Using a Genetic Algorithm to adapt 1D nonlinear matched sieves for pattern classification in images

C. Jeremy Pye, J. Andrew Bangham and Richard Harvey

School of Information Systems, University of East Anglia, Norwich, NR4 7TJ, UK. Tel. +44 (0)603 592245,

Email: cjp@sys.uea.ac.uk, ab@sys.uea.ac.uk, rwh@sys.uea.ac.uk.

Many methods have been developed to recognise objects in a scene, most involving some pre-processing step to extract local information, such as edges, from the image of the scene. Usually these features are compared with some targets and a classification decision is made. A popular pre-processing step is to decompose the image into a set of scale related features. This type of pre-processing is useful because it can be arranged to remove details that would confuse a simple pattern recogniser. Conventionally, these scale related decompositions use linear processes (like wavelets [1] or the difference-of-Gaussians [2]). However, there is no need for such a decomposition to be based on linear operations and, in this paper, we concentrate on one, the recursive median sieve decomposition, that is nonlinear [3].

The recursive median sieve decomposition

The sieve has been shown to be a successful step in machine vision ([3], [4], [5]). Information that is localised in the original image remains localised in the transform domain (called the granularity domain). Furthermore the edges remain localised and not spread to other positions or scales in the granularity domain. A one-dimensional sieve works like this: the data are processing using a recursive median filter of window length three (= $2m$+1, m = 1). The output from this is then fed into a recursive median filter, [6], of length five (= $2m$+1, m = 2) and so on. The granularity decomposition is defined as the difference between these signals. In Figure 1, which shows such a cascade of filters, the two spikes of unit width have been separated into the granularity-one line. Each line represents a different scale. The important feature of the sieve is the cascade of operations. Sieves with open-close structuring elements, median filters and max./min. operations are also good at detecting edges. The primary difference lies in their noise immunity: Median based sieves are the most robust.

In linear systems the corruptions are often well modelled as being additive and Gaussian. This may not be a good model in image processing systems. There may be some additive noise but also there may be, for example, replacement noise in which samples are randomly replaced with random or fixed values. A special case is impulsive noise where one pixel is replaced with a random value. Impulsive noise is particularly damaging in linear filters since they operate on weighted sums of the signal. Sieves are very effective at removing impulsive noise.

Whilst the sieve is useful for analysing images, it needs some additions to form a recognition system. The matched sieve, [7], is an object recognition system based on the sieve. The object to be recognised is first sieved (horizontally or vertically) and the granules at each scale with largest amplitude identified (we use the top 20%). These granules are used to form a template. The template contains granules that are either zero or one (the granules are set to one where the original object decomposition contained a significant granule). This template is passed over an image and at every position a score is generated. The score is formed by adding the values resulting from a point-by-point multiplication of the template with the selected part of the image. Where the score is high we presume to have found the object. Figure 2 shows an example of this. Part of an image containing a picture of the head of one of the authors was separated from Figure 2A and used to form a template. Figure 2B was used as a test image. Figure 2C shows the score (where white indicates a high score). Figure 2D shows the result of inverse sieving the template at positions where the score was high. The head is located. This technique has been shown to work well for a variety of images.

Although the matched sieve is effective, constructing it is tiresome since it requires intervention. If the inputs are constantly changing then an automatic and adaptive method of choosing the appropriate matched sieve is needed.

The adaptive sieve

The adaptive sieve works by having an additional signal containing the desired objects. The adaptive sieve uses the desired signal to construct a sieve that removes the occlusions and noise from the actual signal. After the training is complete the system can be frozen and used as a recogniser. Alternatively, if the object to be recognised is constantly changing, then the filter can be left to adapt continuously and a low error can be interpreted as a recognition.

One adaptation scheme is the Widrow-Hoff LMS algorithm [8]. A schematic diagram of a system based on this is shown in Figure 3 (the system relies on the invertability of the sieve transform, [9], the theory for which has recently been proved). The system is trained by applying an input signal X to the system input. The signal is sieved to produce a vector of granules$G$, in which each member represents the component of the signal at an individual mesh m, and the weighted summation of these WG, produces the output Y. This is compared to the desired output T, and the error E is used to modify the weights on each input, so that the next time the same input vector is applied, the output Y is nearer the desired output T. Note that, although the filtering is performed in the granularity domain, the error is measured in the signal domain. The error is computed as $E = Y - T$ and the weight update equation is

$$W_j^{t+1} = W_j^t + 2\mu.E.X_j^t \tag{1}$$

An example of operation of the LMS sieve is shown in Figure 4. In Figure 4A the desired signal (shown in Figure 4B) occludes or has been (partially) occluded by some other objects. Figure 4C is a density plot of the filter weights (one for each mesh). In this case the signal only contained information at mesh 8. The bright line at mesh 8 demonstrates that the filter has converged. A more testing problem is shown in Figure 5 (which will be described later).

A significant problem with this configuration is impulsive noise. The least-squares error metric associated with the Widrow-Hoff algorithm may mean that large amounts of noise will perturb the sieve coefficients significantly. To this end we have developed an error metric that is sensitive to the relative position of the granules but insensitive to their amplitudes. It is the Hamming distance between the hard-clipped output of the adaptive sieve and the hard-clipped sieve transform of the desired signal. By hard-clipping we mean that all non-zero data values are set to one. The effect of this is to encourage the system to ignore the intensity of the features and concentrate the adaptive effort on positional information.

The use of an error measure like the Hamming distance brings problems: the filter is much harder to adapt using a conventional adaptation scheme. To overcome this problem an alternative adaptive sieve based on the Genetic Algorithm (GA) has been developed. Referring to Figure 6, this system operates to select the appropriate meshes that are then inverted to form the adaptively filtered output. We hope that the GA will act so that the output will have less noise than the input. The GA is driven by the new error signal, E, which can be written

$$E = \sum_{k=n}^{n+15} N_k\left(\tilde{\mathbf{y}}_k \oplus \tilde{\mathbf{d}}_k\right) \tag{2}$$

where the symbol $\oplus$ denotes modulo-2 arithmetic and the operator $N_k(\bullet)$ is the Hamming weight. In this system the Hamming weight is the number of ones in the binary word unless the noisy data vector $\tilde{\mathbf{x}}_k$ is all zeros. This measure, the Hamming distance, indicates the number of bits that are dissimilar in $\tilde{\mathbf{y}}_k$ and $\tilde{\mathbf{d}}_k$. Here, as in Figure 6, $\mathbf{y}_k$ is the output of the adaptive sieve at time k (which in this system is the same as the output from the GA) and $\mathbf{d}_k$ is the sieve transform of the desired output at time k.. The tilde denotes the hard-clipping operation. In Figure 6 the lines from the GA to the AND gates can be thought of as binary weights. If the weight is one then a particular mesh is selected but if it is zero then granules at that mesh are not included in the inverse transform. These weights are called *genes* and the vector of genes is called a *chromosome*. For simplicity Figure 6 shows only one error signal and one chromosome, but in our GA there are actually many. For the m^{th} chromosome we can define an error, E_m, as in (2). The *fitness* of the m^{th} chromosome is defined as

$$F_m = 1 - \frac{E_m}{\sum_{n=1}^{N_c} E_n} \tag{3}$$

where N_c is the number of chromosomes in the *population*. The next stage is to refine the population to form a new collection of chromosomes known as the *breeding population*. The breeding population is formed by selecting chromosomes at random from the population with a probability proportional to their fitness.

The breeding population is processed using two mechanisms, *mutation* and *crossover*. In mutation, genes are switched randomly throughout the breeding population with a probability P_M. In crossover, the chromosomes in the breeding population are examined pairwise and at some random position they are cut in two, with probability P_X, and their tails are swapped. A maximum of one crossover per chromosome is permitted. Crossover is an attempt to change fit chromosomes into other fit chromosomes. *Children* are then selected at random and used replace a portion of the main population to maintain diversity.

The GA used here is novel because we do not use uniform crossover and mutation probabilities. This is because the algorithm operates across the granularity spectrum. This means that at mesh six for instance (where the granules have a width of six samples) there are six chances of crossover or mutation of the mesh-six gene. This is unfair since the smaller granules are of equal importance in an object description. To avoid this the actual probability of gene crossover and mutation $p_X^{(m)}$ and $p_M^{(m)}$ at mesh m are

$$p_X^{(m)} = 1 - \left(1 - P_X\right)^m$$
$$p_M^{(m)} = 1 - \left(1 - P_M\right)^m \tag{4}$$

In our system we have $P_X = 0.1$ and $P_M = 0.02$.

The data used to test the Genetic Algorithm sieve contain small sequences of fifteen samples duration that we call *targets* and also variable length replacement noise pulses. The scheme for generating the signal is as follows. Initially all the data are zero. At every sample the probability of initiating a noise or target event is P_i=0.025. If an event is selected then a random integer is generated from a random number generator with a uniform distribution lying in the interval [0,55]. If the number lies in the interval [31,49] or is 15 then a target is initiated. Otherwise a noise pulse of random amplitude is initiated. The length of the pulse is given by the value of the random number unless that number lies in the range [50,55] in which case the pulse is double the length of the number. This process is repeated another three times. The effect of this scheme is to corrupt the signal with a wide variety of replacement pulses. We have manipulated the noise to avoid pulses of length 15 (the same length as the signal).

An example of some uncorrupted target data are shown in Figure 7A. It shows five target objects. Beneath these data is shown the sieve transform showing meshes 1 to 30. As expected all the targets have the same shape in the granularity domain even though all the targets have different amplitudes. Figure 7B shows the corrupted target data and its corresponding granularity spectrum. When a noise pulse overlaps the target there is some corruption of the signal granularity spectrum but the main features remain. Figure 7C shows the output of an adaptive sieve that has converged using the Genetic Algorithm. Most of the granules have been removed and most of the signal granules are intact.

Results

In practice the genetically adaptive filter needs to converge to the optimum solution. Figure 8 shows an example of this adaptation. Figure 8A shows the uncorrupted signal - the x-axis scale means that the detail has been lost and the targets appear as spikes. Figure 8B shows the corrupted signal and Figure 8C shows the output (the same data as in Figure 5). In the these figures the mean squared error (MSE) and mean absolute error (MAE), over the chromosome population, are given at the end of the sequence. Both the population and the breeding population contain 20 chromosomes. The output of the adaptive sieve (Figure 8C) does not contain the long noise pulses that corrupt the input to the genetically adaptive filter (Figure 7B). Figure 8D is a plot of the fittest chromosome as a function of iteration number. Recall that the fittest chromosome is a binary word used to select the which of the meshes are to be combined to form the filtered signal. In Figure 8D the black regions indicate where the chromosome is one and the white areas show where the chromosome is zero. Initially the chromosomes are chosen randomly and so the fittest chromosome is the one that is closest to the target granularity spectrum.

Figure 5 shows the same thing but using the Widrow-Hoff sieve. The error at the end of the run is greater but at this stage the comparisom between the two schemes is ongoing and it is hard to declare an outright winner.

The signal has large gaps between the target features (this is seen clearly in Figure 8B). In these gaps the filter will try to adapt to minimise the noise whereas when a target feature is present the filter adapts to minimise the Hamming distance between the signal and the target features. In a conventional adaptive filter this would be a serious problem since the optimal filter is different in these two regions but with a Genetic Algorithm this is easily handled since the GA copes with multiple minima naturally. Occasionally, when there is noise present, the GA will find that another chromosome (one that has genes that are zero where there are noise granules) gives a lower error than the chromosome closest to the optimal matched sieve. This is seen in as occasional breaks in the lines shown in Figure 8D.

By the end of the adaptation period the Genetically Adaptive Filter has converged to a solution close to that which would have been chosen for a matched sieve. It differs for two reasons: the target signal occasionally contains targets that overlap each other; there may not have been sufficient noise at a particular mesh to force that gene to zero.

Discussion

In the cameo problem we have studied here the filter was required to adapt to filter optimally for one feature. It should be simple to modify it so that it filters the signal adaptively for more than one feature. This could be done by adding the desired granularity spectrum as pre-defined chromosomes to the population. These need not match the target granularity spectrum exactly since the GA will adapt them.

A further refinement would be to increase the size of the chromosome to allow the GA to operate on a time window of the granularity spectrum of the corrupted data. This would be helpful if the system was used as part of a recognition system since it is the interrelationship between granules that defines an object. Although this would mean that the system is more complicated and hence slower we are hopeful that the performance degradation will not be serious. This because the sieve is an efficient algorithm (order *N*) [10] and our experiments have shown that adaptation using complicated targets is generally faster than for simpler ones since there is a better chance of chromosome including some of the target granules.

References

1. I.Daubechies, *The wavelet transform, time-frequency localisation and signal analysis*, IEEE Trans. Inf. Th., 36, 1990, pp 961 - 1005.

2. D.Marr, Vision, W.H.Freeman and Company, New York, 1982.

3. J.A.Bangham, T.G.Campbell, R.V.Aldridge, *Multiscale median and morphological filters used for 2D pattern recognition, Signal Processing*,

4. C.J.Pye, J.A.Bangham, *2D pattern recognition using 1D multiscale sieve decompostion*, Proc EUSIPCO '94.

5. J.A.Bangham, *Using the root datasieve to recognise and locate patterns in a data sequence*, Proc. EUSIPCO '92, 1992, pp 983-987.

6. I. Pitas and A.N.Venetsanopoulos, Nonlinear Digital Filters, Kluwer Academic Publishers, Norwell Mass., 1990.

7. C.J.Pye, J.A.Bangham, .J.Impey, R.V.Aldridge, *2D Pattern recognition using multiscale median and morphological filters*, Proc. Image Processing TH, and App., San Remo, Italy, June '93, pp 309 - 312.

8. B. Widrow and S. D. Stearns, Adaptive Signal Processing, Prentice-Hall International, Englewood Cliffs, 1985.

9. P.Chardaire, J.A.Bangham, C.J.Pye, D.Q.Wu, *Properties of multiscale morphological filters namely the morphological decomposition theorem*, SPIE Nonlinear signal processing conference, 1994, 2180, pp 148-154.

10. J.A.Bangham, S.J.Impey, F.W.D.Woodhams, *A fast 1D sieve transform for multiscale signal decomposition*, EUSIPCO '94, 1994

.