradient and thus promotes the propagation of information along level lines in a smooth manner.

Low Curvature Image Simplifiers (LCIS) [25, 27] is a fourth-order inpainting method that provides many fine-scale features that are lost in methods such as H^1 and TV inpainting. The inpainted version follows the evolution

$$u_t = -\nabla \cdot (g(\Delta u) \nabla \Delta u) + \lambda(f - u),$$

where $g(s) = \frac{1}{1+s^2}$ is a ‘‘conductance threshold.’’ For high values of Δu , $g(\Delta u)$ is small, so that there is little evolution across high gradients. However, low values of Δu give large values of $g(\Delta u)$, promoting the propagation of information.

LCIS is based on the Perona-Malik equation [23], a second order PDE often used for image denoising tasks because it propagates information via anisotropic diffusion. Thus, edge sharpness is preserved while noise is

smoothed. Unfortunately, the Perona-Malik equation is ill-posed in continuous space, making the model somewhat theoretically unsatisfying. On the other hand, LCIS preserves the anisotropic diffusion properties of the Perona-Malik model while being globally well-posed [4, 16] and making more realistic curvature connections.

The results of several inpainting reconstructions are shown in Fig. 1.6 and Fig. 1.7 for an image of lipid bilayer domains that has been subsampled by factors of 2 and 4 on the vertical axis.

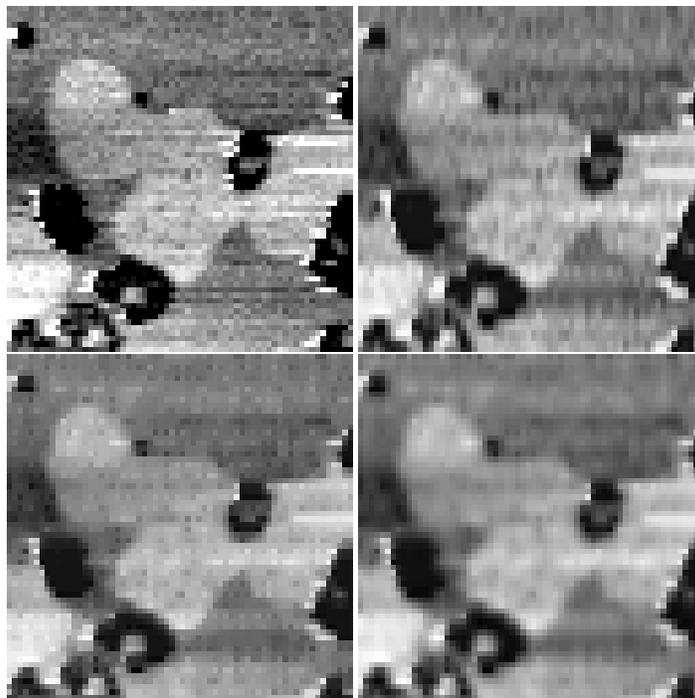


Fig. 1.6. Inpainting reconstructions from a section of a lipid bilayer image, subsampled by a factor of 2 on the vertical axis. Left to Right, Top to Bottom: Original image; H1 inpainting; TV inpainting; LCIS inpainting.

1.6 Reconstructing Damaged Scan Lines

With many AFM images, there are some artifacts related to the process of raster scanning. After each scan line is complete, flattening is done in order to adjust for effects such as tilt and thermal drift, mostly linear in their effects. In this way, a first order polynomial is subtracted from each scan line. Flattening generally works well in compensating for tilt and thermal drift, but some errors still occur, particularly relating to streaks.

These streaks can occur for various reasons. In the course of scanning, the probe may be damaged, be temporarily changed from the addition of material from the sample, or be changed when thermal excitations cause jumps between stable states in the governing equations of the probe-sample interactions.

Another source of the streaks is when anomalous features are detected within a given scan line. In general, since the anomalous features are part of the image, it is useful to keep these in a processed image. One problem with this is that due to the flattening process, streaks can occur in scan lines directly following these contaminants. Since flattening is done by subtracting a polynomial function from each scan line, this can result in shifting the data around the feature. One further challenge is designing an automatic detection method that can distinguish between streaks due to features in the sample and streaks due to mistakes in the scanning process.

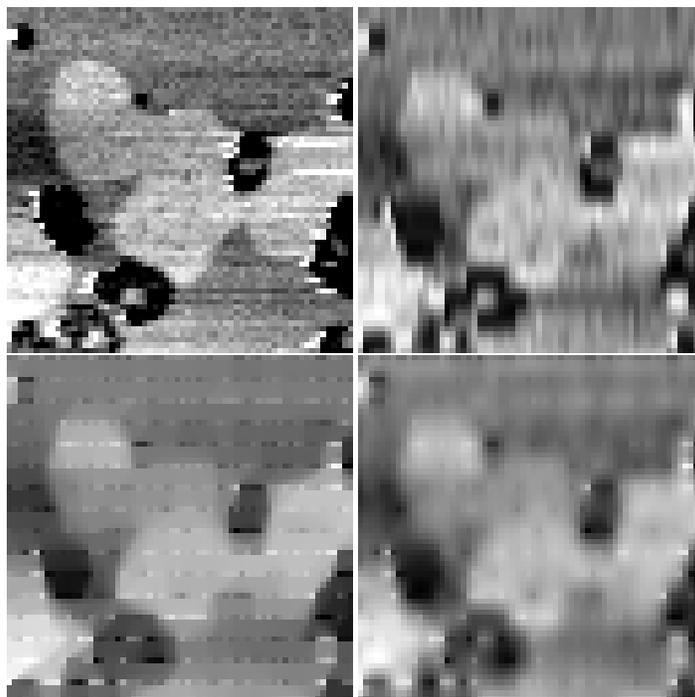


Fig. 1.7. Inpainting reconstructions from a section of a lipid bilayer image, subsampled by a factor of 4 on the vertical axis. Left to Right, Top to Bottom: Original image; H1 inpainting; TV inpainting; LCIS inpainting.

One of the current standard techniques to deal with these streak artifacts is removal of the entire line, followed by an average of the neighboring scan lines. This can cause some distortions, particularly near edges. Fig. 1.8 shows the removing of streaks from the image by manual identification of the inaccurate parts, followed by inpainting of the identified regions by LCIS inpainting and by averaging. The results look comparable, since the unknown regions are small. However, inpainting algorithms work slightly better near edges and when there are multiple streaks nearby.

1.7 Reconstructing Important Features

In this section, various image features are examined more closely and the various algorithms are compared and contrasted. Fig. 1.9 shows zoomed versions of the most common type of edge, one that forms the boundary between two regions of contrasting intensity.

Since the nonlinear interpolation methods explicitly detect the orientation of edges, they are generally able to reconstruct sharper edges than the linear interpolation methods. Additionally, the staircasing effect is reduced as well. The edge is reconstructed more sharply in the PLI algorithm than in contour stencils at the expense of more artifacts.

Analogously, the TV and LCIS inpainting methods result in better edge reconstructions than the H1 method because information is designed to propagate along edges and not isotropically. These result in edges that are comparable to the nonlinear interpolation techniques in sharpness.

A second type of edge is that of a trench separating two regions of similar intensity. These types of edges are typically much more difficult for inpainting and interpolation methods due to the trench values making up a smaller portion of the neighborhood around an edge. Thus, both interpolation and inpainting methods tend to blur these edges by averaging with the surrounding values. Additionally, variational inpainting places an extra penalty

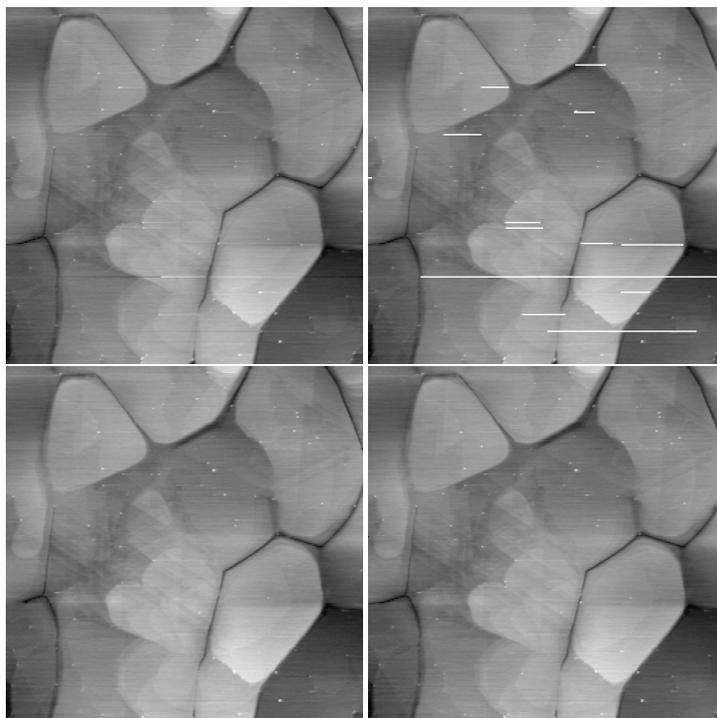


Fig. 1.8. Inpainting damaged scan lines from the “Annealed Gold” image. Top Left: Original image. Top Right: Manual identification of the damaged areas is displayed in white. Bottom Left: Averaging of damaged lines. Bottom Right: Recovery by LCIS inpainting.

on having a double-sided gradient, so that edges often remain disconnected. The PLI algorithm connects this type of edge but at the expense of significant artifacts.

Example reconstructions on this type of edge are in Fig. 1.10.

Staircasing of diagonal edges occurs with both the interpolation and inpainting methods. Fig. 1.11 shows a comparison between the recovery of several different interpolation and inpainting algorithms. The TV inpainting algorithm performs very well in straightening all edges, due to its tendency to connect edges in straight lines. Most of the algorithms give minimal staircasing for the longer edge of the InP nanowire, but do much worse along the shorter side of the nanowire. This is due to the fact that the longer edge lies mostly along the vertical direction. In the space of the recovered image, the points along the edge are separated by a shorter distance and thus more easily connected.

1.7.1 Difference Images

Difference images can be helpful in determining where the largest errors in the reconstruction take place. They are computed by taking the absolute value of the difference between the reconstructed image and the ground truth image. Not surprisingly, they often occur near edges and noise. The lightest parts often indicate systematic errors in a certain method. Fig. 1.12 and Fig. 1.13 show the difference images formed from the various methods on the InP nanowire and lipid bilayer images.

The various algorithms can also be measured objectively through the calculation of the Peak Signal-to-Noise ratio (PSNR), which uses the root mean squared error (RMSE):

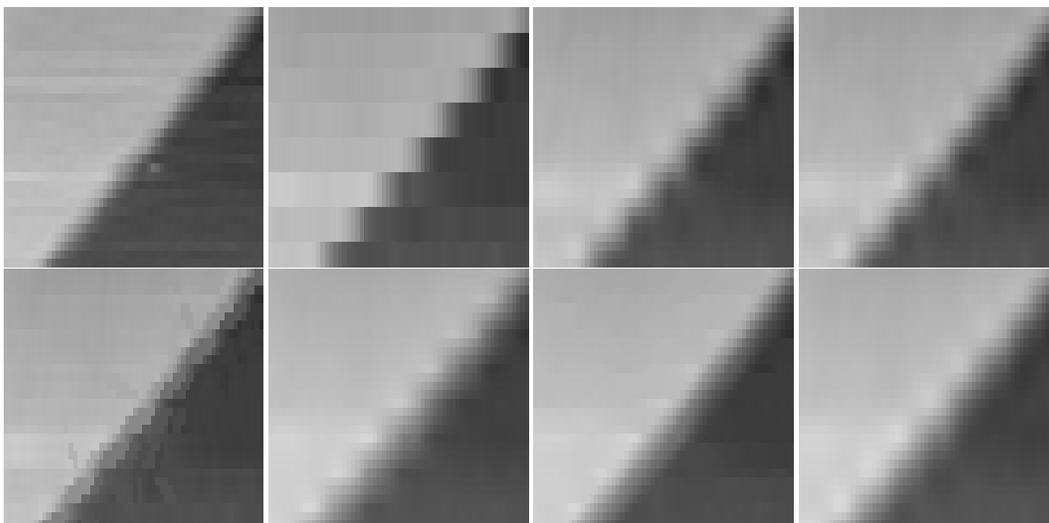


Fig. 1.9. A high-contrast edge from an image of annealed gold, subsampling by 4 in the vertical direction. Left to Right, Top to Bottom: Original image; nearest neighbor interpolation; bicubic interpolation; contour stencil interpolation; PLI; H1 inpainting; TV inpainting; LCIS inpainting.

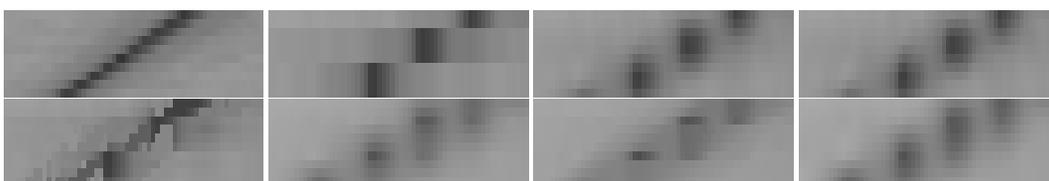


Fig. 1.10. Edges from a trench between two regions of similar intensity from an image of annealed gold, subsampling by 4 in the vertical direction. Left to Right, Top to Bottom: Original image; nearest neighbor interpolation; bicubic interpolation; contour stencil interpolation; PLI; H1 inpainting; TV inpainting; LCIS inpainting.

$$RMSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |I_{m,n} - \tilde{I}_{m,n}|^2.$$

Then the PSNR is defined as:

$$PSNR = 20 \cdot \log_{10} \left(\frac{1}{RMSE} \right).$$

Another method that compares the quality of an image reconstruction is the mean Structural SIMilarity (MSSIM) index, which measures the similarity between two images by comparing luminance, contrast and structure [28].

These measures give some indication of the effectiveness of a method, but they can sometimes be misleading as well. If an algorithm performs well near important features such as edges but badly in the background, its performance indicators will be worse than for an algorithm that does well in the background and worse near edges. Yet the former might be preferable in that the features of interest are reconstructed well.

A table of the PSNR and SSIM on the various reconstructions of the lipid bilayer image is shown in Table 1.1 and the same table is shown for the InP nanowire image in Table 1.2.

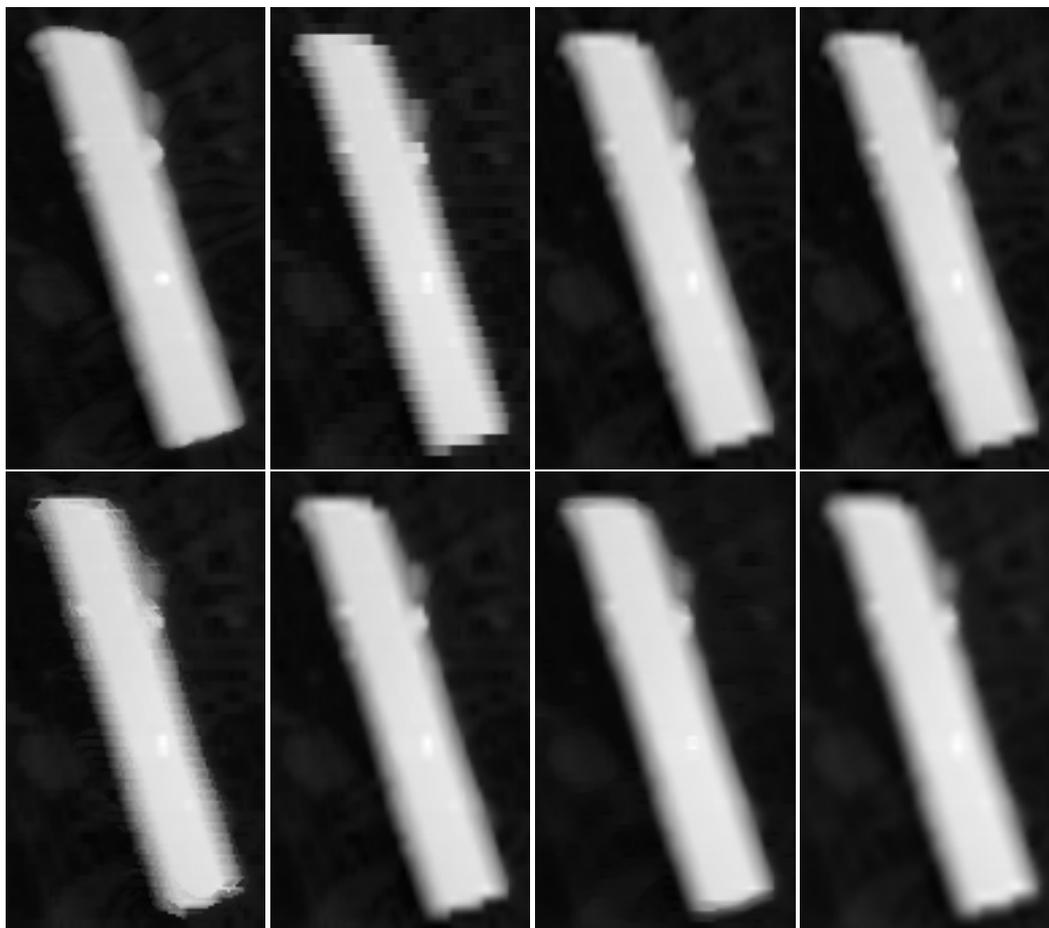


Fig. 1.11. A comparison of edge quality in the recovery from various inpainting and interpolation algorithms on the InP nanowire image, subsampled by a factor of 4 in one direction. Left to Right, Top to Bottom: Original image; nearest neighbor interpolation; bicubic interpolation; contour stencil interpolation; PLI; H1 inpainting; TV inpainting; LCIS inpainting.

Table 1.1. A comparison of peak signal-to-noise ratio (PSNR) and mean Structural SIMilarity (SSIM) for various recovery algorithms on the lipid bilayer image. The best performing algorithm in each column is in bold.

Objective Measures of an Algorithm's Effectiveness		
Method	PSNR	MSSIM
Nearest Neighbor	14.8636	0.4494
Bicubic	15.6739	0.4489
Contour Stencils	15.1333	0.4396
PLI	14.9638	0.4728
H1	16.0301	0.4398
TV	16.5837	0.4356
LCIS	16.6496	0.4292

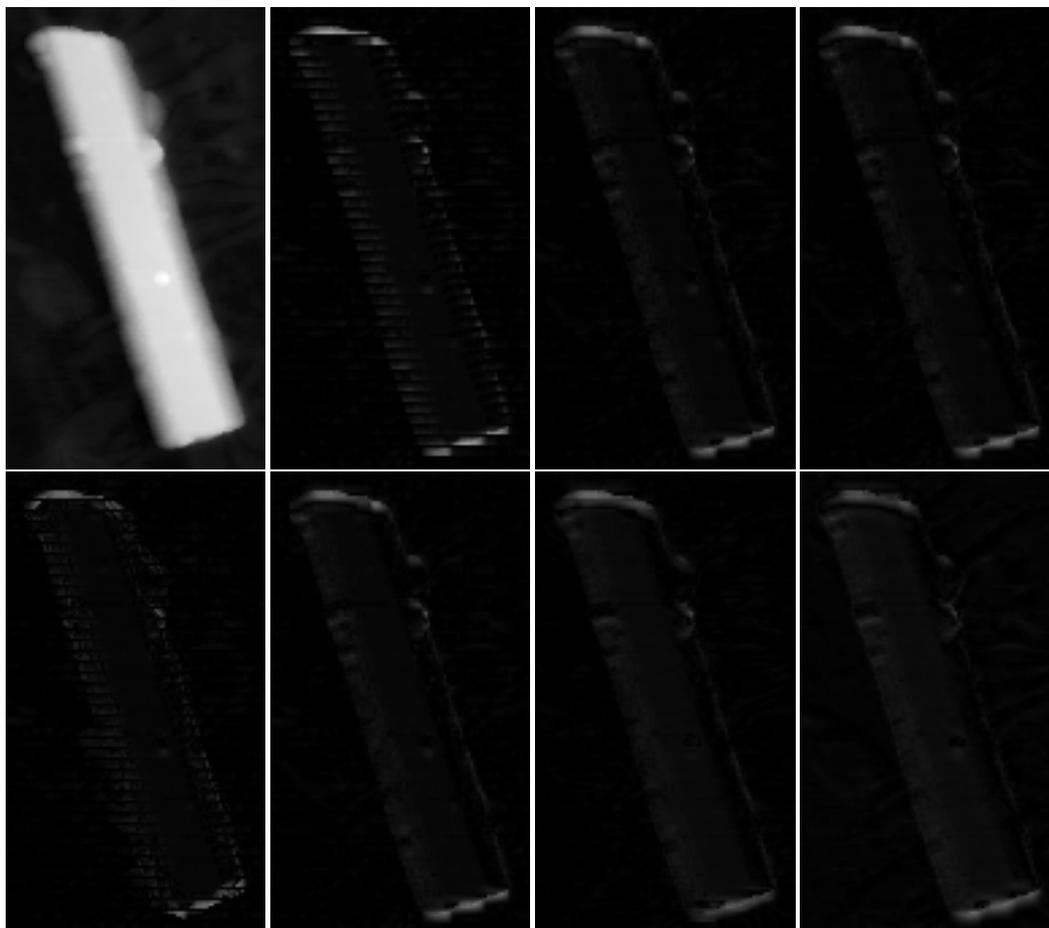


Fig. 1.12. Difference images of various inpainting and interpolation algorithms on the InP nanowire, subsampled by a factor of 4 in the vertical direction. The original image is shown in the top left corner for comparison. Left to Right, Top to Bottom: Original image; nearest neighbor interpolation; bicubic interpolation; contour stencil interpolation; PLI; H1 inpainting; TV inpainting; LCIS inpainting.

Table 1.2. A comparison of peak signal-to-noise ratio (PSNR) and mean Structural SIMilarity (MSSIM) for various recovery algorithms on the InP nanowire image. The best performing algorithm in each column is in bold.

Objective Measures of an Algorithm's Effectiveness		
Method	PSNR	MSSIM
Nearest Neighbor	26.3990	0.9104
Bicubic	27.9555	0.9367
Contour Stencils	28.0161	0.9364
PLI	28.5414	0.9262
H1	27.8737	0.9383
TV	27.9498	0.9422
LCIS	27.1007	0.9292

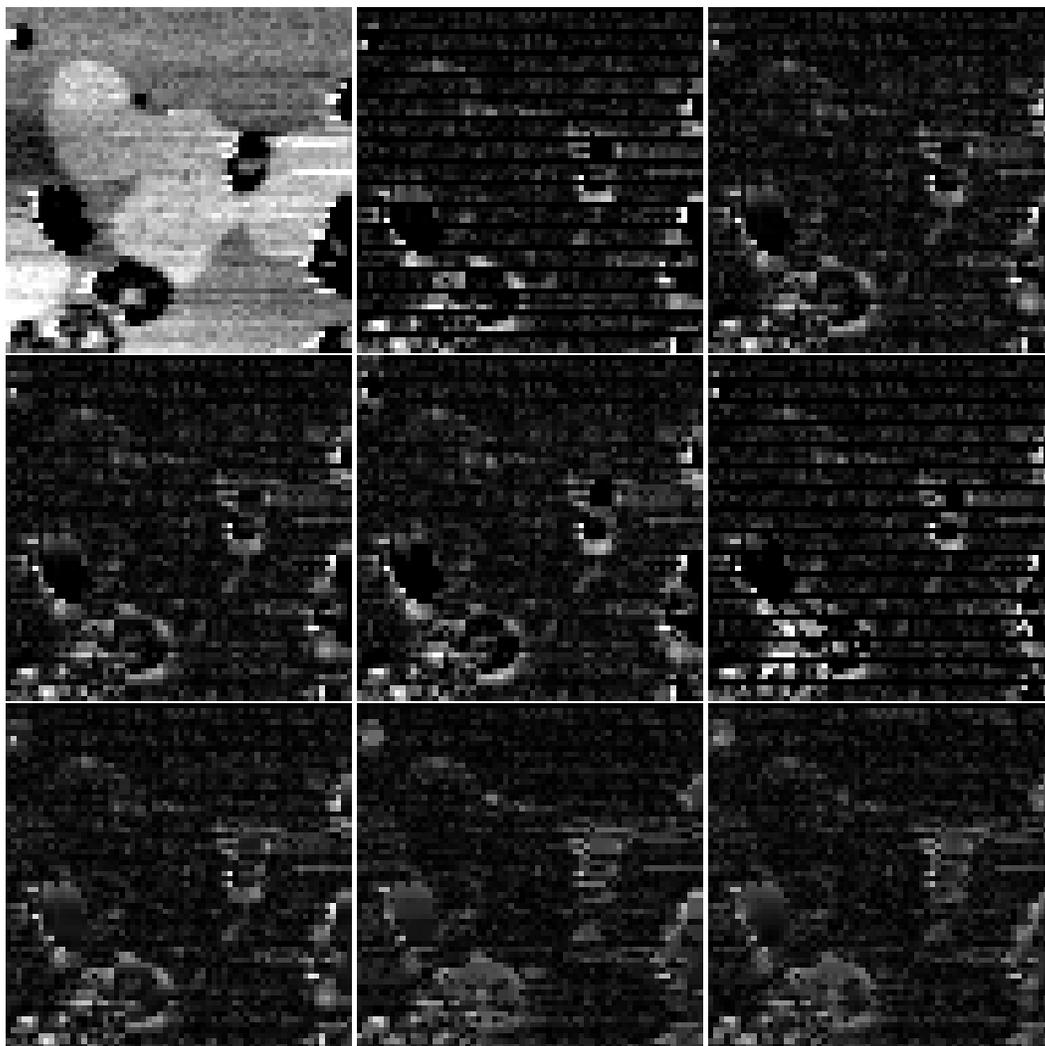


Fig. 1.13. Difference images of various inpainting and interpolation algorithms on the lipid bilayer image, subsampled by a factor of 4 in the vertical direction. The original image is shown in the top left corner for comparison. Left to Right, Top to Bottom: Original image; nearest neighbor interpolation; Lanczos-3 interpolation; bicubic interpolation; contour stencil interpolation; PLI; H1 inpainting; TV inpainting; LCIS inpainting.

Acknowledgement. The authors would like to thank Todd Wittman, Jef Huang and Kevin Thompson for useful conversations on the AFM and inpainting. This research is supported by NSF grant CBET-0940417. Work at the Molecular Foundry was supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Image of lipid bilayer domain sample provided by Elaine DeMasi.

References

1. Ando, T., Uchihashi, T., Kodera, N., Yamamoto, D., Miyagi, A., Taniguchi, M., Yamashita, H.: High-speed AFM and nano-visualization of biomolecular processes. In: *Pflügers Arch - Eur. J. Physiol.*, vol. 456, pp. 211–225. Springer (2008)
2. Ashby, P., Lieber, C.: Ultra-sensitive imaging and interfacial analysis of patterned hydrophilic SAM surfaces using energy dissipation chemical force microscopy. *J. Am. Chem. Soc.* **127**, 6814–6818 (2005)
3. Bertozzi, A., Esedoglu, S., Gillette, A.: Inpainting of binary images using the Cahn-Hilliard equation. *IEEE Transactions on Image Processing* **16**(1), 285–291 (2007)
4. Bertozzi, A.L., Greer, J.B.: Low curvature image simplifiers: global regularity of smooth solutions and Laplacian limiting schemes. *Comm. Pure Appl. Math.* **57**(6), 764–790 (2004)
5. Binnig, G., Quate, C., Gerber, C.: Atomic force microscope. *Physical Review Letters* **56**, 930–933 (1986)
6. Chan, T., Kang, S., Shen, J.: Euler’s elastica and curvature-based inpainting. *SIAM J. Appl. Math.* **63**(2), 564–592 (2002)
7. Chan, T., Shen, J.: Variational image inpainting (2005)
8. Chasiotis, I.: Atomic force microscopy in solid mechanics. In: W.N.S. Jr. (ed.) *Springer Handbook of Experimental Solid Mechanics*, pp. 409–443. Springer, New York (2008)
9. Darbon, J., Lefebvre, S., Chan, T., Esedoglu, S.: TV optimization and graph-cuts. *Proceedings in Applied Mathematics and Mechanics* **7**(1), 1042,303–1042,304 (2007)
10. Dobrosotskaya, J., Bertozzi, A.: A wavelet-laplace variational technique for image deconvolution and inpainting. *IEEE Transactions on Image Processing* **17**(5), 657–663 (2008)
11. Esedoglu, S.: Blind deconvolution of bar code signals. *Inverse Problems* **20**, 121–135 (2004)
12. Florin, E., Moy, V., Gaub, H.: Adhesion forces between individual ligand-receptor pairs. *Science* **264**, 415–417 (1994)
13. Getreuer, P.: Contour stencils for edge-adaptive image interpolation. In: *Proc. SPIE*, vol. 7246 (2009)
14. Getreuer, P.: Image zooming with contour stencils. In: *Proc. SPIE*, vol. 7257 (2009)
15. Goldstein, T., Osher, S.: The split Bregman algorithm for L1 regularized problems. *SIAM Journal on Imaging Sciences* **2**(2), 323–343 (2009)
16. Greer, J.B., Bertozzi, A.L.: Traveling wave solutions of fourth order PDEs for image processing. *SIAM J. Math. Anal.* **36**(1), 38–68 (2004)
17. Hansma, P., Cleveland, J., Radmacher, M., Walters, D., Hillner, P., Bezanilla, M., Fritz, M., Vie, D., Hansma, H., Prater, C., Massie, J., Fukunaga, L., Gurley, J., Elings, V.: Tapping mode atomic force microscopy in liquids. *Surface Science Letters* **64**(13), 1738–1740 (1994)
18. Kodera, N., Yamamoto, D., Ishikawa, R., Ando, T.: Video imaging of walking myosin v by high-speed atomic force microscopy. *Nature* **468**, 72–76 (2010)
19. Lapshin, R.V.: Feature-oriented scanning methodology for probe microscopy and nanotechnology. *Nanotechnology* **15**, 1135–1151 (2004)
20. Martin, Y., Wickramasinghe, H.: Magnetic imaging by “force microscopy” with 1000 Å resolution. *Applied Physics Letters* **50**, 1455–1457 (1987)
21. Morel, J., Yu, G.: Is SIFT scale invariant? *Inverse Problems and Imaging* **5**(1), 115–136 (2011)
22. Noy, A., Vezenov, D., Kayyem, J., Meade, T., Lieber, C.: Stretching and breaking duplex DNA by chemical force microscopy. *Chem. Biol.* **4**, 519–527 (1997)
23. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Machine Intell.* **12**(7), 629–639 (1990)
24. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**, 259–268 (1992)
25. Schoenlieb, C.B., Bertozzi, A.: Unconditionally stable schemes for higher order inpainting. *Comm. Math. Sci.* **9**(2), 413–457 (2011)
26. Thévenaz, P., Blu, T., Unser, M.: Image interpolation and resampling. In: *Handbook of medical imaging*. Academic Press, Inc., Orlando, FL, USA (2000)
27. Tumblin, J., Turk, G.: LCIS: A boundary hierarchy for detail-preserving contrast reduction. In: *Siggraph, Computer Graphics Proceedings*, pp. 83–90 (1999)

28. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity **13**(4), 600–612 (2004)
29. Zhong, Q., Innis, D., Kjolle, K.: Fractured polymer/silica fiber surface studied by tapping mode atomic force microscopy. *Surface Science Letters* **290**, L688–L692 (1993)