A COMPARISON OF LINEAR AND NONLINEAR SCALE-SPACE FILTERS IN NOISE

Richard Harvey, Alison Bosson, J. Andrew Bangham
School of Information Systems, University of East Anglia,
Norwich, NR4 7TJ, UK.
Tel: +44 1603 593257; fax: +44 1603 593345
e-mail: {rwh,bosson,ab}@sys.uea.ac.uk

ABSTRACT

The properties of two scale-space systems are compared by examining their performance in noise. It is found that in Gaussian noise linear diffusion and a new type of filter called the area sieve have similar performance but in impulsive noise of random amplitude the area sieve is superior.

1 INTRODUCTION

The advantages of scale-space have been well reported [1, 2], and working systems that adopt a scale-space approach have started to appear in the literature [3, 4]. The fundamental requirement is that, at increasing scale, the number of extrema in the image should decrease so that, at large scales, the task of processing the image is reduced. Most reported scale-space processors are based on linear, or nonlinear, diffusion equations but these have problems. Linear diffusion systems smear the edges which means that large-scale segmentations do not coincide with small-scale ones. Nonlinear diffusion systems overcome this, but are slow to compute and require a carefully chosen diffusion function. Neither is scale-calibrated: the scale parameter, s, does not give the true size of the objects in an image smoothed to particular scale.

An alternative is to use a morphological scale-space preserving filter [5, 6, 7]. These filters are scale-calibrated since the scale parameter is associated with an integer number of pixels. Furthermore, they are often quick to compute [8, 9]. This paper discusses the performance of a new filter called a sieve that decomposes an image by area. The filter is characterised by examining its performance in noise and comparing it to that achieved by the linear diffusion system.

2 DESCRIPTION OF THE PROCESSORS

Linear diffusion-based scale-space processors are well documented elsewhere [1]. In this paper the scale-space was generated using separable filters (γ = 0) where the image at scale s is computed as

\[ f^{(s)}(x,y) = \sum_{m=-\infty}^{\infty} T(m;s) \sum_{n=-\infty}^{\infty} T(n;s) f(x-m,y-n) \]

(1)

where \( f(x,y) \) is the pixel value at position \((x,y)\) and \( f^{(s)}(x,y) \) is the pixel value after smoothing to scale \( s \). \( T(n;s) \) is the discrete approximation to the Gaussian kernel

\[ T(n;s) = e^{-n I_n(s)} \]

(2)

and \( I_n(s) \) is the modified Bessel function of the first kind. The scale-selection surface \([1] p. 323\)

\[ d = s^2 \left( f_{xx}^{(s)} + f_{yy}^{(s)} \right)^2 \]

(3)

is used to locate the scale-space estimate.

The area-sieve is also documented elsewhere [10] but the basic of the algorithm is to consider an image as a graph \( G = (V,E) \). The set of edges \( E \) describes the adjacency of the pixels (which are the vertices \( V \)). The algorithm proceeds by defining a region, \( C_r(G,x) \) over the graph that encloses the pixel (vertex) \( x \),

\[ C_r(G,x) = \{ \xi \in C_r(G) | x \in \xi \} \]

(4)

where \( C_r(G) \) is the set of connected subsets of \( G \) with \( r \) elements. Thus \( C_r(G,x) \) is the set of connected subsets of \( r \) elements that contain \( x \). For each integer \( r \geq 1 \) the operators \( \psi_r, \gamma_r, M^r, N^r : \mathbb{Z}^V \to \mathbb{Z}^V \) are defined as

\[ \psi_r f(x) = \min_{\xi \in C_r(G,x)} \max_{u \notin \xi} f(u), \]

(5)

\[ \gamma_r f(x) = \max_{\xi \in C_r(G,x)} \min_{u \notin \xi} f(u), \]

(6)

\[ M^r = \gamma_r \psi_r, \quad N^r = \psi_r \gamma_r. \]

(7)

\( M^r \) is a greyscale opening followed by a closing defined over a region of size \( r \) and \( N^r \) is a greyscale closing followed by an opening over the same region.

The types of sieve known as \( M \)- or \( N \)-sieve are formed by repeated operation of the \( M \) or \( N \) operators. An \( M \)-sieve of \( f \) is the sequence \( \{ f^{(r)} \}_{r=1}^{\infty} \) given by

\[ f^{(1)} = M^1 f, \quad f^{(r+1)} = M^{r+1} f^{(r)}, \quad r \geq 1 \]

(8)
The N-sieve is defined similarly. The output of an area sieve is usually taken to be the set of granule functions

\[ d^{(r)} = f^{(r)} - f^{(r+1)} \quad \text{for each integer } r \geq 1 \] (9)

These form the scale selection surface and non-zero connected regions within granule functions are called granules. Each granule has sharp edges and, at a particular scale, all granules have the same area. In this sense the sieve is *scale calibrated*.

An example of the operation of the area sieve is shown in Figure 1 and 2. Figure 1 shows a twenty-five pixel image with one maximum in the centre and one minimum in the top left.

The first application of the sieve yields \( f^{(1)} = f \) and so is not shown. The top left of Figure 2 shows \( f^{(2)} \) which is \( f^{(1)} \) with all the extrema of unit area removed. The top right of Figure 2 shows \( |d^{(1)}| \). There are granules where there was a maximum and a minimum. The remaining feature has area of nine pixels so \( f^{(2)} \ldots f^{(9)} \) are identical and are not shown. The top left of Figure 2 shows \( f^{(10)} \) in which central nine-pixel maximum has been removed and on the bottom right of Figure 2 is \( |d^{(9)}| \) which shows the corresponding granule.

For the simple image shown in Figure 1 the sieve decomposition has only a few images but for a 100 by 100 pixel image there are \( 10^4 \) possible scale-space filtered images. So, for the purposes of display and for approximate scale determination, the outputs over a range of scales are summed to form a channel. Figure 3 shows the channels from an image of person. Features of different sizes appear in different spatial channels and those that had sharp edges retain them. The nostrils, eyes and mouth are localised in scale and space and such a decomposition is a good starting point for an accurate segmentation.

### 3 RESULTS

Real images are corrupted by noise and contain distortions such as ‘glint’. In an attempt to simulate some of these effects the images were corrupted with either Gaussian or impulsive noise. The target image consisted of a disc or square of amplitude 144 in the centre of a 100 by 100 pixel image with background 112. To this, was added either uncorrelated Gaussian noise (\( \mu = 0, \sigma = 24 \)), or alternatively, pixels were replaced with a random value in the range \((-6, 6)\) with a probability of 0.2. The resulting image was clipped into the range \((0, 255)\). The impulsive noise was chosen to give spikes of similar amplitude and occurrence probability to the glints observed in real images. Figure 4 shows
examples of the target images.

For both systems an iterative search was conducted over all scales to find the maximum. Scattergrams of results from 150 trials of the image corrupted by Gaussian noise for the two systems are shown in Figure 5. Each graph shows the estimates as circular markers with their co-ordinates being the estimates of \( x \), \( y \) and \( s \). The light-grey dots are the projection of the points in three-space onto each pair of axes. The linear scale-space processor has been scale-calibrated by normalising the mean of the cluster to equal the true area of the disc.

In the linear scale-space processor the peak was the maximum value of the normalised Hessian (3). To improve the precision of the position estimates a quadratic surface was least-squares fitted to the peak and its four-connected neighbours. The position of this peak gave the \((x, y)\) position estimate. Since the area sieve operates by “slicing off” the peaks and troughs, at a particular scale, the extrema are large and flat. So, to obtain the position of the disc, the centroid of all the pixels that had the value as the maximum at a particular scale was computed.

Exmaining Figure 5 shows that, in Gaussian noise, the two systems have comparable performance. Studies with the equivalent one-dimensional filters [7] showed that the linear diffusion system was more sensitive to Gaussian noise than the sieve. In two-dimensions this is not the case, probably because the scale-selection surface (3) in two-dimensions performs more spatial averaging.

For impulsive noise, Figure 6, the systems have very different behaviours. The sieve is hardly affected. This is an important property since in real vision systems impulses, glint and occlusion are commonplace.

The results are summarised in Table 1 which gives the sample standard deviations of the scale and position estimates.

<table>
<thead>
<tr>
<th></th>
<th>Gaussian noise</th>
<th>Impulsive noise</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Diffusion</td>
<td>Sieve</td>
</tr>
<tr>
<td>Disc ( x )</td>
<td>0.643</td>
<td>0.280</td>
</tr>
<tr>
<td>( y )</td>
<td>0.629</td>
<td>0.243</td>
</tr>
<tr>
<td>( s )</td>
<td>32.4</td>
<td>55.0</td>
</tr>
<tr>
<td>Square ( x )</td>
<td>0.738</td>
<td>0.280</td>
</tr>
<tr>
<td>( y )</td>
<td>0.646</td>
<td>0.288</td>
</tr>
<tr>
<td>( s )</td>
<td>37.7</td>
<td>50.2</td>
</tr>
</tbody>
</table>

| Table 1: Standard deviations of the estimates of \( x \), \( y \) and \( s \) |

4 DISCUSSION

The performance of the sieve in impulsive noise is unsurprising since it is known that, in one dimension, \( M \) and \( N \) filters behave similarly to median filters which have
good performance in impulsive noise. The sieve retains its performance in Gaussian noise probably because the cascaded operation described in (8) means that the output at large scales has a large support providing spatial averaging.

Since the sieve operates on area it is not affected by the geometry of the objects in the image. For example, if these estimation experiments are repeated with a rectangular target then the diffusion processor behaves poorly unless the image is simplified using an anisotropic diffusion system in which the anisotropy matches the aspect ratio of the rectangle.

References


