SMOOTHING OF DIGITAL IMAGES USING THE CONCEPT OF DIFFUSION PROCESS

S. BISWAS, N. R. PAL* and S. K. PAL

Machine Intelligence Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700 035, India

Abstract—An adaptive smoothing algorithm has been described which is capable of performing various tasks, such as removing salt and pepper noise, preserving roof edges, stretching (enhancing) step edges and reducing variations in low intensity varied regions. While iteration advances, it approximates both isotropic and anisotropic heat diffusion processes in performing these tasks. A region topography index has been defined for guiding the algorithm under different situations. Further, an image quality index is proposed which provides a criterion for automatic termination of the algorithm. This criterion can also be used with other iterative smoothing algorithms. The superiority of the method over some other similar techniques has been established for both synthetic and real images.

Adaptive smoothing   Isotropic/anisotropic diffusion   Edge stretching   Topography index
Quality index.

1. INTRODUCTION

Smoothing is an important image processing operation. Smoothing operation is necessary to reduce noises and to blur the false/stray contour fragments in order to enhance the overall visual quality of the degraded image. In order to clean an image and enhance its features, either spatial or frequency domain techniques can be used. The frequency domain smoothing uses filtering in the Fourier domain. Spatial domain techniques, on the other hand, normally employ linear or nonlinear spatial operations. Many efficient techniques have been developed in spatial domain. The simplest smoothing technique uses (unweighted) averaging over a predefined neighborhood. This reduces noise significantly, but at the same time it blurs the edges of objects. Thus, the overall image quality deteriorates. With the increase of the neighborhood size, blurring becomes more prominent. Some of the weighted averaging techniques, which have been proposed to reduce blurring, can be found in references (1–3). Weights play a significant role in the smoothing operation and hence their determination is an important task. One of the techniques of selecting weights is to use the local mean and variance.[4–6] Wang et al.[7] have published a good survey on weighted averaging and enhancement techniques. One of the drawbacks of these fixed weighted methods is that they cannot remove noise as efficiently as the unweighted averaging schemes.

To make the smoothing schemes more efficient, iterative weighted techniques have been reported.[8,9] The weighting coefficients are proportional to the

gradient inverses between the central point and its neighbors. The convergence of these methods is not known. Nagao and Matsuyama[10] used a simple technique to smooth images, preserving edges. They rotated a mask inside a 5×5 window about the center pixel. For every position of the mask, two regions may exist. They calculated the variances of all such regions due to all possible rotations of the mask and replaced the gray value of the center pixel by the average gray value of the region having the minimum variance. The process is repeated iteratively until all the gray levels in the image do not change much. Unfortunately, this algorithm assumes the difference of the average gray levels of the two regions is large, which may not be always true. This may seriously damage the image, particularly when roof edges are present in the image.

Recently, Marc et al.[11] have proposed an iterative weighted averaging scheme, which both sharpens and smooths. The method implements anisotropic diffusion[12] and its iterative behavior has also been discussed. The algorithm considers only the step edges, preservation of roof edges has not been taken into account. Since the weighting coefficients are based on the gradients at all 3×3 neighborhood points, the averaging result is influenced not only by the neighborhood points, but also by some other points beyond it. This sometimes deteriorates the image quality. Furthermore, the number of iterations required for different operations, e.g. edge detection, is heuristic in nature. Human intervention is also needed for termination of the algorithm and to judge the image quality. Otherwise, the noise cleaning may become insufficient or there may be excessive useless iterations.

We have proposed here a smoothing algorithm which implements both isotropic and anisotropic

---

* Author to whom correspondence should be addressed.

diffusion processes. The isotropic diffusion helps in preserving the roof edges and removing noises, while the anisotropic diffusion takes care of sharpening of step edges and reducing of low gray variations within regions. A region topography index guides the diffusion processes.

To keep the influence of neighbors on the computation of weights restricted only within a size of 3×3 of the central pixel, we consider the difference between the central pixel and a neighboring pixel as the gradient at the location of the neighboring pixel. For the central point, we take the usual gradient. Each gradient at a point determines the weight of the pixel at that point using a polynomial function such as that for low gradients, weights are high and *vice-versa*. For automatic termination of the smoothing algorithm we have defined an image quality index (IQI) which gives an estimate of the average contrast (with respect to background) per pixel in the image. The effectiveness of the algorithm, alongwith its comparison with that of Marc *et al.*[11] and Gaussian smoothing, has been demonstrated on both synthetic and real images. The performance of mean and median filtering as well as of Nagao and Matsuyama's edge preserving smoothing algorithm[10] has also been examined.

2. DESCRIPTION OF THE ALGORITHM

The proposed algorithm is based on an iterative weighted averaging technique. As mentioned before, it is intended not only to remove the salt-and-pepper-type noise, but also to preserve roof edges, stretch step edges and to reduce the gray variations in low intensity varied regions. In other words, we keep our attention to the following major tasks while formulating the algorithm:

- Removing salt and pepper noise
- Preserving roof edges
- Reducing variations in low intensity varied regions and enhancing step edges

In our subsequent discussions, the term smoothing will refer to any of these effects. In order to achieve this our weighted averaging scheme employs three different types of weights, depending on the surface topography within a small window.

Let the digital image be defined as $F[f(x, y)]_{m \times n}$, where $f(x, y) \in \{1, 2, \ldots, L\}$ is the set of gray levels. Let us consider a 3×3 neighborhood, $N_3(i, j)$, of a pixel at the position (i, j). At the $(t + 1)$th iteration the pixel intensity at the (i, j)th location in the smooth image is given by:

$$g^{(t+1)}(i, j) = \frac{\sum_{u=-1}^{1} \sum_{v=-1}^{1} g^{(t)}(i + u, j + v) w^{(t)}(i + u, j + v)}{\sum_{u=-1}^{1} \sum_{v=-1}^{1} w^{(t)}(i + u, j + v)}. \tag{1}$$

where $g^{(0)}(x, y)$ is the same as $f(x, y)$. The weights $w^{(t)}$ under the aforesaid three different situations are as follows:

For cleaning noise

$$w_n^{(t)}(i + u, j + v) = \begin{cases} 0 & \text{when } (u, v) = (0, 0) \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

For preserving roof edges

$$w_r^{(t)}(i + u, j + v) = \begin{cases} 1 & \text{when } (u, v) = (0, 0) \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

It is seen that w_n [equation (2)] helps in cleaning salt-and-pepper noises by averaging with unity weights, whereas w_r [equation (3)] ignores the effect of neighborhood in maintaining roof edges; in other words, equation (3) does not have any filtering effect. In the next section we will be explaining how the weight w_s which can take care of both stretching of step edges and reducing variance of low intensity varied regions, can be determined. Here, we will be explaining, first of all, how w_s can be selected. Then we will explain the behavior of the iterative algorithm in the light of anisotropic diffusion process in order to show that the same w_s can also perform enhancement of step edges.

3. WEIGHT FOR SMOOTHING LOW INTENSITY VARIED REGIONS

In order to obtain the weights w_s for smoothing we consider that the weight is inversely proportional to the image gradient α, i.e. for higher values (magnitudes) of α, w_s should be low, while for smaller α values, w_s should be high. However, α can have both positive and negative values. Therefore, the weight function should be symmetric with respect to α. As a simple case, the function may be of the form shown in Fig. 1. In terms of polynomial function, we may write:

$$w_s = (A\alpha^2 + B\alpha + C)^p, \tag{4}$$

where $p > 0$ is a constant.

Note that when $\alpha = 0$, the weight should attain its maximum value of unity, i.e. $w_s = 1$. This implies $C = 1$. Further, for $\alpha = 0$ we must have $dw_s/d\alpha = 0$, i.e.

$$p(A\alpha^2 + B\alpha + 1)^{p-1}(2A\alpha + B) = 0.$$

From this we obtain $B = 0$. So, $w_s = (A\alpha^2 - 1)^p$.

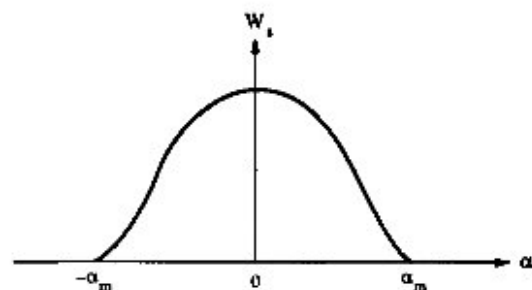Now we want (as mentioned before) $w_s = 0$ when α attains the maximum value α_m (say). This means

Fig. 1. Typical behavior of the weighting function with respect to gradients.

$A = -(1/x_m^2)$. Therefore:

$$w_s = \left[1 - \left(\frac{x}{x_m} \right)^2 \right]^p. \tag{5}$$

3.1. Criteria for computing α

Consider a neighborhood $N_3(i, j)$ of a pixel at (i, j). The central pixel has its gradient α_1 directed along the line with an angle θ equal t $\tan^{-1}(f_y/f_x)$, where f_x and f_y are the derivatives along the x and y directions, respectively, at the (i, j)th point.

For a pixel in the neighborhood $N_3(i, j)$, if we use a similar expression for its gradient computation then the smoothing operation is also influenced by the pixels beyond the 3×3 neighborhood and this may result in sometimes undesirable blurring of the image. To keep the effect restricted only within a 3×3 neighborhood for computing the gradient of a neighborhood pixel, we consider α to be equal to the difference between its intensity and that of the central pixel. The direction of the gradient will be along the line joining the concerned pixel and the central pixel. Thus, we write:

$$\alpha_k = \begin{cases} \sqrt{f_x^2 + f_y^2} & \text{for } k=1, \text{ i.e. for} \\ & u=0, \ v=0 \ \text{(central pixel)} \\ f(i+u, j+v) - f(i,j), & \text{for } k = 2, 3, \ldots, 9 \\ & \text{(for neighboring pixels),} \end{cases} \tag{6}$$

where

$$f_x = [f(i, j+1) - f(i, j-1)]/2$$

and

$$f_y = [f(i+1, j) - f(i-1, j)]/2.$$

Note that the parameter p plays a significant role in smoothing images. Its importance with respect to smoothing as well as enhancing step edges has been described in Section 3.3.

3.2. Behavior of the algorithm: isotropic and anisotropic property

In order to explain the isotropic and anisotropic behavior of the algorithm as iteration advances, we consider [equation (1)], i.e. the smoothed intensity at the (i, j)th point. Equation (1) can be written as:

$$g(i, j) = \sum_{u=-1}^{1} \sum_{v=-1}^{1} m(i+u, j+v) f(i+u, j+v), \tag{7}$$

where $m(i+u, j+v) = \dfrac{w(i+u, j+v)}{\sum \sum w(i+u, j+v)}$ and $\sum \sum m = 1$. The contribution of each pixel in $N_3(i, j)$ in smoothing $g(i, j)$ is, therefore:

$$\beta(i+u, j+v) = m(i+u, j+v)f(i+u, j+v). \tag{8}$$

We now show that the proposed smoothing procedure, when iteratively applied, implements an anisotropic diffusion process[12] for points where the smoothing weights w vary with respect to time (i.e. w_s

for reducing variation in low varied regions and enhancing step edges) and isotropic diffusion process for the points where the smoothing weights do not vary with respect to time (i.e. w_n and w_r for removing salt and pepper noise and preserving roof edges).

Considering the points where w_s [equation (5)] is applicable, the iterated value at time $t + 1$ can be written as:

$$g^{(t+1)} = m^t(i-1, j-1)g^t(i-1, j-1) + m^t(i-1, j)g^t(i-1, j)$$
$$+ m^t(i-1, j+1)g^t(i-1, j+1) + m^t(i, j-1)g^t(i, j-1)$$
$$+ m^t(i, j-1)g^t(i, j+1) + m^t(i+1, j-1)g^t(i+1, j-1) \tag{9}$$
$$+ m^t(i+1, j)g^t(i-1, j) + m^t(i+1, j+1)g^t(i-1, j+1)$$
$$+ m^t(i, j)g^t(i, j).$$

Therefore:

$$g^{t+1}(i, j) - g^t(i, j)$$
$$= \{m^t(i+1, j+1)[g^t(i+1, j+1) - g^t(i, j)]$$
$$- m^t(i-1, j-1)[g^t(i, j) - g^t(i-1, j-1)]\}$$
$$+ \{m^t(i-1, j+1)[g^t(i-1, j+1) - g^t(i, j)]$$
$$- m^t(i+1, j-1)[g^t(i, j) - g^t(i+1, j-1)]\}$$
$$+ \{m^t(i, j+1)[g^t(i, j+1) - g^t(i, j)] \tag{10}$$
$$- m^t(i, j-1)[g^t(i, j) - g^t(i, j-1)]\}$$
$$+ \{m^t(i+1, j)[g^t(i+1, j) - g^t(i, j)]$$
$$- m^t(i-1, j)[g^t(i, j) - g^t(i-1, j)]\};$$

since

$$\sum \sum m = 1.$$

Equation (10) is the discrete approximation of:

$$\frac{\partial g}{\partial t} = \nabla_1(m(\nabla_1 g)) + \nabla_2(m(\nabla_2 g)), \tag{11}$$

where ∇_1 and ∇_2 are the spatial operators with respect to a rectangular frame of axes and its rotated frame (rotated by $45°$), respectively. The weights vary both in spatial location and in time. Therefore, it implements an anisotropic wave diffusion process.[12]

On the other hand, in the vicinity of points where w_n and w_r [equations (2) and (3)] are applicable (i.e. where weights do not vary at all with respect to time) we have:

$$g^{t+1}(i, j) - g^t(i, j)$$
$$= m^t[g^t(i+1, j+1) - 2g^t(i, j) + g^t(i-1, j-1)]$$
$$+ m^t[g^t(i-1, j+1) - 2g^t(i, j) + g^t(i+1, j-1)]$$
$$+ m^t[g^t(i, j+1) - 2g^t(i, j) + g^t(i, j-1)]$$
$$+ m^t[g^t(i-1, j) - 2g^t(i, j) + g^t(i-1, j)]. \tag{12}$$

This is nothing but the discrete approximation of the following isotropic heat diffusion equation with respect to the aforesaid two frames of axes:

$$\frac{\partial g}{\partial t} = m(\nabla_1^2 g + \nabla_2^2 g), \quad \text{where } m = 1. \tag{13}$$

3.3. Relation between critical gradient and parameter p

To determine the effect of the parameter p on the gradients of step edges and for the regions of low gray

variations, we restrict ourselves to the one-dimensional case. From equation (10) we obtain:

$$g^{t-1}(i,j) - g^t(i,j) = m^t(i, j+1)[g^t(i, j+1) - g^t(i,j)]$$
$$- m^t(i, j-1)[g^t(i,j) - g^t(i, j-1)],$$

which implements the anisotropic diffusion equation (in one dimension):

$$\frac{dg}{dt} = \nabla(m^t \nabla g). \tag{14}$$

Now $m^t \nabla g$ can be written as $\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^p g_x^t$.

Thus:

$$\frac{dg^t}{dt} = \frac{d}{dx}\left(\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^p g_x^t\right)$$

$$= g_{xx}^t\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^p$$

$$+ g_x^t p\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^{p-1}\left(-2\left(\frac{g_x^t}{\alpha_m^2}\right)g_{xx}^t\right)$$

$$= g_{xx}^t\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^{p-1}\left\{1 - \left(\frac{g_x^t}{\alpha_m}\right)^2 - 2p\left(\frac{g_x^t}{\alpha_m}\right)^2\right\}.$$

Now for $t = 0$, $g_x = f_x = \alpha_1$. Since g^t is a differentiable function, the order of differentiation (with respect to x and t) can be interchanged. Therefore, we can write:

$$\frac{d}{dt}\left(\frac{dg^t}{dx}\right) = \frac{d}{dx}\left(\frac{dg^t}{dt}\right)$$

$$= \frac{d}{dx}\left\{g_{xx}^t\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^{p-1}\right.$$

$$\times \left.\left\{1 - \left(\frac{g_x^t}{\alpha_m}\right)^2 - 2p\frac{g_x^t}{\alpha_m}\right)^2\right\}\right\}$$

$$= g_{xxx}^t\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^{p-1}\left\{1 - \left(\frac{g_x^t}{\alpha_m}\right)^2(1 + 2p)\right\}$$

$$+ g_{xx}^t(p-1)\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^{p-2}\left(-2\frac{g_x^t}{\alpha_m^2}g_{xx}^t\right)$$

$$\times \left\{1 - \left(\frac{g_x^t}{\alpha_m}\right)^2(1 + 2p)\right\}$$

$$+ g_{xx}^t\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^{p-1}\left\{-2\frac{g_x^t}{\alpha_m^2}g_{xx}^t - 4p\frac{g_x^t}{\alpha_m^2}g_{xx}^t\right\}.$$

In the neighborhood of step edges g_x is locally maximum or minimum. Therefore, g_{xx} is always zero, $g_{xxx} < 0$ when g_x is maximum and $g_{xxx} > 0$ when g_x is minimum. Therefore, we have two conditions, $g_x > 0$, $g_{xx} = 0$ and $g_{xxx} < 0$. Also, $g_x < 0$, $g_{xx} = 0$ and $g_{xxx} > 0$. For edge sharpening due to the first condition $\frac{d}{dt}\left(\frac{dg^t}{dx}\right)$ must always be greater than zero. Now $\frac{d}{dt}\left(\frac{dg^t}{dx}\right) > 0$ implies $\left\{1 - \left(\frac{g_x^t}{\alpha_m}\right)^2(1 + 2p)\right\} < 0$, be-

cause $\left[1 - \left(\frac{g_x^t}{\alpha_m}\right)^2\right]^{p-1}$ is always positive. For:

$$1 - \left(\frac{g_x^t}{\alpha_m}\right)^2(1 + 2p) < 0, \quad \text{we must have} \quad \left(\frac{g_x^t}{\alpha_m}\right)^2$$
$$(1 + 2p) > 1$$

$$\text{or,} \quad g_x^t > \frac{\alpha_m}{\sqrt{(1 + 2p)}}. \tag{15}$$

Note that we obtain the same equaton for p due to the second condition because for edge sharpening, here $\frac{d}{dt}\left(\frac{dg^t}{dx}\right)$ must always be less than zero. Now $\frac{d}{dt}\left(\frac{dg^t}{dx}\right) < 0$ also implies $\left\{1 - \left(\frac{g_x^t}{\alpha_m}\right)^2(1 + 2p)\right\} < 0$. This is because g_{xxx} is positive in this case. Hence, stretching occurs in the neighborhood of step edges when $g_x^t > \frac{\alpha_m}{\sqrt{(1 + 2p)}}$. When $g_x^t \leq \frac{\alpha_m}{\sqrt{(1 + 2p)}}$, we obtain the smoothing effect in the regions of low varied intensity. The process is an approximation because for real images g_{xx} may not be exactly equal to zero due to the quantization error. However, the terms involving g_{xx} are very small since p is reasonably large even for a 32 level gray image and can therefore be assumed to be zero.

It may be mentioned that the same effect also holds good when g is a function in two dimensions. The effect of p is similar to that of the smoothing parameter described in reference [11].

4. REGION TOPOGRAPHY INDEX

Since the algorithm performs different processes (isotropic and anisotropic) for points over noisy, edgy and low intensity regions, it is, therefore, necessary to detect the nature of points (whether noisy or edgy) before we select the weight for obtaining pixel's smoothed value. To discriminate between noisy regions and regions with roof and step edges, we define a region topography index (RTI) in terms of its boundary and within region characteristics.

Consider a small region Ω about the $(i, j)$the pixel consisting of its neighboring pixels in $N_j(i, j)$. The sum of the magnitudes of gradients α_k, $k = 2, 3, \ldots, 9$, i.e. $SG_\alpha(= \sum|\alpha_k|)$ can be used to indicate the degree of flatness of the region Ω. As SG_γ increases (say, for edgy and noisy regions), the homogeneity of Ω decreases.

Let us now consider the boundary pixels of Ω. If SG_B is the sum of the magnitudes of $G_x(x$ gradient) and G_y (y gradient) of boundary pixels, then:

$$SG_B = \sum_{\partial\Omega}|G_x| + |G_y|$$

($\partial\Omega$ being the boundary of Ω) indicates the degree of flatness of the boundary of Ω. For example, $SG_B \approx 0$ implies $\partial\Omega$ is a flat boundary. Note that neither SG_B nor SG_α can distinguish between noisy and edgy re-

gions. In order to discriminate between them we introduce the concept of the region topography index (RTI) of Ω as:

$$RTI = \frac{SG_\alpha}{SG_B}.$$

The discriminating characteristics of RTI will now be explained. For a perfectly homogeneous region we have both $SG_\alpha = 0$ and $SG_B = 0$. Therefore:

$$RTI = \frac{0}{0} = 0 \quad \text{(let us assume)}.$$

Let us consider two types of step edge within a 3×3 mask (as shown below).

$$
\begin{array}{ccc}
L_1 & H_1 & H_2 \\
L_2 & H_3 & H_4 \\
L_3 & H_5 & H_6
\end{array}
\qquad
\begin{array}{ccc}
H_1 & L_1 & L_2 \\
H_2 & L_3 & L_4 \\
H_3 & L_5 & L_6
\end{array}
\ .
$$

H_is and L_is represent almost equal high and low values, respectively. Assuming $|H_i - H_j| \approx |L_i - L_j| \approx l$ for all i and j and $|H_i - L_j| \approx h$ for all i and j, we obtain for either type:

$$SG_B = \sum_{\partial\Omega}|G_x| + \sum_{\partial\Omega}|G_y| \approx 2h + 6l$$

and

$$SG_\alpha \approx 3h + 5l.$$

Therefore:

$$(RTI)_{\text{step edge}} = \frac{SG_\alpha}{SG_B} \approx \frac{3h + 5l}{2h + 6l} \approx 1.5,$$

as $l \approx 0$.

For a salt-and-pepper type noisy patterns (as shown below):

$$
\begin{array}{ccc}
L_1 & L_4 & L_6 \\
L_2 & H & L_7 \\
L_3 & L_5 & L_8
\end{array}
\qquad
\begin{array}{ccc}
H_1 & H_4 & H_6 \\
H_2 & L & H_7 \\
H_3 & H_5 & H_8
\end{array}
\ .
$$

$SG_\alpha \approx 8h$. Thus:

$$(RTI)_{\text{noise}} \approx \frac{8h}{8l} \gg 1.5, \ \text{i.e.} \ (RTI)_{\text{noise}} \gg (RTI)_{\text{stepedge}}.$$

Let us now consider roof edges (as shown below):

$$
\begin{array}{ccc}
L_1 & H_1 & L_4 \\
L_2 & H_2 & L_5 \\
L_3 & H_3 & L_6
\end{array}
\qquad
\begin{array}{ccc}
H_1 & L_1 & H_4 \\
H_2 & L_2 & H_5 \\
H_3 & L_3 & H_6
\end{array}
\ .
$$

In each case $SG_B \approx 4h + 4l$ and $SG_\alpha \approx 6h + 2l$. Therefore, $(RTI)_{\text{roof edge}} = \frac{SG_\alpha}{SG_B} \approx \frac{6h - 2l}{4h + 4l} \approx 1.5$ $(RTI)_{\text{noise}} \gg (RTI)_{\text{roof edge}}$. Since $(RTI)_{\text{noise}}$ is seen to be reasonably larger than those of step and roof edges, it can be used as a quantitative index for discriminating noisy and edgy regions.

RTI cannot discriminate between roof and step edges. However, since SG_B for roof edges is approximately twice the value for step edges. SG_B alone can be used to distinguish them.

To illustrate the discriminating characteristics of

RTI, we consider the following two sets of two image blocks and a noisy (spot noise) block. Each set describes a step edge and a root edge.

$$
\begin{array}{ccc|ccc}
3 & 3 & \boxed{4} & 12 & 14 \\
3 & 3 & 5 & 11 & 13 \\
4 & 3 & 5 & 10 & 12 \\
3 & 3 & 6 & 9 & 11
\end{array}
$$

Consider the 3×3 windows as indicated in the first two blocks. They indicate step and roof edges. In the first block, $SG_B = 20$ and $SG_\alpha = 27$. Therefore,

$$(RTI)_{\text{stepedge}} = \frac{SG_\alpha}{SG_B} = 27:20 = 1.35.$$

In the second block, $SG_B = 58$ and $SG_\alpha = 85$, which means $(RTI)_{\text{root edge}} = 1.46$. Now, for noisy block $SG_B = 6$, $SG_\alpha = 79$ and so $(RTI)_{\text{noise}} = 13.16$. Thus, $(RTI)_{\text{noise}} \gg (RTI)_{\text{stepedge}}$ and also $(RTI)_{\text{noise}} \gg (RTI)_{\text{roof edge}}$. Therefore, RTI can discriminate between step or roof edge and noise.

Furthermore, $(SG_B)_{\text{root edge}}/(SG_B)_{\text{stepedge}} = 2.9$. This helps to distinguish between step and roof edges. Similar discrimination is also seen in the third and fourth block. For these blocks, $(RTI)_{\text{stepedge}} = 1.531$ and $(RTI)_{\text{root edge}} = 1.20$. Therefore, $(RTI)_{\text{noise}} \gg (RTI)_{\text{stepedge}}$ and $(RTI)_{\text{noise}} \gg (RTI)_{\text{roof edge}}$. Also $(SG_B)_{\text{roof edge}}/(SG_B)_{\text{step edge}} = 2.56$.

5. IMAGE QUALITY INDEX (IQI) AND STOPPING CRITERION

One of the major problems of iterative smoothing algorithms is "when to stop the algorithm", i.e. to determine the approximate time for its termination. Marc et al.[11] have discussed the convergence property and the iterative behavior of their algorithm. The convergence of their algorithm takes a fairly long time and it has been found that the smoothing as required for different applications (edge detection, segmentation, etc.) reaches the desired level long before the algorithm converges. This makes many iterations useless and hence a wastage of computing time. They also conducted an experiment in which the termination of the algorithm after a satisfactory result was done arbitrarily, keeping the number of iterations small. However, this requires human intervention to judge the output image quality.

In our method we have proposed an image quality index (IQI) which reflects the average contrast (with respect to background) per pixel in the image for

termination of the algorithm. The IQI does neither depend on the size of objects nor on the number of objects in the image. It is also independent of the dimension of the image.

To determine IQI we find, first of all, the total contrast K of the image. For an $m \times n$ image K may be defined as:

$$K = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \qquad (16)$$

where c_{ij} is the contrast of the (i, j)th pixel. Using the concept of human psychovisual perception, the contrast c_{ij} of a gray level image at the point (i, j) can be written as:[13]

$$c_{ij} = \frac{|B - B_{ij}|}{B}$$

$$= \frac{|\Delta B_{ij}|}{B}, \qquad (17)$$

where B is the immediate surrounding luminance of the (i, j)th pixel with intensity B_{ij}.

From equations (16) and (17) we note that the contrast of pixels in a perfectly homogeneous region is zero everywhere except near the boundary points. The contribution to K of the image, therefore, comes mainly from its noisy pixels and contrast regions.

Therefore, average contrast per pixel may be defined as:

$$IQI = \frac{K}{n_k},$$

where $n_k = mn - n_h$, n_k = total number of significant contrast points, n_h = total number of significant homogeneous points and mn = number of pixels in the image. The average is taken over only those pixels which contribute mainly to contrast measure; the pixels of homogeneous regions being least contributory have been discarded.

To find n_h, we define the homogeneity h_{ij} of the (i, j)th pixel as:

$$h_{ij} = \frac{\sum_{r=1}^{8} \exp - |B_{ij} - B_r|}{8}, \qquad (18)$$

where B_r indicates the intensity of a background (neighboring) pixel in $N_3(i, j)$ of B_{ij}. From equation (18) it is seen that when each background pixel is equal to the central pixel then the tiny region Ω around the central pixel is perfectly homogeneous and the homogeneity measure at the central pixel is equal to unity. For other cases, the homogeneity value of a pixel drops down exponentially with its difference from the background intensity.
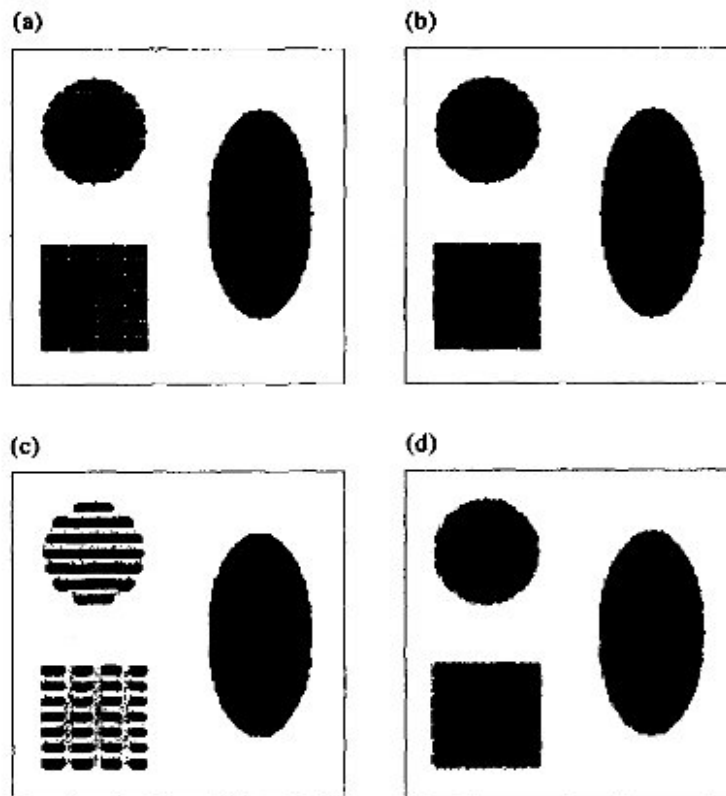

Fig. 2. Noise-free synthetic image with grid and lines: (a) input, (b) output obtained by the proposed method, (c) output obtained by Marc *et al.* and (d) output by Gaussian smoothing ($\sigma = 1$).
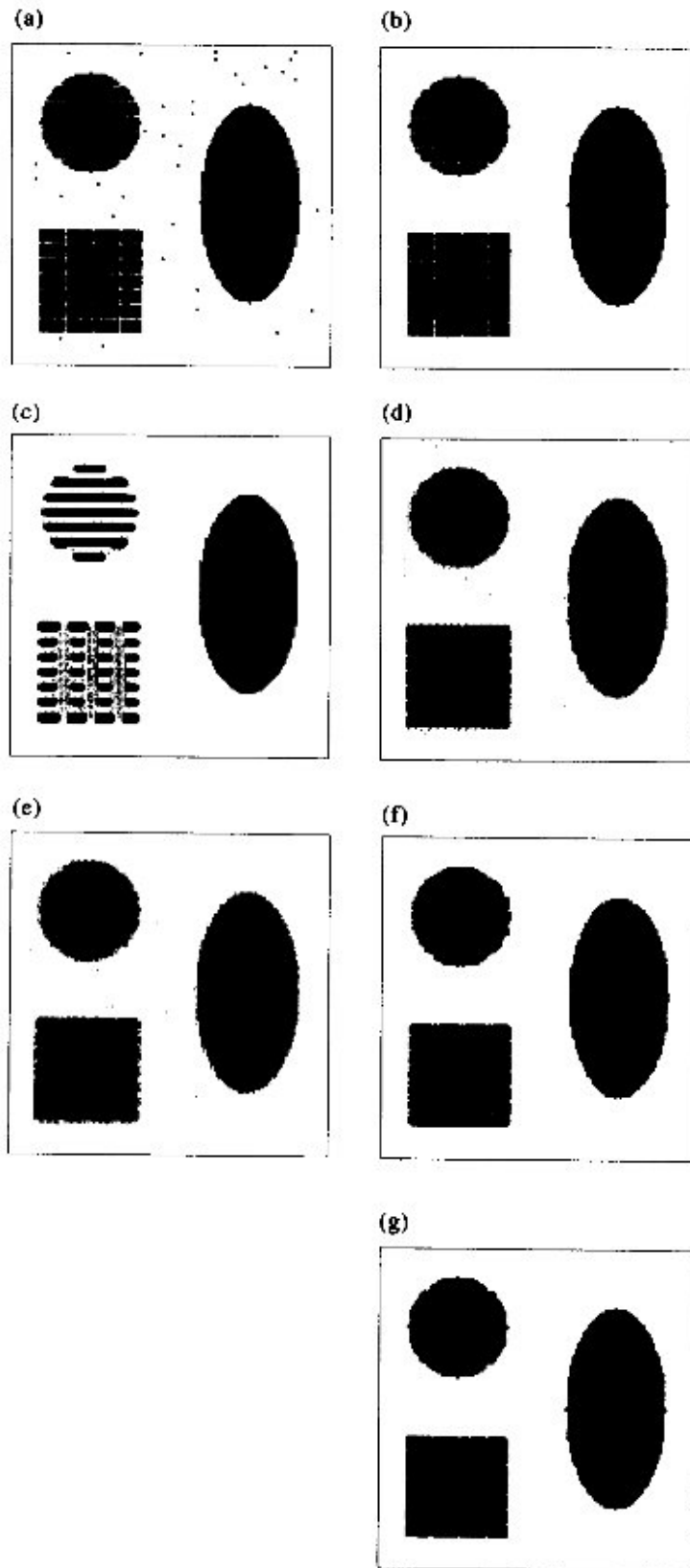
Fig. 3. Noisy (salt-and-pepper) synthetic image with grid and lines: (a) input, (b) output obtained by the proposed method, (c) output obtained by Marc et al., (d) output by Gaussian smoothing ($\sigma = 1$), (e) output obtained by mean filtering, (f) output obtained by median filtering and (g) output obtained by Nagao and Matsuyama's algorithm.
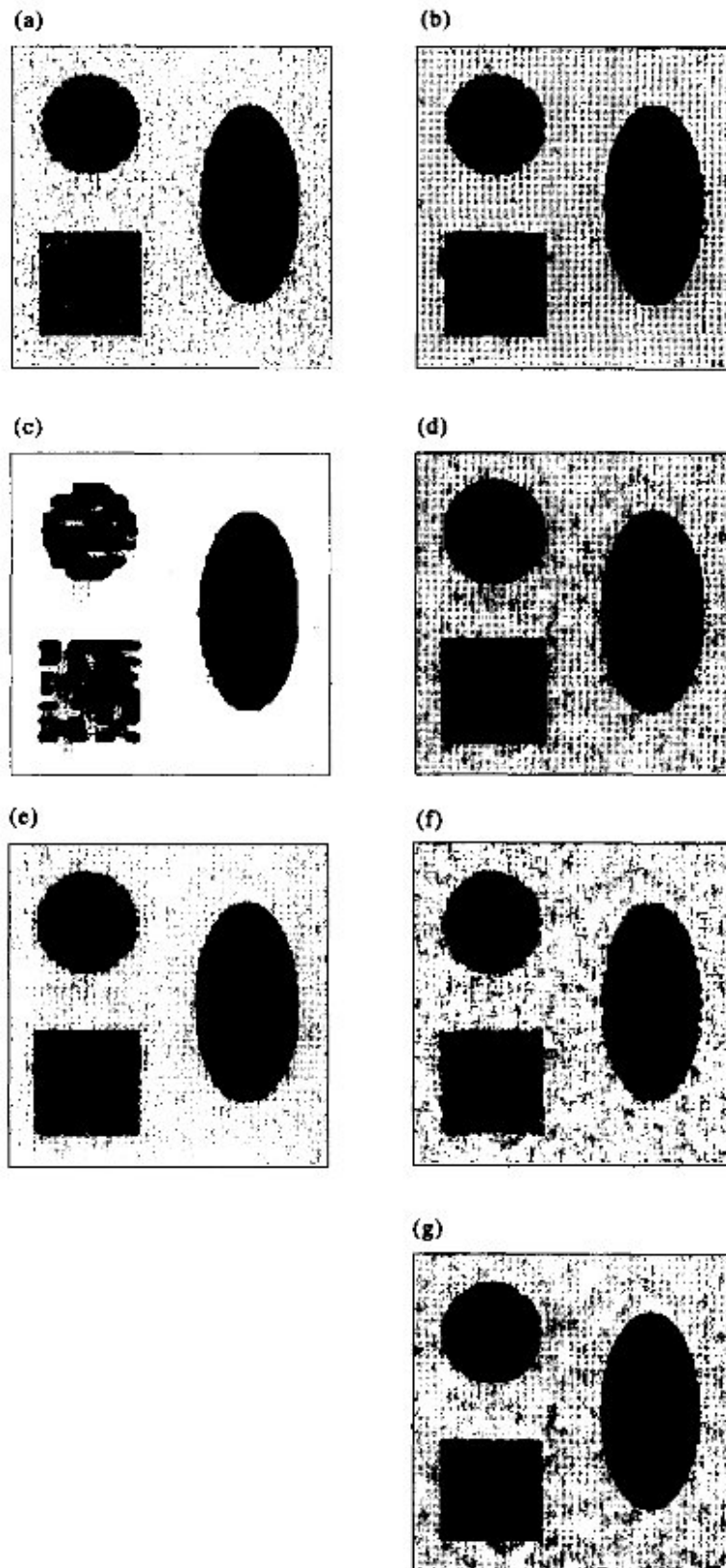
Fig. 4. Noisy (Gaussian, $\sigma = 5$) synthetic image with grid and lines: (a) input, (b) output obtained by the proposed method, (c) output obtained by Marc *et al.*, (d) output by Gaussian smoothing ($\sigma = 5$), (e) output obtained by mean filtering, (f) output obtained by median filtering and (g) output obtained by Nagao and Matsuyama's algorithm.

Equation (18) is a choice for homogeneity. The visual response curve $(\Delta B - B)^{(1/3)}$ can be well approximated by the exponential function. Since homogeneity can be considered as the inverse of contrast, it is reasonable to assume an exponential function for h_{ij} of the background intensity, B.

Therefore, if we compute total homogeneity of an image as:

$$H = \sum_{i=1}^{m} \sum_{j=1}^{n} h_{ij}, \tag{19}$$

then the major contribution to H comes only from the pixels which lie in perfectly homogeneous regions. Thus, n_g can be considered to be approximately equal to H and we have

$$IQI = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} |\Delta B_{ij}|/B}{mn - \sum\sum h_{ij}}. \tag{20}$$

As the iteration advances, the noisy points are cleaned up and hence their contribution to c_{ij} decreases. Consequently, the numerator in the expression of IQI decreases. Also, when homogeneity increases with iteration the denominator of IQI decreases. However, the rate of decrease of the numerator is more than that of the denominator. As a result, IQI decreases with iteration. Therefore, for the termination of the algorithm one can check if the change in IQI, (Δ_{IQI}), is less than a pre-assigned positive number ε. To avoid arbitrary (heuristic) selection of ε one may consider ε be equal to the theoretically possible minimum change in contrast/pixel, i.e. (Δc_{min}) in an image. In other words, terminate the algorithm when Δ_{IQI} attains Δc_{min}. The minimum change in contrast/pixel, Δc_{min}, in an L-level image is $(1/L(L-1))$ (proof is given in Appendix I).

6. RESULTS AND DISCUSSION

To examine the performance of the proposed algorithm, we have used a set of synthetic images and a set of real images. The synthetic images have been used to check the behavior of the algorithm under a known environment, while the application of the algorithm on real images shows its performance under an unknown environment. The synthetic images are of size 128×128 while the real images are 64×64. Each image is of level 32. The critical gradient for these
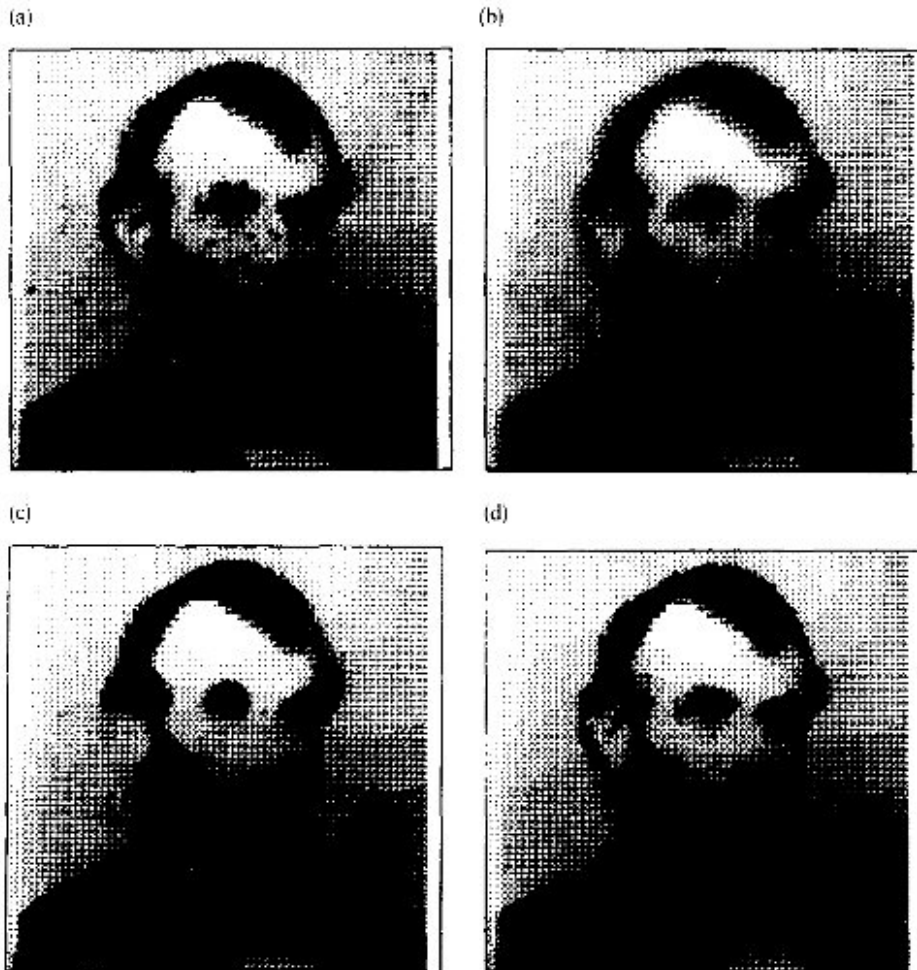

Fig. 5. Lincoln image: (a) input, (b) output obtained by the proposed method, (c) output obtained by Marc et al. and (d) output by Gaussian smoothing ($\sigma = 1$).

images was chosen to be 3 so that the value of p is almost 53 [equations (15)]. For the purpose of comparison of the proposed method we have also implemented the algorithm of Marc *et al.*[11] and Gaussian smoothing. The smoothing parameter for the algorithm in reference (11) was taken to be 3. As a result, critical gradient about which stretching occurs remains the same in both cases.

Figure 2(a) displays a noise-free synthetic input image with step and roof (grid and line structures) edges. The output obtained by the proposed method [Fig. 2(b)] is seen to preserve the grid and line structures very well. On the other hand, the method of Marc *et al.* maintains the step edges but affects the grid and line structures [Fig. 2(c)]. The Gaussian smoothing even with a low sigma value ($\sigma = 1$) is not very effective [Fig. 2(d)], because it blurs both the grid and line structures as well as the step edges present in the input.

Figure 3(a) is a noisy version of Fig. 2(a), the noise is

of salt-and-pepper type. Outputs [Fig. 3(b)–(d)] show the performance of all three algorithms. In the case of Gaussian smoothing noise is reduced with some blurring, while the algorithm of Marc *et al.* removes salt-and-pepper noise completely, but it damages the grid-and-line structures. On the other hand, the proposed algorithm removes the salt-and-pepper noise and maintains the grid-and-line structures. In this context, we also examine the results of mean and median filtering, and Nagao and Matsuyama's edge preserving smoothing algorithm.[10] Figure 3(e)–(g) denote the respective outputs. Following reference (10), we iterated the edge-preserving smoothing process until all the pixels do not change much. Nagao and Matsuyama's[10] algorithm fully removes the salt-and-pepper noise, but it damages grid-and-line structures. All the step edges are maintained. This is because the grid-and-line structures are basically the roof edges, i.e. the regions on two sides of the edge have
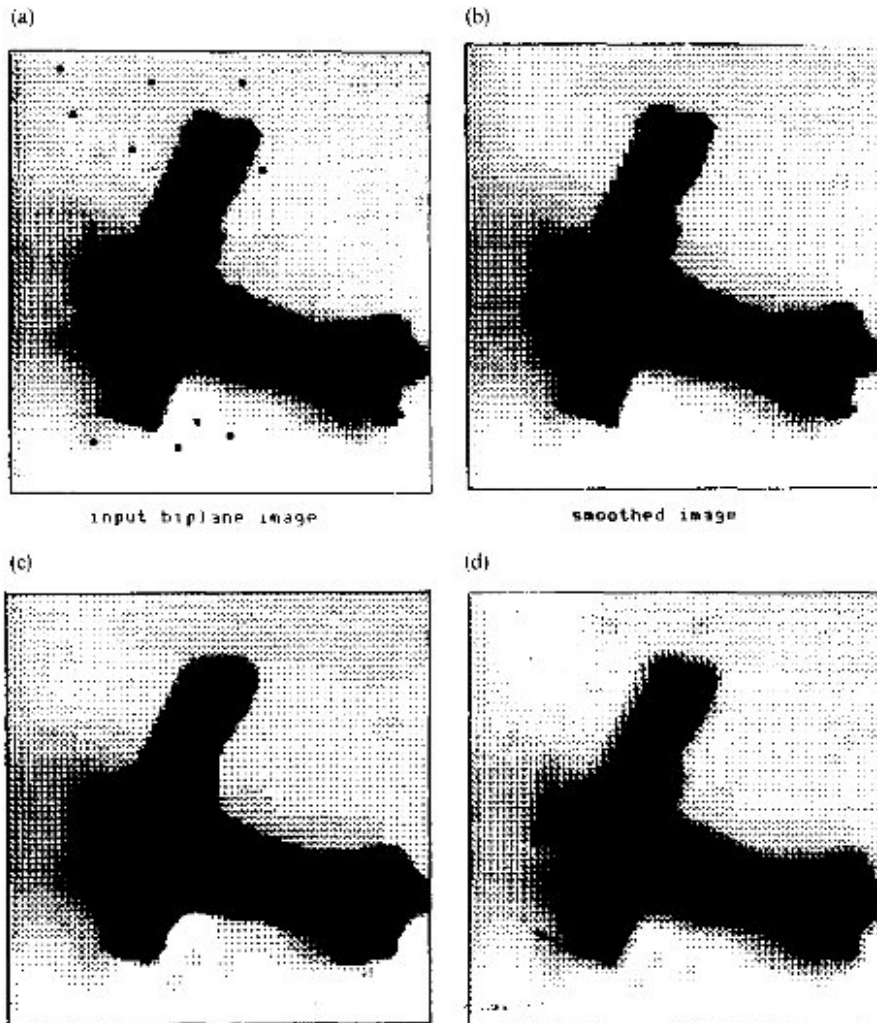

Fig. 6. Biplane image: (a) input, (b) output obtained by the proposed method, (c) output obtained by Marc *et al.* and (d) output by Gaussian smoothing ($\sigma = 1$).
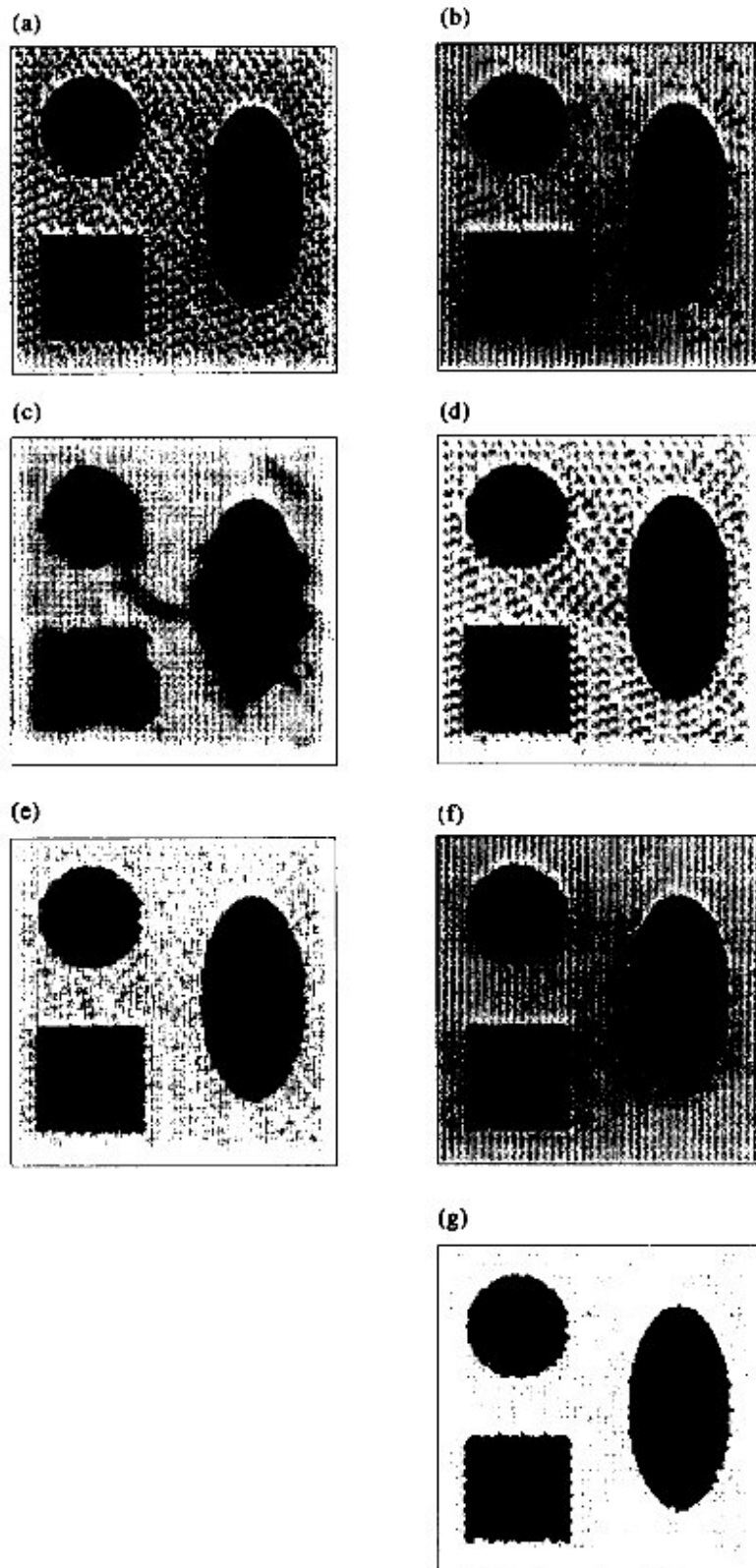
Fig. 7. Noisy (structured noise) synthetic image with grid and lines: (a) input, (b) output obtained by the proposed method, (c) output obtained by Marc *et al.*, (d) output by Gaussian smoothing ($\sigma = 1$), (e) output obtained by mean filtering, (f) output obtained by median filtering and (g) output obtained by Nagao and Matsuyama's algorithm.

Table 1. Change in ΔIQI for different images

	Input images	Total No. of iterations	ΔIQI_{in}	ΔIQI_f	Output images
Proposed algorithm	Fig. 3(a)	3	1.84E-02	2.93E-05	Fig. 3(b)
	Fig. 4(a)	10	4.97E-02	3.15E-04	Fig. 4(b)
	Fig. 5(a)	8	3.45E-02	9.56E-04	Fig. 5(b)
	Fig. 6(a)	6	2.73E-02	6.71E-04	Fig. 6(b)
	Fig. 7(a)	7	3.56E-01	9.58E-05	Fig. 7(b)
Algorithm of Marc *et al.*	Fig. 3(a)	20	1.17E-01	9.78E-04	Fig. 3(c)
	Fig. 4(a)	14	2.65E-01	9.13E-04	Fig. 4(c)
	Fig. 5(a)	32	6.47E-02	9.99E-04	Fig. 5(c)
	Fig. 6(a)	4	4.09E-02	1.15E-04	Fig. 6(c)
	Fig. 7(a)	15	5.10E-01	9.95E-04	Fig. 7(c)

almost equal mean values. Median filtering removes the salt-and-pepper noise and the line structures. The grid structures are also affected. Mean filtering blurs the edges and structural details providing smeared pictures.

Figure 4(a) is a noisy input, generated with additive Gaussian noise on Fig. 2(a). All three algorithms are found to be capable of reducing the Gaussian noise [Fig. 4(b)–(d)], but except for our algorithm, the line and grid structures are distorted. This distortion is highly noticeable in Marc *et al.*'s algorithm. Figure 4(e)–(g) indicate the outputs of mean, median filtering and Nagao and Matsuyama's algorithm. Mean filtering blurs the sharp edges and reduces noise in the background to some extent. Median filtering also damages the line-and-grid structures. The background noise is not as reduced as by the mean filtering algorithm. Nagao and Matsuyama's algorithm, on the other hand, nicely maintains all the step edges and cleans the background noise, but it destroys the roof edges to some extent. Figure 5(a) is the input Lincoln image, while Fig. 5(b)–(d) depict, respectively, the outputs by the proposed method, the method of Marc *et al.* and the Gaussian smoothing technique. A comparison of the outputs shows that both in Fig. 5(c) and (d) the nose and some portion of the lips and ear are distorted, while those are well preserved in Fig. 5(b). In Fig. 5(c), although most of the features are clean and sharp, the nose, lips and part of the ear are found to be seriously affected. This may be attributed to the presence of roof edges and the fact that the central pixel is influenced not only by its 3×3 neighborhood, but also by pixels beyond that.

For the Biplane image [Fig. 6(a)], the propeller is nicely preserved by our method [Fig. 6(b)], while it disappears for the other two methods [Fig. 6(c) and (d)].

In order to examine the effect of the proposed algorithm on structured noise, Fig. 2(a) has been corrupted with structured noise of one pixel, two pixels and three pixels. The results of different algorithms are shown in Fig. 7. It is noted that the algorithm of Marc *et al.* cleans the noise, but due to the interaction of the background pixels and those on the roof edges the line-and-grid structures are highly distorted. The ellipse in the figure is also distorted, but it is free from noise. Median filtering produces excellent results from the viewpoint of noise cleaning. Line structures are completely absent, while the grid structures are affected. Mean filtering does not clean the noise. Moreover, the edges are drastically blurred. Nagao and Matsuyama's algorithm removes the lines and damages the grid structures. Background noise is not noticeably removed. The proposed algorithm, on the other hand, preserves the structural information. Noise in the ellipse is cleaned. Some of the noisy pixels in the background are removed, while some are very prominent. This is due to the fact that three pixels of structured noise may sometimes represent meaningful regions and carry adequate information.

Finally, Table 1 indicates the total number of iterations required for automatic termination of the algorithm and the initial and final values of the image quality index.

The number of iterations in the algorithm of Marc *et al.* is large compared with that in the proposed algorithm. This is due to the interaction between the roof edge and background pixels. It causes ΔIQI to fluctuate a little and damages the smoothed version of the corresponding image. For the Biplane image, however, the iteration numbers for the two algorithms are almost equal. This investigation clearly establishes the superiority of the proposed method over the other two schemes. Note that for the other algorithms such as Gaussian, mean and median, considered in the present investigation, are not iterative in nature and hence they are not included in Table 1.

7. CONCLUSIONS

An iterative smoothing algorithm capable of performing various tasks such as cleaning salt-and-pepper noises, preserving roof edges, stretching step edges and reducing low intensity variations has been developed.

In order to determine the type of operation to be performed, an index characterizing the topography of a region has been defined. This index depends on the concept of both within region and boundary gradients. An explanation for critical gradient for the purpose of enhancing weak edges has been provided based on the concept of anisotropic diffusion process.

An index for image quality has been defined which makes the termination of the algorithm fully automatic. This criterion is applicable for any other iterative smoothing algorithm.

The superiority of the method over Gaussian smoothing and the method of Marc et al.[11] has been established for different synthetic and real images. Comparison, in some cases, shows that it is also performs better relative to mean, median and Nagao and Matsuyama's edge-preserving smoothing algorithm.[10] Further investigation is required for automatic selection of the critical gradient.

REFERENCES

1. D. W. Brown, Digital computer analysis and display of radionuclide scan, J. Nucl. Medicine 7, 740–745 (1966).
2. R. E. Graham, Snow removal—a noise stripping for picture signal. IRE Trans. Inform. Theory 8, 129–144 (1962).
3. S. K. Pal and R. A. King, Image enhancement using smoothing with fuzzy sets, IEEE Trans. Syst. Man Cybernet. 11, 494–501 (1981).
4. J. S. Lee. Digital image enhancement and noise filtering by use local statistics, IEEE Trans. Pattern Anal. Mach. Intell. 2, 165–168 (1980).
5. J. S. Lee. Refined filtering of image noise using local statistics, Comput. Graphics Image Process. 15, 380–389 (1981).
6. J. S. Lee, Digital image smoothing and sigma filter, Comput. Graphics Image Process. 24, 255–264 (1983).
7. D. C. C. Wang, A. H. Vagnucci and C. C. Li, Digital image enhancement: a survey, Comput. Graphics Image Process. 24, 368–381 (1983).
8. A. Lev, S. M. Zucker and A. Rosenfeld, Iterative enhancement of noisy images, IEEE Trans. Syst. Man. Cybernet. 7, 435–441 (1977).
9. D. C. C. Wang, A. H. Vagnucci and C. C. Li, Gradient inverse weighted smoothing scheme and the evaluation of its performance. Comput. Graphics Image Process. 15, 167–181 (1981).
10. M. Nagao and T. Matsuyama, Edge preserving smoothing, Comput. Graphics Image Process. 9, 394–407 (1979).
11. P. S. Marc, J. S. Chen and G. Medioni, Adaptive smoothing: a general tool for early vision, IEEE Trans. Pattern Anal. Mach. Intell. 13, 514–529 (1991).
12. P. Perona and J. Malik, Scale space and edge detection using anisotropic diffusion, Proc. IEEE Workshop Comput. Vis. 23–28 (1987).
13. E. H. Hall, Computer Image Processing and Recognition. New York: Academic Press (1979).

APPENDIX 1

Let the contrast at the (i, j)th pixel at iteration t and $(t + 1)$ be $c^t = (x_1/y_1)$ and $c^{t+1} = (x_2/y_2)$ [equation (17)], respectively. For digital images, $1 \le y_1, y_2 \le L$, and $0 \le x_1, x_2 \le L - 1$. It should be noted that x_1, x_2, y_1, y_2 are all integers. Since we are interested in $\Delta c = |c^t - c^{t+1}|$, without loss of generality we can assume $c^t > c^{t+1}$. We also assume that $\Delta c > 0$, i.e. there is a change in contrast (avoiding the ideal condition of $\Delta c = 0$). This implies $x_1 = 0$ and $x_2 = 0$ are also avoided. Thus, we obtain $1 \le x_1, x_2 \le L - 1$ (avoiding the ideal condition). Then the change in contrast of the pixel is given by:

$$\Delta c = \frac{x_1}{y_1} - \frac{x_2}{y_2}$$

$$= \frac{x_1 y_2 - y_1 x_2}{y_1 y_2}.$$

The minimum of Δc, can be achieved by minimizing the numerator and maximizing the denominator. On a digital grid, the minimum of $x_1 y_2 - y_1 x_2$ should be unity and the maximum of $y_1 y_2$ is $L(L - 1)$. Therefore, Δc_{min} becomes $(1/L)$. This is clear from the following two cases.

Case 1. $y_1 = y_2 = y$ (say).

Then $\Delta c_1 = (x_1 - x_2)/L$. Considering minimum numerator of unity we obtain $\Delta c_{1min} = 1/L$.

Case 2. $y_1 \ne y_2$.

Then the largest value for the denominator is $y_1 y_2 = L(L - 1)$ with $y_1 = L - 1$ and $y_2 = L$. Under this situation we obtain:

$$\Delta c_2 = \frac{Lx_1 - (L-1)x_2}{L(L-1)}$$

$$= \frac{L(x_1 - x_2) + x_2}{L(L-1)}.$$

To attain unity for the numerator we can consider $x_1 = x_2 = 1$. Therefore, $\Delta c_{2min} = 1/(L(L-1))$. Comparing cases 1 and 2 we obtain:

$$\Delta c_{2min} < \Delta c_{1min}.$$

Hence, the minimum change in contrast of a pixel in a digital image is:

$$\Delta c_{min} = \Delta c_{2min}$$

$$= \frac{1}{L(L-1)}.$$

Therefore, the termination criterion reduces to:

$$\Delta_{IQI} < \frac{1}{L(L-1)}. \tag{A1}$$

Note that this terminating criterion can be used with any iterative smoothing algorithm.

About the Author—SAMBHUNATH BISWAS obtained the M.Sc. degree in Physics from the University of Calcutta in 1973. He was in electrical industries in the beginning as a graduate engineering trainee and then as a design and development engineer. He was a UNDP fellow at MIT, U.S.A., studying machine vision in 1988. He is now a programmer at the Machine Intelligence Unit in Indian Statistical Institute, Calcutta. His research interests include image processing, machine vision, computer graphics and artificial intelligence.

About the Author—NIKHIL R. PAL obtained the B.Sc. degree with honors in Physics and Master of Business Management from the University of Calcutta in 1979 and 1982, respectively. He obtained the M.Tech (Comp. Sc.) and Ph.D. degree (Comp. sc.) from the Indian Statistical Institute in 1984 and 1991, respectively. Currently he is an Associate Professor in the Machine Intelligence Unit of Indian Statistical Institute, Calcutta. During September 1991–February 1993 and July 1994–December 1994 he was with the Computer Science Department of the University of West Florida. He was also a guest faculty of the University of Calcutta. His research interest include image processing, fuzzy sets theory, measures of uncertainty, neural networks and genetic algorithms. He is an Associate Editor of the International Journal of Approximate Reasoning and IEEE Transactions on Fuzzy Systems.

About the Author—SANKAR K. PAL is a Professor and Founding Head of Machine Intelligence Unit at the Indian Statistical Institute, Calcutta. He obtained the M.Tech. and Ph.D. degrees in Radiophysics and Electronics in 1974 and 1979, respectively, from the University of Calcutta, India. In 1982 he received another Ph.D. in Electrical Engineering along with DIC from Imperial College, University of London. In 1986 he was awarded Fulbright Post-doctoral Visiting Fellowship to work at the University of California, Berkeley, and the University of Maryland, College Park, U.S.A. In 1989 he received an NRC-NASA Senior Research Award to work at the NASA Johnson Space Center, Houston, Texas, U.S.A. He received the 1990 Shanti Swarup Bhatnagar Prize in Engineering Sciences, 1993 Jawaharlal Nehru Fellowship, 1993 Hari Om Prerit Vikram Sarabhai Research Award, 1993 NASA Tech Brief Award and the 1994 IEEE Transaction on Neural Networks outstanding paper Award. He is a co-author of the book Fuzzy Mathematical Approach to Pattern Recognition, John Wiley & Sons (Halsted), N.Y., 1986 and a co-editor of the book Fuzzy Models for Pattern Recognition, IEEE Press, NY, 1992. Professor Pal is a Fellow of the IEEE, U.S.A., Indian National Science Academy, National Academy of Sciences, India, Indian National Academy of Engineering, Institute of Engineers, India, and the IETE; Associate Editor, IEEE Transactions Neural Networks, Pattern Recognition Letters, Neurocomputing, Applied Intelligence, Information Sciences (C) and Far-East Journal of Mathematical Sciences and an Executive Advisory Editor, IEEE Transactions on Fuzzy Systems and International Journal of Approximate Reasoning, North Holland.