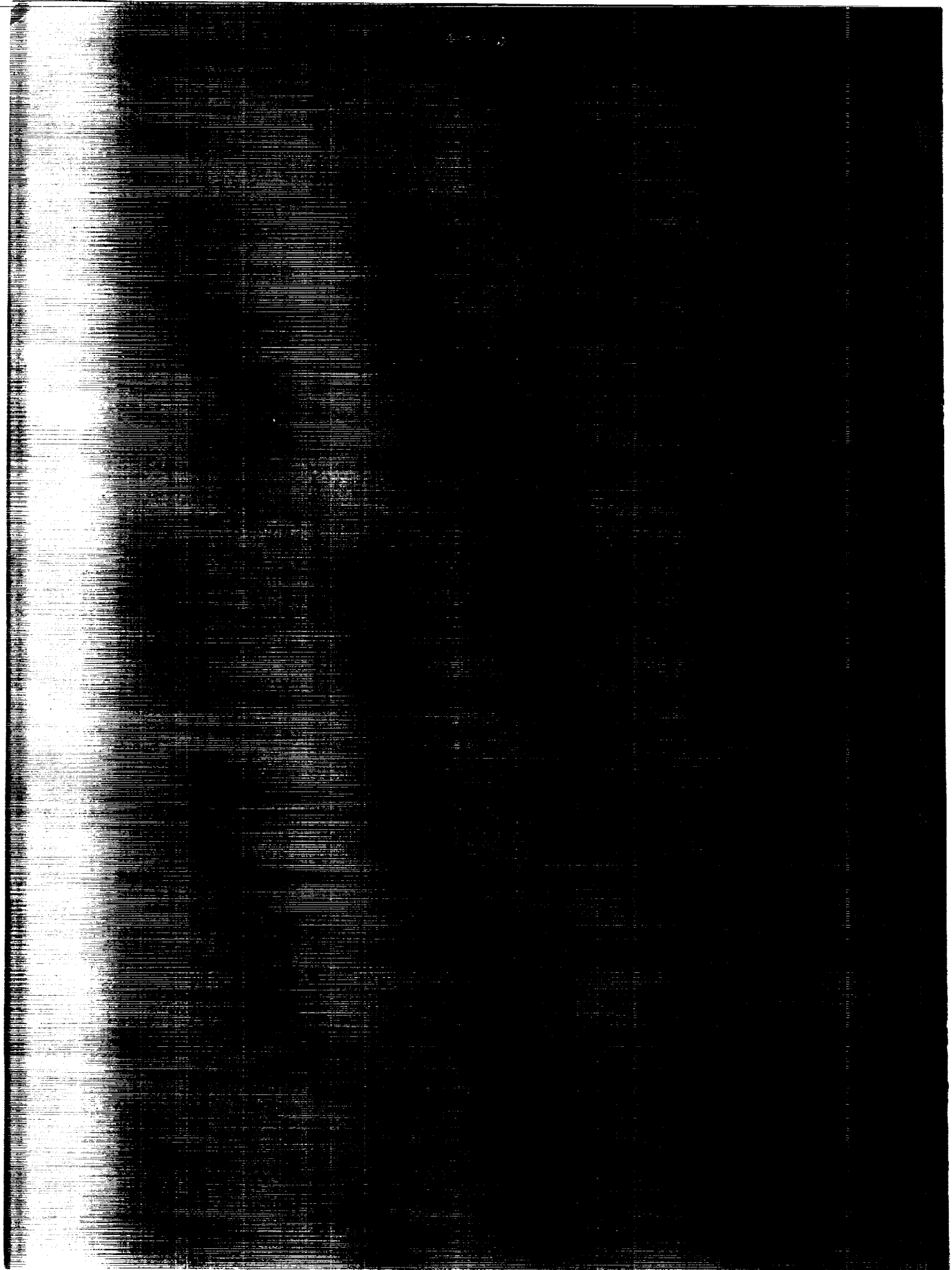
NASA Technical Memo[illegible]311

A Superior Edge [illegible]ving Filter With a Systemat[illegible]lysis

Kenneth W. Holladay and Dou[illegible]

AUGUST 1991



NASA Technical Memorandum 4311

A Superior Edge Preserving Filter With a Systematic Analysis

Kenneth W. Holladay
University of New Orleans
New Orleans, Louisiana
and
Lockheed Engineering & Sciences Company
Stennis Space Center, Mississippi

Doug Rickman
John C. Stennis Space Center
Stennis Space Center, Mississippi

National Aeronautics and
Space Administration

Office of Management

Scientific and Technical
Information Program

1991

INTRODUCTION

Smoothing noise and preserving edges are competing and discordant objectives for an image filter. Spatially invariant linear filters cannot satisfactorily resolve this conflict; they generally emphasize smoothing noise at the expense of preserving edges. The search for filters that preserve edges better has led to nonlinear filters, often called adaptive filters, that adjust their calculations according to the local properties of an image. In this paper we propose a new edge preserving filter, which we call the contiguous K-average filter, that is related to the K-average filter first introduced by Davis and Rosenfeld [8]. As is customary, we will compare our filter to other edge preserving filters using test data. [see for example 6,8,10,12,20,24,26,27,40] However, we would like to move beyond the narrow assessment that such comparison tests provide. We must begin to analyze filters as tools that are used as a part of the restoration, enhancement and analysis of images. Every filter has capabilities and limitations. It is only by identifying the individual properties of a given filter that we can hope to know when this filter is an appropriate tool for some specific task. Therefore we are not merely reporting studies on several classes of edge preserving filters; we are also proposing some methods to be used for the study of all such filters.

The contiguous K-average filter was developed by Ronnie Pearson at NASA's Science and Technology Laboratory of the John C. Stennis Space Center. The filter was intended for use in processing remote sensing images. This context has several important consequences for image processing.

For such data there are a variety of different sources of degradation that produce significantly different kinds of noise. Some of these kinds of noise are only vaguely characterized and some change from one data acquisition to the next. Techniques that require a precise description of well defined noise lose some effectiveness with such vaguely characterized and variable noise [4,12,20,27,44]. The contiguous K-average filter does not make specific assumptions about the statistical properties of the image noise and it does not require estimating any parameters of the noise distribution. The method does have parameters, the window size and the number of neighbors, but these are best set using information about the content of the image, or the exact purpose for using the filter rather than the the nature of the noise.

Many sources of remote sensing images produce multiple channel data. The noise is frequently not well correlated across channels whereas the true data normally is well correlated. Algorithms like the contiguous K-average that can take advantage of this correlation will perform better than algorithms that can not [18].

Geographic images are not pictures of a few well defined objects against a simple background, ie they lack image wide coherence. They contain many small local subpopulations, and do not satisfy the underlying assumptions of methods that use global statistics [18,30]. For example, the histogram of a geographic image merges so many different subpopulations that it can not reliably be used to identify which neighbors of a pixel come from the same subpopulation. For this reason the contiguous K-average filter uses local properties of the image. It assumes the pixels of an object have similar gray level values and form a connected shape.

In practice we want to use edge preserving filters to help delineate the component objects of an image. In many cases the objects in an image are separated from one another by the edges formed by their boundaries. Hence clarifying the edges will make the objects more distinct. The "objects" in a remotely sensed geographic image are often areas delineated by abstract concepts like land use, geology and plant cover. For this reason we will use the term feature instead of object. Another advantage of this usage is that it will remind the reader of the following discussion. A feature is a connected set of pixels of the image. The gray levels of the pixels in a feature are more or less constant, but perturbed by noise, and hence actually are not all the same. Finally, adjacent features have significantly different average gray levels.

Conceptually, we are modeling an image as a piecewise constant function that has been degraded by noise.

The piecewise constant image model has motivated the design of many filters, including ours. But real world images frequently do not conform well to this model. Strong trends in the gray level of an object are common. Boundaries between features are often genuinely extended and vague. Images will contain regions where the true scale of the features is so small that the region appears to be a jumble of different values. See [14] for a filter based on a piecewise linear image model. See [27] for a filter based on higher degree polynomials.

The new filters in this paper enhance features by removing noise using controlled local averaging. Averaging is well established as a technique for removing noise. If several values are taken from a population, their average will be a better estimate of the population mean than any single value. The problem is how to compute an average using only pixels from a single homogeneous subpopulation. See [11,25,29,39,and 43] for several approaches to this problem. Of course we must not be too dogmatic here, since sometimes there will be pixels that do not really come from any well defined subpopulation; and an algorithm must still do something reasonable for these pixels. See [32 and 41] for different approaches to adaptively computing local averages. The following three principles of filter design could be viewed as heuristics for choosing a set to average with the goal of the selection being to increase the chances of choosing members for the set that are from a single subpopulation (if a suitable subpopulation exists.) The first principle is that the members of the set to average should be pixels near the pixel whose filtered value is being computed. We implement this principle by using a moving window algorithm. See [38] for filters that adaptively change the size of the averaging neighborhood. See [12,21,22,42] for several algorithms that use local statistics in a fashion different from from our use. The second principle is that the members of the set should have gray levels near the gray level of the pixel whose value is being computed. See [24,23] for a filter, the sigma filter, based on this principle. The third principle, which characterizes our new filter, is that the set chosen should be connected.

We will compare filters that use different combinations of the three principles. The simplest class of filters to be discussed are the fixed template filters. These filters choose a set to average by selecting the pixels in a fixed set of positions relative to the pixel being updated. Thus the first principle, choosing nearby pixels, is the only principle used by fixed template filters. These conceptually simple, nonadaptive filters provide a theoretically tractable baseline case for comparison with the other, more complex filters. The K-average filter [8] uses the first two principles. The set chosen to average is selected from a window around the pixel to be updated. The second principle guides the selection of pixels from the window. A predetermined number ("K") of pixels are selected by taking the pixel itself and the K-1 pixels with gray levels nearest to the gray level of the pixel to be updated. The new filter introduced in this paper, which we will call the contiguous K-average filter, uses all three principles. The K pixels to be averaged are selected from a window, with the pixels selected having gray levels as close as possible to the gray level of the pixel being updated but with the selection restricted by the requirement that the selected set be connected. This quick description of the algorithms leaves open many details that will be attended to in the next section.

Many nonlinear filters, including ours, show a threshold effect or cutoff size. Features smaller than the cutoff are greatly attenuated or even eliminated by the filter whereas features larger than the cutoff are substantially preserved. A simple example of this is provided by the median filter, a rank operator. If a feature is too small to make up a majority of the window, pixels in the feature often will not be updated with a value from the feature; and the feature may be removed. We will see examples of this in section six. See [16] for examples using other rank filters. Linear

filters do not show this kind of threshold effect and many of the advantages of nonlinear filters over linear ones are related to the presence of the cutoff size.

Information on the cutoff size of a size gives insight on how to best use the filter. Filters are often given as parameterized families. If the dependence of cutoff size on the parameters is known, we have a basis for selecting the appropriate parameters for a given task. One generally has some idea what the smallest meaningful features in an image are and this gives an upper bound on the cutoff size. Filters are normally used to remove noise, but within a parameterized family of filters noise cleaning ability generally correlates with cutoff size. More noise cleaning forces a larger cutoff. Thus we probably want to select the parameters to give as large a cutoff as the upper bound from the smallest features allows.

The noise cleaning performance of a filter can be improved by applying the filter repeatedly. For linear filters this is equivalent to another linear filter with a larger window, but for nonlinear filters with a sharp cutoff size, iteration can be a real improvement over single passes of any filter in the same family. Two passes of a filter with a sharp cutoff will produce an effect that still has a good cutoff that is still about the same size. Thus we can increase the noise cleaning without a comparable increase in cutoff size.

We see that measurements of a filter's cutoff size and noise cleaning ability are fundamental descriptive parameters. This is particularly true of nonlinear filters that cannot be described by a transfer function using Fourier techniques. In this paper, we will define our filters and then begin the task of describing their characteristics.

The second section of this paper is devoted to the definitions of the various filters and some general discussion of how to use the filters. The basic concept of the contiguous K-average filter can be realized by several distinct algorithms that will produce slightly different results starting from the same image. We will discuss some of the possible variants. We will relate the filter's parameters to information about the size and geometry of the features of interest in an image.

The third section develops some of the basic theorems and properties needed for measuring and describing the performance of filters. Filters operating on a fixed pattern or template are used in the development. In the fourth section, we develop a technique for measuring the noise removal performance of a filter within a homogeneous feature. This technique does not measure the filter's ability to preserve edges. What we are studying is the effect of the parameters K and window size on the amount of noise removal. This investigation leads to several interesting open questions.

In the fifth section, we study root structures and iteration. A root structure for a filter is an image that is not changed by the action of the filter. In addition to their importance for the study of the effects of iteration a filter, root structures are of some interest in their own right. Iteration, repeated use of a filter, is a standard technique for improving the performance of a filter. We will give some theoretical results involving root structures and some empirical results involving iteration applied to real images.

In the sixth section, we give three demonstrations involving the contiguous K-average filter. The first demonstrations is a comparison test involving several filters action on a synthetic image with noise added to it. This test compares the filters' ability to preserve edges. The second demonstration is also a comparison test. We consider the effect of filtering as a preliminary step before an analysis algorithm; in this case a thresholding operation and then an interface length calculation. The third demonstration shows the ability of the contiguous K-average filter to use information from other channels of multichannel input to guide the smoothing away of noise.

DEFINITION OF THE CONTIGUOUS K-AVERAGE FILTER

In this section we will define several variant forms of the contiguous K-average filter, as well as the class of fixed template filters, and for completeness, the K-average filter. All these filters are transformations acting on the set of images. An image is a real valued function defined on an N by M grid. The positions of the grid are ordered pairs, (x,y), of integers with x between 1 and N and y between 1 and M. These positions are usually called pixels. The value, P(x,y), of the image P at the pixel (x,y) is usually called the gray level of the image at the pixel or just the gray level of the pixel. For simplicity, we will often speak of single channel data as above, but in practice our images are often multichannel data, and the gray level of a pixel will be a vector of reals rather than just a single real.

Notice that we are defining the gray levels to be real numbers. This is to simplify arithmetic. In practice, the gray levels would normally be discretized to a finite number (such as 256) of possible values. We will dispense with this level of realism for most of the paper because the resulting complications obscure the points we are trying to make.

All three types of filter operate according to the following scheme. The pixel whose value is to be updated is at the center of a W by W window. Some of the pixels in the window are selected. The updated value is a function of the gray levels of the selected pixels. There are two possible sources for variation. One is the function.

In general, the function is some sort of average of the values. The pixels selected are supposed to be from a single subpopulation and the value derived from them by the function is the "typical" gray level of the subpopulation. Two usual candidates for the function are the arithmetic mean, which we will hereafter call the average, and the median. Except in section six, we will confine our attention to the average.

The other and more important source of possible variation is the rule for selecting pixels from the window. The simplest rule is always to take the same pixels, relative to the window. This leads to fixed template rules which we will discuss below. As one would imagine, fixed template rules are too rigid to be really good filters. If we wish to preserve edges, the filter must be adaptive, selecting pixels based on what is in the window.

The K-average filter is one good way to do this. The algorithm selects the K pixels in the window whose gray level vectors are the least distance from the gray level vector of the center pixel. The justification is clear; pixels in the same feature should have gray level vectors that are very close together. If the data are assumed to be multichannel, so that each pixel has a vector of gray levels associated with it, the algorithm smooths one channel at a time but it uses information from all channels to make pixel selections.

A possible problem is also clear. Suppose that the center pixel's gray level has been badly perturbed by noise. Its gray level vector will thus be far from most of the other members of the feature that are around it. But there may either be other similarly perturbed pixels scattered around the window, or another feature in the window whose characteristic gray level more closely matches that of the perturbed pixel. The K-average algorithm will pick up other outliers, or the other feature, before taking the more typical pixels of the true feature. Since noise is not usually well correlated between channels, this is mostly a problem for one channel data. The problem can be partially countered by using a small window and iterating the filter. This is not a satisfying solution, for reasons of time, and because it adds the question of how many iterations are needed or can be tolerated. These problems are weaknesses of the K-average filter, which the contiguous K-average filter fixes.

The selection logic of the contiguous K-average filter is an extension of the logic of the K-average filter. A contiguous K-average algorithm selects the averaging set inductively, building one pixel at a time starting from the center pixel. To add

another pixel, the algorithm first computes the average vector of the values of the pixels already selected. It then looks at all currently nonselected pixels within the window that are adjacent to a selected pixel. From these it selects the pixel that is closest to the current average vector. Closeness is computed as it was for the K-average filter, using the L1 distance measure. That is, the distance between two vectors is the sum of the absolute values of the differences of corresponding components.

The version of the algorithm used in this paper, module FLSIFT of the package ELAS [2], has a feature that allows the channels of a multichannel image to be weighted. The formula for the distance between two vectors is modified by multiplying the absolute values of the component differences by the appropriate weights before adding them up. The channel weights can be set up to allow one channel to guide the processing of another channel. The guide channel gets a large weight and the other channel gets a low (or zero) weight. We will illustrate this useful technique in section 6. There are obviously several points at which this procedure could be changed. Another distance measure, such as the L2 or L_∞ metric, could be used, or, the distance could be measured from the center pixel's vector rather than the currently selected set's average.

The selection rule described above leads to another possible source of variation. That is the rule for breaking ties when adding pixels to the selected set. If there is more than one pixel at minimum distance from the currently selected set's average, the programs used to produce the data in this paper select the first minimal distance pixel encountered as the neighbors of the selected set are searched. We will call tie break rules of this type first appearance tie break rules. This kind of rule is clearly very easy to program and it works well in practice. There are several variants of this rule because of the different ways to scan the neighbors. Ties will be rare for multichannel data and not all ties affect the final composition of the selected set. However, ties are more common in one channel data and they can be seen to have a small effect on the filter's output for real world images as well as contrived, artificial images.

When breaking ties two kinds of problem arise. When one of two equidistant pixels is added to the selected set, there may be more pixels with values just beyond it. The cluster average is pulled toward these points by the choice of the first point and they may be selected next. This may happen repeatedly so that the other equidistant point is never selected. The effect is to introduce artificial breaks into what should have been smooth gradients. The other problem arises when adding all the equidistant pixels would give more than the required number of neighbors. Picking different subsets of the equidistant points can change the result. This effect can also introduce artificial breaks into smooth gradients.

A type of rule that we will call a balanced tie break rule addresses the above problems and leads to algorithms with slightly better theoretical properties. A balanced rule adds all the equidistant neighbors to the selected set at the same time. If this results in more than the required number of neighbors, the rule attempts to produce a prorated value from the set of equidistant pixels and then use that value with the rest of the selected set. Exactly how this is done depends on the function being used but it is easily accomplished if the function is either average or median.

A final source of variations involves the behavior of the algorithm at the boundary of the image. For a given window size, we define the boundary of an image as the set of pixels where the full sized window will not fit within the image. The version of the algorithm used to produce the data in this paper simply computed no result for boundary pixels. To compensate for this limitation, the images used in this paper had the boundary trimmed off after filtering and before statistics were computed for them. There are several reasonable approaches to deal with boundary pixels. One, which we will call the truncated window boundary rule, is to proceed as usual using the part of the window that falls within the image. If more neighbors are needed than the window

contains (for normal processing, this would only happen at the four corners of the image) then simply use all the pixels in the window. Another approach, which we will call the proportional boundary rule, is to reduce the number of neighbors used so that the number used is, as nearly as possible, the same proportion of the window as the full number of neighbors is in a full window. When the number of neighbors has been chosen to preserve a certain size feature, the proportional boundary rule will give a number of neighbors that is appropriate for the intersection of the desired size feature with a truncated window. Thus features of the desired size will be still be preserved within the boundary area. As with the various tie breaking rules, the choice of boundary rule is often not of great practical significance since for many images the boundary is ignored anyway.

The major disadvantage of the contiguous K-average algorithm is its time to execute. On the Laboratory's computer, a Concurrent 3280, a moderate sized image, a 1000 rows by a 1000 columns, can take several hours to process when K is in the range of 7 to 11. Compensating for this is the fact that processing additional channels or bands does not result in a proportional increase in processing time. This is because the time consuming construction of the cluster is only performed once. The algorithm as implemented in the Laboratory also retains a memory of which pixels were first added to a new cluster. If the next pixel to process is one of the early ones added to the current cluster, it assumes the cluster pattern will be the same and reproduces the final value for the next pixel. This yields approximately a 15% decrease in processing time in real applications.

Action of the contiguous K-average filter is controlled using the window size and K, the number of neighbors to cluster. The parameter K controls the amount of smoothing. The relative sizes of K and the window determine the type of blurring that occurs. The larger K is, the more neighbors are used to compute the new value at a point and the greater the smoothing. If all these neighbors can be taken from the same feature then a large set of them will give a value that is more reflective of the whole feature. Of course window size influences the size set that can be picked from one feature, but even more important is the nature of the feature itself. The filter moves from just smoothing noise out, to blurring the feature, as K exceeds the number of pixels in the window which are also in the feature. Then some of the neighbors selected will not belong to the feature and the filter's value will not be as representative of the feature. To summarize, a good strategy is to pick the window size first and then K. The window size should be large enough to contain several points of the smallest features of interest but not much larger than the size of the smallest features of interest. The parameter K can then be set to a little bit less than the size of the intersection of the window with the smallest features of interest.

Edge preserving filter is the traditional term for the class of filters discussed in this paper, but that term is a serious over-simplification. A distinction must be made between edges and small detail. Abstractly, an edge occurs where one feature or subpopulation changes to another. In practice, an edge is indicated by a gradient in the local average of the gray level and this signals the location where the one feature changes to another. The term edge usually suggests that the boundary between the regions is relatively straight at the scale of the window. The speed with which the change between regions occurs affects the way that a filter treats the edge. For our purposes, we define an edge as a discontinuity in the local average gray level that happens in the span of one or two pixels. Slower changes will be called ramps in this paper. We make this distinction because of the behavior of different filters with such data. Some filters, including the contiguous K-average filter, will see ramps as a series of parallel edges. There are other kinds of features that appear in images that cannot be treated as edges when using the contiguous K-average filter. Examples include corners, one pixel wide linear features such as roads and bridges, small rectangular shapes such as buildings,

small circles such as center pivot irrigation fields, and highly curved thin features such as meandering rivers and streams.

Categories of features occupy characteristic proportions of a window. When the center pixel is on an edge in the above sense, it will have just over 50% of the window in the same region as it is. Specifically, for a W by W window, where W=2r+1 about (r+1)/W of the window will be in the same feature as the center pixel. A contiguous K-average with K less than (r+1)*W should only be averaging the center pixel with other pixels from the same region. Thus we say, in simplified form, K must be less than 50% to completely preserve edges. A different example is one pixel wide roads. A pixel on a road is part of a feature whose relative size is determined by the width of the window, usually being the window diameter W. Unlike edges, this is not a fixed percentage of the window. A final example is rectangular buildings. An aircraft scanner image of a city at 5 meter resolution will have many rectangular features ranging in area from 4 to 20 or more pixels. Only when the window is at least 5 by 5, will the larger features have a fixed size in the window.

FIXED TEMPLATE FILTERS

When comparing the performance of the contiguous K-average filter to other filters, the K-average filter would be a natural starting point. However, we find that a simpler class of filters is a more instructive place to begin. The fixed template filters simply average over K pixels in a fixed set of positions. The invariance of the selected positions leads to filters with behavior that is easier to analyze, and still provides a useful baseline for the more difficult analysis of the contiguous K-average filters.

A template is a set T of ordered pairs of elements of {-l..l} X {-l..l} for some integer l. Think of the elements of T as being offsets from the center of a (2l+1) X (2l+1) window. Here K is merely a formal window size and it could be larger than the smallest radius of a window that contains T. We define the actual window radius needed as follows: let

$$R(T) = \max (\max (|i|,|j|))$$

where the outer max is over all pairs (i,j) in T. A template is said to be anchored if and only if (0,0) is in the template. (0,0) is called the center of the template. Anchored templates contain their centers. We are mostly interested in anchored templates but we do not always need to exclude templates that are not anchored.

Before defining template filters we need to define some sets that help specify a filter's behavior near the boundary of an image. The set of allowable positions of T at (x,y) is

$$A(T;x,y) = \{ (i,j) \in T \mid 1 \le x+i \le N \text{ and } 1 \le y+j \le M \}.$$

A(T;x,y) is the set of template pairs that actually fit inside the image when the template is placed at the location (x,y). The template filter for T applied to an image P gives the following value at pixel (x,y)

$$F(T:P;x,y) = \begin{cases} 0 & A(T;x,y) \text{ empty} \\ \dfrac{1}{|A(T;x,y)|} \displaystyle\sum P(x+i,y+j) & \end{cases}$$

where the sum is over the elements (i,j) of A(T;x,y). We use the notation |S| to denote the number of elements in the set S. Note the case with value 0 will not occur for anchored templates since A(T;x,y) will always contain at least the pair (0,0). See [5,16] for some work with non-square templates using order statistics rather than averages.

The template filter F(T:P) is clearly linear in the image variable P but it is not spatially invariant because the sum changes near the boundary of the image. This means that the standard Fourier transform theory for linear spatially invariant filters cannot be directly applied. Nonetheless we can deduce many properties of these filters. We start by examining the effect of the template filters on the average gray level of an image and on the standard deviation of the set of gray levels. The hit set of T at the pixel (x,y) is

$$H(T;x,y) = \{ (i,j) \text{ pixels} \mid (x-i,y-j) \in T \}.$$

H(T;x,y) is the set of pixels (i,j) such that the template hits (x,y) when the template is placed at (i,j). We can now define the weight function image of the template T

$$W(T;x,y) = \sum \frac{1}{|A(T;i,j)|}$$

where the sum is over the elements (i,j) of H(T;x,y). Note that W(T) is an image; some of its properties will be given after the theorem below.

The mean value, Mn(P), of image P is defined in the obvious manner

$$Mn(P) = \frac{1}{NM} \sum P(x,y)$$

where the sum is over all pixels (x,y). The standard deviation can then be defined as

$$SD(P) = \sqrt{Mn((P - Mn(P))^2)}$$

Here Mn(P) is interpreted as an image with all gray levels equal Mn(P) and then the images P - Mn(P) and $(P - Mn(P))^2$ are computed by pointwise operations.

THEOREM 1 Mn(F(T:P)) = Mn(W(T)P). The product W(T)P is performed pointwise.

PROOF.

Mn(F(T:P))

$$= \frac{1}{NM}\sum \frac{1}{|A(T;i,j)|}\sum P(x+i,y+$$

$$= \frac{1}{NM}\sum\sum \frac{P(x+i,y+j)}{|A(T;i,j)|}$$

In both expressions the outer sum is over all pixels (x,y) and the inner sum is over all pairs (i,j) in A(T;x,y). We want to rewrite the sum so that it is over (a,b) where a=x+i and b=y+j. In this form the inner sum over (i,j) is rewritten as being over (a-x,b-y) and the condition is that this be in T. But that is equivalent to (x,y) being in H(T;a,b).

Mn(F(T:P))

$$= \frac{1}{NM}\sum P(a,b) \sum \frac{1}{|A(T;i,j)|}$$

$$= \frac{1}{NM}\sum P(a,b)W(T;a,b).$$

Here again the outer sums are over all pixels (x,y) and the inner sum in the middle expression is over H(T;a,b).

Corollary 2 If T is an anchored template then Mn(W(T)) = 1.

PROOF. Define a constant image P where all pixels have value 1. Then F(T:P) = P because the average value over each A(T;x,y) is clearly 1 and A(T;x,y) is never empty because T is anchored. We can find the mean of W(T) as follows:

$$\begin{aligned} 1 &= Mn(P) \\ &= Mn(F(T:P)) \\ &= Mn(W(T)P) \\ &= Mn(W(T)). \end{aligned}$$

THEOREM 3 If $R(T) < x \leq (N - R(T))$ and $R(T) < y \leq (M-R(T))$, then W(T;x,y) = 1.

PROOF. If x and y satisfy the indicated inequalities then the template T can not reach the image boundary when it is located at (x,y). Thus H(T;x,y) has the full |T| elements and A(T;i,j) = T for each (i,j) in H(T;x,y). Thus

$$W(T;x,y) = |T| * \frac{1}{|T|} = 1.$$

Thus if N and M are large with respect to R(T) then W(T) is 1's except right around the boundary.

THE EFFECTS OF FILTERS ON STANDARD DEVIATION

Removing noise from an image and also preserving features seem to be conflicting goals. To the extent that these goals do conflict, a filter must make trade-offs between methods for reducing noise and those for retaining edges and small details. To understand how noise removal is being traded for feature preservation, we must be able to measure each separately. A figure of merit that combines the amount of noise cleaning with the amount of feature preserving can be fine tuned for a specific task, but a different task will require a new figure of merit. See Pratt [37] and others [12,19,36] for examples of figures of merit and their use. In this section,we will measure the noise removing ability of our filters applied to uniform white noise images [1,15,16,27,31]. As for feature preservation, we discussed the influence that feature size and geometry should have upon parameter selection in section 2. Later, in section 6, we will give some test results comparing the ability of several filters to preserve and enhance edges that have been degraded by noise.

We want to model the interior of a single homogeneous feature. In the following, we use a normal model where the pixels are samples from a normally distributed population with some fixed mean and variance. We generated test images using this model and ran the filter with varying values of K and window size. To avoid boundary effects, the images were trimmed after filtering and then statistics were obtained for the trimmed images. After picking a window size, we filtered the image with K, the number of neighbors, varying from 1 to the entire window. We computed statistics for the filtered image and graphed the results against K.

The results for a fixed template filter provide the basis for comparison. For a template T, theorem 1 from section 3 shows that the expected value of the image mean of F(T:P) will be the same as the mean of the original image P. We would presume that the same would be true of the K-average and contiguous K-average filters and that is what we found.

The behavior of the standard deviation is more interesting. As expected, for all three types of filters, the filtered image standard deviation showed a strong and consistent decrease as K increased. In order to study the size of this decrease, we compared the decrease in variance with increased K for the contiguous K-average and the K-average filters to the decrease in variance obtained by the fixed template filters. A similar idea, using ratios of mean local variances can be found in [15]. The expected value of the image standard deviation of a fixed template filter can be found by an elaboration of the method used to derive the standard deviation of the sample mean of fixed sized samples from a distribution. We find that

$$E(SD(F(T:P))) = \frac{E(SD(P))}{\sqrt{|T|}}.$$

Let us denote the result of applying the contiguous K-average filter to image P using K neighbors as Sift(k,P). Then Sift(1,P) = P and Sift(W*W,P) is a fixed template filter whose template is the entire W X W window. For fixed template filters, the effect on the image standard deviation of using a template of size $|T|$ is to divide the standard deviation by the square root of $|T|$. Therefore we normalize the standard deviations of the filtered images as follows:

$$NSD(K,P) = \frac{SD(Sift(K,P)) * \sqrt{K}}{SD(P)}.$$

NSD is thus exactly 1 for K = 1. When K = W*W the filter is a fixed template filter and the expected value of the standard deviation will be the original standard deviation divided by W. Thus NSD returns to value 1 on average when K = W*W and in fact it is usually very close to 1 in practice. The shape of the graph of NSD against K is very consistent. The graph

rises from 1 to a maximum at 3/8 of the full window and then falls back to 1 again. The value of the maximum depends on the window size.

We would like to study the expected shape of the NSD curve and make comparisons between curves for different window sizes. To do this we will further normalize the curve and talk about the expected values of the functions rather than the values observed for any fixed random normal image. We do not have a theoretical derivation of these expected values at this time, so we can only discuss the expected values by studying the distribution of observed values. We normalize the domain of our function from [1,W*W] to [0,1] by converting from the independent variable K to the normalized variable t where

$$t = \frac{K-1}{W^2-1} .$$

The maximum now occurs at t = 3/8 for all window sizes. The next thing to do is normalize the actual value of the maximum. Let us define M(W) as the expected value of NSD at t = 3/8 for an W by W window. Next we define an image statistic

$$H(W,t,P) = \frac{NSD(K,P)-1}{M(W)-1}.$$

We can now define the function H(W,t) as the expectation of H(W,t,P) as P ranges over the given set of normal images. The notation shows no dependence on the mean and standard deviation of the normal distribution used to produce the set of images; this is because the functions do not depend on the distribution parameters. Since the K-average and contiguous K-average filters scale linearly, NSD and hence M and H are scale invariant. This means that the model image standard deviation has no effect on NSD (which was part of the point of its definition.) The two filters also shift with offsets of the data so that the distribution mean does not affect NSD or M or H. Notice that the notation does show a continued dependence on the window size W. We had hoped that there would be no dependence on W but the results reported below show that is not the case. Fixing W for a moment, we state the properties of the function H. The properties that do not follow from the definition are hereby offered up as conjectures. H(0) = 0, this is by definition. H(1) = 0, this would follow from the result on the expectation of the standard deviation of a fixed template filter. H(t) is unimodal with its maximum occurring at t = 3/8. H(3/8) = 1 by definition.

We have made many runs to begin to estimate the functions M(W) and H(W,t). Since there is necessarily variance in the results, the values obtained and the conclusions we can draw have some range of uncertainty. Let us discuss the values for M(W) first. We generated a normally distributed random image of size 250 by 1000 using a random normal generator with mean 128 and standard deviation 30. We processed the complete image with t = 3/8 for odd sized windows from 3 by 3 to 19 by 19. The results are recorded in Table 1. A border of width 10 was removed from the processed image before the statistics in the table were computed. The version of the filter used did not have a boundary rule to apply to these points. Also in the table are partial results for 21 by 21 and 31 by 31. The 21 by 21 result is based on 10580 points and the 31 by 31 result is based on 23980 points with the border increased to width 15. The program is quite slow by the time we are clustering 166 neighbors for the 21 by 21 window and 361 neighbors for the 31 by 31 window, so we only ran a sample to check extrapolations based on the other data. Also shown in the table are the corresponding values for the K-average filter. These were obtained from a 200 * 200 file, with a border of width 10 removed from the processed image prior to computation of statistics. Thus 32,400 points are used in each determination.

M(W) is roughly linear for both filters. If the filtered standard deviations did not change as W increased, the NSD's would be almost exactly linear. For the K-average filter, the filtered standard deviations do not change much for W greater than 9, but a linear fit to the whole data set would

make errors larger than the uncertainty in numbers. A quadratic fits the whole data set to adequate precision but should not be used for extrapolations. The values for the contiguous K-average filter are significantly sublinear since the filtered standard deviation continues to decrease. As for the K-average, a linear fit to the data set, up to 19, gives errors that are too large, but a quadratic gives an adequate fit. Test extrapolations using the values for W=21 and W=31 show that quadratic and even cubic fits do not give good enough predictions for W=31, although W=21 is satisfactory.

To study H(W,t), we ran many small random images rather than one big image, so that variance could be studied. Figure 1 shows the results for window sizes 3, 9 and 21 done with the contiguous K-average filter (labeled SIFT). Also shown are the results for K-average with a window size of 7. Data for the contiguous K-average, for $W \leq 9$, was produced using 100 by 100 images with a half window width trimmed after processing. H(W,t) does change systematically with W. The shape of the function for other W values can be interpolated from the curves given. Curves for the K-average filter are similar in appearance to those of the contiguous k_average filter. For $t < 3/8$, the K-average values closely match the contiguous K-average values, but are about 0.02 smaller. For $t > 3/8$, the K-average curves move up from the curve for W=3, whereas the contiguous K-average curves move down. Note that the two filters are the same when W=3.

The H(W,t) values can be approximated to an accuracy comparable to the observed standard errors using a four dimensional space of cubic splines. Let B(i,t) be the i^{th} normallized B-spline of the order 4 for the knot sequence (0.0, 0.0, 0.0, 0.0, 0.1, 0.6, 1.0, 1.0, 1.0, 1.0). See [3, p 108] for the definition of B-splines used. We can write an approximation

$$H(W,t) \approx \sum_{i=2}^{5} p(i,W)\, B(i,t)$$

where the p(i,W) are the following quadratic polynomials in W:

$$p(2,W) = (-8.73E\text{-}4)W^2 + (3.704E\text{-}2)W + 0.1865$$

$$p(3,W) = (1.27E\text{-}4)W^2 + (-3.702E\text{-}3)W + 1.1596$$

$$p(4,W) = (-4.33E\text{-}4)W^2 + (8.918E\text{-}3)W + 1.0030$$

$$p(5,W) = (5.83E\text{-}4)W^2 + (-3.451E\text{-}2)W + 0.3588.$$

This approximation is good for W ranging from 3 to 21. Significant errors occur only for $W > 9$ and $T < 0.02$. For an extreme example, with W=21 and t=0.0045, the approximation is 56% of the observed value. We comment though, for t values this small, the size of the window is irrelevant and the standard deviation is best estimated by un-normallizing the values for a smaller W.

STABLE IMAGES AND ITERATION

One property of nonlinear filters that should be studied is the question of stable images or root structures. A stable image is an image that is not changed by application of the filter. If the filter is considered as a transformation of the space of images, then stable images are the fixed points of the transformation. One reason for studying stable images is the dynamical systems viewpoint. Iterating the filter produces a dynamical system on the space of images. Typical behavior for such systems is that repeated passes through the filter produce a series of images that converge to a fixed point of the transformation. This is assuming that the dynamical system has no attractors other than its fixed points. [13] proves that the one dimensional median filter has convergence to fixed point after a bounded number of iterations. In [33] this analysis is extended to one dimensional rank order filters other then the median. The only stable images for the rank order filters, other than the median, are constants and there is again convergence to a fixed point. They also show that the recursive median filter produces a fixed point after one pass and they comment on some the difficulties on extending this analysis to two dimensional filters. The two dimensional separable median filter was analyzed in [34]; while most images converge to a fixed point under iteration, stable cycles of length two are possible. See Tyan in [17] for further discussion of root structures for the two dimensional filters.

The stable images for some fixed template filters can easily be determined. We would expect that constant images are the only stable images for fixed template filters. There is a complication however. The problem can be illustrated using the game of chess. To a chess bishop, the chess board breaks up into two separate domains, the white squares and the black squares. Although the white and black squares seem to be close together, the diagonal moves of the bishop make them separate worlds. Similarly the pixels of the image grid will break up into mutually unreachable sets for certain templates. It is also possible that information from pixel x can reach pixel y but not vice versa.

To make this precise, we define a reachability graph for anchored templates. The reachability graph is a directed graph whose vertices are the pixels. This is illustrated in figure 2 (b), where the pixels have been labeled 1 through 12. The template appears in part (a) with (0,0) marked by an asterisk. The other two cells of the template are (0,1) above and (-1,-1) below and to the left. Placements of the template on the grid give the edges of the reachability graph. A placement of the template produces edges directed from the center to each pixel covered by the template (including a loop which has been omitted that goes from the center to itself because the template is anchored.) Part (c) of the figure shows the resulting directed graph. A directed graph is said to be strongly connected if any vertex can be reached from any other vertex by following a directed path.

THEOREM 4 If the reachability graph of a grid for an anchored template is strongly connected then the only stable images for the fixed template filter are the constant images.

PROOF Look at the maximum value of a stable image. This maximum is supposed to be the average of the values covered by the template when centered at the maximum pixel. The only way the average can equal the maximum is if all the pixels covered take that same maximum value. We have now propagated the maximum value one step on all directed paths from the original pixel. By induction, we can continue down all the directed paths and by hypothesis we will get the entire image this way.

If the reachability graph is not strongly connected it is still possible that the only stable images are the constant images. This is the case for the template in figure 2. A complete analysis of the stable images for a general template could be done using Markov Chain theory, but that would take us too far afield. Also, note that some

unanchored templates have no stable images except the constant zero map. The left shift (with $T=\{(1,0)\}$) is an example.

This sort of result leads one to ask how the value of the constant in a limiting constant image depends on the original image. This question can be reformulated as a standard mathematical task, the computation of eigenvectors. Let us represent the image's gray levels as a row vector of values and the anchored fixed template filter as a matrix, F, that acts on the image vectors by multiplication on the right. The columns of F will be all zeros except for $a=|A(T;x,y)|$ entries each of which is $1/a$. Thus if J is a row vector with all entries equal 1, we obtain $J*F = J$. This shows that F has a left eigenvalue of 1 with eigenvector J. Since the template is anchored, F has entries $1/a$ on it diagonal. If we apply the column form of Gershgorin's theorem [see 35, p317, and p321 problem 6], we see that the Gershgorin circles have centers $1/a$ and radii $(a-1)/a$. Therefore 1 is the dominant eigenvectors of F and there are no other eigenvalues of F with eigenvalues greater than or equal to 1 in absolute value. Since the right eigenvalues of F are the same as the left ones, we conclude that 1 is also the dominant right eigenvalue. Let C be a right (column) eigenvector for the eigenvalue 1 and suppose that C has been scaled so that the sum of its entries is 1. Then C satisfies $J*C=1$ and $F*C=C$.

LEMMA 5 Suppose that constant images are the only fixed points for the fixed template filter of T on a certain grid and that every image tends to a constant image under iteration by the filter. Then the value of the constant for image A is $A*C$.

PROOF
$$\begin{aligned} A*C &= A*F*C \\ &= A*F*F*C \\ &\cdots \\ &= A*F^n*C. \end{aligned}$$

Let L be the constant value at each pixel of the limit of A under iteration by F. Then

$$\begin{aligned} A*C &= A*F^n*C \\ &= (\lim_{n\to\infty} A*F^n)*C \\ &= (LJ)*C \\ &= L(J*C) \\ &= L. \end{aligned}$$

If T is a template with a strongly connected reachability graph for a given grid then the hypothesis of the above lemma holds. In that case the matrix F is regular and the Perron–Frobenius theorem [see 35, pp 374–376] shows that C is unique and has all entries greater than or equal to 0 (and hence less than or equal to 1.) We can summarize by saying that the limit value of a image A under iteration by F is a convex combination of the values of A. For certain symmetrical templates, the coefficients of this convex combination are easy to describe.

An anchored template T is said to be centrosymmetric if and only if (x,y) in T implies (–x,–y) is also in T. Another way of stating this condition is that the template is fixed by a half turn about its center. Centrosymmetric is equivalent to

$$(x,y) \in H(T;u,v) \Leftrightarrow (u,v) \in H(T;x,y) \quad (1)$$

since (u–x,v–y) in T if and only if (x–u,y–v) in T. Yet another restatement of centrosymmetry is that the reachability graph of T on any rectangular grid is undirected. This follows from the fact that Eq. 1 implies that two vertices of the reachability graph that are joined by an edge in one direction must also be joined by an edge in the reverse direction.

THEOREM 6 A centrosymmetric template T on any grid has a right eigenvector C for eigenvalue 1 whose component for the pixel (x,y) is $|A(T;x,y)|$.

PROOF The column of the matrix F corresponding to the pixel (x,y) has exactly $|A(T;x,y)|$ nonzero entries, each of which is $1/|A(T;x,y)|$. The locations of the nonzero entries are the pixels covered by T when it is placed at (x,y). Consider the row of F

corresponding to (x,y). The nonzero entries of this row are the pixels (u,v) such that T covers (x,y) when T is placed at (u,v). Since T is centrosymmetric, these are exactly the same pixels as those giving the nonzero entries in the column corresponding to (x,y). Consider the product of the row corresponding to (x,y) with the column vector C. Each nonzero entry of the row will have a value $1/|A(T;u,v)|$ and will be multiplied by $|A(T;u,v)|$ giving a product of 1. There are exactly $|A(T;x,y)|$ such products, which yields a total value of $|A(T;x,y)|$. But this is just the value of C again and we have $F*C = C$ as claimed.

This result shows that the limit value of an image P under iteration by a fixed template filter with a centrosymmetric template will be close to the average value of P. The pixels near the boundary will be weighted less than those in the center but for a grid that is much larger than the template, most pixels are away from the boundary. For templates that are not centrosymmetric, the convex combination can be very different from this pattern. If the template is unbalanced with more cells in some direction, the components of the eigenvector will not be all roughly similar in size. Components in the direction opposite that of the template will be very small and components in the direction of the template unbalance will be much larger. The values of the image in the direction of the unbalance will have far greater influence on the limit value than the values of the pixels in the opposite direction.

These results for fixed template filters are in accord with known smoothing and blurring effects of this kind of filter. Repeated passes through a typical fixed template filter produce a series of images that converges to a constant image. These filters are known for their ability to remove noise, the drawback being that they also attempt to remove everything else, too. These filters can still be useful since a pass through one will smooth out the very fine grain detail which may be considered noise and yet not do unacceptable damage to the larger features which are considered to be the true content of the image.

The set of stable images for the contiguous K-average filter depends on K. The larger K is, the smaller the set of stable images. For $K = 1$, all images are stable. For larger K we must specify a tie break rule and a boundary pixel rule to obtain a well defined transformation. The set of stable images for a given K will depend on the choice of tie break rule and boundary pixel rule, as will be illustrated below. The results below are not intended to be a complete analysis of all cases. We have picked some illustrative examples where the stable set has a simple, clean description and the proof is easy. In all the examples below, the function of the K neighbors is their average. The definitions of the various rules are given in section 2.

THEOREM: 7 Let the contiguous K-average filter be specified as using a first appearance tie break rule and the truncated window boundary rule. The stable images for the $K = 2$ case of this filter are those images where every pixel is adjacent to another pixel with the same gray level.

Proof. The specifications for the filter are designed to insure that the filter value is always the average of the value at the pixel and at one of its neighbors. The average of two numbers will only equal one of them if the two numbers are equal to each other. Thus in a stable image, every pixel is adjacent to another with the same value.

The above set of images will be stable images for the $K = 2$ case of almost any specification of the filter that uses averaging, but specifications other than the one above generally allow additional stable images. For example, a boundary rule that causes corner or edge pixels to be updated using only their own value will have stable images that obey the above adjacency rule in the interior but not on the corners or edges. The set of stable images for $K = 2$ is thus seen to be quite large, even though it is of course much smaller than the set for $K = 1$. The idea of theorem 7 does not generalize to values of K greater than 2. For $K = 3$ (and greater) there are stable images with pixels that have a value that is not shared by any other nearby pixel. These pixels sit between groups of pixels with higher values and with low values, and

they balance the two groups to preserve their own value. The idea of theorem 7 is mostly correct, however. Most pixels of a typical stable image are contained in groups of pixels that all have the same value.

At the other end of the range, when K is equal to the full window size, we have a fixed template filter. There is no need for a tie break rule in this case and the proportional boundary rule reduces to the truncated window rule. By theorem 4 the set of stable images is about as small as it possibly can be.

We will now show that the set of stable images is just the set of constant images for values of K much smaller than the full window. In fact, K need only be somewhat over half the window to force stable images to be constant images. In the following, the window size will be W by W, where W=2r+1.

THEOREM 8 Let the filter be specified as using a first appearance tie break rule and the truncated window boundary rule. If K is greater than W*(r+1) then the only stable images are constant images.

PROOF. Let I be a stable image and consider pixels where I attains its maximum value. Since the average of the K neighbors is the maximum observed value, all K pixels must take this maximum value. Take the set of pixels of I that have this maximum gray level and look at a connected component of it. Could a component have a rightmost member before the right edge of the image? No, it cannot. A rightmost member must have at least K elements of the window that are also part of the component. But the maximum number of elements of the window that can all be no farther right than the center is W*(r+1) and we have more than that many. In a truncated window on the boundary, K is an even larger share of the window and so a rightmost member is still impossible except when the truncated window's center pixel is at the actual edge of the image and there are no pixels farther right. We see that the component must extend all the way to the right edge of the image. But the same argument shows that the component must also extend to the left edge of the image and to the top edge and to the bottom edge. Now consider the set of pixels where I takes its minimum value. By the same argument as above, a connected component of minimum value pixels must extend to the right, left, top and bottom edges of the image. This is incompatible with the extent of the maximum values component unless the minimum value equals the maximum value and the image is a constant image.

This result remains true for other specifications of the filter. We used the first appearance tie break rule to insure that the average was always over K numbers except at the boundary. Even with the balanced tie break rule, the average can be the maximum or minimum only if all pixels involved take the same extreme value. Thus the above proof also works for the balanced tie break rule. We used the truncated window boundary rule to insure that the only time the average was over less than K numbers was when it was over the entire window. The proportional boundary rule reduces K for windows in the boundary but it does so in a fashion that leaves the above argument valid. In boundary windows near the top or bottom but away from the right edge, K is reduced but it is still large enough so that K pixels cannot all fit into the window without at least one of them being to the right of the center. For windows in the boundary near the right edge, K may be reduced enough so that all K pixels will fit in the window with none to the right of the window center; however, the gap left is not large enough for the component of minimum value pixels to get past. Thus a different value for maximum and minimum would still leave two components trying to touch or get very close to all four edges and that is still impossible.

All the results above were derived assuming exact arithmetic. In fact, real world image processing uses floating point or even integer arithmetic. Limited precision arithmetic complicates the picture presented above. In limited precision arithmetic the average of two different

numbers could be equal to one of them. And the average of several numbers could be an extreme observed value without all of the numbers also being that same extreme value. The various sets of stable images will therefore be enlarged somewhat by arithmetic oddities when limited precision arithmetic is used.

To show the effect of iterating to stability we give an example using a remote sensing image. The raw image is figure 3a. It is a 200 by 200 portion of channel 6 (11.2 μm to 12.2 μm) of some data acquired by a NASA airborne scanner called TIMS (Thermal Infrared Multispectral Scanner.) This particular acquisition was taken at night over rural Alabama. The data is recorded in an 8 bit format, although the actual dynamic range in this image is only about 5 or 6 bits. The image is very noisy and some of the noise forms coherent features (the many dark, horizontal lines.) We used filter parameters K = 8 and a 5 by 5 window. Thirty iterations produced the stable image in figure 3b. Notice that the noise lines created some artificial horizontal edges that were then built in to the image by the filter. The horizontal striping noise is not nearly as pronounced in the other channels of the data; we could have used multichannel distances if we were really interested in clearing out all of the noise from this image. We did not do this because we wanted to show that a dogged determination to preserve all edges can be taken too far.

In a series of tests with a diverse set of real images, we observed a consistent response to repeated passes through the contiguous K-average filter. This response was a rapid stabilizing upon an image and then all further change being confined to a scattered set of fixed locations with an extremely small total area. We will illustrate this behavior with some examples. In figure 4 we give examples of semilog plots of the number of pixels changed at each iteration of a filter. The measure of change used, the number of pixels changed in value, is a reasonable measure in this case because the images in question are 8 bit data; so that very small changes in value are impossible. If the images had been floating point data, we could not have done this since almost all pixels would have changed their values at every iteration (but most only by a very small amount.)

This measure of change clearly reveals the two phases of the convergence to stability. The initial phase would last from 5 to 15 iterations, depending on how noisy the original image was. During this phase the decline preceded at an approximately geometric rate modeled by a formula of the form

$$C = M R^n$$

where C is the measure of change and n is the iteration number. On a semilog plot like figure 4, a decline at an exactly geometric rate would appear as a straight line. For all images tested, we found that change initially decreased at rates that could be reasonably well modeled by geometric series.

Once the geometric decline phase was finished, further iterations of the filter produced a new pattern of change. This change was confined to a few scattered areas of the image, whose total area was well below 1 percent of the image. For the TIMS-ag data all changes past the twentieth iteration occurred at just two locations. The number of pixels changing in this phase of the example, out of 40,000 total, ranged from 11 down to 1 at the end. For all the images, the changes often followed a pattern where a feature slowly engulfs a small region on its border. We like to think of the process as one in which the filter nibbles away at a little, unstable area, taking out a bite on every iteration.

The major variable controlling the differences between data sets appears to be how heterogeneous are the features in the image. Complex images start with an absolutely greater number of changes per unit area, therefore they require more iterations to reach stability. A second variable may be the nature of the imaging mechanism. Some methods of acquisition are inherently two dimensional, for example airborne and satellite scanners. Others are inherently three dimensional. Specifically, in this case the MRI and the thin section of sandstone. Three dimensional boundaries

produce ramps when projected into a two dimensional cross section. We suggest that the noticeably different slopes between these two data sets and the other data sets is caused by this.

DEMONSTRATIONS

In this section we will give the results of some tests comparing the contiguous K-average filter to other filters. We will also sketch three sample applications using the filter. These are taken from research projects at the Science and Technology Laboratory. Our choices of test filters and of applications were determined by convenience, but we believe that they are representative. The other filters for the comparison tests: averaging, median and K-average , were selected from the filters already available in our image processing package. We believe that these tests give a fair assessment of the capabilities of the contiguous K-average filter and show its power as a edge preserving filter.

Two examples use synthetic images. These were constructed specifically to show the result of executing various filters on selected types of features. The first images are used primarily to illustrate behavior with respect to well defined features. The second example illustrates behavior with respect to edges in the presence of noise. Employing artificial data makes it simple to generate qualitative measures of each filter's effects. A third example uses multichannel imagery from an airborne sensor. This is given to show the efficacy of "guiding" with real data.

Stripes

Construction

For the first comparison test, two sets of three artificial images were generated. All consist of parallel, diagonal stripes of varying widths against a constant background, see [24 and 26] for similar test images. The stripes are at a 45° angle; background intensity was set to 5. The horizontal spacing between between stripes was set to 4 pixels. The stripes themselves have increasing widths as we scan across the image from left to right. The first stripe has width 1, the next width 2, and so on. We used three different patterns for the gray levels of the stripes; in all cases the gray level within each stripe is constant. The three images in each set differ in how the digital values are given to the stripes. One image has the stripes set at a constant value, a second has increasing values from stripe to stripe, starting just above background; the third has a decreasing pattern starting from an initial high value. To one set of images, noise was added; to a second set, multiplicative noise was applied. The constant images have stripes set to 206 in the additive noise set, and 128 in the multiplicative noise set. In both sets, the increasing image has the value increase 1 count from one stripe to next, starting at 6 (1 above background) for the width one stripe. In the images with a decreasing pattern in the stripes, the value decreased by 1 count from one stripe the next, starting at 206 for the width one stripe, additive noise, starting at 128 for the multiplicative noise. The images have 100 rows and 200 columns. To avoid boundary effects all statistics were computed using only rows 16 through 85 and columns 16 through 185.

These images were designed to test and demonstrate some critical aspects of filters in actual applications: the relationship between feature size and window size, and contrast between feature and background. Retention of features which are one pixel wide and one count above background simulates some of the most rigorous real world requirements for filters. Spacing between stripes was set so that 3x3 operators, working on a stripe would not receive interference from other stripes, but larger filters would.

The noise added to both sets came from a random noise image having statistics as follows. The 5762 points in the background area had mean 127.4599 and standard deviation 29.5148. The 6138 points in the stripes area had mean 127.3763 and standard deviation 29.8912. Additive noise was inserted according to the equation,

$$DV_{out} = DV_{in} + (DV_{noise} - 128) * 0.04$$

This produced a noisy image with background mean 4.9039 and standard deviation 1.2136 and with stripes mean 205.9110 and standard deviation 1.2303.

Multiplicative noise was inserted according to

$$DV_{out} = DV_{in} + (DV_{noise} - 128) * \frac{DV_{in}}{105}$$

This produced a noisy image with background mean 5.0252 and standard deviation 1.4366. The statistics for the stripes area depended on the pattern of values for the stripes. Image mean and standard deviations after filtering are given in Table 2.

Fixed Template, average

The first test filter was a straight average over a W by W window. This is effectively a fixed template filter with the template being a W by W square, anchored at the center. We used W equal 3,5 and 7 and the blurring effects of this kind of filter are readily apparent. The filtered images had profoundly altered means and greatly increased standard deviations. Note that in the stripes area of the uniform pattern, the step from W=5 to W=7 has produced so much more blur that the standard deviation actually decreased! In the increasing pattern, the stripes have very low contrast to began with and the filter virtually obliterates all the narrower stripes. Although the statistics given do not show this, the averaging filter (particularly W=7) has been very successful in removing the fluctuations produced by the noise. However, the price in edge destruction has been high. When the standard deviation is measured within features, the decreases in standard deviation from noise removal are swamped by the increases caused by blurring in other populations.

Fixed Template, median

The next test filter, a median filter, is a decent edge preserving filter. As above, we used square windows with sides W equal 3,5 and 7. For W equal 3 and 5, the filter performs well, significantly decreasing the fluctuations and irregularities introduced by the noise without significantly damaging the edges except the first stripe in the W=5 case. For W=5, the first stripe is too small to make a majority in the window and so the filter cannot "see" the stripe and it is wiped off the image. This same phenomenon shows up strongly in the W=7 case. Here the background between the stripes is too small to be "seen" by the filter and the background areas are obliterated, with the result being the dramatic increases recorded in the table.

K–average

We will discuss the last two filters, K-average and contiguous K-average together. The general size of the standard deviations shows that either filter can give reasonable results for any size window provided an appropriate size K is selected. The two parameters working together give good control of the cutoff size. K-average does suffer from a window size problem, however. Note that the W=K=5 case achieved a smaller standard deviation than the W=K=31 case. When W=31, the K-average filter has 961 pixels to search for 30 values similar to the center pixel. Even if the center pixel is somewhat of an outlier, there will be other outliers in the same general vicinity and the algorithm will pick some of them before working its way down to the more frequent, more typical values. The contiguous K-average filter does not suffer from this problem since no matter how big the window is, it must still select pixels next to pixels that were previously selected. Thus the W=K=31 case of the contiguous K-average filter gives a distinct improvement over the W=K=5 case.

K–average, guided

Both filters make substantial gains when they are guided by a noise free channel. The contiguous K-average filter with W=K=5 was improved more by being guided than by increasing the parameters to W=K=31. The K-average filter lost the large window effect and substantially improved from W=K=5 to W=K=31 when guided. For both filters, the guided W=K=31 case produced an image with very little noise left. The most stringent

test for the filters was the first stripe of the increasing pattern. It was only one wide and one above background. For both types of noise, the unguided filters lost the first stripe and most of the second stripe, too. The guided filters almost completely recovered both stripes. Averaging the correct 31 values was easily enough to give a result that rounded to one higher than the result of averaging 31 background values. The guiding technique is very powerful when it can be used; we give an example application below.

Ramps

Construction

In practice images are frequently filtered prior to execution of other operations. Within our organization, visual interpretation frequently is not the reason for filtering data. Therefore, to test and demonstrate the effectiveness of filters we must determine how filters of interest will affect the results of further processing. We feel that tests of a filter as a step to be followed by other types of processing should be part of the standard test suite, and there are moves in this direction [see for example 5,7,9,10,28,41]. One example of a type of further processing is the computation of the boundary length between two classes. This occurs commonly in our work, such as when measuring the nature of the interface between water and land classes in marshes, or in determining the nature of the interface between sand grains and pore space in petroleum reservoir rock.

For this example application, we created a synthetic image and two sets of synthetic noise. The synthetic image could be thought of as parallel strips of land against a background of water. The strips of land have sloping beaches on the left sides and cliffs on the right sides. Specifically, the image consists of vertical bands which, read left to right, go from a low background value (103), fifteen pixels wide, slowly up to a higher value (153 or 163). After a plateau of fifteen pixels at the high value, the level abruptly drops to the background value, and then the cycle of rise, plateau and drop begins again. The slope or rate of change from background to peak goes from 45°, to 60°, then 75°, 80°, and 85°. The visual effect is one of increasingly steep ramps. We made this construction to imitate the common real world situation where the boundaries between materials of interest are defined by gradational changes, not discontinuities.

Two varieties of noise were then added to the image of ramps. One consists of normally distributed random noise; the second consists of a pair of sinusoidal waveforms of unequal amplitude and wavelength. The two noises together very closely mimic the type of noise in some airborne scanner imagery (see figure 7, channel 8 for an example.) Each of the two noise sources were added to the image of the ramps at 5 different intensities, creating 25 images, Figure 5. Each image was then filtered and classified into 2 classes - background ("water") and a peak class ("land") - using a minimum spectral distance algorithm that degenerates into a simple threshold for one channel data such as this. All pixels were forced by the software into one or the other class. The length of the interface between the two classes was then computed. The process was then repeated using different filters.

Results

The values of interface length are shown in Figure 6. The first effect to consider is the peaks for images 1, 6, 11, 16 and 21. These are the cases where the harmonic noise is maximum. It is so large that it caused the K-average and contiguous K-average filters to generate values at the harmonic noise peaks that were large enough to be misclassified as land. These two filters also broke the vertical strips of land up into disconnected islands that were then sometimes connected by horizontal bridges of misclassified water and noise. This created a classified image made up of numerous horizontal islands, with an attendant substantial increase in shoreline length. The average and median filters did not break up the strips of land, but they produced a wavy shoreline whose length was

more than triple the length of the straight shorelines produced when there is no noise.

The other effect to consider is the difference in results between the average and median filters and the K-average and contiguous K-average filters. Thresholding after K-average type filters produced complicated, irregular boundaries with very large interface lengths. The average and median filters restored all but the worst images to smoothly changing gradients that thresholded into relatively straight boundaries with short interface lengths not far above the no noise "true" answers.

CAMS Data

The last application demonstrates how the multichannel form of the filter can use relatively noise free channels to guide the clean up of noisy channels. As indicated in the comparison tests above, this guiding technique can achieve levels of noise reduction that are not otherwise possible. The basic requirement for the technique to work is multiple channel data where the features in each channel exactly correspond to the features in the other channels. There need be no similarity in the actual gray levels of the features, only that the features occupy the same location pixels in each channel. In other words, the channels must be rectified to each other, and where there is a boundary in one channel, there should be a boundary in any channel used.

The requirement for mutually rectified channels is fulfilled in the images produced by the multichannel scanners used in remote sensing. Figure 7 shows three channels of an image acquired by the CAMS scanner that is operated by our laboratory. The device acquires 9 simultaneous channels of imagery from the visible, near infrared and thermal infrared. Image quality and the nature of the noise is highly uneven between channels. The data shown is from a flight along the coast of Louisiana at Atchafalaya Bay. Principal features in the image are the bayous, canals and turbidity patterns in the ocean. We used the relatively low noise visual channels like channel 4 to guide the clean up of the noisy near infrared channels 7 and 8. We must remark that a photograph like figure 7 cannot fully convey the scope of the noise in channels 7 and 8. The contrast between the land and the water is too great to allow a photograph to fully reveal the noise in both.

The contiguous K-average filter was run over channels 2–5, 7, and 8. We did not use channel 1 because of intense haze, which results in a very low contrast image. This violates the second necessary condition. Channel 6 was not used because of its redundancy in this image with channel 5. Channel 9 was not used because it is the thermal infrared channel. Features in the thermal infrared are often poorly correlated with features in the visible and near infrared bands, due to the fundamentally different physics involved. Thus the second basic requirement is not meet.

For this demonstration, the high noise channels, 7 and 8, were given a weight of zero, other channels were each weighted 1. We used a window size of 5 and a K of 8. These choices were based on several considerations. One "noise" has coherence between channels, and appears as linear "features;" these we wanted to try to break up. This requires a K larger than the window size. Further, we felt that no feature of interest was smaller than approximately 8 pixels and that one pixel wide features were only moderately significant. The thin straight features, oil company barge canals, are at least two pixels wide. If we had been more concerned with one pixel wide features, we would have increased the window size to 7.

The filter greatly reduced the noise and significantly improved the quality of channels 7 and 8. For example, notice how the small bayous on the land in the lower left part of channel 7 have been brought out and clarified by the removal of the dark speckle noise around them. Significant noise remains in channel 8. The remaining noise has a "wavelength" greater than that which can be removed by a window of size 5.

SUMMARY

The contiguous K-average algorithm is a superior, edge preserving filter for image processing applications. With suitable parameter values, it can preserve edges and other detail better than the other filters tested and better than any other filter of which we know, while still removing noise at rates that do not lag greatly behind similar sized mean filters. Based on many studies of noise cleaning by local filters, for examples see [6,7,26 and 27], we believe that mean filters can serve as the performance goal for the class of general purpose local filters. Fixed template filters provide a method of adding the parameter K to mean filters so that we can get mean filters the same size as general contiguous K-average filters. Using the comparison to mean filters, we find a consistent relationship of K and window size to reduction of the standard deviation in filtered white noise. The relationship is described using our M and H functions. We are convinced that the behavior we have described is real and highly reproducible. We would like to see some more proofs and, even more, an explanation for this phenomenon.

The filter's two parameters allow a great deal of control over the filter's action without requiring powerful assumptions about the image noise in order to estimate noise parameters. We believe that setting filter parameters using the expected meaningful content of the image, rather than an assumed structure for the image noise, is a positive and rational step. Furthermore, it is a step toward the methods used by the most powerful image processing systems known, the visual system of living creatures.

We have begun the study of the behavior of the contiguous K-average filter under iteration. There is more to do. We would like to know if the dynamical system produced by this filter can have an (irreducible) attractor other than a fixed point. Characterization of the set of fixed points for all filter parameter values would be interesting if the final result is not too ugly. A comparison of the sizes of the regions of attraction for different fixed points is a topic that has not been well studied in general and in this particular case, the answer may be non-trivial.

As illustrated by our results using shore line length,it is not easy to judge the performance of a filter as a preprocessing stage for other operations. However, we feel that it is necessary. Filters are commonly used as preprocessing stages and their evaluation in this context is probably more important for end users than evaluations in isolation.

The technique of using some channels to guide the cleaning of other, noisier channels is a powerful device that deserves more attention. One promising application is using visual imagery to guide the cleaning of radar data.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation for the work done by the numerous programmers, both of NASA and those employed by Lockheed Engineering and Sciences Company, who created the programs used in this research. Special thanks are extended to Ronnie Pearson and David Walters, who made several modifications to the program. Funding for the first author was provided by NASA under programs in the division of Technology Utilization, Office of Commercial Programs, and the division of Earth Science and Applications, Office of Space Science and Applications.

BIBLIOGRAPHY

1 D.G. Bailey and R.M. Hodgson, "Range filters: local-intensity subrange filters and their properties," Image Vision Comput., Vol. 3, 1985, pp. 99–110.

2 A.M. Beverley and P.G. Penton, editors, ELAS, Science and Technology Laboratory Applications Software Vol II, User Reference; Earth Resources Lab. Report No. 183; April 1987; Stennis Space Center, NASA; SSC, Mississippi 39529.; pp. 1-23.

3 C. de Boor, A Practical Guide to Splines, Springer-Verlag, New York, 1978.

4 A.C. Bovik, T.S. Huang and D.C. Munson, "A Generalization of Median Filtering Using Linear Combinations of Order Statistics," IEEE Trans. Acoust., Speech, Signal Processing ASSP-31, No. 6, Dec. 1983, pp. 1342–1350.

5 A.C. Bovik, T.S. Huang and D.C. Munson, "The Effect of Median Filtering on Edge Estimation and Detection," IEEE Trans. on Pattern Anal. Mach. Intell. PAMI-9, No. 2, Mar. 1987, pp. 181–194.

6 R.T. Chin and C.L. Yeh, "Quantitative Evaluation of Some Edge-Preserving Noise-Smoothing Techniques," Computer Vision, Graphics, and Image Processing 23, 1983, pp. 67–91.

7 J.L. Cushnie and P. Atkinson, "Effect of Spatial Filtering on Scene Noise and Boundary Detail in Thematic Mapper Imagery, "Photogrammetic Eng. and Remote Sensing, Vol. 51, No. 2, Sept. 1985, pp. 1483–1493.

8 L.S. Davis and A. Rosenfeld, "Noise Cleaning by Iterated Local Averaging," IEEE Trans. Syst., Man, and Cybern. SMC-8, No. 9, Sept. 1978, pp. 705–710.

9 L.S. Davis, A. Rosenfeld and J.S. Weszka, "Region Extraction by Averaging and Thresholding," IEEE Trans. Syst., Man, and Cybern. SMC-5, May 1975, pp. 383–388.

10 J.M. Durand, B.J. Gimonet and J.R. Perbos, "SAR Data Filtering for Classification," IEEE Trans. Geoscience and Remote Sensing, GE-25, No 5, Sept. 1987, pp. 629–637.

11 J.O. Eklundh and A. Rosenfeld, "Image Smoothing Based on Neighbor Linking," IEEE Trans. on Pattern Anal. Mach. Intell. PAMI-3, No. 6 Nov. 1981, pp. 679–683.

12 V.S. Frost, J.A. Stiles, K.S. Shanmugan and J.C. Holtzman, "A Model for Radar Images and Its Application to Adaptive Digital Filtering of Multiplicative Noise," IEEE Trans. on Pattern Anal. Mach. Intell. PAMI-4, No. 2, Mar. 1982, pp. 157–166.

13 N.C. Gallagher and G.L. Wise, "A Theoretical Analysis of the Properties of Median Filters," IEEE Trans. Acoust., Speech, Signal Processing ASSP-29, No. 6, Dec. 1981, pp. 1136–1141.

14 R.M. Haralick and L. Watson, "A Facet Model for Image Data," Computer Graphics Image Processing 15, 1981, pp. 113–129.

15 G. Heygster, "Rank Filters in Digital Image Processing," Computer Graphics and Image Processing 19, 1982, pp. 148–164.

16 R.M. Hodgson, D.G. Bailey, M.J. Naylor, A.L.M. Ng and S.J. Mc Neill, "Properties, implementations and applications of rank filters," Image Vision Comput., Vol 3, 1985, pp. 3–14.

17 T.S. Huang, Ed., Two-Dimensional Signal Processing: Transforms and Median Filters, Springer-Verlag, New York, 1981.

18 L. Kitchen, M. Pietikainen, A. Rosenfeld and C.Y. Wang, "Multispectral Image Smoothing Guided by Global Distribution of Pixel Values," IEEE Trans. Syst., Man, and Cybern. SMC-13, No. 4, July 1983, pp. 626–631.

19 L. Kitchen and A. Rosenfeld, "Edge Evaluation Using Local Edge Coherence," IEEE Trans. Syst., Man, and Cybern. SMC-11, No. 9, Sept. 1981, pp. 597–605.

20 D.T. Kuan, A.A. Sawchuk, T.C. Stranda and P. Chavel, "Adaptive Noise Smoothing Filter for Images with Signal-Dependent Noise," IEEE Trans. on Pattern Anal. Mach. Intell. PAMI-7, No. 2, Mar. 1985, pp. 165–177.

21 J.S. Lee, "Digital Image Enhancement and Noise Filtering by Use of Local Statistics," IEEE Trans. on Pattern Anal. Mach. Intell. PAMI-2, No. 2, Mar. 1980, pp. 165–168.

22 J.S. Lee, "Refined Filtering of Image Noise Using Local Statistics," Computer Graphics Image Processing 15, 1981, pp. 380–389.

23 J.S. Lee, "A Simple Speckle Smoothing Algorithm for Synthetic Aperture Radar Images," IEEE Trans. Syst., Man, and Cybern. SMC-13, No. 1, Jan. 1983, pp. 85–89.

24 J.S. Lee, "Digital Image Smoothing and the Sigma Filter," Computer Vision, Graphics, and Image Processing 24, 1983, pp. 255–269.

25 A. Lev, S.W. Zucker and A. Rosenfeld, "Iterative Enhancement of Noisy Images," IEEE Trans. Syst., Man, and Cybern. SMC-7, No. 6, June 1977, pp. 435–442.

26 G.A. Mastin, "Adaptive Filters for Digital Image Noise Smoothing: An Evaluation," Computer Vision, Graphics, and Image Processing 31, 1985, pp. 103–121.

27 P.H. Mowforth and Z.P. Jin, "Implementation for noise spuppression in images," Image Vision Comput., Vol 4, No. 1, Feb 1986, pp. 29–37.

28 P.W. Mueller and R.N. Hoffer, "Low-Pass Spatial Filtering of Satellite Radar Data," Photogrammetric Eng. and Remote Sensing, Vol.55, No. 6, June 1989, pp. 887–895.

29 M. Nagao and T. Matsuyama, "Edge Preserving Smoothing," Computer Graphics Image Processsing 9, 1979, pp. 394–407.

30 K.A. Narayanan and A. Rosenfeld, "Image Smoothing by Local Use of Global Information," IEEE Trans. Syst., Man, and Cybern. SMC-11, No. 12, Dec. 1981, pp. 826–831.

31 P. Narendra, "A Separable Median Filter for Image Noise Smoothing," IEEE Trans. on Pattern Anal. Mach. Intell. PAMI-3, No. 1, Jan. 1981, pp. 20–29.

32 A. Nieminen, P. Heinonen and Y. Neuvo, "A New Class of Detail-Preserving Filters for Image Processing," IEEE Trans. on Pattern Anal. Mach. Intell. PAMI-9, No. 1, Jan. 1987, pp. 74–90.

33 T.A. Nodes and N.C. Gallagher, "Median Filters: Some Modifications and Their Properties," IEEE Trans. Acoust., Speech, Signal Processing ASSP-30, No. 5, Oct. 1982, pp. 739–746.

34 T.A. Nodes and N.C. Gallagher, "Two-Dimensional Root Structures and Convergence Properties of the Seperable Median Filter," IEEE Trans. Acoust., Speech, Signal Processing ASSP-31, No. 6, Dec. 1983, pp. 1350–1365.

35 B.Noble and J.W.Daniel, Applied Linear Algebra, 3rd Ed., Prentice-Hall, Englewood Cliffs, NJ, 1988.

36 T. Peli and D. Malah, "A Study of Edge Detection Algorithms," Computer Graphics and Image Processing 20, 1982, pp. 1-21.

37 W.K. Pratt, Digital Image Processing, Wiley, New York, 1978, pp. 495–501.

38 S. Ranade and M. Shneier, "Using Quadtrees to Smooth Images," IEEE Trans. Syst., Man, and Cybern. SMC-11, No. 5, May 1981, pp. 373–376.

39 A. Scher, F.R. Dias Velasco and A. Rosenfeld, "Some New Image Smoothing Techniques," IEEE Trans. Syst., Man, and Cybern. SMC-10, No. 3, Mar. 1980, pp. 153–158.

40 I. Scollar and B. Weidner, "Image Enhancement Using the Median and the Interquartile Distance," Computer Vision, Graphics, and Image Processing 25, 1984, pp. 236–251.

41 F. Tomita and S. Tsuji, "Extraction of Multiple Regions by Smoothing in Selected Neighborhoods," IEEE Trans. Syst., Man, and Cybern. SMC-7, Feb 1977, pp. 107–109.

42 R. Wallis, "An Approach to the Space-Variant Restoration and Enhancement of Images," in Image Science Mathematics, C.O. Wilde and E. Barrett, Eds, Proc. Symp. on Current Math. Problems in Image Science, Monterey, Cal., Nov. 1976, Western Periodicals, North Hollywood, 1977, pp. 10–12.

43 D.C.C. Wang, A.H. Vagnucci and C.C. Li, "Gradient Inverse Weighted Smoothing Scheme and the Evaluation of its performance," Computer Graphics Image Processing 15, 1981, pp. 167–181.

44 J.W. Woods and J. Biemond, Comments on "A Model for Radar Images and Its Application to Adaptive Digital Filtering of Multiplicative Noise," IEEE Trans. on Pattern Anal. Mach. Intell. PAMI-6, No. 5, Sept 1984, pp. 658–659.

TABLE 1

Estimated Values for M(W)

Window Width	Number of Neighbors	Contiguous K-average Standard Deviation	Contiguous K-average NSD	K-average Standard Deviation	K-average NSD
3	4	21.8664	1.4558	22.0817	1.4642
5	10	18.2429	1.9204	20.8624	2.1872
7	19	16.2509	2.3581	20.4732	2.9586
9	31	14.9150	2.7644	20.3138	3.7497
11	46	13.9198	3.1428	20.2248	4.5477
13	64	13.1390	3.4991	20.1578	5.3464
15	85	12.4863	3.8322	20.1315	6.1533
17	109	11.9479	4.1527	20.1069	6.8954
19	136	11.4855	4.4588	20.0959	7.7697
21[1]	166	11.0544	4.7503	-	-
31[1]	361	9.4795	6.0245	-	-

Values for M(W) estimated from a 250 by 1000 image with border trimmed after processing. The standard deviations used to compute the NSD are taken over the same trimmed area for both the original and processed image. The standard deviation of the trimmed original data was 30.0399. 32,400 points, after trimming, were used to compute the values for the K-average filter.

[1] The data for W=21 and W=31 are over partial images with 10580 and 23980 points respectively. The W=21 has a width 10 border trimmed and the W=31 has a width 15 border trimmed.

TABLE II

BACKGROUND CLASS

Additive Noise; 5762 points

Pattern→ / Filter↓		Constant mean	Constant std.dev.	Increasing mean	Increasing std.dev.	Decreasing mean	Decreasing std.dev.
Fixed Template							
Ave.	W=3	31.89	27.37	5.84	1.31	31.08	85.32
	W=5	51.52	37.96	6.60	1.83	50.12	36.71
	W=7	64.03	48.82	7.14	2.40	62.11	47.00
Med.	W=3	5.20	.70	5.19	.70	5.20	.70
	W=5	5.57	.73	5.55	.73	5.57	.73
	W=7	89.29	98.14	7.99	3.69	85.88	94.17
K-average							
Ave.	(5,5)	4.91	1.02	4.93	1.04	4.91	1.02
	(31,31)	4.91	1.16	4.91	1.17	4.91	1.16
Ave.G.	(5,5)	4.90	.61	4.90	.61	4.90	.61
	(31,31	4.98	.15	4.98	.15	4.98	.15
Contiguous K-average							
Ave.	(5,5)	4.91	.93	4.92	.96	4.91	.93
Med.	(5,5)	4.91	.93	4.93	.95	4.91	.93
Ave.	(31,31)	4.91	.68	4.94	.74	4.91	.68
Ave.G.	(5,5)	4.90	.61	4.90	.61	4.90	.61
	(31,31)	4.97	.19	4.97	.19	4.97	.19

Multiplicative Noise; 5762 points

Pattern→ / Filter↓		Constant mean	Constant std.dev.	Increasing mean	Increasing std.dev.	Decreasing mean	Decreasing std.dev.
Fixed Template							
Ave.	W=3	21.49	17.31	5.98	1.39	20.70	16.46
	W=5	33.55	23.60	6.72	1.90	32.13	22.33
	W=7	41.22	30.17	7.22	2.47	39.35	28.41
Med.	W=3	5.37	.81	5.33	.80	5.37	.81
	W=5	5.82	.89	5.71	.84	5.82	.89
	W=7	32.31	33.61	6.57	1.91	30.65	31.49
K-average							
Ave.	(5,5)	5.03	1.20	5.06	1.25	5.03	1.20
	(31,31)	5.02	1.38	5.04	1.41	5.02	1.38
Ave.G.	(5,5)	5.03	.70	5.03	.70	5.03	.70
	(31,31)	5.01	.19	5.01	.19	5.01	.19
Contiguous K-average							
Ave.	(5,5)	5.02	1.09	5.05	1.14	5.02	1.09
Med.	(5,5)	5.03	1.09	5.06	1.13	5.03	1.09
Ave.	(31,31)	5.03	.77	5.08	.87	5.02	.77
Ave.G.	(5,5)	5.02	.71	5.02	.71	5.02	.71
	(31,31)	5.01	.23	5.01	.23	5.01	.23

STRIPE CLASS

6138 points

Noise→ / Filter↓		Additive mean	Additive std.dev.	Multiplicative mean	Multiplicative std.dev.
Fixed Template					
Ave.	W=3	180.57	31.16	113.09	22.62
	W=5	162.16	36.33	101.75	23.49
	W=7	150.42	32.39	94.55	20.60
Med.	W=3	203.40	21.10	119.52	23.78
w/o first 3		205.70	.69	122.17	19.28
	W=5	194.11	46.18	108.13	30.75
w/o first 3		205.47	.64	114.87	17.55
	W=7	191.70	50.30	102.52	31.00
w/o first 3		205.32	.64	109.57	17.38
K-average					
Ave.	(5,5)	205.91	1.01	127.82	31.35
	(31,31)	205.91	1.16	128.34	1.16
Ave.G.	(5,5)	205.92	.62	128.77	16.59
	(31,31)	205.97	.16	128.78	6.49
Contiguous K-average					
Ave.	(5,5)	205.92	.92	127.97	28.62
Med.	(5,5)	205.92	.91	127.78	28.64
Ave.	(31,31)	205.90	.69	128.12	21.86
Ave.G.	(5,5)	205.92	.62	128.55	16.45
	(31,31)	205.97	.18	128.60	6.84

Table 2 Mean and standard deviation data for striped test image. Additive noise and multiplicative noise have been added to three original images, then filtered using various methods. The templates are centered squares with side length W. For the K-average and contiguous K-average filters, the G signifies guided. For the guided runs, the (noise free) original image is one channel with weight 1 and the noisy image is the other channel with weight 0. In the K-average and contiguous K-average filter descriptions, the first number of the pair is the window size and the second number is k, the number of pixels clustered. For the median filter applied to the stripe class, the first three stripes were destroyed because of their small width and this effect produced most of the variance in the additive noise case. We also give the statistics for all the stripes except the first three to show the difference.

CAPTIONS

Figure 1

Estimated values for H(W,t). Data for contiguous K-average, labelled SIFT, and for K-average, labeled KAVG. Error bars are not plottable at this scale. See text for discussion.

Figure 2

Reachability Graph:
(a) The template with (0,0) marked by an asterisk.
(b) This shows the labels for the pixels of the grid.
(c) This is the resulting reachability graph. Note that it is not strongly connected.

Figure 3

a. Left image This image is a raw image produced by TIMS. See section 5 for further description. Note the intense horizontal noise pattern.
b. Right Image Data from after 30 iterations of contiguous K-average, W=5, K=8. Note retention of some horizontal noise and non-noise features.

Figure 4

Percent of data changed per iteration for four dissimilar data types: human head (MRI), agriculture, marsh/water, ocean, rock. The human head data was provided by Dr. Michael Vannier of the Mallinkrodt Inst. of Radiology, Washington University. It is a vertical section, frontal view of the brain and neck. It was processed by contiguous K-average, labeled SIFT, and by K-average, labeled KAVG. All other images were processed only by contiguous K-average. Agricultural areas are a portion of the data used in figure 3 and some Thematic Mapper imagery over a portion of NW Mississippi. The marsh/water and ocean images came from the CAMS scanner, see discussion in section 6 of text. The target areas were near the mouth of the Atchafalaya River. The rock is an image of a thin section of St. Peter Sandstone, provided by Dr. Dale Morgan, Texas A&M University.

Figure 5

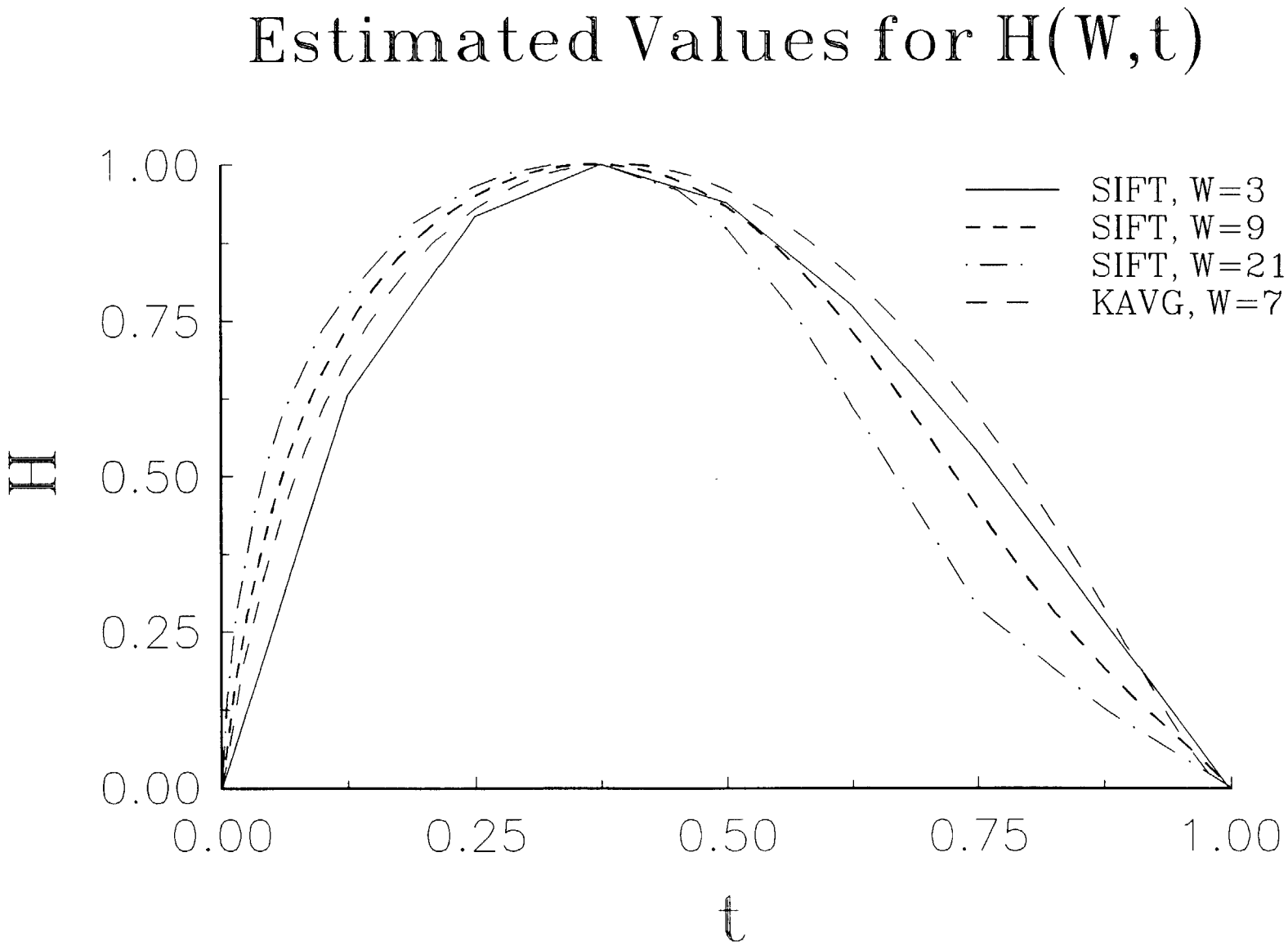
Synthetic image with admixtures of normal random and harmonic noise of various intensities. From right to left the images have increasing harmonic noise. This corresponds to the minor cycles in figure 6. From top to bottom they have increasing random noise, corresponding to the major cycles in figure 6. These are the limits and the central points of the 5 X 5 array of images used to demonstrate the use of filters as a preprocessing step.

Figure 6

Effect of Five Filters on Interface Length, with differing types and levels of noise. Refer to figure 5 for samples of the images processed. Interface length was computed using the ELAS module SLIN [2] after previously filtering each image.

Figure 7

CAlibrated Multispectral Scanner, CAMS, data of Salt Point, Atchafalaya Bay, Gulf Coast of Louisiana. Resolution is approximately 30 meters. Raw data is shown in on the left. The corresponding channels, after a multichannel filtering with the contiguous K-average filter, are shown on the right. From top to bottom the channels shown are 4 (0.63–0.69μm), 7 (1.55–1.75μm), and 8 (2.08–2.35μm).


Estimated Values for H(W,t)
1.00
0.75
0.50
0.25
0.00
H
0.00
0.25
0.50
0.75
1.00
t
SIFT, W=3
SIFT, W=9
SIFT, W=21
KAVG, W=7

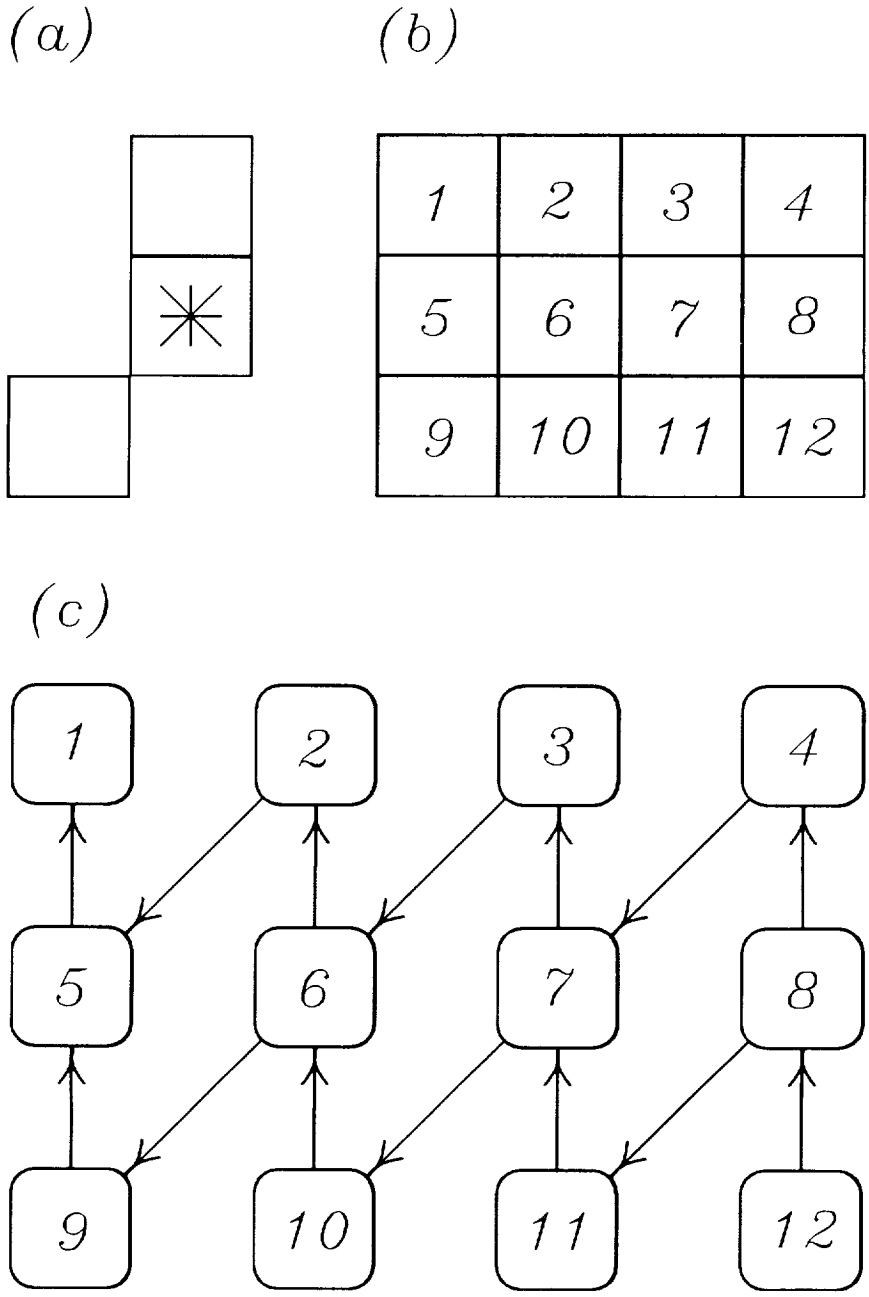


Figure 2

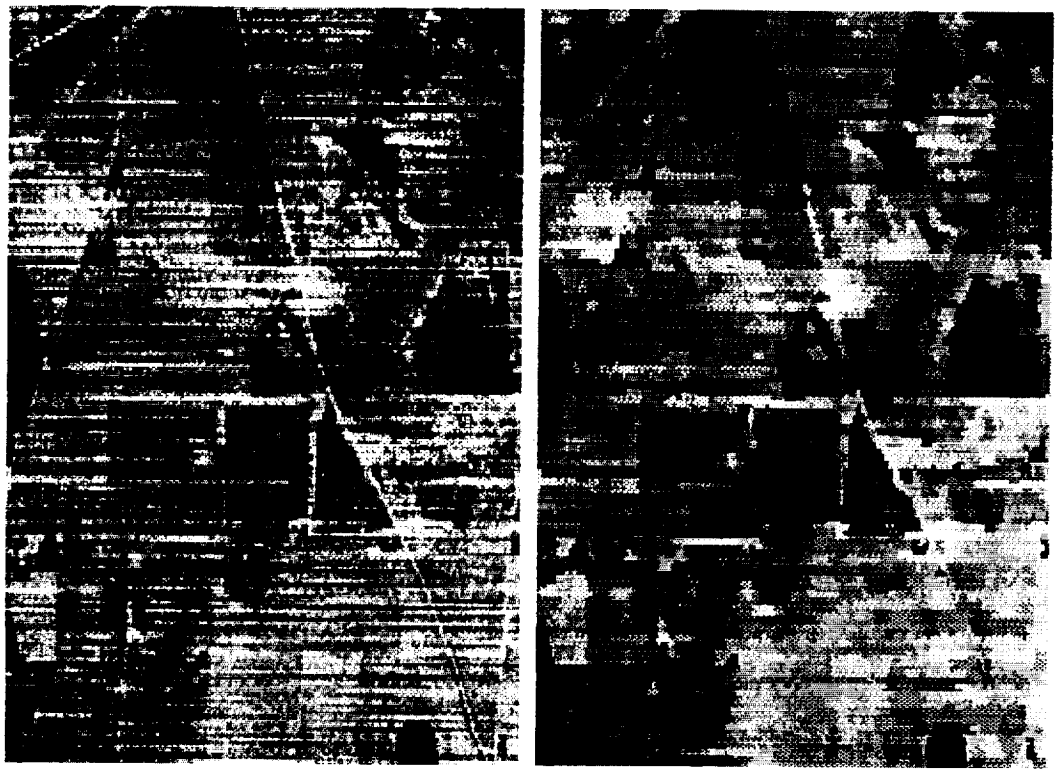
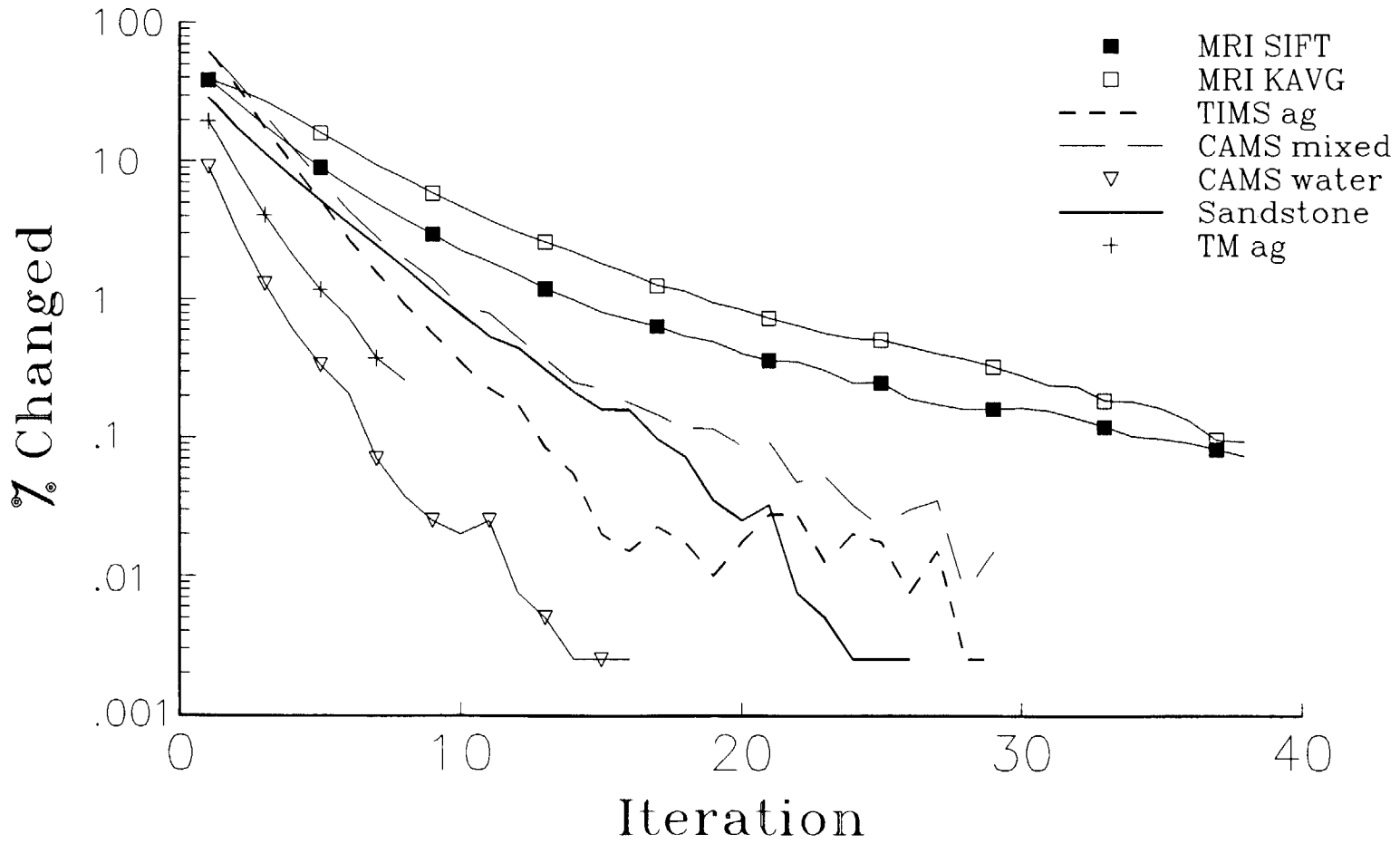


Figure 3

Change per Iteration
CAMS, TIMS, MRI, and Sandstone



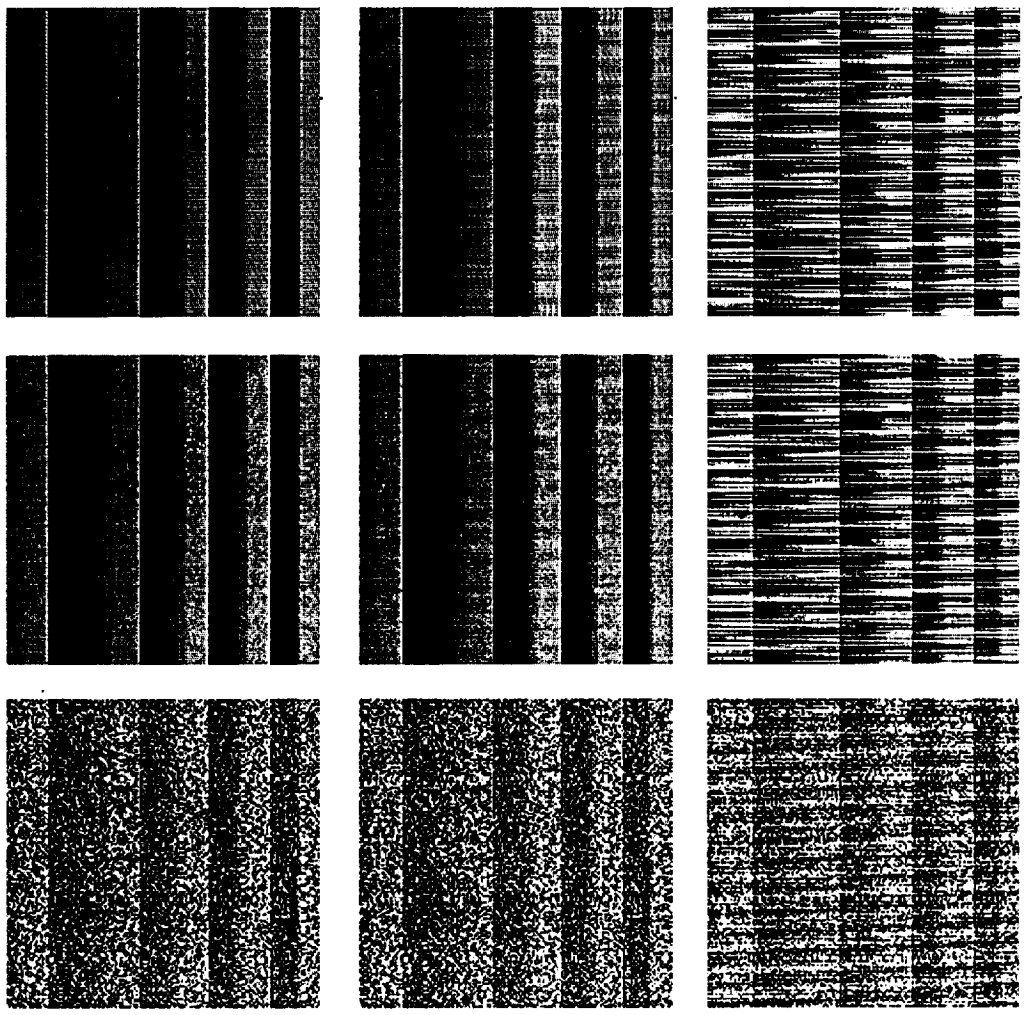
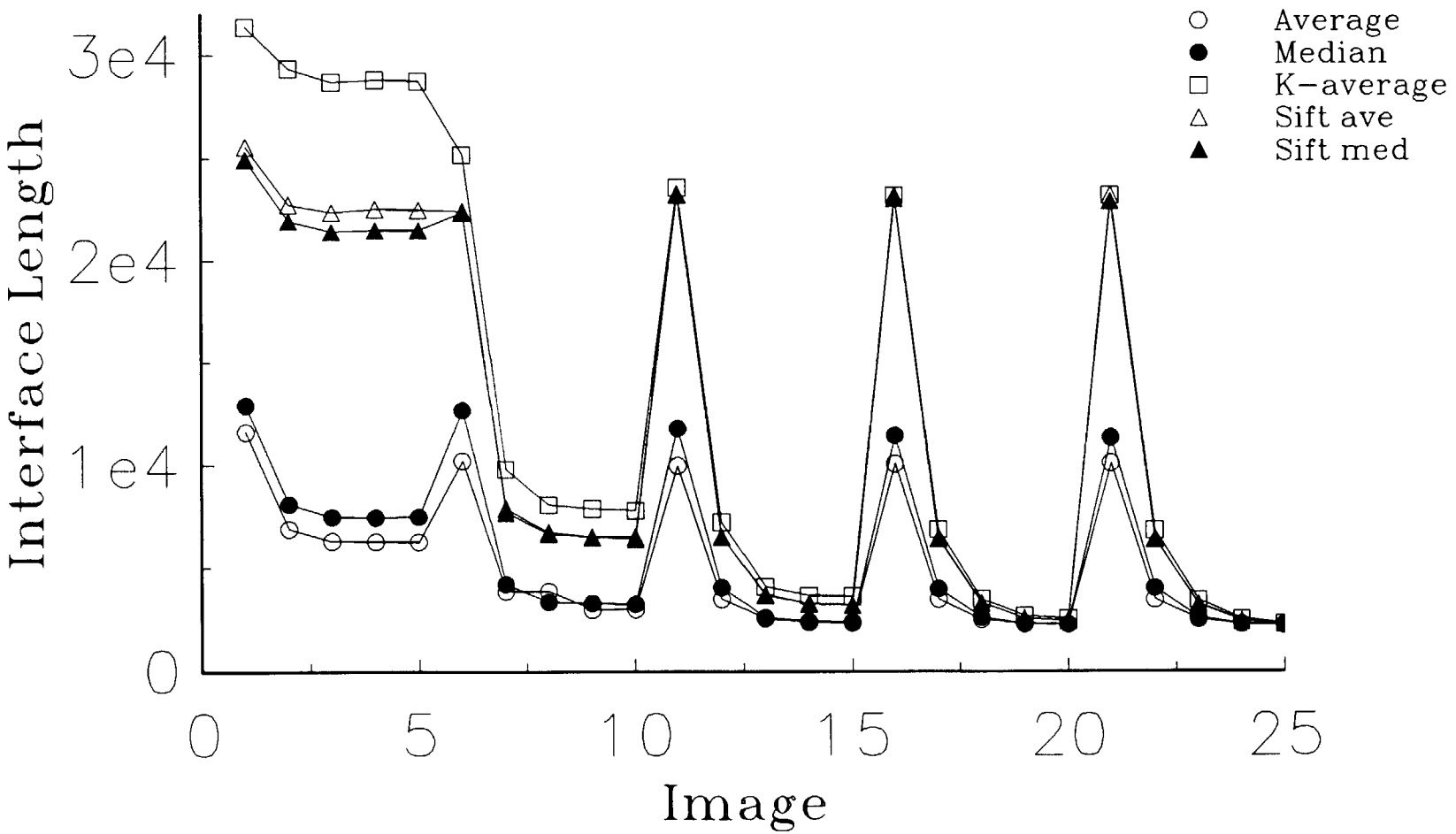


Figure 5

Effect of Five Filters on Interface Length With Differing Types and Levels of Noise



Major Cyles – Decreasing random noise

Minor Cycles – decreasing harmonic noise

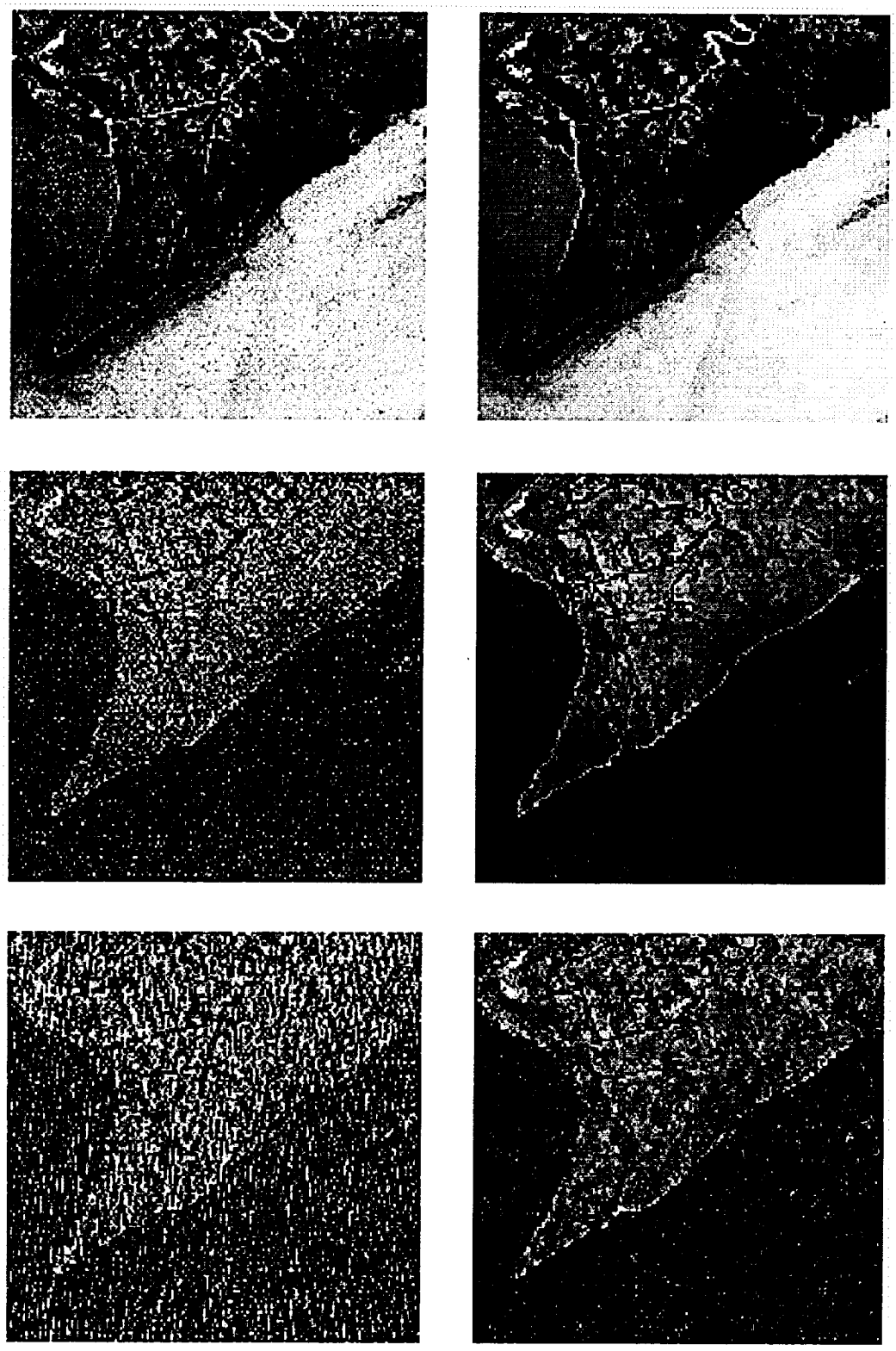


Figure 7

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-4311		

4. Title and Subtitle	5. Report Date
A Superior Edge Preserving Filter With a Systematic Analysis	August 1991
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Kenneth W. Holladay and Doug Rickman FTS 494-3830 OR 601-688-3830	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
NASA Science and Technology Laboratory John C. Stennis Space Center Stennis Space Center, MS 39529	13. Type of Report and Period Covered

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration	

15. Supplementary Notes

Kenneth W. Holladay: University of New Orleans, New Orleans, Louisiana, and Lockheed Engineering & Sciences Company (currently at John C. Stennis Space Center, Stennis Space Center, Mississippi).

Doug Rickman: John C. Stennis Space Center, Stennis Space Center, Mississippi.

16. Abstract

We report a new, adaptive, edge preserving filter for use in image processing. It has superior performance when compared to other filters. Termed the contiguous K-average, it aggregates pixels by examining all pixels contiguous to an existing cluster and adding the pixel closest to the mean of the existing cluster. The process is iterated until K pixels have been accumulated. Rather than simply compare the visual results of processing with this operator to other filters, we develop some approaches which allow quantitative evaluation of how well any filter performs. Particular attention is given to the standard deviation of noise within a feature and the stability of imagery under iterative processing. Demonstrations illustrate the performance of several filters to discriminate against noise and retain edges, the effect of filtering as a preprocessing step, and the utility of the contiguous K-average filter when used with remote sensing data.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Adaptive filtering image processing basic theory	Unclassified - Unlimited Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	40	A03

NASA FORM 1626 OCT 86

NASA-Langley, 1991