Electronic Transactions on Numerical Analysis.
Volume 13, pp. 22-37, 2002.
Copyright © 2002, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

IMAGE RESTORATION THROUGH SUBIMAGES AND CONFIDENCE IMAGES[*]

JAMES G. NAGY[†] AND DIANNE P. O'LEARY[‡]

Abstract. Some very effective but expensive image reconstruction algorithms cannot be applied to large images because of their cost. In this work, we first show how to apply such algorithms to subimages, giving improved reconstruction of regions of interest. Our second contribution is to construct confidence intervals for pixel values, by generalizing a theorem of O'Leary and Rust to allow both upper and lower bounds on variables. All current algorithms for image deblurring or deconvolution output an image. This provides an estimated value for each pixel in the image. What is lacking is an estimate of the statistical confidence that we can have in those pixel values or in the features they form in the image. There are two obstacles in determining confidence intervals for pixel values: first, the process is computationally quite intensive, and second, there has been no proposal for providing the results in a visually useful way. In this work we overcome the first of those limitations and develop an algorithm called Twinkle to overcome the second. We demonstrate the usefulness of these techniques on astronomical and motion-blurred images.

Key words. image restoration, regularization, confidence intervals, confidence images, motion blur, conjugate gradient method.

AMS subject classifications. 65F10, 65F20, 65F30.

1. Introduction.

Image deconvolution or deblurring can be accomplished by a variety of methods, some simple and some quite expensive. Linear methods such as (linear) least squares or back-projection methods often provide adequate results, and their cost is quite low. In contrast, constrained least squares algorithms, which impose the side conditions that pixel values lie within a given range, are more expensive but sometimes yield better results.

In many cases, after a reconstruction of the image, there may be a need to further improve a small part of the image. In astronomical imaging, for example, this may be a star cluster; in medical imaging, it may be an area in which a tumor is suspected. We develop the techniques required to extract subimages and apply more sophisticated processing techniques to them.

Our basic idea can be outlined as follows.

1. Given an image restoration problem involving a measured image y and a point-spread function (PSF) $\mathcal{K}$, we use a basic technique to compute an initial reconstruction $\hat{x}$ of the true $n \times n$ image x.
2. Using the initial reconstruction $\hat{x}$, we identify a region of interest for further improvement. We denote this true subimage by x_s and apply a more sophisticated technique in order to obtain improved estimates $\tilde{x}_s$ on this subimage.

There are numerous variants on this basic scheme. As an example, a few steps of preconditioned conjugate gradients (PCG) can be applied to the problem

$$\min_x \|Kx - y\|_2,$$

where K is the matrix corresponding to the PSF $\mathcal{K}$ and the vectors x and y contain the pixel values of the true image and the measured image, respectively. The result of this computation is an estimate $\hat{x}$, and then we can compute confidence intervals for each pixel value in some subimage.

[*] Received April 5, 2001. Accepted for publication February 1, 2002. Recommended by D. Calvetti.

[†] Department of Mathematics, Emory University, Atlanta, Georgia 30322. E-mail: nagy@mathcs.emory.edu. This research was supported by the National Science Foundation under Grant DMS 00-75239.

[‡] Department of Computer Science, University of Maryland, College Park, Maryland. E-mail: oleary@cs.umd.edu. This research was supported by the National Science Foundation under Grant CCR 97-32022.

As a second example, filtered backprojection can be used in Step 1, and we can use a nonnegatively constrained least squares algorithm in Step 2.

Basically, our approach combines the power of an inexpensive method (such as PCG or filtered back projection) to obtain an initial (global) reconstruction of the image, and then further improve small subimages of interest using a nonlinear method that gives better reconstructions. Because the improvements are done on small subimages, the nonlinear methods become economical to implement.

The improvement of subimages has been considered in other work in the context of improving efficiency of expensive global methods. Some examples of this work include the sectioning methods developed by Trussell and Hunt [12], a scanning SVD method suggested by Fish, Grochmalicki and Pike [3], and more recently a piecewise polynomial SVD method suggested by Hansen [5]. These methods, though, require that the size of the image blocks be at least as large as the support of the PSF. This is a severe restriction in cases where the image contains a substantial blur. The approach considered in this paper does not have this restriction, and thus we are able to solve a wider class of problems.

In section 2 we describe how we construct the subproblem data. We illustrate the use of nonlinear methods on such subproblems in order to improve the reconstruction over the subimage. In section 3 we describe how confidence images can be used to compute approximations to discrete ill-posed problems. We illustrate the effectiveness of our approach by computing and displaying confidence images for some image restoration examples. Section 4 presents some summary comments.

2. Construction of Subimage and Submatrix.

In this section we describe how to construct the small subproblem

$$K_s x_s \approx y_s \, .$$

We'll assume that the original image has dimension $n \times n$, while the subimage has dimension $r \times c$. We note that for large image restoration problems, the matrix K is not formed explicitly, so extracting a submatrix from it is not necessarily a trivial operation. In fact, the efficient formation of K_s requires some background material involving PSFs and Kronecker products, but before we present that, let's consider the idea itself. We begin with the problem

$$Kx \approx y \, ,$$

where K is the $N \times N$ matrix corresponding to the PSF, y is a vector containing the measured values of the pixels (including noise), and x is the unknown true image. Note that $N = n^2$, where n is the dimension of the true image. Let E be the matrix with N rows and $S = rc$ columns, each column a unit vector corresponding to a pixel in the subimage. Let $\bar{E}$ have columns equal to the other $N - S$ unit vectors. Then

$$\begin{aligned} Kx &= (K \begin{bmatrix} E & \bar{E} \end{bmatrix})(\begin{bmatrix} E^T \\ \bar{E}^T \end{bmatrix} x) \\ &\equiv \begin{bmatrix} \hat{K}_s & \hat{K}_t \end{bmatrix} \begin{bmatrix} x_s \\ x_t \end{bmatrix} \\ &= \hat{K}_s x_s + \hat{K}_t x_t, \end{aligned}$$

where x_s contains the pixels in the subimage. The idea is to improve the entries of x_s while leaving those of x_t at their current values, so our resulting problem is

$$\hat{K}_s x_s \approx y - \hat{K}_t x_t \, .$$

Since most of the rows of $\hat{K}_s$ are zero (because the PSF typically has small support), we can drop down to a smaller dimensional problem, dropping the unwanted rows

$$K_s x_s \approx y_s\,.$$

Our discussion is therefore divided into several subsections, beginning with a discussion of how we form y_s.

2.1. Constructing the Subimage. We begin by considering a simple example. Suppose we partition the problem $Kx = y$ as

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix},$$

and suppose the subimage that we would like to improve is $x_s = x_2$. We'll assume that the row partitioning of K has been done so that $K_{12} = 0$, $K_{32} = 0$, and there are no zero rows in K_{22}. Then

$$\begin{bmatrix} K_{12} \\ K_{22} \\ K_{32} \end{bmatrix} x_s = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} - \begin{bmatrix} K_{11} & K_{13} \\ K_{21} & K_{23} \\ K_{31} & K_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}.$$

A subproblem involving x_s can then be formed as

$$K_s x_s \approx y_s\,,$$

where $K_s = K_{22}$ and $y_s = y_2 - K_{21}x_1 - K_{23}x_3$.

Of course since x is not known, we cannot compute y_s. However, if we assume that a good initial approximation, $\hat{x}$, can be computed using PCG or some other method, then we can replace x_1 and x_3 with $\hat{x}_1$ and $\hat{x}_3$, and this gives us an estimate of y_s.

With this simple example in mind, we can generalize the approach to construct y_s. Recall that we have defined E be the matrix with N rows and S columns, each column a unit vector corresponding to a pixel in the subimage, and $\bar{E}$ has columns equal to the other $N - S$ unit vectors. Note that

$$x_s = E^T x$$

and

$$x = \begin{bmatrix} E & \bar{E} \end{bmatrix} \begin{bmatrix} E^T \\ \bar{E}^T \end{bmatrix} x\,,$$

since the product of those matrices is the identity. Let $\hat{E}$ be the matrix with N rows and $S + s$ columns, each column a unit vector corresponding to a row of K_{22}. Then, assuming $\hat{x}$ has been previously computed, the vector y_s can be obtained this way:

1. Form a vector z that is equal to zero in the subimage and equal to $\hat{x}$ outside that subimage:

$$z = \begin{bmatrix} E & \bar{E} \end{bmatrix} \begin{bmatrix} 0 \\ \bar{E}^T \end{bmatrix} \hat{x}\,.$$

2. Compute $\hat{y} = y - Kz$.
3. Then $y_s = \hat{E}^T \hat{y}$.

The *efficiency* of this approach depends on the cost of forming a matrix-vector product with K. The *effectiveness* depends on the error contained in y_s and how it is magnified by solving the subproblem. Fortunately, the subproblem is better conditioned than the original problem, since the singular values of the submatrix K_s are bounded above and below by the singular values of the original matrix K. The right-hand side y_s, however, has somewhat more error than the original y, and the distribution of this error has also been changed in a manner dependent on how the estimates of the x variables were made.

Our next goal is to construct the submatrix $K_s = \hat{E}^T K E$.

2.2. Construction of the Submatrix. The matrix K is defined through PSFs that represent the blur of a point source. These functions can be determined by an idealized model of the blur, or experimentally by physically recording one or more images of a point source. If the blur is *spatially invariant*, then the blur is independent of position, and one PSF completely describes the blurring operation. In this case, fast Fourier transforms can be used to efficiently form matrix-vector multiplications involving K. The matrix K is never formed explicitly; only the single PSF is needed.

The situation is more difficult if the blur is *spatially variant*. In this case, the blur depends on the position of the point source, and therefore a single PSF does not completely describe the blurring operation. If the blur is assumed to be locally spatially invariant, then efficient matrix-vector multiplies can be implemented and effective preconditioners for PCG can be constructed; see [7, 8] for further details.

In the rest of our discussion, we assume that the blur is possibly spatially variant, but that locally the blur is spatially invariant (this is a common assumption in image restoration; see [7] and the references therein). Therefore, when we apply PCG to the large image, we use a spatially variant model of the blur, but when we consider the subproblem, we use a spatially invariant model.

In the spatially invariant case, the matrix K has an interesting decomposition. Let P be a $p \times p$ array containing the coefficients of the PSF. Then the $n^2 \times n^2$ matrix K can be decomposed into

$$K = \sum_{i=1}^{k} A_i \otimes B_i \,,$$

where k =rank(P), and A_i and B_i are banded $n \times n$ Toeplitz matrices [6]. The notation $\otimes$ denotes Kronecker product:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & & \vdots \\ a_{n1}B & \cdots & a_{nn}B \end{bmatrix} .$$

We remark that in our experience, most blurs have PSFs with rank very small compared to (and independent of) the dimension of the image (e.g., rank$(P) \leq 5$). The cost of computing A_i and B_i is $O(p^3)$.

Analogous to the Kronecker product decomposition of K, the matrix E has a Kronecker product decomposition:

$$E = E_p \otimes E_q,$$

where E_p is an $n \times r$ matrix with each column a unit vector corresponding to a row in the two-dimensional image that is in the subimage. Similarly, E_q is an $n \times c$ matrix of unit vectors corresponding to columns of the image that are in the subimage. We also decompose $\hat{E}$ as

$$\hat{E} = \hat{E}_p \otimes \hat{E}_q \,.$$

Thus the submatrix K_s can be computed as

$$\begin{aligned} K_s &= (\hat{E}_p \otimes \hat{E}_q)^T K (E_p \otimes E_q) \\ &= (\hat{E}_p \otimes \hat{E}_q)^T \left(\sum_{i=1}^{k} A_i \otimes B_i \right) (E_p \otimes E_q) \\ &= \sum_{i=1}^{k} (\hat{E}_p^T A_i E_p) \otimes (\hat{E}_q^T B_i E_q) \,. \end{aligned}$$

This is quite efficient to compute if the dimension $S = rc$ of the subimage, and the rank k are both small. If k is too large, we can truncate the summation to obtain the best such approximation in the Frobenius norm; see [6] for more details.

It may be possible to exploit further structure of K_s when using it in computations, but since this matrix is relatively small, we feel that such (possibly unstable) techniques are not necessary. We also point out that if the subimage is near the boundary of the image domain, then consideration should be given to choice of correct boundary conditions; see, for example, [9]. However, if the subimage is contained within the image domain, boundary conditions are easily taken into account by extending the subimage boundaries sufficiently away from the object of interest.

2.3. An Example: Hubble Space Telescope image, with a spatially variant PSF. We obtained data from ftp://ftp.stsci.edu/software/stsdas/testdata/restore/sims/star_cluster, intended to simulate a star cluster as it would appear to the Hubble Space Telescope Wide-Field Planetary Camera before its repair. First, we begin with only one PSF, assuming that the blur is spatially invariant. Figure 1 shows the true star cluster image, the blurred image, the PCG restored image (using 3 iterations), and the subimage we attempt to improve. The preconditioner used here is an optimal circulant approximation to the PSF matrix, K, and can be efficiently implemented using fast Fourier transforms; see [1, 4, 7] for further details. In the larger images in this plot, the subimage of interest is outlined by a white box. This 16×16 subimage corresponds to row pixels 45 to 60, and column pixels 175 to 190. We use nonnegatively constrained least squares (as implemented in the algorithm `NNLS` of Matlab) to improve our estimates over this subimage.

In Figure 2 we compare the true, blurred, PCG restored, and NNLS improved subimages. Figure 3 shows the same images, but as mesh plots. The NNLS improvement is better than the PCG restoration in this subimage, but still we are not able to determine that only one star is in this subimage.

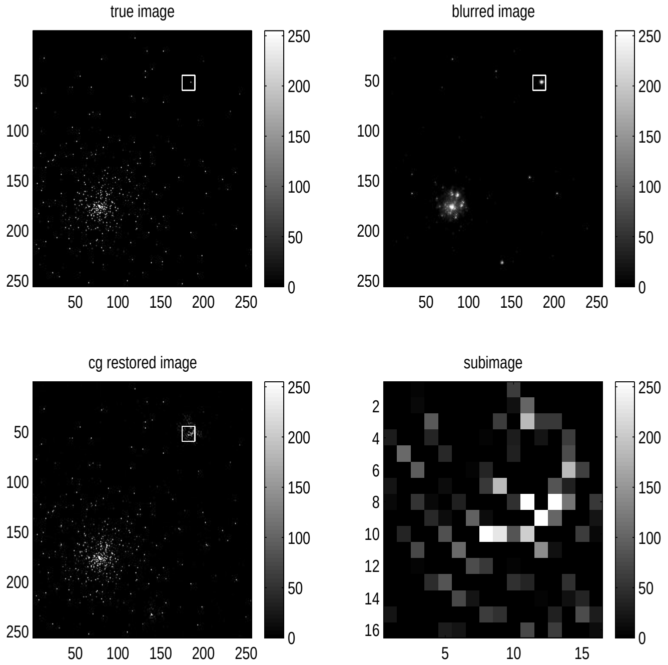


FIG. 1. *Satellite data and restoration using one PSF in PCG.*

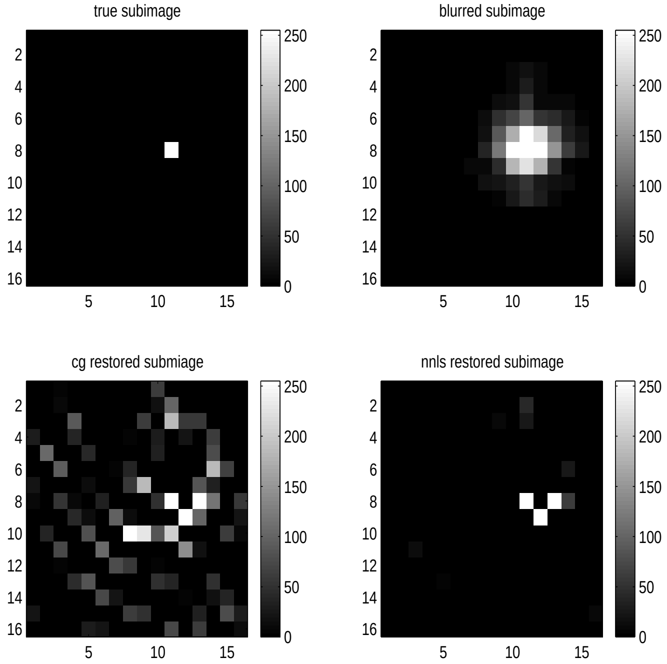


FIG. 2. *Subimages, using one PSF for PCG.*

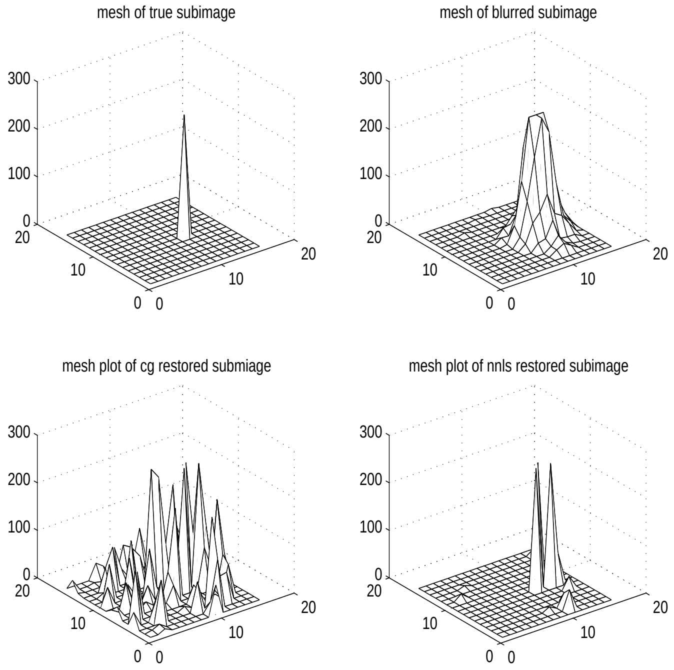


FIG. 3. *Mesh plot of subimages, using one PSF for PCG.*

Since the blur is actually spatially variant, we hope that we can improve the picture using more PSFs. From our earlier work [8], we found that good PCG restorations could be obtained using four PSFs.

Figure 4 shows the subimages of the true, blurred, PCG restored image after three iterations, and NNLS improved subimages in this case. Figure 5 shows the same images, but as mesh plots. We see that using the spatially variant blur, we are able to do much better. Note that we used multiple PSFs in the PCG restoration, but needed only the one local PSF to construct the submatrix used by NNLS.

Thus, we can quite effectively use expensive algorithms to reconstruct a subimage, once an inexpensive algorithm has been applied to find a region of interest. It should be noted that the coefficient matrix in this particular example is fairly well-conditioned, and therefore a good restoration can be computed without any regularization. However, in many image restoration problems, regularization is necessary, and this issue will be addressed in the next section.

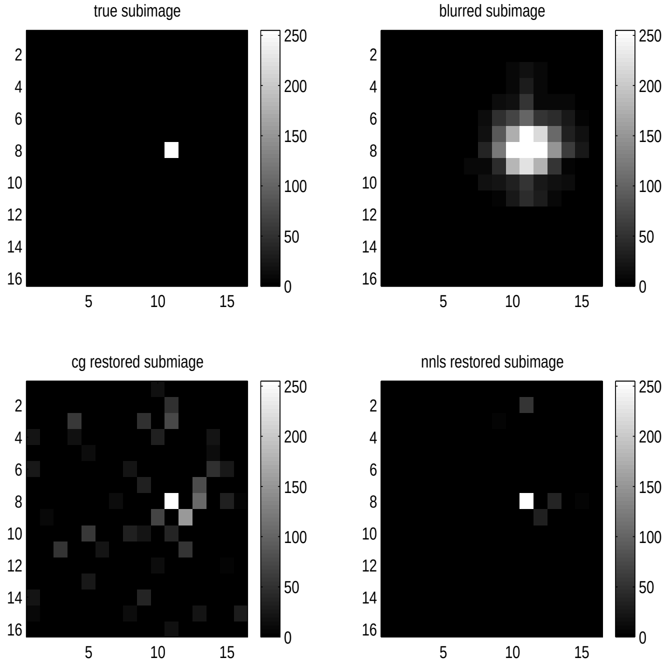


FIG. 4. *Subimages, using four PSFs for PCG.*

3. Confidence Intervals for Ill-Posed Problems. Now that we have the machinery to extract subproblems involving the reconstruction of subimages, we develop an algorithm to construct and display confidence intervals for the pixels of the subproblem

$$K_s x_s \approx y_s ,$$

where the right-hand side contains error, and where we know that the values x_s satisfy $x_s \geq 0$.

3.1. Computing Confidence Images. Computing confidence intervals for unconstrained least squares problems $K_s x_s \approx y_s$ is a standard problem in statistics. It can be done either without a distribution assumption on the unknown errors, or by assuming a normal distribution.

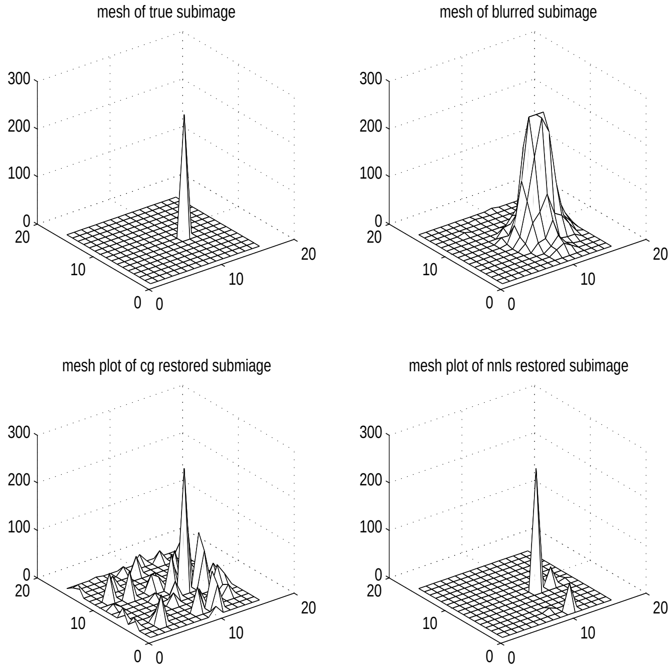


FIG. 5. *Mesh plot of subimages, using four PSFs for PCG.*

Unfortunately, confidence intervals computed in the usual way are often so pessimistic that they have no value. The trouble is that ill-conditioning in the matrix K causes them to be quite wide, since they are not constrained by side information such as bounds on the pixel values.

When there are upper and lower bounds x_{up} and x_{lo} on the values, the problem is somewhat more difficult computationally. However, the inclusion of bound constraints in the computation provides a natural regularization procedure [2], and therefore often produces much better results, especially for severely ill-conditioned problems. We make use of the following result:

THEOREM 3.1. *Assume that the noise η is normally distributed with mean zero, standard deviation S. Then the probability that x_i, a component of x, is contained in the interval $[\ell_i, u_i]$ is greater than or equal to α, where*

$$\begin{aligned}\alpha &= \int_0^{\gamma^2} \chi^2(\rho) d\rho\,, \\ \beta &= \min_x \|Kx - y\|_S^2\,, \\ \mu^2 &= \beta + \gamma^2\,, \\ \ell_i &= \min\{x_i \,:\, \|Kx-y\|_S \le \mu,\, x_{lo} \le x \le x_{up}\}\,, \\ u_i &= \max\{x_i \,:\, \|Kx-y\|_S \le \mu,\, x_{lo} \le x \le x_{up}\}\,,\end{aligned}$$

where $\|z\|_S^2 = z^T S^{-2} z$ and χ^2 is the probability density function for the chi-squared distribution with the number of degrees of freedom equal to the rank of K.

Proof. This result is a minor extension of a theorem of O'Leary and Rust [10, 11] for problems with nonnegativity constraints, and its proof requires changing $x \geq 0$ in their proof to $x_{lo} \leq x \leq x_{up}$ everywhere it appears. □

These confidence intervals also have *joint* probability α. Therefore, confidence intervals for p pixel values can be computed. Each confidence interval is computed by root finding, where function evaluation is the solution of a quadratic programming problem. This is expensive, but possible for modestly-sized subimages.

There is a similar result for non-normal noise, but the resulting confidence intervals are typically wider. Again the result is a minor extension of a theorem of O'Leary and Rust [11].

3.2. Displaying Confidence Images. Suppose that for each pixel in an image, we compute an estimate x_i of its true value as well as a confidence interval $[\ell_i, u_i]$ which is guaranteed with $\alpha\%$ certainty. In other words, if we repeated the data gathering 100 times, we would expect that α% of the images would have each pixel value within its confidence interval. The collection of confidence intervals forms the *confidence image*. We are left with the task of displaying a confidence image in some useful way.

To do this, we developed an algorithm called Twinkle_image. We generate a sequence of images, each with pixel values contained in the ranges defined by the confidence image. Thus, in each of these images, pixel value i is taken to be a random value chosen from the interval $[\ell_i, u_i]$. We display this sequence of images as a movie, running the frames at a rate so that the change in frame is barely perceptible to the viewer. By comparing the movie with the image x, the viewer can conclude with $\alpha\%$ confidence that features that persist in the frames of the movie are real. Those that appear to flicker (or "twinkle") could be either real or artifact.

We show three examples for illustration. We include frames from the movie of our algorithm as figures; the movies can be viewed at
`http://www.mathcs.emory.edu/~nagy/Twinkle/`.

Jorge E. Pinzón (private communication) tells us that he is using a similar technique to display uncertainties in the characteristics of ground cover, estimated by satellite images at NASA Goddard.

3.2.1. An Illustrative Example. Figure 6 is an example computed using Matlab. Suppose that medical imaging has produced this image. A physician might reasonably conclude from this data that there are three masses, and a treatment option appropriate to this conclusion would be chosen.

If the confidence intervals are such that the images in the left side of Figure 7 are typical of those within the 95% confidence level, then this conclusion would be justified. The movie of such images has barely perceptible flicker, and the three masses are persistent features. But if there is more noise in the measurements, or more uncertainty in the image reconstruction, then perhaps the images on the right side of Figure 7 are typical. In this case, the existence of three masses is far from certain. As the physician views this movie of confidence images, at least two suspected masses flicker, sometimes distinctly visible, but other times disappearing, and this would demonstrate the uncertainty in their existence.

In both scenarios, the clinician can make better decisions if presented with confidence images rather than just the reconstructed image of Figure 6.

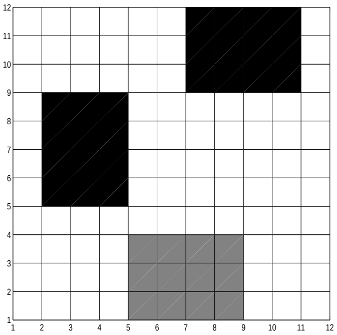


FIG. 6. *A model medical image*

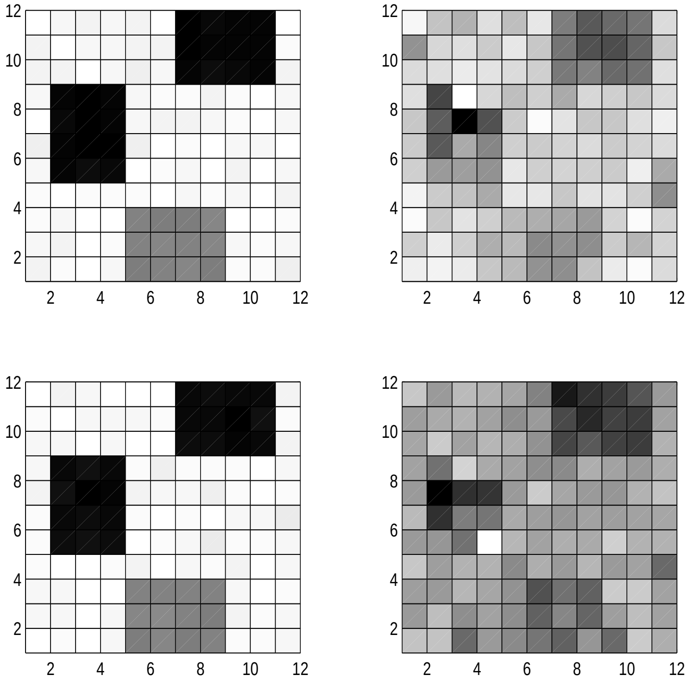


FIG. 7. *Two low-noise reconstructions of the medical image (left) and two high-noise reconstructions (right).*

3.2.2. Star Intensity Example. We took the subimage used in Section 2.3 and computed confidence intervals for it, assuming that the error was normally distributed with mean zero and standard deviation one, and that pixel values are constrained to be nonnegative. Because the pixels surrounding the subimage were computed to be black, the errors in these border values have negligible contributions to the error distribution. We chose a value of γ corresponding to a 99.99% confidence interval.

The data is shown in Figure 8 and the result of such a reconstruction is shown in Figure 9. Bright stars can oversaturate an image, washing out dimmer stars and other objects, so we display images of the actual pixel values along with images of the log of the pixel values. The top two images show the true subimage, the second two are the reconstructed one, while the remaining are 99.99% confidence images, where the noise has been assumed to be normally distributed with mean zero and standard deviation one. Although the images containing the log pixel values show some change in the confidence images, the confidence images of the actual pixel values are virtually indistinguishable from the reconstructed image, giving high confidence in the reconstruction of the star and enabling estimates of derived quantities such as star intensity, as well as error estimates for these quantities.

The full movie produced by the Twinkle algorithm can be viewed at http://www.mathcs.emory.edu/~nagy/Twinkle/.

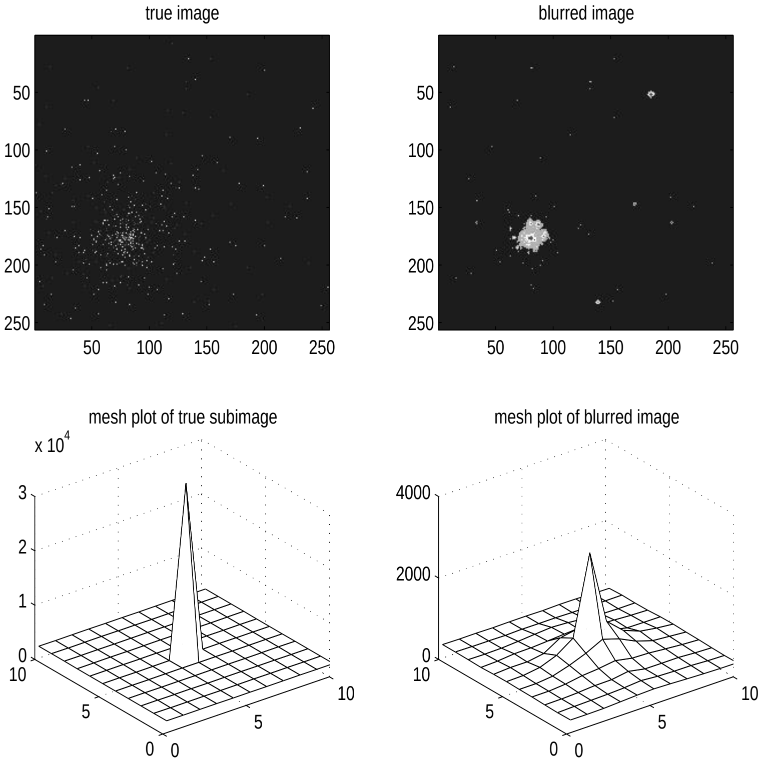


FIG. 8. *True star image and blurred image, with subimage extracted from the upper right corner.*

3.2.3. A Motion Blur Example. Cameras are used as a means of law enforcement by capturing the license plates of cars that fail to obey speed limits or traffic signals. The images acquired may be contaminated by motion blur, so we investigate the uncertainties involved in processing such images.

We used a digital camera to image a stationary white van. If the van were moving directly away from the camera, then the motion blur would be in the vertical direction in the image. We applied such a blur, spreading each pixel to the adjacent 19 pixels, and then added a noise image with each component chosen from a normal distribution and with the norm of the noise equal to 6.0×10^{-5} times the norm of the image. We then cropped the image to size 36×66, to isolate the license plate. Each column of pixels gives an independent reconstruction problem of rank 36. The results are in Figure 10. The original and blurred images are in the top row. The second row contains the reconstructed images, using the assumptions of nonnegativity of pixel values and bounds of $[0, 1]$ respectively. The tighter bounds give much better reconstructions. The third row contains two confidence images for

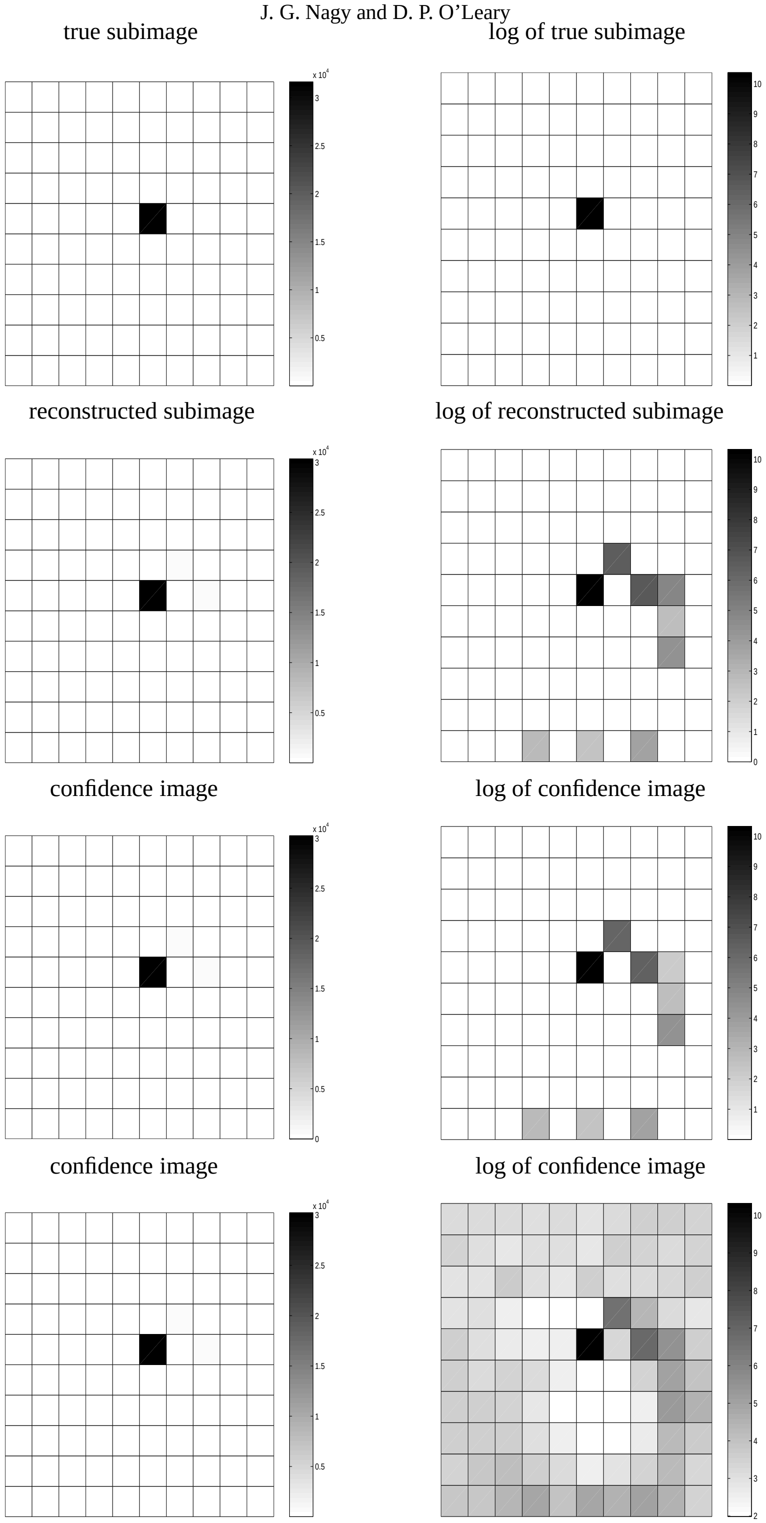
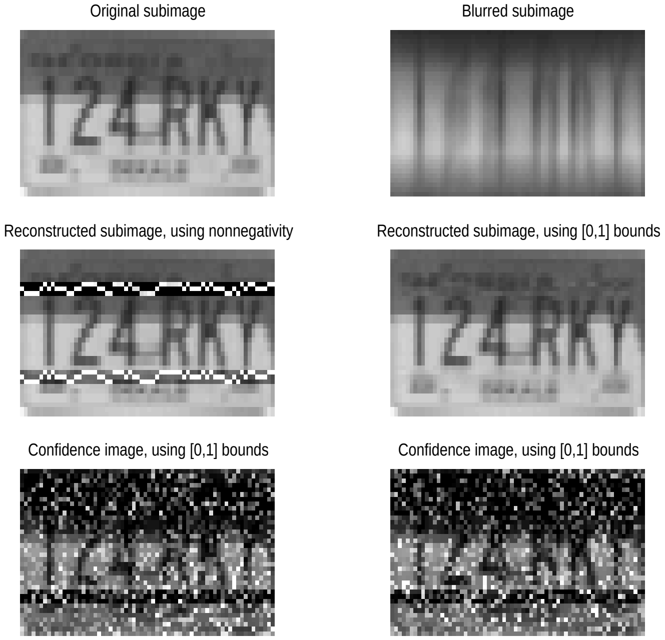


FIG. 9. *The true and reconstructed star subimages, with two confidence images. Pictures on the left show the actual pixel values, and images on the right show the log of the pixel values.*

the upper and lower bound reconstruction. Although the reconstructed image is quite clear, the confidence images are not as definitive. In this case, the pixels surrounding the subimage are not known exactly, so the errors in these border values contribute to the error distribution, although this additional error has been neglected in the computation of confidence intervals. Our boundary conditions were quite crude: we implicitly assume that the pixel values in the 18 rows below this subimage are black. If we allow these values to vary, by increasing the size of the subimage, then the Chi-squared parameter must be increased because the rank of the matrix increases, and the bounds worsen.

The full movie produced by the Twinkle algorithm on the license plate can be viewed at http://www.mathcs.emory.edu/~nagy/Twinkle/.



Original subimage

Blurred subimage

Reconstructed subimage, using nonnegativity

Reconstructed subimage, using [0,1] bounds

Confidence image, using [0,1] bounds

Confidence image, using [0,1] bounds

FIG. 10. *The images from the license plate example.*

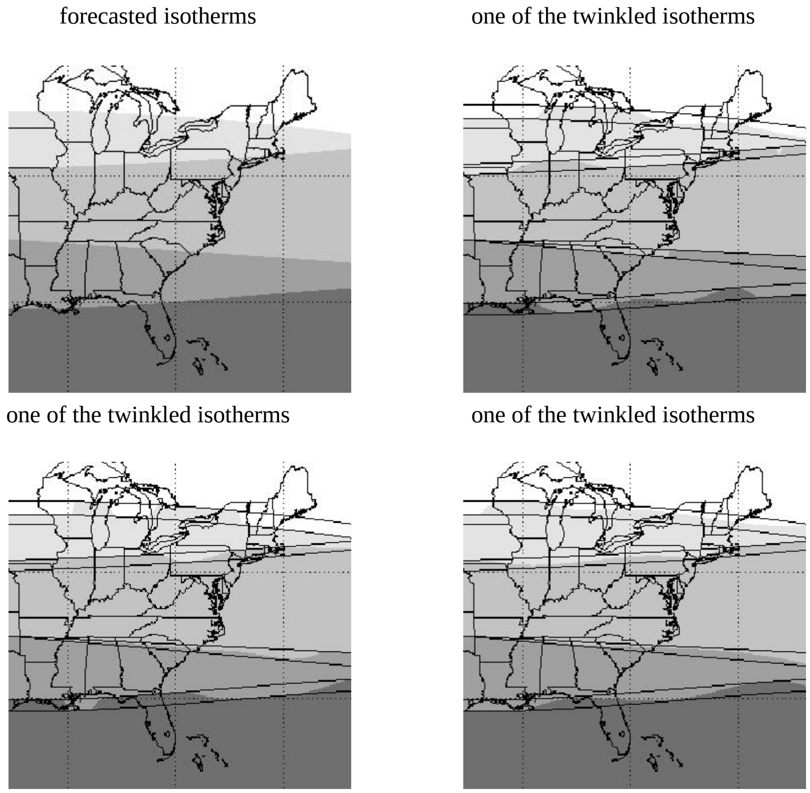
3.3. Displaying Confidence Images Based on Feature Estimates.

The previous version of Twinkle is useful when we have confidence intervals on the pixel values in the image. In some cases, though, it is the *features* in the image that have uncertainty. For example, a fluid dynamics computation might produce flow lines that have uncertainty. Or a topographical map or estimate of a function might have uncertainties in the locations of contour lines.

In such cases, we apply a variant of the algorithm, Twinkle_feature. Each feature is represented by its distribution. For instance, we might know that the location of a contour has some expected value, some variance, and that the uncertainty is normally distributed. Twinkle_feature would then generate a sequence of contours chosen from this distribution and display them as a movie.

3.3.1. A Weather Forecasting Example.

Weather forecasts of such things as high/low pressure regions, temperatures, precipitation, etc., are often displayed as isopleths; that is,


forecasted isotherms

one of the twinkled isotherms

one of the twinkled isotherms

one of the twinkled isotherms

FIG. 11. *Forecasted temperatures (top left), and three images illustrating forecast uncertainties.*

lines connecting points of equal value of the particular variable in question. These isopleths are usually shown on maps as fixed lines which do not indicate any uncertainty in the forecast. An isotherm example is shown in the top left image of Figure 11. Uncertainty in temperature is not easy to visualize from this single image. Suppose, as an example, that the vertical uncertainty in the isotherms is

- 5 pixels in the top isotherm
- increasing from 0.2 to 5.6 pixels across the second isotherm,
- 0.05 times the y-coordinate of the third isotherm, and
- 6 vertical pixels in the bottom isotherm.

We display the bounds for the isotherms as solid lines. By generating sample locations of these isotherms within these bounds, we obtain pictures like the other three images of Figure 11. Displaying them as a movie gives the viewer a much clearer idea of what will be the expected temperature. Note, for example, that the uncertainty in northern Florida is nicely illustrated by the movie samples.

4. Conclusions. We have presented techniques useful in obtaining more accurate reconstruction of subimages, and in providing more information about the reconstruction, such as confidence images. We have also demonstrated a useful tool, Twinkle, for visualizing statistical confidence that we can have in pixel values or in the features they form. The technique is useful whether the confidence information is in the form of a confidence interval or a distribution of possible values.

The major obstacle in constructing valid confidence intervals for images is handling the boundary conditions in a statistically valid way. We emphasize that this can be easily done for some images (e.g., the astronomical images) but not for all.

REFERENCES

[1] T. F. CHAN AND J. A. OLKIN, *Preconditioners for Toeplitz-block matrices*, Numer. Algorithms, 6 (1993), pp. 89–101.

[2] L. ELDÉN, *Algorithms for the regularization of ill-conditioned least squares problems*, BIT, 17 (1977), pp. 134–145.

[3] D. A. FISH, J. GROCHMALICKI, AND E. R. PIKE, *Scanning singular-value-decomposition method for restoration of images with space-variant blur*, J. Opt. Soc. Amer. A, 13 (1996) pp. 1–6.

[4] M. HANKE, J. G. NAGY, AND R. J. PLEMMONS, *Preconditioned iterative regularization*, in Numerical Linear Algebra, L. Reichel, A. Ruttan, and R. S. Varga, eds., de Gruyter, Berlin, 1993, pp. 141–163.

[5] P. C. HANSEN, *Local deblurring with PP-TSVD*, in Seventh SIAM Conference on Applied Linear Algebra, Raleigh, NC, October 2000. Available on line from http://www.imm.dtu.dk/˜pch/Talks/index.html.

[6] J. KAMM AND J. G. NAGY, *Optimal Kronecker product approximation of block Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 22 (2000), no. 1, pp. 155–172.

[7] J. G. NAGY AND D. P. O'LEARY, *Fast iterative image restoration with a spatially varying PSF* in Advanced Signal Processing Algorithms, Architectures, and Implementations VII, vol 3162, F. T. Luk, ed., SPIE, 1997, pp. 388–399.

[8] J. G. NAGY AND D. P. O'LEARY, *Restoring images degraded by spatially-variant blur*, SIAM J. Sci. Comput., 19 (1998), no. 4, pp. 1063–1082.

[9] M. K. NG, R. H. CHAN, AND W.-C. TANG, *A fast algorithm for deblurring models with Neuman boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–886.

[10] D. P. O'LEARY AND B. W. RUST, *Confidence intervals for inequality-constrained least squares problems, with applications to ill-posed problems*, SIAM J. Sci. Statist. Comput., 7 (1986), no. 2, pp. 473–489.

[11] B. W. RUST AND D. P. O'LEARY, *Confidence intervals for discrete approximations to ill-posed problems*, J. Comput. Graph. Statist., 3 (1994), no. 1, pp. 67–96.

[12] H. J. TRUSSELL AND B. R. HUNT, *Image restoration of space-variant blurs by sectional methods*, IEEE Trans. Acoust. Speech, Signal Process., 26 (1978), pp. 608–609.