A New Algorithm for Image Noise Reduction using Mathematical Morphology *

Richard Alan Peters II
Department of Electrical Engineering
Vanderbilt University
Nashville, TN 37235
(615) 322-7924
rap2@vuse.vanderbilt.edu

IEEE Transactions on Image Processing
Volume 4, Number 3, pp. 554-568, May 1995

Abstract

Morphological openings and closings are useful for the smoothing of grayscale images. However, their use for image noise reduction is limited by their tendency to remove important, thin features from an image along with the noise. This paper is a description and analysis of a new morphological image cleaning algorithm (MIC) that preserves thin features while removing noise. MIC is useful for grayscale images corrupted by dense, low-amplitude, random or patterned noise. Such noise is typical of scanned or still-video images. MIC differs from previous morphological noise filters in that it manipulates residual images – the differences between the original image and morphologically smoothed versions. It calculates residuals on a number of different scales via a morphological size distribution. It discards regions in the various residuals that it judges to contain noise. MIC creates a cleaned image by recombining the processed residual images with a smoothed version. This paper describes the MIC algorithm in detail, discusses the effects of parametric variations, presents the results of a noise analysis and shows a number of examples of its use, including the removal of scanner noise. The paper also demonstrates that MIC significantly improves the JPEG compression of a grayscale image.

1 Introduction

Image enhancement through noise reduction (also called image cleaning) is a fundamental problem in image processing. As such, it has been studied for many years [1, 2]. Most general texts in image processing include chapters on it [16, 20, 8, 5]. Noise reduction is an image restoration problem in that it attempts to recover an underlying perfect image from a degraded copy. This problem is intractable unless one makes assumptions about the actual structure of the perfect image. Various noise reduction techniques make various assumptions depending on the type of imagery and the goals of the restoration. Consequently, one technique is not useful for all applications, and many techniques have been devised.

This paper is a description and analysis of a morphological image-cleaning algorithm, the MIC algorithm. MIC was designed primarily to enhance scanned images and still-video images. The digitization processes introduce noise, usually of low amplitude, which nevertheless can degrade significantly the appearance of the imagery. Moreover, the noise can reduce the efficiency of image compression schemes. Transform coding schemes, notably JPEG, [26] can use quite a lot of storage space to code the noise in an image. That is, a noisy version of a clean image compresses to a larger size than the clean version. MIC removes image noise; this improves the subjective appearance of scanned and still-video imagery and improves the performance of JPEG compression.

*This work was supported by the US Air Force Office of Scientific Research Grant No. F49620-88C-0053

1.1 Goal of the MIC Algorithm

The goal of "cleaning up" images so that they look better to most people is admittedly subjective. Nevertheless, images that most people regard as "clean" possess two common characteristics

(1) Edges, thin lines, and small features are sharp and clear.

(2) Areas between these features are smoothly varying.

These two assumptions are fundamental to many image restoration techniques and image models. Hence they are the goals of many such procedures, including MIC.

Many scanned images and still-video images look less than perfect because visible noise prevents the image from having characteristic (2). Yet in a noisy image (providing the noise is not of very high amplitude) a dark or bright line often appears to be relatively noise free. Presumably, this is because of the characteristics of human perception and not because the noise stops on the line. Perhaps edges, thin lines, and small features are less affected perceptually by the noise because these features are effectively one dimensional. They have small area or no area at all. Noise, however, may require *area* to be perceptually significant. Whether or not this is true under all circumstances, experience has shown it to be so with common digital imagery containing visible noise of relatively low amplitude. (High amplitude noise can break up thin lines, displace edges, and mask small features.) The technique of this paper exploits the idea that the perception of noise in images is largely restricted to 2D features. The quality of the results supports the hypothesis.

1.2 Previous Work

Many techniques for noise reduction replace each pixel with some function of the pixel's neighborhood. Because 1D features and 2D noise usually have common frequency components, they are not separable in the frequency domain. Hence, linear filters seldom can meet goals 1 and 2 simultaneously. Linear filters tend either to amplify the noise along with the 1D features, or to smooth out the noise and blur the 1D features.

To minimize the conflict between goals 1 and 2 above, researchers have introduced a number of adaptive noise reduction algorithms [27, 15, 19]. Essentially, these attempt to detect non-noise edges in the image. The algorithms modify their behavior near the edges in order to preserve them. Edge detection is, however, a difficult problem in its own right. Enhancement algorithms that depend on edge detection incorporate all the problems associated with edge detection – undetected edges, spurious edges, misplaced edges, etc. Nevertheless, the idea of adapting the smoothing procedure near 1D features so that the procedure does not blur them could, in theory, lead to an algorithm that meets both goals.

Morphological filters are, perhaps, the most well-known nonlinear filters for image enhancement [6, 4, 11, 12, 14, 21, 22, 23]. These include erosions, dilations, openings, closings, and rank filters including the median filter. The action of a morphological filter depends on its structuring element, a small, quasi-image that defines the operational neighborhood of a pixel. The median filter is very good at removing some types of noise (notably shot noise or "salt and pepper" noise) while preserving some edges (perfect step edges). It is not so good, however, at removing dense noise, and it degrades thin lines and small features (smaller than half the area of its structuring element). In [3], Bovik provides a detailed analysis of the artifacts introduced by median filters.

Sternberg [25] introduced the idea of image noise reduction through iterative application of openings and closings with successively larger structuring elements. This technique, called an "alternating sequential filter" (ASF) by Serra [23], is good for recovering some approximation of a structure that is nearly invisible in dense, high amplitude noise. It is inherently incapable of restoring image structures that are thinner than the largest structuring element used.

Song and Delp [24] have devised a technique which they call the "generalized morphological filter". It takes linear combinations of the results of openings and closings with multiple structuring elements. This filter works well in the presence of impulsive noise. However, in the presence of dense noise there is a trade-off between noise smoothing and detail preservation. Moreover, there is no systematic approach for the exact specification of the structuring elements. Good results in different images probably depend on the choice of structuring elements.

The ASF and the Song-Delp algorithms remove noise from images because openings and closings annihilate image features whose support (i.e., area in the image) does not cover the structuring element. That

is, they eliminate small features and thin features. On the other hand, these operations do preserve features that can contain the structuring element. Since features partition an image, opening and closing preserve those edges which are the boundaries between sufficiently large features. Edges that are boundaries between regions are preserved if they have zero width. That is, they are the visual result of one smooth region abutting another. The boundary itself has no support. The 1D features which are annihilated by opening and closing are thin features that do have support in the image, such as a dark line across a bright region.

Linear filters always affect edges, whether boundaries or lines. In particular, smoothing filters always blur them. Thus in processing boundaries, morphological filters – in partricular, openings and closings – are superior to linear filters.

Because of their actions, the ASF and the Song-Delp obliterate small or thin features along with the noise. If there were some way to adapt a morphological smoothing filter so that it would not eliminate these features then it could, theoretically, meet both the goals for image cleaning. This idea, together with the observation that 1D objects appear relatively insensitive to low levels of noise inspired the MIC algorithm.

1.3 Outline

A rough sketch of the MIC algorithm follows: Consider a noisy grayscale image I. Let S be the result of smoothing I with openings and closings. Assume S is noise free. Then the difference image, $D = I - S$ contains all the noise in I. But, S cannot contain any features with nonzero support that are thinner than the structuring elements used to create it. Thus, D contains features as well as noise. If the noise in I has a smaller dynamic range than the thin features, then D will contain noise at lower amplitude levels and features at higher amplitudes. Experience with many scanned and still-video images has demonstrated that this is often the case (See section 4). If D is thresholded (actually, center-clipped since D is a signed image) at a value greater than the amplitude of the noise, the result is a support map or mask of the thin features in the image. The mask after further manipulation can be used to recombine the thin features in D with S while leaving the noise behind. The result is an image that is smoothly varying except for edges, thin lines and small spots. The algorithm described below is an elaboration of this idea using a morphological size distribution to isolate features from noise on different scales.

Section 2 is a detailed description of the new morphological image cleaning program. Following that, in section 3, is a discussion of the free parameters in the algorithm, including recommended values and an analysis of the effect of variations in the parameter values. Section 4 compares the new algorithm with the result of median filtering and the result of the Song-Delp algorithm. The section also has an example of how the algorithm can be used to remove patterned noise from a poorly scanned image. Section 4 concludes with results from example images. Section 5 is a statistical analysis of the noise reduction capabilities of the algorithm. Section 6 demonstrates that the algorithm leads to improved JPEG compression.

2 The MIC Algorithm

The fundamental idea behind the morphological image-cleaning algorithm is to segment into features and noise, the residual image that is the difference between an original image and a smoothed version. The features from the residual are added back to the smoothed image. Ideally, this results in an image whose edges and other one dimensional features are as sharp as the original yet has smooth regions between them. Figures 1 and 2 demonstrate that this can be quite effective.

Figure 1 is the 256 graylevel luminance image extracted from a 256 color *gif* format image, "elle.gif," taken from the alt.binaries.pictures newsgroup of USENET during the summer of 1991. The image is of relatively high quality. But like many scanned images, elle.gif is corrupted with low level, visible noise. This image of model Elle McPherson is a good test image for MIC because of its many thin features, most notably the hair on the model's face and the specular highlights on her skin. These features, which are very important to the perceived quality of the image, are either blurred or lost completely by many other noise reduction algorithms. Figure 2 is the result of processing this image. Note that few of the thin features are lost and that the smoothly varying regions are largely noise free.

Because morphological openings and closings do not blur some edge-like features, the algorithm uses them to smooth the image. Morphology also provides the tools for differentiating between noise and features

in the residual. It must be acknowledged that such differentiation by MIC seldom will match exactly the subjective judgement of a human observer. In fact, different observers will have different opinions about which pixels constitute a feature, and which, noise. All the problems associated with image segmentation can (and do under certain circumstances) affect this step. However, MIC performs well under the conditions for which it was designed – for images corrupted by visible, yet relatively low levels of spatially dense noise, Section 5 will suggest that the technique is suited for images where the standard deviation of the noise is less than half the standard deviation of the features.

2.1 Morphological Smoothing of Images

Let I represent an image. Let I_Z represent the opening of the image with structuring element (SE) Z. Let I^Z represent the closing of the image with Z. Let

$$\mathrm{OCCO}(I;Z) = \frac{1}{2}(I_Z)^Z + \frac{1}{2}(I^Z)_Z \tag{1}$$

That is, the OCCO filter is the pixelwise average of the open-close of I and the close-open of I. (See [11] for properties of open-close and close-open). $\mathrm{OCCO}(I;Z)$ is approximately smooth with respect to the SE and, unlike its constituent operators, unbiased with respect to graylevel.

MIC uses an OCCO filter for smoothing. The algorithm has been designed and tested using only SE's with disk-shaped support. Thus, MIC's OCCO filter has two parameters associated with its structuring element, its size (i.e. diameter) d, and the choice of whether to use a flat or a curved top (i.e. function-set processing or function-function processing as described in [11, 12]. Section 3 discusses choice the flat *vs.* curved SE. It suggests that slightly smoother results can be had using the curved SE.

The choice of SE size is of fundamental importance. In fact, no single size will do. It is necessary to filter the image with a range of sizes and operate on the various residuals separately.

It appears to be a common characteristic of the images for which this algorithm was designed that the visible noise is size limited. That is, if one smooths an image with successively larger OCCO filters, one of the smoothed images, for some SE diameter d, appears to be noise free whereas its predecessor (with a smaller SE diameter) shows evidence of noise. By "noise free" it is meant that large, constant or smoothly varying regions of the smoothed image show no small perturbations from the overall graylevel trend of the region. Granted, this is subjective. However, the goal here is to improve the subjective appearance of an image. Hence for this purpose, if an image appears to be noise free, then it is. The point is: a sufficiently large SE appears to filter all the noise out of the image.

Thus, for a sufficently large diameter SE, the residual difference image contains all the noise as well as all the features that were too small to be approximated by the structuring element. Under the conditions set out in section 1.3, to preserve the small features it is necessary only to segment the residual into features and noise. However, for most images, the SE diameter d needed to smooth the image, is so large that the residual contains much of the important feature information. The more features there are in the residual, the more difficult it is to segment. For a large enough structuring element, the residual is the original image itself!

In the course of the research leading to the MIC algorithm, th following was observed: In the residual, I minus a small diameter OCCO filtered image, the noise regions and feature regions are clearly identifiable and easily separable by the technique to be described below. But, the amount of noise present in the small SE OCCO image causes the resultant image to remain noisy. This noise is visible in smoother regions of the image. Thus, if d is large it is difficult to segment the residual, and if d is small the algorithm fails to clean the image.

The work-around is to generalize the procedure. Smooth the image with several OCCO filters of different diameters. The result is a morphological size distribution [21, 13]. Let S_j be the result of smoothing I with an OCCO filter with SE diameter d_j. Let $S_0 = I$. Assume $d_1 < d_2 < \cdots < d_k$ and that $d_0 = 1$ since the OCCO of I with $d_0 = 1$ is I itself. Then,

$$D_j = S_{j-1} - S_j \tag{2}$$

is the jth residual image. D_j has, for the most part, no features larger than d_j and none smaller than d_{j-1}. That is, none of the features in D_j could cover the SE with diameter d_j; they all can cover the SE with

diameter d_{j-1}. (Some features that violate this are possible because in general OCCO(I) is not both opened and closed [4].) Each D_j is an intermediate residual image that constitutes a "size-band."

2.2 Processing of the Residuals

Each of the residuals D_j is assumed to contain both features and noise. The qualitative difference between D_j and S_{j-1} is that D_j is a "flattened out" S_{j-1}. The large amplitude regions of D_j indicate large variations in S_{j-1} from the behavior of S_j. Such regions are undoubtedly of more visual significance than the small amplitude regions. If the amplitude standard deviation of the noise in I is smaller than that of the features then it is reasonable to assume that this is true within the various size-bands. It is reasonable for white noise since its spatial frequency characteristic is flat. (The assumption may not be true across all size-bands for images corrupted with patterned noise; in this case one size-band may be dominated by the noise, masking all features. But this is an advantage when it comes to removing patterned noise.) If the noise standard deviation is smaller than the amplitudes of the most important features on a given scale, it is possible to remove the noise from D_j by thresholding.

Because residual D_j is the difference between two OCCO filter outputs, it is a signed image. The zero regions of D_j indicate areas where S_{j-1} and S_j coincide. The positive regions of D_j are the areas in S_{j-1} that are brighter than S_j. The negative regions are dark areas in S_{j-1} with respect to D_j. The light features may be perceptually more important than the dark features or vice-versa. For example, in figure 1, the hair on the models face, the dark features, are dominant. Thus, residual processing is best handled in two separate, light and dark bands. Call the positive residual the "tophat" image, T_j (because that is the term normally used for the difference between an image and its opening by a flat topped structuring element).

$$T_j(x,y) = \begin{cases} D_j(x,y) & \text{if} \quad D_j(x,y) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Call the negative residual, B_j, the "bothat" image.

$$B_j(x,y) = \begin{cases} -D_j(x,y) & \text{if} \quad D_j(x,y) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Note that B_j is a nonnegative image. MIC performs the same operation on both T_j and B_j using (possibly) different parameters.

Let R_j represent either T_j or B_j. The goal of the processing of R_j is to segment it into feature regions and noise regions and then to discard the noise. Because of the assumptions about the characteristics of the noise and the important features in the image, and because of the way that R_j is constructed, the first step in the segmentation of R_j is to threshold it. Automatic threshold selection is an art in itself (See, for example [28, 29, 7]). Like many other algorithms, MIC computes thresholds from $p_{R_j}(n)$, the graylevel probability distribution of R_j. Let m_{R_j} be the square root of the second moment of $p_{R_j}(n)$

$$m_{R_j} = \sqrt{\sum_{n=0}^{n=255} n^2 p_{R_j}(n)} \tag{5}$$

Experimentation has shown that thresholds $t = f \cdot m_{R_j}$ where $f \in [1,2]$ yield the best results for most images. The value of f is a free parameter in the MIC algorithm which defaults to one. See the discussion of this in section 3.

Let X_{R_j} be an image and t a threshold such that

$$X_{R_j}(x,y) = \begin{cases} 1 & \text{if} \quad R_j(x,y) \geq t \\ 0 & \text{if} \quad R_j(x,y) < t \end{cases} \tag{6}$$

Then X_{R_j} is the support map of those pixels in R_j that are equal to or greater than the threshold.

In general X_{R_j} will contain many isolated ones (ones surrounded by zeros in a 3×3 neighborhood) and many regions of only two adjoining ones. Isolated one or two pixel regions are more likely to be seen by a person as noise than features. Therefore the algorithm removes them. It does this as follows: First it applies

a rank filter to X_{R_j} to find the locations of all 3×3 neighborhoods with 3 or more ones. Then it dilates the location map with a 3×3 square of ones and forms the logical AND of the result with X_{R_j}. The result of this is a support map of the pixels in R_j that are at least as large as the threshold and are near at least two other such pixels in a 3×3 neighborhood. There can be isolated pixels in this map. Consider the following 5×5 region:

$$\begin{array}{ccccc} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{array}$$

The 3×3 neighborhoods in the each of the corners has four ones in it, yet the one in the center of the 5×5 region is isolated. Such isolated pixels are removed with a simple deletion operation. The isolated delete can cause some neighborhoods to have less than three ones. Thus it is necessary to iterate the rank filtering and isolated deletions until there are no more isolated pixels. The minimum number of support pixels (i.e. 1's) inside each 3×3 neighborhood is another free parameter of the algorithm. It defaults to three pixels. See section 3 for a discussion of the effects of variations in the parameters.

At this point, it would be possible to use X_{R_j} to select those pixels from R_j which correspond, presumably, to important features. These features could be added back to S_j to restore some of its detail. However, thresholding R_j trims off the "bases" of large features. When the features are added back to S_j, the trimming introduces distortion that was not present in the original image. Figure 3 is a 1D example of this. (It uses opening and the tophat to simplify the demonstration of the problem. The distortion with the OCCO transform is analogous.)

(a) is the original 1D image.

(b) and (c) show the opening of (a) by a flat topped structuring element.

(b) shows the opening and the original superimposed.

(c) is the opening alone.

(d) is the tophat image showing the threshold (dotted line).

(e) is the thresholded tophat.

(f) shows the result of adding (e) directly to (c).

Notice the distortion of the large peak; essentially the middle of it has been removed. A way around this problem is to grow back the bases of those peaks which survived the thresholding procedure.

(g) is the thresholded tophat with the bases of the peaks restored.

(h) shows the sum of (g) and (c).

Notice that the distortion of (f) is not present in (h). The large features are complete.

To recover the bases of the features, MIC expands the nonzero regions of X_{R_j} through the processing of their morphological skeletons [10, 23]. The skeleton of a region is its medial axis. The procedure uses the skeleton algorithm in [4] (pp. 120-125). It dilates the skeleton with a binary structuring element of diameter d_j. This is a reasonable approach because of the characteristics of the size-band, D_j, from which X_{R_j} is computed. The largest feature in D_j must be smaller than the structuring element of radius d_j. Thus by dilating the skeleton with the structuring element of radius d_j, the algorithm finds the support of the largest possible area that could include both the peaks and their bases. This unavoidably reintroduces the support of some noise pixels. To complete the processing of the residual, the algorithm zeros any pixel in R_j whose support is not in the dilated skeleton.

Finally the algorithm recombines the information in the cleaned up residuals with the smooth image. Assume that the algorithm computed smooth images $S_1, \ldots, S_k$ using structuring elements with diameters $d_1, \ldots, d_k$ such that $d_1 < \cdots < d_k$. Assume it removed the noise from tophats, $T_1, \ldots, T_k$ and from bothats $B_1, \ldots, B_k$. The most coarsely smoothed image, namely S_k is the base image in the reconstruction. The

bright features are put back in S_k by adding to it the sum of all the cleaned-up tophats. The dark features are put back in S_k by subtracting from it the sum of all the cleaned-up bothats. That is

$$J = S_k + \sum_{j=1}^{k} \hat{T}_j - \sum_{j=1}^{k} \hat{B}_j \tag{7}$$

where J is the cleaned image, $\hat{T}_j$ is the jth noise cleaned tophat and $\hat{B}_j$ is the jth noise cleaned bothat. Note that if nothing were done to the original T_j's and B_j's, then $J \equiv I$.

2.3 Formal Statement of the Algorithm

The following is a formal statement of the image cleaning algorithm. It is divided into three parts; a main routine and two subroutines. The main routine controls the decomposition of the image into size-bands, creates the residuals, sends them off to be cleaned, and recombines the results. The first subroutine is the image smoothing procedure. The second is the residual processor.

Algorithm 1 Morphological Image Cleaning (MIC)

Let I be a grayscale image and let $\hat{I}$ be a cleaned version of I. Let $S = \mathrm{OCCO}(I; d)$ be the pixelwise average of the openclose of I and the closeopen of I where the morphological operations are performed with a structuring element with disk-shaped support of diameter d. $\mathrm{OCCO}(I; d)$ is computed by subroutine 1. Let T be the tophat of I with respect to S and let B be the bothat of I with respect to S. Let $\hat{T} = \mathrm{TOPBOT}(T; d, f, s)$ be the cleaned up tophat T, and let $\hat{B} = \mathrm{TOPBOT}(B; d, f, s)$ be the cleaned up tophat B. Parameter f is a factor by which to multiply the square root of the second moment when calculating the threshold, and s is the minimum area per 3×3 neighborhood of a region in the cleaned up residual. TOPBOT() *is computed by subroutine 2. Let T_{acc} be the accumulated cleaned tophat and let B_{acc} be the accumulated cleaned bothat. Let k be the number of size-bands to be computed. Let $\{d_1, \ldots, d_k\}$ be integers such that $d_1 < \cdots < d_k$. Let j be an index variable. All additions and subtractions of images are pixelwise. The usual choices of parameters are $d_j = 2^{j+1}+1$, $f = 1$, $s = 3$. Occasionally, it is useful to specify one value of f for the tophat and another for the bothat. The same applies to s.*

1. *initialize variables and images:* $j \leftarrow 0$; $T_{acc} \leftarrow 0$; $B_{acc} \leftarrow 0$;
2. $j \leftarrow j + 1$;
3. $S \leftarrow \mathrm{OCCO}(I; d_j)$;
4. *for all pixels* (x, y),
$$T(x,y) \leftarrow \begin{cases} I(x,y) - S(x,y) & \text{if} \quad I(x,y) - S(x,y) > 0 \\ 0 & \text{if} \quad I(x,y) - S(x,y) \le 0 \end{cases};$$
5. $\hat{T} \leftarrow \mathrm{TOPBOT}(T; d_j, f, s)$;
6. $T_{acc} \leftarrow T_{acc} + \hat{T}$;
7. *for all pixels* (x, y),
$$B(x,y) \leftarrow \begin{cases} S(x,y) - I(x,y) & \text{if} \quad S(x,y) - I(x,y) > 0 \\ 0 & \text{if} \quad S(x,y) - I(x,y) \le 0 \end{cases};$$
8. $\hat{B} \leftarrow \mathrm{TOPBOT}(B; d_j, f, s)$;
9. $B_{acc} \leftarrow B_{acc} + \hat{B}$;
10. $I \leftarrow S$;
11. *if* $j < k$ *goto 2*

12. $\hat{I} = T_{acc} + S - B_{acc}$;

Algorithm 2 OCCO(I, d)

Let I, C, O, CO, OC, and S represent images. Let OPEN(I, d) *and* CLOSE(I, d) *represent morphological opening and closing, respectively, using a a structuring element with disk-shaped support of diameter d.*

1. $O \leftarrow$ OPEN(I, d);
2. $OC \leftarrow$ CLOSE(O, d);
3. $C \leftarrow$ CLOSE(I, d);
4. $CO \leftarrow$ OPEN(C, d);
5. *for all pixels* (x, y), $S(x, y) = \frac{1}{2}[OC(x, y) + CO(x, y)]$;

Algorithm 3 TOPBOT(R, d, f, s)

Let R designate the residual image (either a tophat or a bothat). Let d be the diameter of the structuring element used to create the smooth image. Let histo*(R) be the graylevel histogram of R. Let $E\{n^2\}$ be the second moment of the graylevel probability distribution of R. Let* THRESH(R, t) *represent the thresholding of R at t as defined in eqn. (6). X_i is a binary image mask. Parameter f is a real number in the interval* $[1, 2]$*, and s is an integer in* $[1, \ldots, 9]$*.* RANK(X_i, s) *is the rank filter of order s for sets. (This returns a binary image X_{i+1} such that if $X_{i+1}(x, y) = 1$ then $X_i(x, y)$ is the center of a 3×3 neighborhood in X_i that contains at least s white pixels. If $X_{i+1}(x, y) = 0$ then the 3×3 neighborhood in X_i centered at (x,y) contains fewer than s pixels.)* DILATE(X_i, d) *is the dilation of (X_i) with a disk of radius d (when $d = 3$, the "disk" is a 3×3 square).* AND *is the pixelwise logical AND operator.* ISODEL(X_i) *removes isolated pixels.* SKELETON(X_i) *computes the morphological skeleton.*

1. $h \leftarrow$ histo*(R)*;
2. $m \leftarrow \sqrt{E\{n^2\}}$ *(calculated from h);*
3. $t \leftarrow f \cdot m$;
4. $X_0 \leftarrow$ THRESH(R, t);
5. $X_1 \leftarrow$ RANK(X_0, s)
6. $X_2 \leftarrow$ DILATE$(X_1, 3)$
7. $X_3 \leftarrow X_0$ AND X_2
8. $X_4 \leftarrow$ ISODEL(X_3)
9. *If there were deletions in the last step, let $X_0 \leftarrow X_4$ and GOTO 5.*
10. $X_5 \leftarrow$ SKELETON(X_4)
11. $X_6 \leftarrow$ DILATE(X_5, d)
12. $\hat{R} \leftarrow R$ AND X_6

Figure 4 is a block diagram of the main algorithm loop. Figure 5 has diagrams of the two subroutines.

2.4 Effects of the Algorithm on an Example Image

Recall that Figure 1, "elle.gif" is luminance band from a 256 color *gif* format image that is, presumably, a scanned magazine picture. The image exhibits the dense, low variance noise typical of scanned images. Figure 2. is the result of cleaning "elle.gif" with the following parameters: $d_1 = 5$, $d_2 = 9$, $d_3 = 17$; for the tophats $f = 1.5$ and $s = 4$; for the bothats $f = 1.1$ and $s = 3$. These values were arrived at through experimentation.

Figure 6 (a) shows an enlarged section of the face from "elle.gif". The remaining 5 panels are OCCO smoothed versions of (a). They have been smoothed with flat-topped disk-shaped structuring elements with diameters (number of pixels across) (b) 3, (c) 5, (d) 9, (e) 17, (f) 33. These are the outputs of the OCCO subroutine. Note that (d) appears to be mostly noise free whereas (c) appears to be noisy. Panels (e) and (f) appear to be completely noise free. Notice how the fine features progressively disappear in smoother images.

Figure 7 shows the residuals for figure 6. Panel (f) in fig. 7 is the difference between panels (e) and (f) of fig. 6. Panel (e) in fig. 7 is the difference between panels (d) and (e) of fig. 6, and so on through panel (b) in fig. 7 which is the difference between panels (a) and (b) in fig. 6. Panel (a) of fig. 7 is the sum of residuals (b) through (f). All of the residuals in fig. 7 have been scaled so that zero is mapped to medium gray (8-bit pixel level 128). The tophats are the bright regions; the bothats, the dark regions. In this figure, the tophats and bothats have been expanded into a full range of graylevels to make them easier to see. The differences between important visual features and noise are especially apparent in panels (b) and (c). Panel (a), the sum of all the residuals, is also the difference between the original image and the most highly smoothed image (fig. 6(f)). Notice that segmentation of this image (fig. 7(a)) into features and noise would not be trivial since much of the noise is superimposed on important 2D features. Note in particular the cheek areas. There is important graylevel curvature in panel (a). Compare this with panel (b) where much of the noise is present but there is little curvature.

Figure 8 shows the cleaned up residuals. Panels (b) through (f) are the results of applying TOPBOT to the respective panels of fig. 7. A comparison between figures 7 and 8 clearly shows what MIC considers to be features and noise. Panel (a) of figure 8 is the cumulative cleaned up residual (i.e. the sum of panels (b) through (f)). The sum of fig. 8 (a) and fig. 6(f) is the cleaned image $\hat{I}$. (recall that fig. 8 (a) is signed with medium gray the zero level, bright regions the positive tophat and dark regions the negative bothat.)

3 Effects of Parameters

There are a number of free parameters in the MIC algorithm.

- k, the number of size-bands to compute
- d_k, the sizes of the structuring elements;
- f, the residual threshold multiplication factor;
- s, the minimum number of nonisolated pixels in any 3×3 neighborhood of the cleaned up residuals;
- both f and s can be specified separately for the tophats and the bothats;
- the structuring elements can be flat or hemispherical on top.

Experience gleaned from applying the algorithm to over 100 images (from about 256^2 to 1000^2 in size) suggests that good default values are $k = 3$ with SE sizes 5, 9, and 17 $f = 1$ and $s = 3$. The values are good for images of up to about a million pixels in area to be viewed on a computer monitor screen.

This section discusses the various parameters and gives examples of the effects of their variation.

3.1 Number of Size Bands and Structuring Element Size

Section 2.1 explains that there is a trade-off between degree of smoothing and degree of feature preservation when processing an image in a single size-band. Figure 9 demonstrates this for the highly detailed face from "elle.gif". Panel (a) is the original. The others are single size-band smoothings of the image with SE

diameters and threshold factors as follows: (b) $d = 3$, $f = 1.00$, (c) $d = 5$, $f = 1.25$, (d) $d = 9$, $f = 1.50$, (e) $d = 17$, $f = 2.00$, and (f) $d = 33$, $f = 2.5$.

With a single band, to get any noise reduction, it is necessary to increase the threshold factor f along with d. Otherwise, the thresholded residual has such dense support that dilating the support skeleton causes the algorithm to recombine most of the residual with the smoothed image. If this happens, the processed image is almost identical to the original. Increasing f causes the algorithm to discard more of the residual so that more of the smoothed image shows through.

Figure 9(b) shows that a small diameter SE preserves detail, but the smoother regions are still noisy. At the other extreme, figure 9(f) is too smooth in some places and identical to the original in large, almost random patches. Among the five smoothed images, figure 9(d) provides the best trade-off between detail and smoothness.

The next figure (10) demonstrates multiple size-band smoothing with various combinations of various SE diameters. Panel (a) is the original. Panels (b) through (f) were all generated with $f = 1.5$ to make the underlying smoothed images more visible. The size-bands used were: (b) $d = 5$ and $d = 9$, (c) $d = 5$, $d = 9$, and $d = 17$, (d) $d = 3$ and $d = 5$, (e) $d = 3$, $d = 5$, and $d = 9$, (f) $d = 3$, $d = 5$, $d = 9$, and $d = 17$. This figure shows that adding larger size-bands improves the smoothing without diminishing the feature preservation of the smaller bands. Note, however, that (b) and (c) look better than (d), (e), and (f). Experience has shown that for most images, $d = 3$ is too small for any significant smoothing. Moreover, due to the extreme flattening effect of the tophat and bothat at $d = 3$, the noise is often amplified with respect to the features. Thus, it is difficult to segment the $d = 3$ residual with a simple threshold.

In the course of implementing MIC, one has to decide on appropriate shapes and sizes for the structuring elements. The most useful SE support shape for arbitrary images is the disk because it is nondirectional. (Imagery with limited shape content, such as pictures of printed circuit boards, might be better processed with other structuring elements, for example squares or hexagons.) All of the research for this paper has been done with disk shaped SE's. Structuring element size is a major issue when using a single size-band smoothing. The only way to reliably find the optimum size is through trial and error. Size is less of an issue when using multiple size-bands. A natural visual progression is to have successive bands double in size. Experimentation has shown this to be adequate. The algorithm defaults to structuring elements of diameter $2^j + 1$ starting with $j = 2$. The odd size is necessary for symmetry (so that the SE origin is in its center).

If all the parameters are chosen carefully, a single size-band smoothing can do an adequate cleanup in some areas of an image. It is very likely, however, that other areas will not look as good. The best results are had using multiple size-bands. Figure 11 is a side-by-side comparison of the best single band smoothing with the best multiple band smoothing. Both cases were computed with tophat parameters $f = 1.5$, $s = 4$, and bothat parameters $f = 1.1$, $s = 3$ to favor dark features over light. Panels (a) and (c) were smoothed with one SE of diameter $d = 9$; panels (b) and (d) were processed with 3 SE's of diameters $d = 5$, $d = 9$, and $d = 17$. The multiple size-band image is smoother than the single size-band image while retaining more features. Experimentation has suggested that 3 size-bands of diameters 5, 9, and, 17, are good defaults for the algorithm.

3.2 Residual Thresholds

The noise cleaning algorithm automatically chooses a separate threshold for the tophat and for the bothat of every size-band. It uses the square root of the second moment of the graylevel histogram of the tophat or bothat as its nominal threshold. This has been found to be a good, if conservative, choice. In general, the higher the amplitude of the noise, the higher each threshold must be set with respect to the second moment. The program that implements the algorithm accepts either one or two threshold multiplication factors. If one factor is input, then all the nominal thresholds are multiplied by this value. If two factors are input, the first multiplies the bothat thresholds and the second multiplies the tophat thresholds. This enables dark features and light features to be treated separately.

Figure 12 demonstrates the effect of increasing the threshold multiplication factor. Panel (a) is the original image. The others are: (b) $f = 1.00$, (c) $f = 1.25$, (d) $f = 1.50$, (e) $f = 1.75$, and (f) $f = 2.0$. It is apparent that the (a) panel contains spurious noise, whereas in the (f) panel the face has lost much detail. For smooth regions, larger values of f have no effect, once the noise is gone.

Experience has shown that a good way to select a threshold multiplication factor is to use none at first

($f = 1$) and if the noise cleaning is inadequate, try $f = 1.25$ or $f = 1.5$. Seldom is $f \geq 2.0$ required. In fact, as figure 12 demonstrates, as f gets close to 2, most of the residual is discarded, and the image begins to lose important features. In general, if $f > 2$ is necessary to clean the image then MIC is probably not the right approach. Under these circumstances, the Song-Delp algorithm is likely to be superior.

3.3 Residual Support Size

The noise in some images is characterized by small clusters of pixels with similar graylevel. Such clusters will sometimes exceed the residual thresholds. Then MIC treats them as features and puts them back into the cleaned image. This usually enhances their visibility in the image and can effectively negate any positive results from the algorithm. Often these noise clusters can be eliminated by increasing the minimum support size per neighborhood (parameter s in the algorithm). The way the algorithm is now written, s can range from 1 to 9. The default value is $s = 3$ since this will preserve a 1 pixel-wide line crossing a 3×3 neighborhood. Increasing the minimum support size removes noise clusters at the expense of features. Figure 13 shows this. Panel (a) is the original. The others are (b) $s = 3$, (c) $s = 4$, (d) $s = 5$, (e) $s = 7$, and (f) $s = 9$. It would appear that for "elle.gif" the only useful values are 3 and 4.

Residual support size is calculated on a *per neighborhood* basis. The algorithm would likely be more effective if steps 5 through 9 in algorithm 3 (TOPBOT) were replaced with a connected component labeler that then deleted all connected support regions with less than some given area. This was not done because of the complexity it would add to the algorithm and because the simpler approach works well much of the time.

3.4 Structuring Element Curvature

Section 2.1 indicated that either flat-topped or curved-topped structuring elements would be appropriate to use with this algorithm. Each type has its advantages. A curved-topped SE will produce a somewhat smoother opening or closing than a flat-topped SE; but it also can blur slightly the boundary between large regions. An opening or closing by a flat-topped SE not only preserves some interregion boundaries without smoothing, but also it is much faster to compute than with the same sized curved-topped SE. Usually, the visible differences between the two approaches are minimal. Figure 14 attests to this. Panels (a) and (d) are from the original image. Panels (b) and (e) are the result of the algorithm with flat-topped SE's. Panels (c) and (f) are the result using curved-topped SE's. The curve-topped SE's were hemispheres; the difference in gray-level from center to edge was equal to the radius in pixels of the support of the SE. The sky is noticeably smoother in panel (c) than in panel(b). Likewise the face in panel (f) is smoother than that in (e), but it is not clear which one actually looks better. Because of the small difference in appearance between the two, and because of the computational advantage of the flat-topped SE, it is the default for the algorithm.

4 Effects on Example Images

After comparing the result of the new algorithm with that of a median filter and the result of the Song-Delp generalized morphological filter, this section shows how the new algorithm can remove patterned scanner noise. Then follow small sections of other images that demonstrate the general utility of the algorithm.

4.1 Comparison with Other Methods

Figure 15 shows the sky and face from "elle.gif" as processed by a 3×3 median filter (panels (a) and (d)), the new algorithm (panels (b) and (e)), and the Song-Delp generalized morphological filter (panels (c) and (f)). The median filter smooths the image somewhat, but it obliterates the 1D features. The overall appearance is suggestive of linear blurring although it is not as severe. The Song-Delp filter was an averaging filter configured as the authors suggest in figure 4 of their paper [24]. The structuring elements used were the four line SE's, $k_1, \ldots, k_4$, suggested in figure 6 of [24] The result is quite smooth and undoubtedly superior to the median filter. However, as they suggest in their paper, some of the image detail is lost. The result of MIC is smoother than both the others in the smooth regions and retains noticeably more detail. Results (b) and (e), however, do contain some spurious noise artifacts.

4.2 Removal of Scanner Noise

Figure 16 shows portions of two poorly scanned images. Panel (a) exhibits a high frequency cross-hatch pattern. Panel (b) contains both high frequency cross-hatching and a highly visible low frequency banding like a moire pattern. Panels (c) and (f) show the result of 3×3 median filtering the two images. The median filter has done a reasonably good job of removing the cross-hatch pattern in both images at the expense of fine detail. However, it had no effect on the moire pattern of panel (d). Panel (b) is the result of a straight application of MIC with SE diameters 5, 9, and 17, and $f = 1.5$. This has removed the cross-hatching everywhere except in the vicinity of retained features.

Panel (e) shows the result of MIC applied to (d). The removal of the moire pattern required some special treatment. Analysis of the individual size-bands showed that the pattern was completely contained in the band bounded by $d = 9$ and $d = 17$, where it was the dominant feature. It was visible in both the tophat and the bothat. The thresholds for this size-band were chosen very high at $f = 2.5$. However, the remaining size-bands were thresholded comparatively lower, at $f = 1.25$. Recombining all the thresholded residuals with a $d = 33$ OCCO smoothed version produced the image of panel (e). A straightforward application of MIC with $f = 1.25$ did not remove the moire pattern; using $f = 2.5$ removed most all the features along with the pattern. By thresholding the residuals at different levels MIC was able to remove the pattern.

4.3 General Applicability

Figure 17 is a composite of sections of 8 different images. All the images were processed using 3 size-bands with structuring elements diameters 5, 9, and 17. Only one parameter was adjusted for the various images; the threshold multiplication factor, f was set individually. In each pair, the original image is on the left, and the cleaned image is on the right. These figures demonstrate that MIC is applicable to a range of imagery beyond faces.

5 Noise Analysis and Image Fidelity

To understand quantitatively the capabilities and limitations of a noise reduction algorithm one must analyze the algorithm under controlled conditions. To test MIC, a noise free image was found. A set of images was created by adding noise of different variances to the original image. MIC was applied to the individual images. The images were median filtered for comparison. A low variance (nearly constant) area was selected in the original image. The noise was measured in the corresponding area in each of the noisy images before and after processing. Also, each of the processed images was compared to the original using an image similarity metric.

The noise free test image was the 8-bit, grayscale, luminance component of a 24-bit, truecolor, raytraced image of some mirrored balls with the Sun Microsystems logo on them. The image is extraordinarily clean with smooth regions and fine details, attributes which made it a good test for MIC. The noise corrupted images were created using the program, *cantata* from the University of New Mexico *khoros* system [17, 18]. Thirteen noisy images were made as follows: The original image was loaded into *cantata* and converted to floating point. Thirteen zero mean, floating point, gaussian distributed, noise images were made using the routine *vggauss*. Each image had a different noise variance. The variances ranged from 1 to 4096 in powers of 2. A noisy image was made by adding one of the pure noise images to the floating point original. Each noisy floating point image was converted back to a byte image through simple truncation. Pixel values less than zero were set to zero, and values greater than 255 were set to 255.

The thirteen images were each processed by MIC 4 times with different parameters. These were

1. 3 size-bands, $d = 5, 9, 17$, $f = 1.5$, $s = 3$
2. 4 size-bands, $d = 3, 5, 9, 17$, $f = 1.5$, $s = 3$
3. 4 size-bands, $d = 3, 5, 9, 17$, $f = 2.0$, $s = 3$
4. 4 size-bands, $d = 3, 5, 9, 17$, $f = 2.0$, $s = 4$

Table 1: Noise Standard Deviation

Nominal	Estimated	3-level (1.5, 3)	4-level (1.5, 3)	4-level (2, 3)	4-level (2, 4)	median
1.0000	1.1459	0.5793	0.6522	0.5577	0.5577	0.7596
1.4142	1.4244	0.5953	0.7898	0.5699	0.5479	0.7943
2.0000	1.9641	0.7055	0.7926	0.6307	0.5302	0.9713
2.8284	2.8143	1.4688	1.5521	1.0671	0.8010	1.2308
4.0000	4.0529	2.2429	2.3876	1.4302	0.9196	1.7732
5.6569	5.7875	3.2408	3.3579	1.9347	1.0198	2.3226
8.0000	8.0634	5.4310	4.8034	3.1769	1.4446	3.3188
11.3137	11.3702	8.3889	7.2983	4.8895	2.3229	4.8278
16.0000	15.5071	10.8020	9.0510	5.8622	3.2126	6.2250
22.6274	23.1456	17.3927	15.2191	10.3464	5.3158	9.6864
32.0000	31.8216	24.1878	21.0138	12.9798	7.0790	13.3041
45.2548	44.3234	34.4294	29.4147	21.8658	8.5438	18.5423
64.0000	62.1826	52.4358	45.6681	34.7874	17.6056	26.8120

(In the following exposition, a type j image ($j = 1, \ldots, 4$) refers to an image processed with parameters j above). Each of the images was also processed with a 3×3 median filter. Figure 18 shows (a) a section from the original image, (b) the type 3 cleaned up version of (c), (c) the original image with additive gaussian noise of variance 32, and (d) the 3×3 median filtered (c). Two different measurements were made. The first was a measure of the noise standard deviation, σ, in a nearly constant 50×50 region of each image. The second was a comparison of each image with the original using a similarity metric. The metric is the energy of the difference between the laplacians of the two images divided by the energy in one of them (the reference image). This metric is suggested by Pratt [16] as one which follows human perception of image similarity. Let I be the uncorrupted original image, Let P be either a noisy image or a noise reduced image. Let $\mathcal{L}\{I\}$ represent the laplacian of I. Then the similarity metric is

$$e(I,P) = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\mathcal{L}\{I\}(i,j) - \mathcal{L}\{P\}(i,j)\right]^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\mathcal{L}\{I\}(i,j)\right]^2}} \tag{8}$$

Table 1 lists the measured noise standard deviations. The pairs of numbers in the headers are (f, s), where f is the threshold multiplication factor and s is the neighborhood support size. Column 1 lists the nominal σ of the noise added to the clean image. Column 2 lists the σ measured in the 50×50 square in the noisy image. The measured σ's of the last two images differ significantly from their nominal value because the pixel values were clipped at the extrema when converting from floating point to 8-bit grayscale. Columns 3-6 list the measured σ's of the processed images of types 1-4 respectively. Column 7 is the measured σ in the square in the median filtered image. As one would expect, the best noise reduction performance is achieved by the type 4 images, since these have had the least amount of residual added back to them; they are the smoothest images. For the same reason, the type 3 images have smaller σ's than the type 2 images. Note that for low noise levels, the 3 size-band (type 1) image is cleaner than the 4 size-band (type 2) image. The type 1 and type 2 images have smaller σ's than the median filter only for the lower noise levels. The type 3 images beat the median filter in noise levels through $\sigma = 4$. The type 4 images always do better than the median filter.

Table 2 lists the similarity of the the various images to the original, as calculated by the metric, eqn. (8. The table shows that for small σ's the type 1 image is the most similar to the original. At two intermediate values of σ the type 3 images are most similar. For higher values of the standard deviation, the type 4 images are closest in the metric to the original. With the appropriate parameters, MIC always produces an image closer in the metric to the original than the median filter.

Figure 19 shows a section of the noisy images for (a) $\sigma = 2$, (b) $\sigma = 4$, (c) $\sigma = 8$, and (d) $\sigma = 16$. The

Table 2: Normalized Energy Difference from Original Image

Noise σ	Unprocessed	3-level (1.5, 3)	4-level (1.5, 3)	4-level (2, 3)	4-level (2, 4)	median
1.0000	0.3082	0.2666	0.2835	0.3053	0.4235	0.5206
1.4142	0.4144	0.2950	0.3174	0.3250	0.4357	0.5316
2.0000	0.5694	0.3355	0.3579	0.3540	0.4552	0.5501
2.8284	0.7946	0.4427	0.4677	0.4030	0.4838	0.5838
4.0000	1.1158	0.5913	0.6204	0.4929	0.5305	0.6371
5.6569	1.5789	0.8323	0.8509	0.6416	0.5895	0.7275
8.0000	2.2279	1.4171	1.1967	0.8367	0.6737	0.8642
11.3137	3.1376	2.0871	1.6797	1.1486	0.7874	1.0706
16.0000	4.2248	2.9048	2.3342	1.5522	0.8798	1.3612
22.6274	5.2909	3.6111	3.0978	2.0268	1.0337	1.8110
32.0000	6.2104	4.5665	3.7611	2.4848	1.2068	2.4452
45.2548	6.9256	5.3289	4.4758	2.9587	1.3960	3.3045
64.0000	7.4614	6.0741	5.2049	3.5858	1.7472	4.2515

first column on the left is the unprocessed noisy image. The second column from the left is the output of the median filter. The third column is the type 3 output and column 4 is the type 4 output of MIC. The median filter destroys the fine features as it smooths the noise. The type 3 output looks good in (a), (b), and (c). But in (d), it incorrectly interprets noise patches as features. The result is an image that looks worse than the median filtered version. The type 4 image is smoother throughout, but it does not have as many of the 1D features as the type 3 image.

The results of the metric test are misleading. They do not jibe with visual evaluation of the images. If one looks at all the images, the ones produced by MIC look better than the median filtered ones up to about $\sigma = 6$. For larger standard deviations, the median filter produces better results. One can calculate the standard deviation of the features in the original, clean test image, I, by subtracting from it OCCO(I,17), the openclose-closeopen average computed with a structuring element diameter $d = 17$. The residual can be considered to contain features alone because I is noise free and $d = 17$ is the diameter of the largest SE used in the 4 cleanings above. This residual, hence the features, had $\sigma = 10.6750$. From this value and the perception that the MIC algorithm outperforms the median filter on the noisy test images for σ's up to about 6, one could state that MIC works well on images for which the the noise σ is no greater than one half of the feature σ. Of course, such a measurement *a priori* is usually impossible.

6 Application: Improved JPEG Compression

A useful aspect of MIC is that it enhances JPEG compression. The image compression algorithm known as JPEG [26] is a transform coding scheme. It partitions an image into blocks, computes the discrete cosine transform (DCT) of each block and codes each DCT component according to a quantization scheme as a function of the magnitude of the component. The compression is greatest for constant or slowly varying blocks since these can be described by just a few DCT components. If an image is noisy, even slightly, then all its constant or slowly varying regions are degraded. More DCT coefficients are necessary to code the block. This results in a larger compressed file. In effect, JPEG devotes a significant portion of the resulting file to coding the noise.

For example, in the test images of the previous section the percent increase in the size of the JPEG file is approximately an exponential function of $2\log_2\sigma$ for $\sigma \in (1, 6)$ and a linear function of $2\log_2\sigma$ for $\sigma \in (6, 64)$, where σ is the standard deviation of the noise added to the clean image. Variance 1 noise caused a 2% increase in the size of the compressed file whereas $\sigma = 4$ resulted in a 43% increase and $\sigma = 64$ caused the JPEG file to be 5 times as large as that of the clean image. [All the JPEG compressed files mentioned

in this section were computed with program *cjpeg* [9] at the default quality setting of 75. A number of the images were recomputed with program *alchemy* [30] to verify the compressions. Both programs created files of similar size.]

Clearly any procedure that will reduce the noise in an image is bound to improve the compression. To test MIC in this capacity, 42 scanned images were taken from the alt.binaries.pictures.misc newsgroup of usenet. It is presumed that all of the images were scanned. None of them were images created directly by a computer. MIC processed the images with two size-bands, $d = 5$ and $d = 9$, and used $s = 3$. The algorithm was run twice; once with $f = 1.0$ and once with $f = 1.5$. Table 3 shows the results. With $f = 1$ MIC is very conservative; its noise reduction, although pronounced, is at a minimum. Under these conditions, the average reduction in file size was 12%. With $f = 1.5$ there are obvious changes in the images, favorable in many cases, but not all. In this threshold factor, the average reduction in file size was 22%.

7 Conclusion

This paper has presented a new algorithm, MIC, for image noise reduction based on morphological size distributions. MIC smooths the image in a number of size-bands, subtracts these bands out of the image to create residual images, segments the residuals into features and noise, and adds the features back to the smoothed image. The algorithm was shown to be useful in removing noise and scanner artifacts in images where the standard deviation of the noise is not large. In a test case where it was possible to compute the relative standard deviations, it was shown that the standard deviation of the noise should not exceed one half of the standard deviation of the features in the residual. Such small noise levels are often the case in scanned and still video images.

Successful use of MIC requires a user to set three parameters, the number of size-bands, a residual segmentation threshold multiplication factor, and a segmentation neighborhood support size. Default values for these parameters were suggested, and the effects of parametric variations were reviewed. The results of the algorithm were compared to those of median filtering and the Song-Delp generalized morphological filter and shown to be superior when the noise conditions are met by the input imagery. The results on 24 test images were exhibited. Two examples of removing scanner artifacts were given. A noise analysis was performed under controlled conditions of adding noise to a noise free test image. The algorithm was shown to useful for the preprocessing of images for JPEG compression where it resulted in an average size reduction of 12% when applied with its most conservative parameters.

Negative aspects of MIC include: Parameters have to be set by trial and error to get the best results. It is relatively computationally expensive compared to a median filter; a megapixel image can require 30 minutes (of real time) for processing on a SparcStation1 when computed with 3 size-bands (speedups are possible). If the noise content of an image is high, MIC does not perform as well as the Song-Delp algorithm. In this situation it can actually highlight some of the noise features making the result look (possibly) worse than the original. MIC has been tested on grayscale, luminance images only. An extension to color may or may not be straightforward.

Two extensions of MIC are planned. One is to color imagery where obvious extensions to RGB and HSI spaces will be explored. The effect of color quantization will also be considered. The other plan is to generalize the algorithm into three dimensions for use in volumetric imagery and as a spacetime filter for moving imagery. Here the key problems include structuring element design and event detection in segmentation of the spacetime residual.

Table 3: Normalized Energy Difference from Original Image

image name	rasterfile size	jpeg size	clean $f = 1.0$ jpeg size	percent change	clean $f = 1.5$ jpeg size	percent change
a7-e.gif	256800	37627	28850	23	26011	31
airport.gif	308000	51661	43816	15	38544	25
aliens.gif	308000	56803	53110	7	50999	10
astronaut.gif	308000	68519	60315	12	56519	18
baboon.gif	262944	73586	65113	12	59420	19
balticbrown.gif	1037600	317080	291231	8	256114	19
butfish1.gif	303200	81914	71730	12	64822	21
cannibal.gif	323388	90496	85397	6	63655	30
clshutle.gif	308000	51452	43703	15	42088	18
dollar.gif	507962	129362	117333	9	111543	14
earth3.gif	443684	109472	87352	20	79431	27
eclipse.gif	969504	28933	20768	28	18003	38
einstein.gif	168328	52898	47083	11	40390	24
elle.gif	254996	45565	32032	30	28679	37
elvis-n-nixon.gif	289400	47031	42011	11	40666	14
fishy.gif	256800	34788	32271	7	30006	14
flower.gif	308000	70472	60445	14	52790	25
froggy.gif	256800	22389	21405	4	20491	8
jean.gif	121600	13182	9906	25	9108	31
jlennon.gif	164000	34670	29502	15	27306	21
judith_henry.gif	192752	48238	37789	22	33844	30
jupiter_bw.gif	640800	25755	25871	0	25548	1
lighthouse.gif	787232	229508	199959	13	182410	21
lori2.gif	71200	12535	10450	17	9337	26
lush.gif	787232	287944	264839	8	239184	17
mercury.gif	719568	201206	181534	10	165675	18
michaela.gif	295520	29870	22239	26	19535	35
neptunec.gif	250800	6504	6348	2	6377	2
ouat.gif	294080	76627	69964	9	61521	20
redrose.gif	64800	5546	5166	7	4923	11
rover2.gif	186656	63005	58744	7	43443	31
sanfran.gif	221984	39364	36058	8	33990	14
shelly.gif	302264	42718	40972	4	35143	18
shuttle15.gif	1037600	198758	159441	20	142922	28
sioct86.gif	182570	26092	22602	13	21439	18
spacewalk.gif	308000	47466	42079	11	40076	16
sr-71.gif	256800	23084	19532	15	16250	30
steph1.gif	440800	67108	54310	19	42305	37
t2-1.gif	155936	24520	20616	16	16944	31
t2-2.gif	172454	20318	18412	9	15236	25
t2-3.gif	308000	48881	40411	17	34178	30
the7.gif	465536	144130	134098	7	102317	29
the7.gif	465536	144130	134098	7	102317	29
average				12		22

References

[1] Andrews, H. C., "Monochrome digital image enhancement," *Applied Optics*, Vol. 15, No. 2, pp. 495-503, 1976.

[2] Andrews, H. C., and B. R. Hunt, *Digital Image Restoration*, Prentice-Hall Signal Processing Series, A. V. Oppenheim, Ed.,, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[3] Bovik, A. C., "Streaking in median filtered images," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, No. 4, pp. 493-503, 1987.

[4] Giardina, C. R. and E. R. Dougherty, *Morphological Methods in Image and Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[5] Gonzalez, R. C., and P. Wintz, *Digital Image Processing*, 2nd ed., Addison-Wesley, Reading, Mass., 1987.

[6] Haralick, R. M., S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, No. 4, pp. 532-550, 1987.

[7] Haralick, R. M., and L. G. Shapiro, *Computer and Robot Vision*, Vols. 1 and 2, Addison-Wesley, Reading, Massachusetts, 1992.

[8] Jain, A.K., *Fundamentals of Digital Image Processing*, Prentice-Hall Information and System Sciences Series, T. Kailath, Ed., Prentice-Hall, Englewood cliffs, New Jersey, 1989.

[9] Lane, T. J., P. Gladstone, and L. Ortiz, "cjpeg," (computer program) *The Independent JPEG Group's JPEG Software*, 1991.

[10] Maragos, P. and R. W. Schafer, "Morphological skeleton representation and coding of binary images," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, No. 10, pp. 1228-1244, 1986.

[11] Maragos, P. and R. W. Schafer, "Morphological filters – part I: their set theoretic analysis and relations to linear shift invariant filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, No. 8, pp. 1153-1169, 1987.

[12] Maragos, P. and R. W. Schafer, "Morphological filters - part II: their relations to median, order-statistic, and stack filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, No. 8, pp. 1153-1169, 1987.

[13] Maragos, P., "Pattern spectrum and multiscale shape representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-11, No. 7, pp. 701-716, 1989.

[14] Maragos, P. and R. W. Schafer, "Morphological Systems for multidimensional signal processing," *Proc. IEEE*, vol. 78, No. 4, pp. 690-710, 1990.

[15] Mastin, G. A. "Adaptive filters for digiital image noise smoothing: an evaluation," *Comp. Vision, Graph., Image Process.*, vol. 31, pp. 103-120, 1985.

[16] Pratt, W. K., *Digital Image Processing*, Wiley, New York, 1978.

[17] Rasure, J., D. Argiro, T. Sauer, and C. Williams, "A Visual Language and Software Development Environment for Image Processing," *Int. J. Imag. Syst. and Technol.*, Vol. 2., pp 183-199, 1990.

[18] Rasure, J. and C. Williams, "A Integrated Visual Language and Software Development Environment," *J. Visual Lang. and Comp.*, Oct. 1991.

[19] Restrepo, A., and A. C. Bovik, "Adaptive trimmed mean filters for image restoration," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-36, No. 8, pp. 1326-1337, 1988.

[20] Rosenfeld, A., and A. C. Kak, *Digital Image Processing*, Vols. 1 and 2, Academic, New York, 1982.

[21] Serra, J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

[22] Serra, J., "Introduction to mathematical morphology," *Comp. Vision, Graph., Image Process.*, vol. 35, pp. 283-305, 1986.

[23] Serra, J., ed., *Image Analysis and Mathematical Morphology, Vol. 2: Theoretical Advances*, Academic Press, New York, 1988.

[24] Song, J., and E. J. Delp, "A study of the generalized morphological filter," submitted to *Circuits, Systems, and Signal Processing*, 1990.

[25] Sternberg, S. R., "Grayscale morphology," , vol. 35, *Comp. Vision, Graph., Image Process.*, pp. 333-355, 1986.

[26] Wallace, G. K., "The JPEG still picture compression standard," Commun. ACM, Vol. 34, No. 4, pp. 31-44, 1991.

[27] Wang, D. C. C., A. H. Vagnucci, and C. C. Li, "Gradient inverse weighted smoothing scheme and the evaluation of its performance," *Comp. Vision, Graph., Image Process.*, vol. 15, pp. 167-181, 1981.

[28] Weszka, J. S., "A survey of threshold selection techniques," *Comp. Graph. and Image Process.*, vol. 7, 259-265, 1978.

[29] Weszka, J. S. and A. Rosenfeld, "Threshold evaluation techniques" *IEEE Trans. Syst, Man, and Cybern.*, vol SMC-8, No.8, pp. 622-629, 1978.

[30] Woehrmann, M. H., A. N. Hessenflow, and D. Ketterman, "Image Alchemy," (computer program), Handmade Software, Los Gatos, CA, 1991.

Figure 1: Original image, "elle.gif."

Figure 2: Result of the new algorithm applied to "elle.gif."

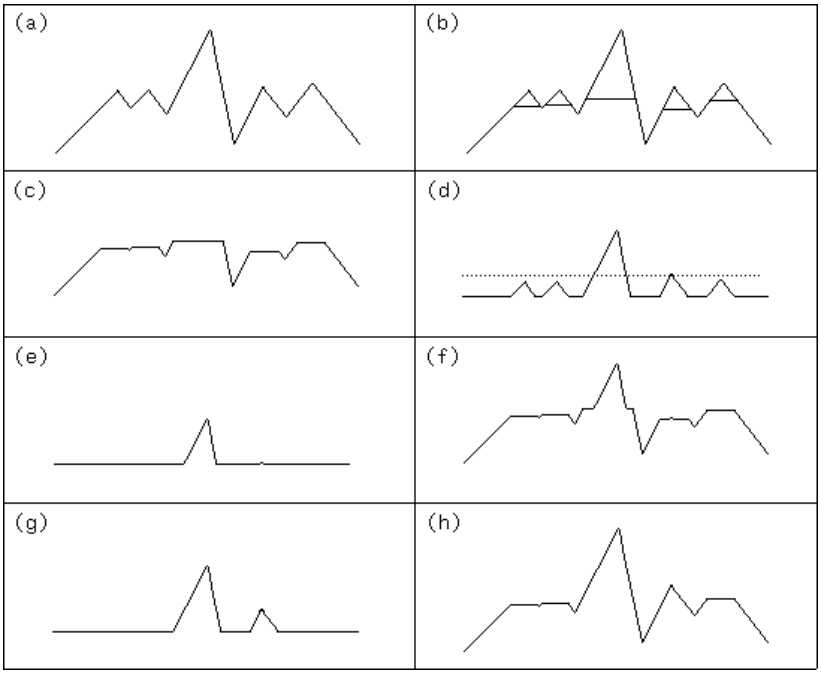

Figure 3: Recovery of the "base" of a center clipped residual.

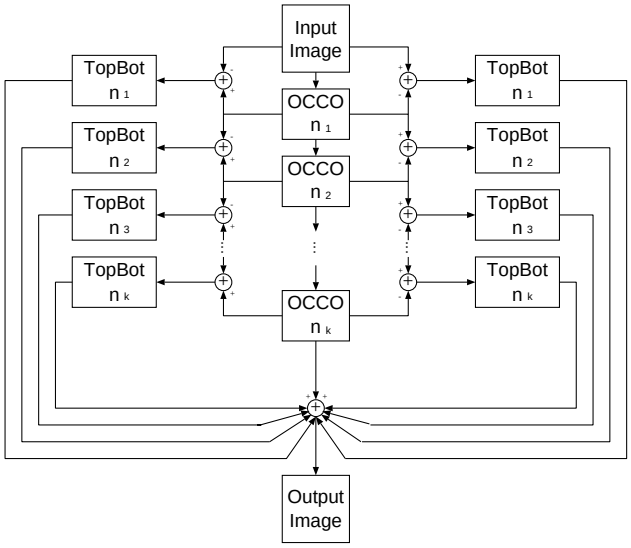

Figure 4: Block diagram of the main loop of the noise reduction algorithm.

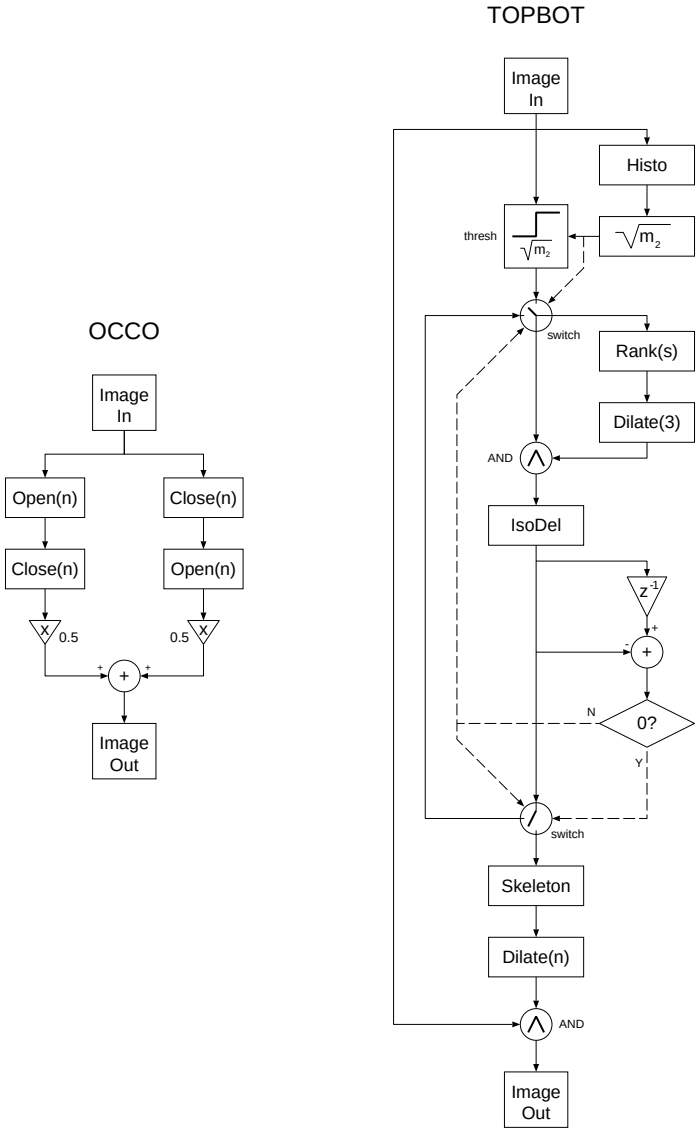

Figure 5: Block diagrams of subroutines OCCO and TOPBOT.

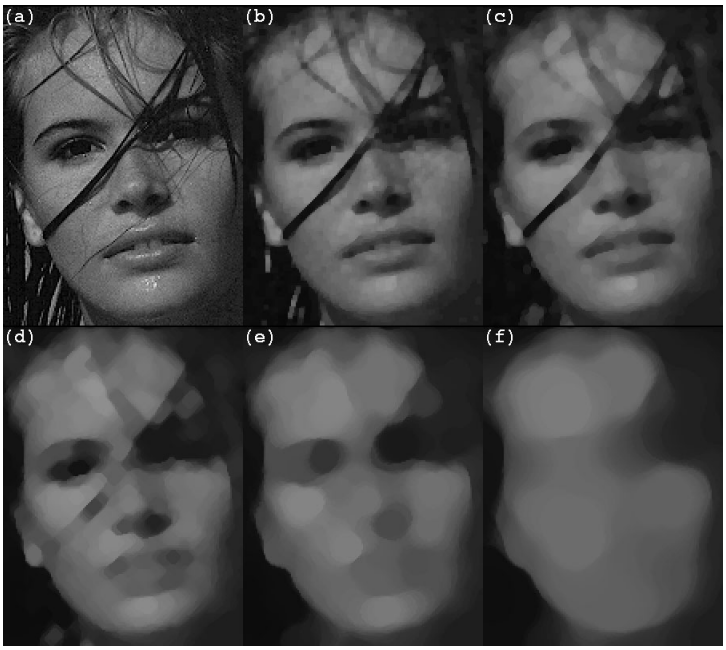

Figure 6: OCCO smoothed images of face.

Figure 7: Residuals of fig. 6.

Figure 8: Noise cleaned residuals from fig. 7.

Figure 9: Single size-band processing of the face.

Figure 10: Multiple size-band processing of the face.

Figure 11: Comparison of the best 1 band and best 3 band results.

Figure 12: Effects of increasing threshold f on face.

Figure 13: Effects of increasing minimum neighborhood support size.

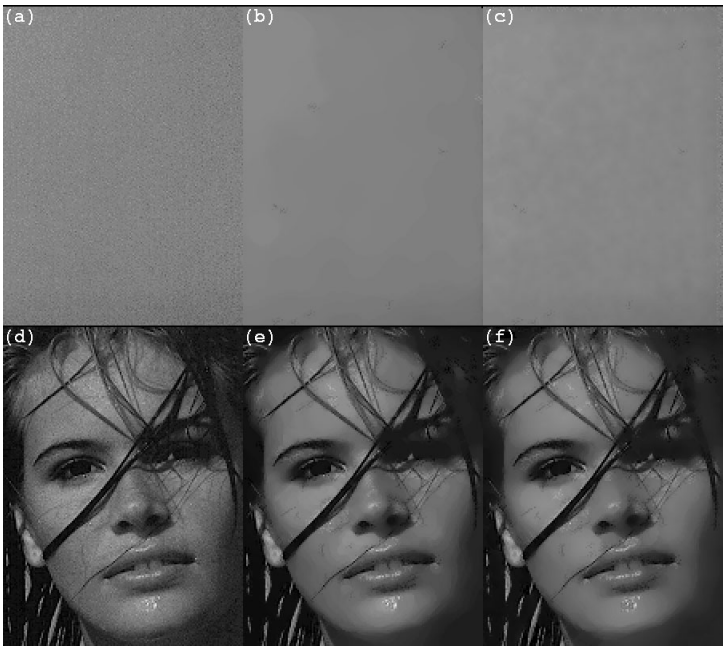

Figure 14: Effects of structuring element curvature.

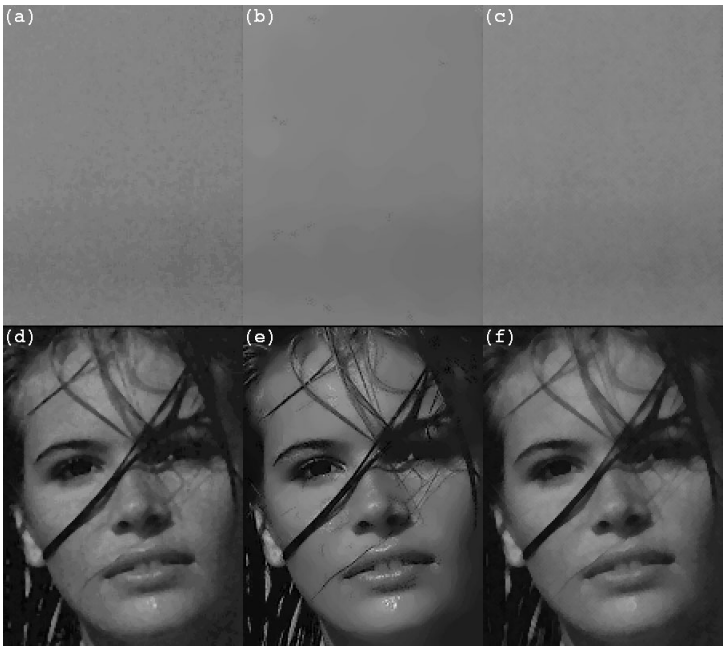

Figure 15: Comparison of a median filter, the new algorithm, and the Song-Delp algorithm.

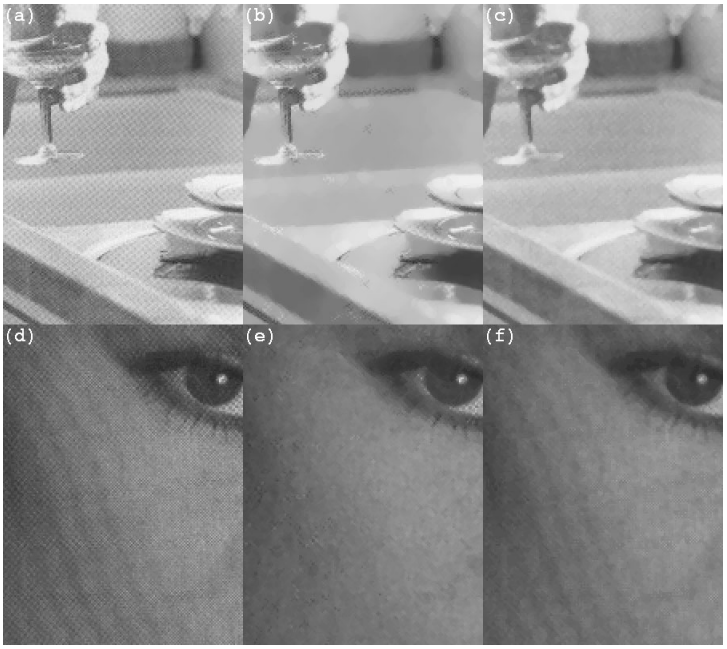

Figure 16: Removal of scanner noise.

Figure 17: Composite image. Originals are on the left.

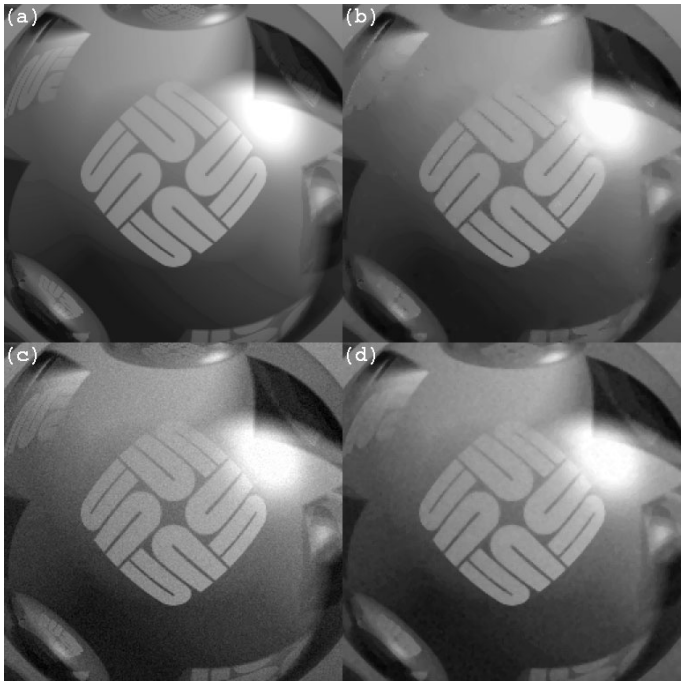

Figure 18: Noise free image, noise currupted image, and cleaned versions

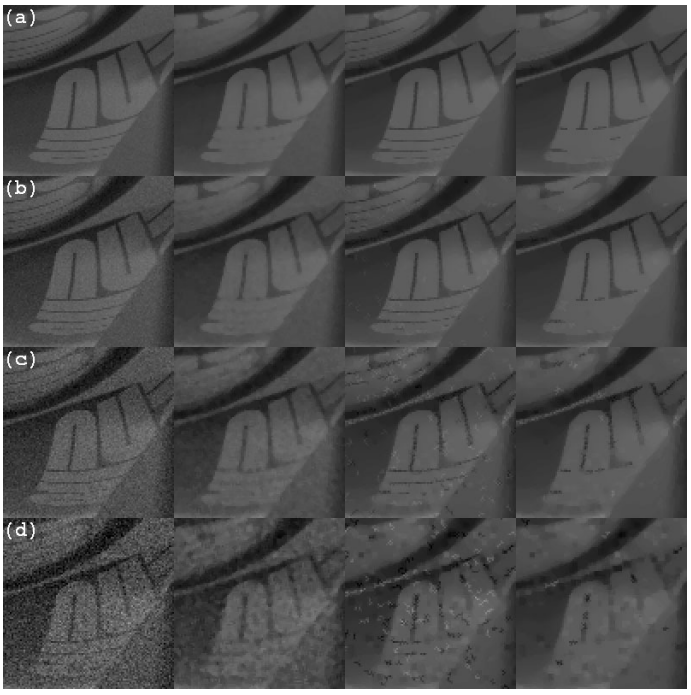

Figure 19: Effects of noise cleaning on images with different noise variances.

List of Figures

1 Original image, "elle.gif." . . . 19
2 Result of the new algorithm applied to "elle.gif." . . . 19
3 Recovery of the "base" of a center clipped residual. . . . 20
4 Block diagram of the main loop of the noise reduction algorithm. . . . 20
5 Block diagrams of subroutines OCCO and TOPBOT. . . . 21
6 OCCO smoothed images of face. . . . 22
7 Residuals of fig. 6. . . . 22
8 Noise cleaned residuals from fig. 7. . . . 23
9 Single size-band processing of the face. . . . 23
10 Multiple size-band processing of the face. . . . 24
11 Comparison of the best 1 band and best 3 band results. . . . 24
12 Effects of increasing threshold f on face. . . . 25
13 Effects of increasing minimum neighborhood support size. . . . 25
14 Effects of structuring element curvature. . . . 26
15 Comparison of a median filter, the new algorithm, and the Song-Delp algorithm. . . . 26
16 Removal of scanner noise. . . . 27
17 Composite image. Originals are on the left. . . . 27
18 Noise free image, noise currupted image, and cleaned versions . . . 28
19 Effects of noise cleaning on images with different noise variances. . . . 28

List of Tables

1 Noise Standard Deviation 13
2 Normalized Energy Difference from Original Image 14
3 Normalized Energy Difference from Original Image 16