

Since the problem of depth estimation can be reduced to a 1-D problem, we consider also for a 2-D case a set of 1-D networks. We apply to the networks the synthetic stereo pair in Fig. 3, in which the boxes are at different distances: the one on the right is at the same distance as the background. The resulting depth map is also depicted in Fig. 3.

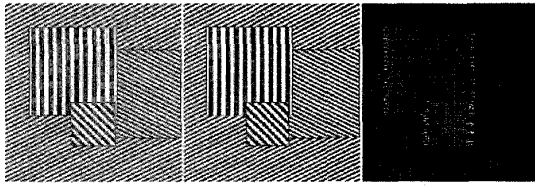


Fig. 3 Two stereo images

The box on the right is at the same distance as the background. The top box is closer to the viewer and the bottom box is still closer. The last image shows the result of the network: bright grey corresponds to objects closer to the viewer.

**Conclusions:** In this Letter, I have presented a resistive network using a feedback mechanism able to extract phase information from an image for depth evaluation. The information, related to a single frequency, can be properly organised by a further network, or the resistive network itself can be tuned to a more complex behaviour (e.g. regularisation of data). The design of this network makes sense if we can integrate the acquisition and elaboration processes on the same substrate (i.e. analogue VLSI implementation). To this end, the complexity of the operations has been reduced to a small number of primitives that can be implemented, as a first approximation, by low-power analogue VLSI techniques [4].

**Acknowledgments:** The author wishes to thank G.M. Bisio, S.P. Sabatini and M. Vanzi for useful suggestions. This research was partially supported by CEC-Esprit-8503 Project CORMORANT.

© IEE 1995

9 August 1995

Electronics Letters Online No: 19951326

L. Raffo (Department of Electrical and Electronic Engineering, University of Cagliari Piazza d'Armi, I-09123 Cagliari, Italy)

## References

- 1 SANGER, T.: 'Stereo disparity computation using Gabor filters', *Biol. Cybern.*, 1988, **59**, pp. 405-418
- 2 DE ANGELIS, G.C., OHZAWA, I., and FREEMAN, R.D.: 'Depth is encoded in the visual cortex by a specialized receptive field structure', *Nature*, 1991, **352**, pp. 156-159
- 3 RAFFO, L.: 'Resistive network implementing maps of Gabor functions of any phase', *Electron. Lett.*, 1995, **31**,
- 4 VITTOZ, E.A.: 'Analog VLSI signal processing: Why, where, and how?', *J. VLSI Signal Process.*, 1994, **8**, pp. 27-44

## Nearest-neighbour multichannel filter

K.N. Plataniotis, D. Androutsos, V. Sri and A.N. Venetsanopoulos

*Indexing terms:* Image processing, Filtering and prediction theory

The authors address the problem of noise attenuation for multichannel data. The proposed filter utilises adaptively determined data dependent coefficients. The special case of colour image processing is studied as an important example of multichannel signal processing.

**Introduction:** Vector processing based on order statistics (OS) is one of the most effective methods available to filter out noise in multichannel signals. In the multichannel case, however, the concept of vector ordering has more than one interpretation and the centremost vector inside a filter window can be defined in more

than one way depending on the distance function selected to measure dissimilarity among multivariate vectors [1]. A number of multichannel filters, such as the vector median filter (VMF) and the vector directional filter (VDF) [2], which utilise correlation among multivariate vectors using distance measures, have been proposed. In this Letter, an adaptive nearest-neighbour filter based on vector directional ordering is introduced. The new filter constitutes a generalisation of the class of vector directional filters.

**The filter:** Let  $y(x): Z^1 \rightarrow Z^n$  represent a multichannel signal and let  $W \in Z^1$  be a window of finite size  $n$  (filter length). The noisy vectors inside the window  $W$  are denoted as  $x_j, j = 1, 2, \dots, n$ . The filter is a weighted average of all input vectors inside the window  $W$ . Therefore, the filter's output at the window centre is

$$\hat{y} = \frac{\sum_{j=1}^n w_j x_j}{\sum_{j=1}^n w_j} \quad (1)$$

Each one of the weights is a function of the distance between the vector under consideration and all other vectors inside the filter window. In this Letter a neighbour weighting function is used to assign weights to each one of the vector inputs. A function similar to the  $k$ -NN rule discussed in [3] is used to regulate the contribution of the vector located at pixel  $i$ , and is defined in the following equation:

$$w_i = \frac{(d_{(n)} - d_{(i)})}{(d_{(n)} - d_{(1)})} \quad (2)$$

where  $d_{(n)}$  is the maximum distance in the filtering window, measured using an appropriate distance criterion, and  $d_{(1)}$  is the minimum distance, which is associated with the centremost vector inside the window. The value of the weight, above, expresses the degree to which the vector at point  $i$  is close to the ideal, centremost vector, and far away from the worst value, the outer rank. Both the optimal rank position  $d_{(1)}$  and the worst rank  $d_{(n)}$  are occupied by at least one of the vectors under consideration. It is evident that the outcome of the filter depends on the choice of the distance criterion selected as a measure of dissimilarity. Since our primary objective is to apply the new filter to colour images, the so-called vector angle criterion is used to calculate distances between the colour vectors [2]. This criterion considers the angle between two vectors as their distance. A scalar quantity

$$d_i = \sum_{j=1}^n A(x_i, x_j) \quad (3)$$

$$A(x_i, x_j) = \cos^{-1} \left( \frac{x_i x_j^t}{|x_i| |x_j|} \right) \quad (4)$$

is the distance associated with the noisy vector  $x_i$  inside the processing window of length  $n$ . An ordering of the  $d_i$ s,

$$d_{(1)} \leq d_{(2)} \leq \dots \leq d_{(n)} \quad (5)$$

implies the same ordering to the corresponding  $x_i$ s:

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)} \quad (6)$$

The proposed filter performs smoothing at all vectors which are from the same region as the vector at the window centre. It is reasonable to make the weights proportional to the difference, in terms of a distance measure, between a given vector and its neighbours inside the operational window. At edges or in areas with high details the filter smooths only over pixels at the same side of the edge as the centremost vector, since vectors with relatively large distance values will be assigned smaller weights and will contribute less to the final filter estimate. Thus, edge or line detection operations prior to filtering can be avoided with considerable savings in terms of computational effort.

Table 1: Noise distributions

Number	Noise model
1	Gaussian ( $\sigma = 30$ )
2	impulsive (4%)
3	Gaussian ( $\sigma = 15$ ) impulsive (2%)
4	Gaussian ( $\sigma = 30$ ) impulsive (4%)

*Application to colour images:* The adaptive nearest-neighbour filter is compared quantitatively with the widely used vector median filter (VMF) and the chromaticity based generalised vector directional filter (GVDF) [2]. The colour RGB test image 'Lena' has been contaminated using various noise source models in order to assess the performance of the filters under different noise distributions (see Table 1). A correlation factor  $\rho = 0.5$  is used in all the experiments to determine the corruption of pixel  $(i, j)$  in a channel, if the same pixel  $(i, j)$  is corrupted in any of the other two channels. The normalised mean square error (NMSE) has been used as a quantitative measure for evaluation purposes. It is computed as

$$NMSE = \frac{\sum_{i=0}^{N1} \sum_{j=0}^{N2} \|y(i, j) - \hat{y}(i, j)\|^2}{\sum_{i=0}^{N1} \sum_{j=0}^{N2} \|y(i, j)\|^2} \quad (7)$$

where  $N1, N2$  are the image dimensions, and  $y(i, j)$  and  $\hat{y}(i, j)$  denote the original image vector and the estimation at pixel  $(i, j)$  respectively. Table 2 summarises the results obtained for the test image 'Lena' for a  $3 \times 3$  filter window. The results obtained using a  $5 \times 5$  filter window are given in Table 3.

**Table 2:** NMSE ( $\times 10^{-2}$ ) for the 'Lena' image, window  $3 \times 3$

Noise model	FVF1	GVDF	VMF
1	0.8516	1.46	1.60
2	0.2667	0.30	0.19
3	0.3785	0.6238	0.5404
4	1.0864	1.982	1.6791

**Table 3:** NMSE ( $\times 10^{-2}$ ) for the 'Lena' image, window  $5 \times 5$

Noise model	FVF1	GVDF	VMF
1	0.6242	1.08	1.17
2	0.4269	0.54	0.58
3	0.4367	0.459	0.5172
4	0.7528	1.1044	1.0377

*Conclusion:* The adaptive nearest-neighbour filter introduced here smooths noise under different scenarios outperforming other widely used multichannel filters. Moreover, the new filter preserves the chromaticity component, which is very important in visual perception of colour images.

© IEE 1995

25 July 1995

Electronics Letters Online No: 19951294

K.N. Plataniotis, D. Androutsos, V. Sri and A.N. Venetsanopoulos (Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario M5S 1A4, Canada)

## References

- 1 PITAS, I., and VENETSANOPOULOS, A.N.: 'Order statistics in digital image processing', *Proc. IEEE*, 1992, **80**, (12), pp. 1893-1921
- 2 TRAHANIAS, P.E., and VENETSANOPOULOS, A.N.: 'Vector directional filters: A new class of multichannel image processing filters', *IEEE Trans.*, 1993, **IP-2**, (4), pp. 528-534
- 3 DUDANI, S.A.: 'The distance-weighted k-nearest neighbour rule', *IEEE Trans.*, 1977, **SMC-15**, pp. 630-636

## Reconstruction of character skeletons using Gabor filter features

A.P. Whichello and Hong Yan

*Indexing terms:* Signal processing, Digital filters, Character recognition

Features derived from Gabor wavelet filters are used to reconstruct the skeletons of handwritten digits. The features are linked into strokes which are suitable as input to character recognisers. Experimental results are given to show the effectiveness of the stroke-linking algorithm as a thinning method, producing well-connected skeletons for broken and noisy digits.

*Introduction:* Optical character recognition (OCR) is now a common tool in use throughout the world. For handwritten and poor quality print with broken characters, there are still formidable problems to be overcome and there is much ongoing research [1]. OCR requires the salient features of the true character to allow unambiguous identification. However, nominating the set of features to be used remains an open question.

Ideal printed text images are binary (black and white), but images produced by scanners are grey-scale because of noise, distortions and other causes in the scanner and the document being scanned. There is also significant loss of information in the thresholding process that subsequently converts the grey-scale image to binary [2]. Most of the databases of hand-written characters that are readily available for research only provide binary information, which is not always perfect. For example, the NIST (US National Institute of Standards and Technology) data contain several examples of wrongly segmented digits. Therefore in the case of processing binary data after imperfect thresholding, we must reconstruct a plausible skeleton by other methods. Moreover, the skeleton extraction method must be robust to cope with a wide range of source material, from high quality print to that with defects caused by different mechanisms. In this Letter it will be shown that feature extraction using Gabor filters produces reliable features for creating skeletons, which may be subsequently used in character recognition.

*Feature extraction using Gabor filters:* In recent years, Gabor filters have been used as models of low-level vision behaviour. Gabor filters are the modulation products of Gaussian and sinusoidal functions, developed originally by Gabor as optimal signal carriers in communications [3]. Marcelja introduced Gabor filters as a mathematical representation of the receptive profiles of visual cortical cells in mammals, performing spatial frequency analysis of visual information [4]. Daugman established that Gabor functions achieve the lower limit of joint uncertainty in the space and spatial frequency domains, producing the best localisation [5]. Others have used Gabor filters for edge detection [6], feature extraction [7] and image compression [8].

For our binary images, we use a feature extraction algorithm developed by Shustorovich as the basis of the feature extraction method [9]. Two dimensional Gabor filters are given by the following equation:

$$g(x, y) = e^{-(x^2+y^2)/\sigma^2} (\cos 2\pi fx + j \sin 2\pi fx) \quad (1)$$

where  $f$  is the frequency and  $\sigma$  the spread of the wavelet. Rotating the  $x$ - $y$  plane by  $\theta$  will produce a Gabor filter at orientation  $\theta$ .

Gabor filters for several orientations and dilations are pre-calculated and the data projected onto them at each processing centre of a regular grid. The features extracted are the amplitude of the filter output with the strongest response, at the orientation of the corresponding sine wavelet. Plotting line segments with lengths proportional to amplitude, at the noted orientation produces a visual reconstruction of the features.

*Method for linking features into strokes:* The line segments found at each local processing centre  $(m, n)$  have two nodes (index  $p = 0, 1$ ). An optimisation procedure, similar to the heuristic used in [10], is used to determine which of the neighbours of the current node represents the best match. The line segments are thus connected by links into a stroke. Results are obtained by calculating a cost function that finds the minimum of a weighted sum  $C_1 + \beta C_2 + \gamma C_3 + \delta C_4$ . The cost components are:  $C_1$ , distance to the next node;  $C_2$ , difference in orientation of the line segments being examined;  $C_3$ , difference in orientation of the line segment or link with the current course; and  $C_4$ , the amount of white space traversed by the line segment or link, (computed from the path integral over the data pixels). The scaling constants  $\beta, \gamma$  and  $\delta$  are chosen to make the cost components comparable. The current course

$$\phi_i = \tan^{-1} \left( \frac{L_i \sin \theta_i + \alpha L_{i-1} \sin \phi_{i-1}}{L_i \cos \theta_i + \alpha L_{i-1} \cos \phi_{i-1}} \right) \quad (2)$$

is the trend of the stroke's direction ( $L_i$  measures the length of the  $i$ th movement,  $\alpha$  puts some stiffness in the stroke as it is traced). The algorithm is implemented to prefer existing line segments over making new links. The steps are