

E 870

RP

JOURNAL ON

COMMUNICATIONS

VOLUME XLV.

JULY-AUGUST 1994

IMAGE PROCESSING II.

REGULAR PAPERS

FILTERING

- Comparison of zero crossing based edge detection filters
in the discrete domain J. De Vriendt and S. Simon 1
- A multi-step fast median algorithm for 2D data of various
word-length L. Alparone, V. Cappellini, and A. Garzelli 3
- Filtering via the discrete Fourier cosine transform T. Trump 7
- A comparison of FIR and IIR filter banks in image subband
coding applications M. Domański and R. Swierczyński 10

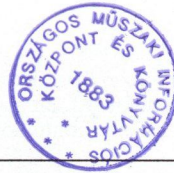
RESTAURATION, RECOGNITION

- Moment-based image reconstruction in the presence of noise R. C. Papademetriou 13
- Stochastic and deterministic methods in motion picture restoration R. D. Morris and W. J. Fitzgerald 17
- Fish species recognition using a Bayesian
classification process N. Gastignolles, M. Cattoen, and M. Larinier 21
- Recognition of printed Berber characters using modified rapid transform J. Turán 24
- Pattern recognition method based on k nearest neighbor rule A. Jóźwik 27

IMAGE ANALYSIS

- Segmentation and counting of ovisacs with morphological filtering
in ultrasound images I. E. Pratikaris, B. G. Mertzio, and J. Cornelis 30
- Estimation of edge parameters and image blur from local derivatives V. Kayargadde and J.-B. Martens 33
- Low cost vision system for quality control J. Climent and S. Alvarez 36
- Crowding evaluation by means of multi-hypothesis modelling C. S. Regazzoni and A. Tesei 38

Followed on the back cover.



COMPARISON OF ZERO CROSSING BASED EDGE DETECTION FILTERS IN THE DISCRETE DOMAIN

J. DE VRIENDT* and S. SIMON

LABORATORY FOR COMMUNICATION ENGINEERING UNIVERSITY OF GHENT
SINT-PIETERSNIEUWSTRAAT 41.
B-9000 GENT, BELGIUM

1. INTRODUCTION

Edge detection is mainly the process that measures, detects and localizes changes of intensity. Edges may or may not correspond to an object boundary, but they have the desirable property of drastically reducing the amount of information to be processed subsequently while preserving information about the shapes of the objects in the scene. Most vision systems use an edge description of the scene as input to higher level image understanding processes.

In literature a number of optimal linear filters have been proposed for edge detection. These filters can be divided into two classes. The first class of filters are first order derivatives of low-pass filters and look for maxima in the output of the filter (cf. Canny [1], Deriche [2], Sarkas and Boyer [11]). The second class of filters are second order derivatives of low-pass filters and detect zero crossings in the output of the filter (Marr and Hildreth [8], Deriche [3], Sarkar and Boyer [10], Shen and Castan [12]). These filters are always obtained in the 1D continuous domain. The criteria used are those introduced by Canny: good detection, good localization and a low spurious response. The influence of nearby edges is not taken into account. An important aspect in obtaining these filters is the natural uncertainty principle. This principle says that the product of the good detection and the good localization criteria is invariant with respect to spatial scale. However, the filtering has to be performed in the discrete domain. Discrete filters are obtained by sampling the continuous filter.

2. 1D EDGE DETECTION IN THE DISCRETE DOMAIN

In this section we will discuss the influence of the sampling process on the performance of the different edge detectors. The performance criteria are similar to those used by Sarkar and Boyer [10] (adapted to the discrete domain): large SNR (i.e. a large slope ratio), a good localization ($Localization=1/localization$ error) and a low multiple response (a large MRC -value will result in fewer spurious responses). In the continuous domain, the product of the SNR and the $Localization$ is independent of the spatial scale of the filter. However, increasing the spatial scale results in an increase of the influence of nearby edges (De Vriendt [5]). This influence should be taken into account both in the continuous and in the discrete domain. Therefore, in the discrete domain we also introduced a parameter d_h , which takes into account the influence of nearby edges. d_h is defined as the minimal distance between two edges with the same intensity step (but with a different sign) such that the localization error has a maximum of 1/2 pixel. The performance ($perf = SNR \cdot Localization \cdot MRC$) of different filters is compared for a given d_h . The derivation of the expression of the SNR , $Localization$ and MRC are found in De Vriendt [6].

The SNR is given by the slope ratio, i.e.

$$SNR = \frac{|y_s(1) - y_s(0)|}{\sqrt{E[(y_n(k) - y_n(k+1))^2]}}, \quad (1)$$

where $y_s(k) = s(k) * h(k)$, $s(k)$ the Gaussian filtered (and sampled) version of the ideal step edge $A \cdot U(x)$ and $h(k)$ the

zero crossing based edge detection filter. The Gaussian filter g_σ , represents the low-pass filtering introduced by the system's optics. $y_n(k) = n(k) * h(k)$, where $n(k)$ is white Gaussian noise, i.e. $E[n(k)n(k+\tau)] = \sigma_n^2 \delta(\tau)$. After some calculations we obtain

$$SNR = \frac{A^2}{\sigma_n^2} \frac{\left| \sum_{m=-\infty}^{+\infty} h(m) \left[\Phi\left(\frac{1-m}{\sigma_s}\right) - \Phi\left(\frac{-m}{\sigma_s}\right) \right] \right|}{2(1-\rho) \sum_{m=-\infty}^{+\infty} h^2(m)}, \quad (2)$$

where

$$\rho = \frac{\sum_{m=-\infty}^{+\infty} h(m)h(m+1)}{\sum_{m=-\infty}^{+\infty} h^2(m)}. \quad (3)$$

The *Localization* is obtained as

$$Localization = \frac{1}{\sqrt{E[\hat{l}^2] - (E[\hat{l}])^2}}. \quad (4)$$

In De Vriendt [4], [6] we have obtained a very good approximation for the distribution of the localization error. From this distribution the variance of the localization error is obtained. The MRC is given by

$$MRC = \frac{d_{zc}}{\bar{W}}, \quad (5)$$

where d_{zc} is the mean distance between zero crossings in $y_n(k)$ and $\bar{W}$ is given by

$$\bar{W} = \sqrt{\frac{\sum_{m=-\infty}^{+\infty} m^2 h^2(m)}{\sum_{m=-\infty}^{+\infty} h^2(m)}}. \quad (6)$$

It is easy to prove that (crf. Papoulis [9])

$$MAC = \frac{\pi}{\arccos \rho} \frac{1}{\bar{W}}. \quad (7)$$

Due to the low-pass filtering introduced by the system's optics, the natural uncertainty principle is no longer valid. The minimum in the standard deviation of the localization error is no longer obtained for a filter width which approaches zero. Furthermore, the sampling of the continuous filters proposed by Sarkar and Boyer [10], Deriche [3] and Shen and Castan [12] leads to aliasing. As a consequence, the response of the filter to a constant signal is no longer zero. The detection filter has to be modified. Alternatively, we can use the projection function $p(x)$ (i.e. the double integral of $h(x)$), sample $p(x)$ to obtain $p(k)$ and finally compute $h(k)$ by convolution of $p(k)$ with a small filter for approximating the second order derivative i.e.

$$h(k) = p(k+1) - 2p(k) + p(k-1). \quad (8)$$

The filter considered in the comparison are in case we start from the projection filters:

$$p_{Gauss}(k) = g_\sigma(k) \quad \text{for } |k| \leq 5.5\sigma \\ = 0 \quad \text{for } |k| > 5.5\sigma \quad (9)$$

$$p_{ISEF}(k) = a^{|k|} \quad (10)$$

$$p_{Sar}(k) = \left(1 + \beta|k| + \frac{\alpha+1\beta^2}{\alpha+2\beta^2} k^2\right) e^{-\beta|k|} \quad (11)$$

$$p_{Der}(k) = (1 + \beta|k|) e^{-\beta|k|} \quad (12)$$

* The author is now with Alcatel Bell

In this case the zero crossing based edge detection filter $h(k)$ are obtained by using equation (8). In case we immediately start from the sampled versions of the continuous edge detection filters we have:

$$h_{Gauss}(k) = \frac{k^2 - \sigma^2}{\sigma^4} g_\sigma(k) + cst \quad \text{for } |k| \leq 5.5\sigma$$

$$= 0 \quad \text{for } |k| > 5.5\sigma \quad (13)$$

where cst is chosen such that $\sum_{m=-\infty}^{+\infty} h(k) = 0$. For the ISEF we define $h(k)$ as (cfr. Shen and Casten [12])

$$h_{ISEF}(k) = \frac{1-a}{1+a} a^{|k|} - \delta(k) \quad \text{with } a \in]0, 1[. \quad (14)$$

This filter (as well as the next 2 filters) can easily be implemented as an IIR filter. As an approximation for the filter of Sarkar and Boyer [11] we use

$$h_{Sar}(k) = \left(1 + \gamma(\alpha\beta|k| - \frac{\alpha+1}{2}\beta^2|k|^2) \right) e^{-\beta|k|} \quad (15)$$

with

$$\gamma = \frac{(1+a)(1-a)^2}{\alpha\beta} \left[\frac{1}{-2\alpha(1-a) + (\alpha+1)\beta(1+a)} \right] \quad (16)$$

$$\text{en } a = e^{-\beta} \quad (17)$$

The value of γ is chosen such that $\sum_{m=-\infty}^{+\infty} h(k) = 0$. Sarkar and Boyer choose $\alpha = 0.312$ because this value maximizes the product of SNR , localization and MRC . The filter proposed by Deriche is a special case of the filter of Sarkar and Boyer (i.e. for $\alpha = -1$).

$$h_{Der}(k) = (1 - \gamma\beta|k|)e^{-\beta|k|} \quad (18)$$

with

$$\gamma = \frac{1 - e^{2\beta}}{2\beta e^{-\beta}}. \quad (19)$$

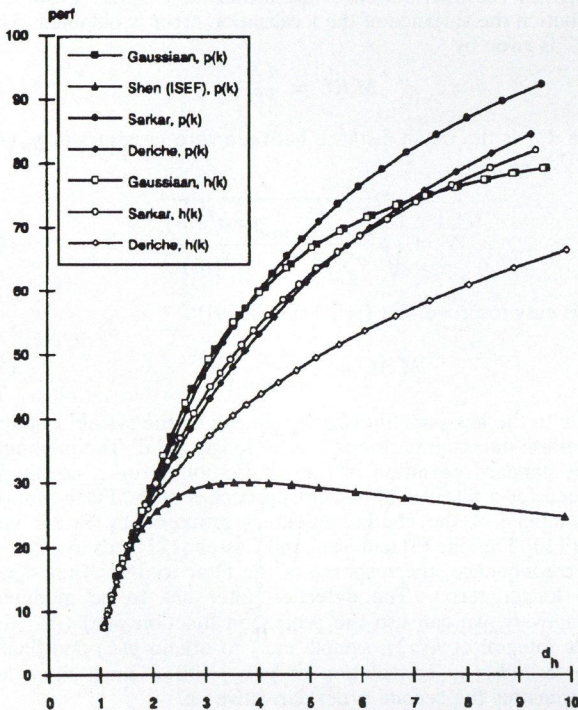


Fig. 1. The performance (perf) as a function of d_h for the different filters in the 1D domain ($d_h = 0.6$)

Again the value of γ is chosen such that $\sum_{m=-\infty}^{+\infty} h(k) = 0$.

In the continuous domain the best performance is obtained for the filter of Sarkas and Boyer, followed by the filter of Deriche, the Gaussian filter and the filter of Shen and Castan. This result is still true for large filter widths i.e. for large d_h . We could expect this result because for large (spatial) widths of a filter, the bandwidth of the filter is very small and consequently there is no

aliasing (except for the filter of Shen and Castan (De Vriendt [6])). However, other conclusions are obtained for smaller d_h (see Fig. 1). For d_h smaller than 3.84, the Gaussian filter gives the best performance. Except for the filter of Shen and Castan, all filters have an increasing performance when d_h increases. Of course, an increasing d_h also results in a larger influence of nearby edges which is not reflected in the performance. The optimum of $perf \cdot d_h^{-1}$ is obtained for the Gaussian filter ($d_h = 2.78$) for a spread of the Gaussian filter of $\sigma = 1.41$. In De Vriendt [7], [6] we obtain the optimal 1D edge detection filter using these criteria.

3. 2D EDGE DETECTION IN THE DISCRETE DOMAIN

A 2D filter can be obtained by convolving a linear edge detection filter aligned normal to the edge direction with a projection function parallel to the edge direction. A substantial saving in computational effort is possible if the projection function is the double integral of the edge detection filter. Indeed, the edges can be detected by first filtering the image with a 2D version of the projection filter and then computing the second order directional derivative in the direction of the gradient. This method is used by Marr and Hildreth [8] (though they compute the Laplacian instead of the directional derivative), Deriche [3], Sarkar and Boyer [10] and Shen and Castan [12].

The performance criteria are the natural extensions of the criteria in the 1D domain. These criteria are: SNR , $Localization$, MRC and d_h . In the 2D continuous domain, the product of SNR and $Localization$ is independent of the spatial scale. On the other hand, in the 2D domain, the SNR increases strongly by increasing the width of the filter, and the $Localization$ is almost constant (slightly increasing) for an increasing width of the filter. If we would only take into account the SNR , $Localization$ and MRC in the performance evaluation, the optimal filter would have an infinite width. Therefore, the parameter d_h is very important in the 2D domain. d_h takes into account the influence of nearby parallel edges. For an edge parallel to a given direction, the length of the edge in that direction will also be limited. Therefore, also the extent of the projection function $p(k)$ should be limited. A new parameter σ_{opt} is introduced as a combination of σ_p , the spread of the projection function, and the spread of the low-pass sampling filter. The performance in 2D is defined as $perf = SNR \cdot Localization \cdot MRC / \sigma_{opt}$. Filter are compared for a given d_h .

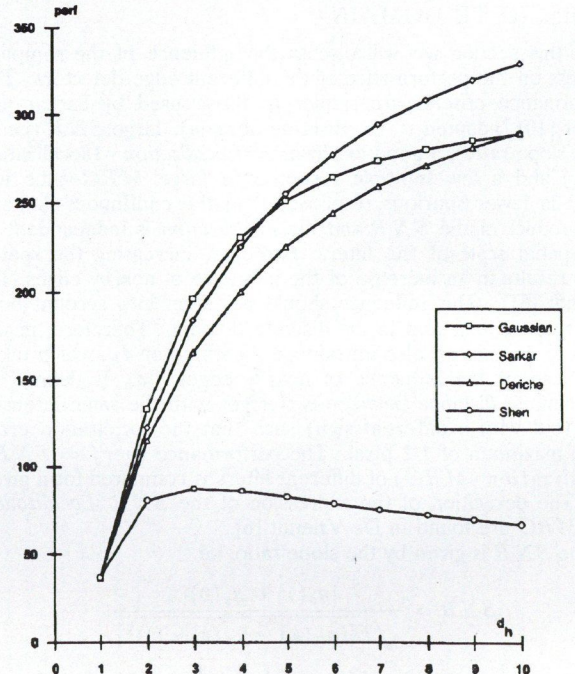


Fig. 2. The performance (perf) as a function of d_h for the different filters in the 2D domain

Also in the 2D domain the Gaussian filter has a better performance than the filter of Sarkar and Boyer for limited d_h ($d_h \leq 4.6$, see Fig. 2). For increasing values of d_h , the performance of the filters of Sarkar and Boyer and Deriche increases much faster than the performance of the Gaussian filter. Again, the filter of Shen and Castan gives the worst performance. In De Vriendt [6] we obtain the optimal 1D edge detection filter using these.

REFERENCES

- [1] J. Canny: "A computational approach to edge detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-8, pp. 679-698, 1986.
- [2] R. Deriche: "Using Canny's criteria to derive a recursively implemented optimal edge detector", *Int. J. Comput. Vision*, pp. 167-187, 1987.
- [3] R. Deriche: "Fast algorithms for low-level vision", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-12, pp. 78-87, 1987.
- [4] J. De Vriendt: "Effect of sampling, quantization and noise on the performance of the second directional derivative edge detector", Rep CV-9302, 1993, accepted for publication in *Multidimensional Systems and Signal Processing*.
- [5] J. De Vriendt: "Accuracy of the zero crossings of the second directional derivative edge detector", *Multidimensional Systems and Signal Processing*, Vol. 4, pp. 227-251, 1993.
- [6] J. De Vriendt: *Randdetectie en bewegingsestimatie in beeldsequenties*, Ph.D. thesis, 1994.

4. CONCLUSIONS

The result of a comparison of the performance of different filters in the continuous domain is not always valid in the discrete domain. For small values of d_h , different qualitative results are obtained in the discrete domain than in the continuous domain. In the continuous domain, the filter of Sarkar and Boyer is much better than the Gaussian filter. However, in the discrete domain, the Gaussian filter has the best performance for small values of d_h . This difference is mainly due to the aliasing that occurs when the continuous filters are sampled in order to obtain discrete filters.

- [7] J. De Vriendt: "Optimal discrete linear filters for step edge detection in images", *Proceedings EUSIPCO-94*, Edinburgh, September 1994.
- [8] D. Mar, E. Hildreth: "Theory of edge detection", *Proc. Roy. Soc. London Ser. B207*, pp. 187-217, 1980.
- [9] A. Papoulis: *Probability, random variables and stochastic processes*, Mc Graw-Hill, New York, 1965.
- [10] S. Sarkar, K. L. Boyer: "Optimal infinite impulse response zero crossing based edge detectors", *Comput. Vision, Graphics, Image Processing*, Vol. 54, pp. 224-243, 1991.
- [11] S. Sarkar, K. L. Boyer: "On optimal infinite impulse response edge detection filters", *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. PAMI-13, pp. 1154-1171, 1991.
- [12] S. Shen, S. Castan: "An optimal linear operator for step edge detection", *Comput. Vision Graphics Image Process.*, Vol. 54, pp. 112-133, 1992.

A MULTI-STEP FAST MEDIAN ALGORITHM FOR 2D DATA OF VARIOUS WORD-LENGTH

L. ALPARONE, V. CAPPELLINI, and A. GARZELLI

DIPARTIMENTO DI INGEGNERIA ELETTRONICA
UNIVERSITY OF FLORENCE
VIA S. MARTA, 3, 50139 FLORENCE, ITALY

1. INTRODUCTION

Two-dimensional median filter is widely recognized as a powerful tool for impulsive noise removal in digital imagery. Its effectiveness depends on that abnormal values due to superimposed noise pulses do not affect the response, as for linear filters, but are simply discarded as extreme local values [1]. In image processing applications of median filtering, a local window is moved from one column to the next along the rows of the image and the median of the pixels contained within the window at each position is computed. Finding the elements of a new window requires the replacement of a number of pixels in the previously displaced window equal to the number of rows in the window. The median obtained through this process is called the *running median* [2]. Since the direct approach of sorting a given set of numbers and finding the median is an extremely time-consuming operation, faster algorithms have been designed in the practical case of image data with a finite number of levels L [3]-[6].

Perhaps the most well-known fast algorithm for two-dimensional *nonseparable* median filter has been proposed by Huang et al. [5], and is based on the concept of local histogram of gray-levels, instead of sorting criteria, to determine the median. The local half-percentile is thoroughly computed only for the first pixel of each row and then upgraded from the update of the local histogram obtained by considering pixels entering and leaving the local sliding window. This scheme heavily reduces the computational burden with respect to implementations based on sorting

concepts, in the hypothesis that only a finite set of values are allowed. In fact the number of operations needed to process a pixel is $O(n)$, for an $n \times m$ window, whereas in conventional sorting implementations this number ranges between $O(n^2 \cdot m^2)$ and $O[n \cdot m(\log n + \log m)]$, and in sorting schemes exploiting 1D recursion $O(n \log n)$ [7], [8].

Other fast algorithms for 1D and 2D median include the scheme developed by Ataman et al. [3], that determines the k th bit of the median by inspecting the k Most Significant Bits (MSB) of the samples. Another algorithm to compute the median, or any other rank-order value, was presented by Danielsson [4]. This algorithm computes the median by examining the bits of the argument columnwise starting with the MSBs. The algorithms discussed in [3] and [4] are based on the binary representation of the data for finding out the median and therefore have a cost logarithmic with the number of levels. They are efficient for *real-time* realizations on array processors or other parallel hardware. However, the implementation of these algorithms on general-purpose computers will not be suitable, since most of the high-level languages, such as *Fortran*, *Pascal* and *C*, do not provide bit manipulation facilities.

A further algorithm for running median computing was developed by Rao and Rao [6], who consider the complement of the local Cumulative Distribution Function (CDF) instead of the local histogram. The scheme is more efficient in the 1D case; while computing the median becomes as bitonic operation and therefore

has a logarithmic cost, the update of the complementary CDF is extremely onerous, especially for 2D windows, being proportional to the number of levels. Finally, a different fast algorithm has been developed by Ahmad and Sundarajan [9], whose execution time is independent of the data precision and is relatively insensitive to the noise level, or to the spatial correlation, in the image. Instead of using a histogram and updating it, the elements inside a window are put into two subsets that are updated for each position of the sliding window. This algorithm has the noteworthy property of exploiting the recursion along both rows and columns, thus reducing the update time, but implementation is critical and requires efficient structures in order to take advantage over histogram-based algorithms, especially for large local windows.

Another drawback of some fast algorithms specifically designed for integer data, like those reported in [5] and [6], is that their computational cost linearly depends on the number of levels L , that is on the length of the local histogram, as well as on the linear size of the local window. While the latter is limited to relatively small values, at least in practical situations, the number of levels of the input image data can be also considerably elevated in applications where high precision values are required, i.e. whenever local fluctuations of small entity compared to the full scale value are regarded as significant, as in medical imaging, or in remote sensing for geology, hydrology, agronomy applications. The invariance, or at least a slow variance of the execution time for a given window size as the data wordlength is increased, is a desirable feature especially in view of the fact that modern high-precision scanners have led to processing images of 10, 12, and 14 bit precision.

In this paper a modified version of the algorithm introduced in [5] is proposed as specifically designed for integer image data with a very large, or even huge number of levels. The algorithm is based on factorizing the number of levels and introducing the concept of *reduced* local histogram relative, in case of binary representation of the data, to some MSBs of the image samples. A *multi-step* algorithm provides a *coarse-to-fine* estimate of the actual median value. The first, or coarse, step works on the most reduced histogram according to the modality stated in [5], which will be reviewed in Sect. 2, when explaining the outline of the novel algorithm. The gross estimate of the median is refined in further phases, involving only limited portions of histograms, relative to LSBs of the samples. Comparisons in terms of theoretical average number of operations, reported in Sect. 3, between the *single* and a *two-step* fast running medians, explain the features of the modified algorithm and its advantages over histogram-based schemes, when dealing with high precision integer data. Running time comparisons, reported in Sect. 4 between the histogram algorithms and *Quicksort*, show that significant advantages are achieved with respect to both Huang's and rank-sorting median, in many practical image processing application cases.

2. OUTLINE OF THE ALGORITHM

Before describing the coarse-to-fine fast median scheme, the algorithm proposed by Huang et al. [5] is summarized in the following steps concerning the retrieval of the running median from the local histogram, and relative to processing of an individual row.

1. Set up the gray-level histogram of the first window of size $n \times m$ and find the median, by making the count $ltmdn$ of the number of pixels with gray-level less than the median ($ltmdn \leq [(n \times m)/2]$).
2. Move to the next window by deleting the leftmost column of the previous window and adding one column to the right. Update the histogram and the count $ltmdn$ from the n leaving and n entering pixels. Now $ltmdn$ stores the number of pixels in the current window having gray-levels less than the median of the previous window.
3. Starting from the previous median, move *up/down* the histogram bins one at a time if the count $ltmdn$ is *not greater/greater* than $[(n \times m)/2]$ and update the count $ltmdn$ until the median bin is reached.
4. Stop if the end of the line is reached; otherwise go to Step 2.

Notice that if in Step 2 the count $ltmdn$ is unchanged with respect to its previous value, then the median is the same as

before; therefore, Step 3 is skipped.

Now, let us introduce a *reduced histogram* H_R of size $L_1 < L$ from the histogram H of size L :

$$H_R(k) = \sum_{l=kL_2}^{(k+1)L_2-1} H(l) \quad k = 0, 1, \dots, L_1 - 1, \quad (1)$$

where $L_2 = L/L_1$. If $B_1 = \log_2(L_1)$, then H_R is the local histogram of W_R , the B_1 -MSB representation of W , the data set inside the local window. The correspondence between the two histograms is graphically evidenced in Fig. 1.

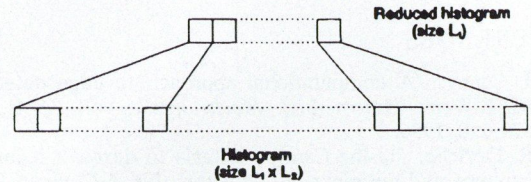


Fig. 1. Whole and reduced gray-level histograms for two-step fast running median computing

The *two-step* algorithm consists of finding $rmdn$, the median of W_R , whose bits are the B_1 MSBs of the median of W $rmdn$ [3], while computing a raw estimate of $ltmdn$ and then of finding out the LSBs of $rmdn$, by refining $ltmdn$, adding the bins of H from the $(rmdn \cdot L_2)$ th to the $[(rmdn + 1) \cdot L_2 - 2]$ th at most. The outline of flow may be condensed in the following five steps, again referring to filtering an individual row.

1. Set up the histogram H and the reduced histogram H_R (1) of the first window of size $n \times m$ and find the median, by making the count $ltmdn$ of the number of pixels with gray-level less than the median.
2. Move the next window by deleting the leftmost column of the previous window and adding one column to the right. Update H , the count $ltmdn$, and H_R (scale pixel values by L_2 to discard their LSBs).
3. Scale the median of the previous window by L_2 , move *up/down* the bins of H_R one at a time if the count $ltmdn$ is *not greater/greater* than $[(n \times m)/2]$ and update $ltmdn$ until the median bin of H_R , namely $rmdn$, is reached.
4. Starting from $rmdn \cdot L_2$ move *up/down* the bins of H one at a time if the count $ltmdn$ is *greater/greater* than $[(n \times m)/2]$ and upgrade $ltmdn$ until the median bin of H is encountered, or until the $[(rmdn + 1) \cdot L_2 - 2]$ th bin is reached. In which case the next bin will necessarily be the median.
5. Stop if the end of the line is encountered; otherwise go to Step 2.

Again, if in Step 2 the count $ltmdn$ is unchanged with respect to its previous value, then we have the same median as before and, therefore, Steps 3 and 4 are skipped. Otherwise, even if $rmdn$ is unchanged from the previous window, at least one iteration of Step 3 will be performed. A *three-step* algorithm may be easily derived; however its application may be recommended only for extremely large level numbers (more than 14-bit data), as it will be experimentally shown in Sec. 4. Notice that even if two (or more) *histogram-updating* as well as *median-upgrading* steps are necessary, there is only one count $ltmdn$.

3. THEORETICAL PERFORMANCE ANALYSIS

The two-step histogram algorithm described in Sec. 2 will now be compared with the classical histogram algorithm in terms of orders of computational cost, and *maximum* and *average* numbers of operations per pixel.

The order of computation time for the running Huang's algorithm depends on two main contributors: update of the local histogram and of the number of pixels lower than the median $ltmdn$; upgrade of the local median from the new histogram. The former depends on the size n of the side of the sliding window perpendicular to the recursion direction, i.e. the number of rows of the window for raster scan of the image data domain, and is $O(n)$. The latter is proportional to the length of the local histogram, that is to the number of gray-levels L , as $O(L)$.

Therefore the overall cost is taken as $O(L)$, in the practical case $n \ll L$. For the coarse-to-fine two-step algorithm, the term relative to median searching is split into $O(L_1 + L_2) = O[\max(L_1, L_2)]$, where L_1 represents the size of the reduced histogram H_R and L_2 denotes the number of bins of the local histogram corresponding to an individual location of H_R , as in Fig. 1. Therefore the global cost is $O[\max(n, L_1, L_2)]$ but, as it will be shown in the following of this section, practically reduces to $O(L_2)$ when image correlation is taken into account.

The formulations expressed in the above paragraph theoretically hold for uniformly distributed and uncorrelated 2D data. More significant for histogram-based algorithms, when dealing with natural images, are the *maximum*, or *worst-case*, and *average* numbers of comparisons. For Huang's scheme, even if the former is $N_{MAX} = 2n + l - 1$, the latter is given by

$$N_A = 2n + \overline{|d|} + 1.5 + 0.5p_0 \quad (2)$$

ere, if d is the difference of two adjacent $n \times m$ medians along the scan line, $\overline{|d|}$ is the average of absolute value of d , and p_0 is the probability of d being zero in one picture. For a given image with L gray-levels the term $\overline{|d|}$ should decrease as the window size increases, since medians of large windows, overlapped except for one column, are likely to be close in value. For 8-bit correlated images and windows usually employed in image processing, the authors have found that the term $\overline{|d|} + 1.5 + 0.5p_0$ is about 10 [5]; therefore $N_A \ll N_{MAX}$.

For the novel scheme the worst case is $N_{MAX} = 2n + L_1 + L_2 - 2$. Even if the number of comparisons (to upgrade l_{tmdn} , which is unique) is not changed, *two* histograms are to be updated for each new position of the sliding window, and this affects computation times, as it will be addressed in Sect. 4. The operation of median retrieval is split into two cascaded steps of maximum cost $L_1 - 1$ and $L_2 - 1$, respectively. Instead, the average number of operations N_A , which is related to the actual computation time, not per pixel but of the entire image, results to be

$$N_A = 2n + \frac{\overline{|d|}}{L_2} + \frac{L_2 - 1}{2} + 1.5 + \frac{p_0}{2} \quad (3)$$

with p_0 as in (2). The assumption is made that the $B_2 = \log_2(L_2)$ LSBs are uniformly randomly distributed among adjacent medians; therefore $\overline{|d|}$ reduces by a factor L_2 , while the average search cost of the median refinement step is $(L_2 - 1)/2$. Such hypothesis is likely for the typically adopted window sizes, provided that the number of levels is conveniently great. For natural images the best factorization of L , that is the choice of either of L_1 or L_2 minimizing N_A will be investigated in Sect. 4, when explaining the experimental results. As a tendency, L_1 should be as greater than L_2 , as smaller $\overline{|d|}$ is.

The major advantage of the two-step algorithm lies in that, differently of logarithmic schemes like [3] and [4], it fully exploits the spatial correlation between two adjacent medians, as for Huang's scheme. In this way the cost of median retrieval from the histogram is intermediate between a linear [5] and a logarithmic scheme, but closer to the latter, at least for reasonable values of L (10-12 bit wordlength), whereas the enormous computational overhead of histogram updating, typical of logarithmic schemes, is strongly reduced.

The analysis reported in this section ignores the comparisons made in finding the median of the first window in each line. The additional number of comparisons can be significant compared to (3) for large number of levels and small image size in the horizontal direction. This effect can be eliminated by applying the algorithm vertically at the end of each line and scanning alternate lines in reverse direction, as suggested in [5].

4. EXPERIMENTAL RESULTS AND COMPARISONS

Computer tests have been performed to outline the features of the coarse-to-fine histogram-based running median algorithm and to assess the analysis presented in Sect. 3 when dealing with true images, as well as to compare computation times with those of the fast algorithm proposed by Huang et al. This is credited to be the most efficient for *off-line* software implementations on general

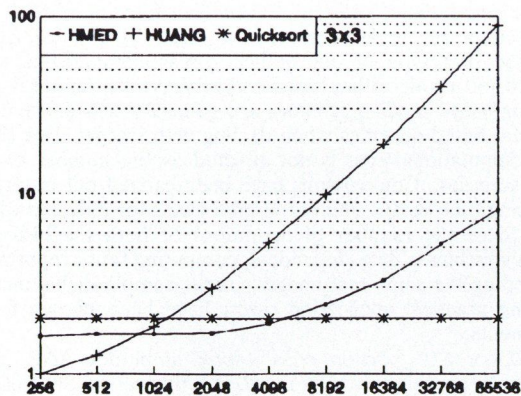
purpose computers of 1D and 2D median filtering of 8-bit data, as well as of any *rank-order* statistics filter [7], [8], [1], besides being the prototype of our outline. Also comparisons with a standard sorting algorithm based on *Quicksort* are reported. Even if sorting-based median schemes are generally less efficient than histogram-based schemes when dealing with integer data [7], [8] their computational cost is not affected as the number of gray-levels increases. Comparisons have been carried out in terms of processing time versus number of gray-levels for different window sizes. Since the number of operations of both the histogram-based algorithms is data-dependent, computing times reported are averaged on the same test images. Three sample digital pictures, reflecting practical application cases, have been chosen for the experiments.

- a A 512×512 portion of a 4-look amplitude SAR image, originally provided with 16 bit/pel, at a theoretical SNR of 6 dB.
- b A 512×512 section of the head, originally provided with 12 bit/pel, not all significant, resulting in a somewhat noisy appearance.
- c A 512×512 portion of a 150 dpi-scanned RX plate of the chest, produced at 12 bit/pel, with superimposed uniformly distributed impulsive random noise with occurrence probability 0.1.

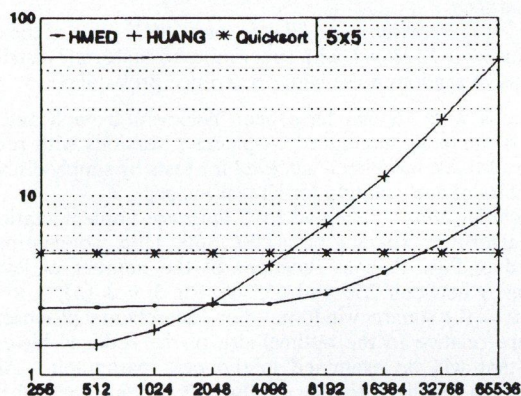
Versions with 16 bits have been achieved from b and c by adding 4 randomly generated bit-planes. Versions with reduced number of levels have been achieved for tests by simply discarding some LSBs of each sample for all test images.

Computing times, relative to C language implementations of all the algorithms running on a DEC 5000 Unix workstation, are reported in Figs. 2a-c as functions of the number of levels L (in octaves between 256 and 65536), for 3×3 (a), 5×5 (b) and 7×7 (c) square windows. For the scheme proposed the times are relative to the optimal size of the reduced histogram, aspect that will be examined in the next paragraph. All the values are normalized to that of Huang's scheme for 256 levels, for an easier comprehension, and expressed in logarithmic scale. In order to process exactly the same number of pixels for any window size, the image data set has been extended by *even-mirroring* the samples neighboring to the four edges. Since all the values reported are normalized to the same unique value, all the plots are comparable to each other. The *Quicksort* median scheme is represented by horizontal lines, since its run-time is independent of the number of gray-levels, depending only on the window size. Due to the reduced bit-plane correlation for decreasing significance from MSB to LSB, the cost of Huang's algorithm has a trend proportional to the number of levels, as theoretically stated in Sect. 3, while for the novel scheme the linear trend starts after the 14-th bit, but could be easily dropped by adding a *third* intermediate step (i.e. using a *triple* histogram). Increasing the window size decreases the advantages of the novel scheme, referred to as HMED (i.e. Huge MEDian), over Huang's scheme, as the time for updating *two* histograms, which significantly influences the overall cost, proportionally increases. This contribute is independent on the number of gray-levels and roughly expressed by the difference of the values relative respectively to HMED and HUANG, in the leftmost part of the curves in Figs. 2a-c, in which the median-upgrading procedure has approximately the same cost for both the histogram schemes.

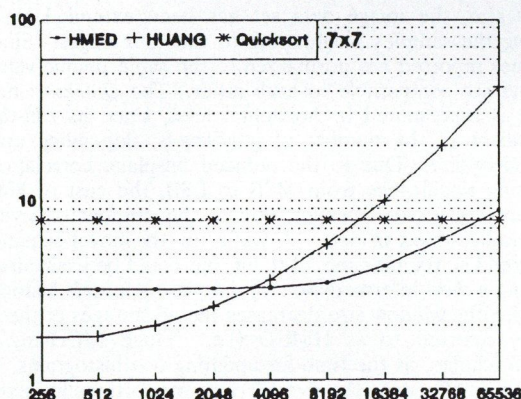
As more than one factorization is feasible for the coarse-to-fine algorithm, also the best choices have been investigated, varying with the window size and the number of levels. Computing time values in seconds are reported in Tables 1 and 2 for 4096 and 16384 levels, respectively, relative to the SAR test image only. It is noteworthy that even if the optimal size of the reduced histogram is critical for the performance, for the same image its value is practically independent of the number of levels and slowly varies with the window size. This behavior might be related to the decaying spatial correlation of the bit-planes, so that the mean correlation length of the *reduced* medians should be the same as for their full representation. Therefore, as a trend a more correlated image would require a greater L_1 , as it has been verified for the RX image.



(a)



(b)



(c)

Fig. 2. Computing times (in log scale) versus number of levels (in octaves) for windows 3×3 (a), 5×5 (b), and 7×7 (c) normalized to the value of Huang's algorithm for 256 levels (1.95s).

We also have investigated the dependence of the optimal choice of the 2D median scheme among Huang's HMED and Quicksort, as a function of the number of gray-levels L and the side n of the square sliding window. By inspecting the plots of Figs. 2 to 4, one can note that in all of them there exists an interval of the L axis in which HMED is more efficient than the other. In Table 3 the minimum run-time scheme, among HMED, HUANG and QUICK, is reported as a function of L and n , for which the most efficient algorithm is derived from the curves of Figs. 2a-c and from other plots relative to a 9×9 window. As it appears, Huang's scheme is better for lower L , as n decreases; Quicksort-based scheme is recommendable for larger L , as n increases; instead, the superiority of the double-histogram scheme holds when both L and n are large, yet not exceeding values corresponding to practical application cases.

Table 1. CPU times (s) of a 12 bit image; number of bits of reduced histogram along row; height of square window along column; minimum values are underlined

	6	7	8	9	10	11
3	10.7	6.1	<u>4.8</u>	5.3	6.9	10.3
5	15.8	9.7	6.6	<u>6.4</u>	7.7	11.7
7	19.7	12.7	8.5	<u>8.2</u>	9.3	13.6

Table 2. CPU times (s) of a 14 bit image; number of bits of reduced histogram along row; height of square window along column; minimum values are underlined

	7	8	9	10	11	12
3	13.0	9.2	<u>8.4</u>	9.1	12.0	17.9
5	16.3	11.1	<u>9.6</u>	9.7	11.3	15.3
7	19.8	13.1	<u>10.9</u>	11.1	12.2	15.4

Table 3. Minimum run-time median scheme, among Huang's (H), two-step (D) and Quicksort (Q), as a function of wordlength along row and window size along column

	8	9	10	11	12	13	14	15	16
3	H	H	<u>D</u>	<u>D</u>	<u>D</u>	Q	Q	Q	Q
5	H	H	H	<u>D</u>	<u>D</u>	<u>D</u>	<u>D</u>	Q	Q
7	H	H	H	H	<u>D</u>	<u>D</u>	<u>D</u>	<u>D</u>	Q
9	H	H	H	H	H	<u>D</u>	<u>D</u>	<u>D</u>	<u>D</u>

5. CONCLUDING REMARKS

A multi-step version of Huang's fast algorithm, exploiting also reduced local gray-level histograms, is proposed for efficient calculation of running median in digital monochrome images whose number of bits is larger than 8. Comparisons in terms of theoretical number of operations and computing times show that a two-step version of the new algorithm is especially efficient, compared to the classic Huang's algorithm, when the number of levels of the input image is greater or equal to 1024. As the number of gray-levels further increases, a standard sorting algorithm (e.g. Quicksort) may be more efficient, at least for relatively small windows; therefore, a three-step algorithm may be recommendable. However, it is experimentally shown that for a wide range of gray-level numbers and window sizes the proposed algorithm outperforms both the other examined schemes. The optimal size of the reduced histogram is steady with the number of levels and the window size, and essentially depends on the spatial correlation of the data field. Also, the simplicity of the outline is far greater than that of schemes based on sorting criteria and implemented through tree or heap structures.

REFERENCES

- [1] I. Pitas, A. N. Venetsanopoulos: "Order Statistics in Digital Image Processing", *Proceedings of the IEEE*, 80, (12): 1893-1921, 1992.
- [2] J. T. Astola, G. Campbell: "On Computation of the Running Median", *IEEE Trans. on Acoustic Speech and Signal Processing*, 37, (4): 572-574, 1980.
- [3] E. Ataman, V. K. Aatre, K. M. Wong: "A fast method for real-time median filtering", *IEEE Trans. on Acoustic Speech and Signal Processing*, 28, (5): 415-421, 1980.
- [4] P. E. Danielsson: "Getting the median faster", *Computer Graphics and Image Processing*, 17, (1): 71-78, 1981.
- [5] T. S. Huang, G. J. Yang, G. Y. Tang: "A Fast Two dimensional Median Filtering Algorithm", *IEEE Trans. on Acoustic Speech*

and *Signal Processing*, 27, (1): 13-18, 1979.

- [6] V. V. B. Rao, K. S. Rao: "A new algorithm for real-time median filtering", *IEEE Trans. on Acoustic Speech and Signal Processing*, 34, (6): 1674-1675, 1986.
- [7] M. Juhola, J. Katajainen, T. Raita: "Comparison of Algorithms for Standard Median Filtering", *IEEE Trans. on Acoustic Speech and Signal Processing*, 39, (1): 204-208, 1991.
- [8] F. Pasián: "Sorting algorithms for filters based on ordered statistics: performance considerations", *Signal Processing*, 14, (3): 287-293, 1988.
- [9] M. O. Ahmad, D. Sundarajan: "A Fast Algorithm for Two-Dimensional Median Filtering", *IEEE Trans. on Circuits and Systems*, 34, (11): 1364-1374, 1987.

FILTERING VIA THE DISCRETE FOURIER COSINE TRANSFORM

T. TRUMP*

ROYAL INSTITUTE OF TECHNOLOGY (KTH)
S-100 44 STOCKHOLM, SWEDEN

1. INTRODUCTION

Filtering with linear finite impulse (FIR) filters is one of the classical subjects of digital signal processing. It is well known that if both the signal to be filtered and the impulse response of the filter are long, considerable computational saving can be achieved by performing filtering in the spectral domain [5]. Straightforward application of spectral domain filtering will, however, result in a so-called cyclic convolution, since the technique implies that the sequences considered are periodic with a period equal to DFT length. To obtain linear convolution, one has to pad both the input sequence and the filter impulse response with zeros. The computational advantage of this approach lies in the existence of efficient algorithms for computing the Discrete Fourier Transform (DFT). For efficient DFT algorithm, see e.g. [1], [6].

In [3], the use of Discrete Cosine Transform which is not the real part of DFT is studied for filtering purposes. The possibility of deriving rather simple relationships, provided that the filter frequency response is real and even, is demonstrated. In a recent related work [4] a usage of a family of discrete cosine transform for filtering purposes is studied and symmetric convolution operation is introduced.

In this paper we investigate a way to perform filtering using the Discrete Fourier Cosine Transform (DFCT) which is just the real part of DFT.

Define DFCT from a real sequence $x(n)$ with length $\frac{N}{2} + 1$, $N = 2^l$, where l is a positive integer as [2]

$$C(k) = \sum_{n=0}^{\frac{N}{2}} x(n) \cos \frac{2\pi nk}{N}, \quad k = 0, 1, \dots, \frac{N}{2}. \quad (1)$$

DFCT of the sequence $x(n)$ is equal to the DFT of a sequence obtained from $x(n)$ by properly scaling and extending it to the interval $[0, N]$ as a periodic function with even symmetry. To see this, define

$$x_p(n) = \begin{cases} x(n) & \text{if } 0 \leq n \leq \frac{N}{2} \\ x(N-n) & \text{if } \frac{N}{2} < n < N. \end{cases} \quad (2)$$

* On leave from Dept. of Radio and Communication Engineering, Tallinn technical University, Estonia.

Compute the DFT from $x_p(n)$

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x_p(n) \exp \left(-j \frac{2\pi nk}{N} \right) \\ &= x_p(0) + \sum_{n=1}^{\frac{N}{2}-1} x_p(n) \exp \left(-j \frac{2\pi nk}{N} \right) \\ &\quad + x_p \frac{N}{2} \cos(k\pi) + \sum_{n=\frac{N}{2}+1}^{N-1} x_p(n) \exp \left(-j \frac{2\pi nk}{N} \right) \\ &= x_p(0) + \sum_{n=1}^{\frac{N}{2}-1} x_p(n) \exp \left(-j \frac{2\pi nk}{N} \right) \\ &\quad + x_p \frac{N}{2} \cos(k\pi) + \sum_{n=1}^{\frac{N}{2}-1} x_p(N-n) \exp \left(-j \frac{2\pi (N-n)k}{N} \right) \\ &= x(0) + 2 \sum_{n=1}^{\frac{N}{2}-1} x(n) \cos \frac{2\pi nk}{N} + x \left(\frac{N}{2} \right) \cos(k\pi) \\ &= \sum_{n=0}^{\frac{N}{2}} x'(n) \cos \frac{2\pi nk}{N}, \end{aligned}$$

where

$$x'(n) = \begin{cases} x(n) & \text{if } n = 0 \text{ or } n = \frac{N}{2} \\ 2x(n) & \text{otherwise.} \end{cases}$$

The last expression is closely related to the definition of DFCT (1). Consequently, it is sufficient to investigate the problem of finding the linear convolution of sequences $x(n)$ and $h(n)$ using the cyclical convolution of extended sequences $x_p(n)$ and $h_p(n)$ having even symmetry.

2. GENERAL CASE

Our objective is to compute the linear convolution of the sequence $x(n)$, $0 \leq n < N_1$ with $h(n)$, $0 \leq n < N_2$. Extend the

sequences into an interval with length $N = 2^l$ as even functions

$$x_p(n) = \begin{cases} 0 & \text{if } n < M \\ x(n-M) & \text{if } M \leq n \leq M+N_1-1 \\ 0 & \text{if } M+N_1 \leq n \leq N-M-N_1 \\ x(N-M-n) & \text{if } N-M-N_1+1 \leq n \leq N-M \\ 0 & \text{if } N-M+1 \leq n \leq N-1 \end{cases} \quad (3)$$

and

$$h_p(n) = \begin{cases} 0 & \text{if } n < L \\ h(n-L) & \text{if } L \leq n \leq L+N_2-1 \\ 0 & \text{if } L+N_2 \leq n \leq N-L-N_2 \\ h(N-M-n) & \text{if } N-L-N_2+1 \leq n \leq N-L \\ 0 & \text{if } N-L+1 \leq n \leq N-1 \end{cases} \quad (4)$$

for some N, M and L . The proper values for the length N, M and L will be determined later.

The cyclical convolution y_p of x_p with h_p can be expressed as

$$y_p(k) = \sum_{n=0}^{N-1} h_p(n)x_p(k-n) \quad (5)$$

$$\begin{aligned} &= \sum_{n=0}^{N-1} [h(n-L) + h(N-L-n)] \\ &\quad [x(k-n+M) + x(k-N+M+n)] \\ &= \sum_{n=0}^{N-1} h(n-L)x(k-n+M) \\ &\quad + \sum_{n=0}^{N-1} h(n-L)x(k-N+M+n) \\ &\quad + \sum_{n=0}^{N-1} h(N-L-n)x(k-n+M) \\ &\quad + \sum_{n=0}^{N-1} h(N-L-n)x(k-N+M+n) \end{aligned} \quad (6)$$

As we can see from this expression, the cyclical convolution y_p consists of four additive terms, where overlapping can be controlled by varying the parameters L, M and N . The first term in (5) represents the linear convolution of $x(n)$ with $h(n)$, spanned in the interval $L+M \leq k \leq N_1+N_2+L+M-2$. The second term is the linear cross-correlation between $h(n)$ and $x(n)$, spanned in the interval $L+1-N_1-M \leq k \leq N_2-M+L-1$. The third term is the linear cross-correlation between $h(-n)$ and $x(-n)$, spanned in the interval $M-N_2-L+1 \leq k \leq N_1+M-L-1$. The last term is the linear convolution of $h(-n)$ with $x(-n)$ and it is spanned in the interval $N-M-N_1-L-N_2+2 \leq k \leq N-M-L$.

By varying the parameters N, L and M , we can cyclically move the location of terms in (5) so that the term of interest is not overlapped with the others. To illustrate this let us see how to extract the first term, i.e. the linear convolution of $x(n)$ with $h(n)$. To do this, demand that the first term is not overlapped with the second and the third ones and that it is overlapped with the fourth term (which represents the mirror image of the same convolution) only in one end-point. Note that the overlapping end-value should be later scaled by 0.5 in order to obtain the correct result. These requirements can be formally written as follows:

$$\begin{cases} L+M > N_2-M+L-1 \\ L+M > N_1+M-L-1 \\ N_1+N_2+L+M-2 = N-M-N_1-L-N_2+2. \end{cases} \quad (7)$$

** Note that all the indexes throughout the paper are computed modulo N due to the cyclic nature of formula (5).

The inequalities require that the first term in (5) must locate right from the second and the third ones. The equality says that the end-point of the first term is overlapped with the first sample of the fourth term. The system above is satisfied, for instance, if

$$\begin{cases} L+M = N_2-M+L \\ L+M = N_1+M-L \\ N_1+N_2+L+M-2 = N-M-N_1-L-N_2+2. \end{cases} \quad (8)$$

For this particular choice, the cross-correlation sequences overlap totally with each other and their sum is located tightly left from the convolution term of interest. Solving with respect to L and M , we obtain

$$\begin{cases} L = \frac{N_1}{2} \\ M = \frac{N_2}{2}. \end{cases} \quad (9)$$

Substitution to the third equation gives that the smallest possible length of extended sequences x_p and h_p is

$$N = 3(N_1+N_2)-4. \quad (10)$$

The location of input and output terms for the case $N = 16$ is shown in Fig. 1. With respect to the conditions (9) and (10) $N_1 = N_2 = 3$ and $L = M = 2$. Observe that just the first 9 samples of the extended sequences not all the 16 are used in actual computations because of the definition of DFCT (1).

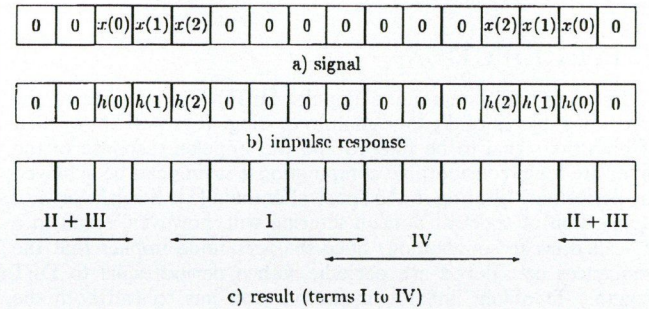


Fig. 1. Location of terms inside a) x_p , b) h_p and c) y_p for $N = 16$. General case.

3. FILTERING WITH TYPE 1 LINEAR PHASE FIR FILTERS

The impulse response of the type 1 linear phase FIR filter [5] has even symmetry and its length N_2 is an odd number. This symmetry fits the symmetry inherent in DFT, and consequently, we can expect existence of a more efficient technique compared to the general case.

Redefine the extended impulse response of filter h_p taking into account its symmetry as follows:

$$h_p(n) = \begin{cases} h(\frac{N_2-1}{2}+n) & \text{if } 0 \leq n \leq \frac{N_2-1}{2} \\ 0 & \text{if } \frac{N_2-1}{2} \leq n \leq N - \frac{N_2-1}{2} \\ h(\frac{N_2}{2}+n-N) & \text{if } N - \frac{N_2-1}{2} \leq n \leq N-1. \end{cases} \quad (11)$$

Observe that we have placed h so that its center of symmetry is located at $n = 0$. Find the cyclical convolution of h_p with x_p defined in (3)

$$\begin{aligned} y_p(k) &= \sum_{n=0}^{N-1} h_p(n)x_p(k-n) \\ &= \sum_{n=0}^{N-1} h_p(n)x(k-n+M) + \sum_{n=0}^{N-1} h_p(n)x(k-N+M+n). \end{aligned} \quad (12)$$

As opposed to the general case, we have just two components here since h_p is a cyclically shifted version of h . The first component is the linear convolution of $h(n)$ with $x(n)$ spanned in the interval $M - \frac{N_2-1}{2} \leq k \leq M + \frac{N_2-1}{2} + N_1 - 1$. The second term represents the linear cross-correlation between $x(n)$ and $h(n)$ and it is spanned in the interval $N-M-N_1+1 - \frac{N_2-1}{2} \leq$

$k \leq \frac{N_2-1}{2} - M$. We can observe that due to symmetry properties of h , the two terms are mirror images of each other. Hence, we can allow their overlapping in end-points

$$\begin{cases} M - \frac{N_2-1}{2} = \frac{N_2-1}{2} - M \\ M + \frac{N_2-1}{2} + N_1 - 1 = N - M - N_1 + 1 - \frac{N_2-1}{2} \end{cases} \quad (13)$$

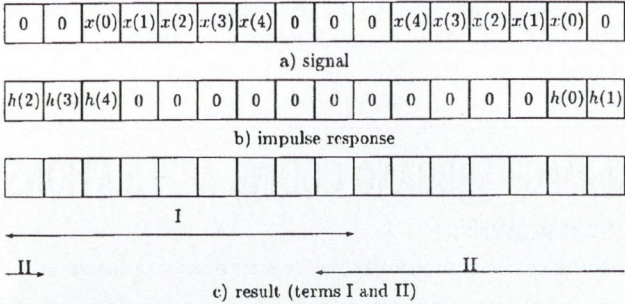


Fig. 2. Location of terms inside a) x_p , b) h_p and c) y_p for $N = 16$. FIR filter case.

Solving for M and N , we obtain

$$M = \frac{N_2 - 1}{2} \quad (14)$$

$$N = 2(N_1 + N_2) - 4. \quad (15)$$

The convolution $y(k)$ is spanned in the interval $0 \leq k \leq \frac{N}{2}$ which is precisely the interval where the computations using DFCT will be carried on.

Note that for type 3 linear phase FIR filters (odd N_2 , odd symmetry), it is possible to find an algorithm involving Discrete Fourier Sine Transform in addition to the DFCT. The symmetry of the type 2 and 4 linear phase FIR filters does not fit the symmetry of DFT and no simplifications compared to the general case are available.

The location of input and output terms for the case $N = 16$ is shown in Fig. 2. To satisfy the condition (15) $N_1 = N_2 = 5$ and $M = 2$.

4. COMPUTATION OF AUTOCORRELATION

In the case of autocorrelation computations $x(n) = h(n)$ and $N_1 = N_2$. Recall that the second and the third terms in (5) represent the cross-correlation between $x(n)$ and $h(n)$ and its mirror image. Since the autocorrelation function of a real sequence has an even symmetry, we can let these terms overlap totally

$$\begin{cases} -N_2 + M - L + 1 = -N_1 - M + L + 1 \\ N_1 + M - L - 1 = N_2 - M + L - 1 \end{cases}$$

which results in $L = M$. Next, we require that the terms corresponding to correlations must not overlap with these corresponding to convolutions. Let the end-point of the correlation terms locate left from the convolution terms

$$-M + L + N_2 - 1 < L + M,$$

we have

$$N_2 < 2M + 1. \quad (16)$$

We also require the starting point of the correlation terms to locate right from the convolution terms

$$N = M - N_1 + L + 1 > N_1 + N_2 + L + M - 2.$$

Since $N_1 = N_2$, as stated earlier, the latter yields to

$$N > 3N_2 + 2M - 3. \quad (17)$$

Finally, combining (16) with (17) we have

$$N > 4N_2 - 3. \quad (18)$$

The location of input and output terms for the case $N = 16$ is illustrated in Fig. 3.

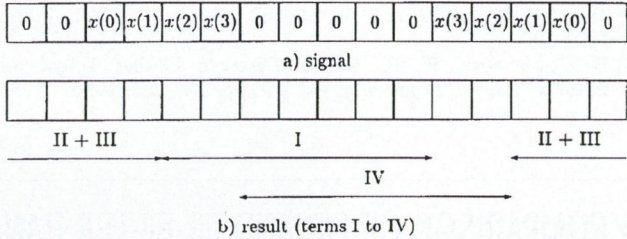


Fig. 3. Computation of autocorrelation. Location of terms inside a) x_p and b) y_p for $N = 16$.

5. COMPUTATIONAL COMPLEXITY

In this section, we present the number of additions/subtractions and multiplications required to implement the algorithms presented in the previous sections. The use of the reduced complexity algorithm to compute DFCT [2] is assumed. In Table 1, the numbers of real additions (ADD), multiplications (MULT) and total amount of computations per output sample (RSUM) are given versus the length of the computed convolution or correlation N_c . It is assumed that the DFCT of the impulse response can be pre-computed and the corresponding computational load is not taken into account. In Table 2, the computational complexity of filtering using DFT approach is presented for comparison. The use of the reduced complexity algorithm for DFT [1] is assumed.

Table 1. Operation counts for DFCT filtering

		General		FIR		Autocor.	
MULT	ADD	N_c	RSUM	N_c	RSUM	N_c	RSUM
19	54	5	14.6	9	8.1	7	10.4
51	144	10	19.5	17	11.5	15	13.0
131	370	21	23.9	33	15.2	31	16.2
323	916	42	29.5	65	19.1	63	19.7
771	2198	85	34.9	129	23.0	127	23.4
1795	5144	170	40.8	257	27.0	255	27.2
4099	11802	341	46.6	513	31.0	511	31.1

Table 2. Operation counts for filtering DFT

N_c	MULT	ADD	RSUM
8	18	46	8.0
16	50	134	11.5
32	130	360	15.3
64	322	902	19.1
128	770	2182	23.1
256	1794	5126	27.0
512	4098	11782	31.0

We can see that the computational load of the general DFCT approach is higher than that of the traditional DFT approach. However, the computational complexity in type 1 linear FIR filtering case is nearly equal to that of DFT approach. The same holds for the computation of an autocorrelation sequence.

REFERENCES

- [1] I. Arro: "A Generalized Algorithm for Reduced Computation of the Discrete Fourier Transform", *Izvestia Vysikh Uchebnykh Zavedenii. Radioelektronika*, Vol. 30, No. 12, pp. 5-10, 1987.
- [2] I. Arro: "Generalized Algorithm of short Discrete Cosine Transform", *Proc. ICA 1988*, Helsinki, June 1988, Vol. 2, pp. 1903-1906.
- [3] B. Chitprasert, K. R. Rao: "Discrete Cosine Transform Filtering", *Signal Processing*, Vol. 19, pp. 233-245, 1990.
- [4] A. Martucci: *Symmetric Convolution and the Discrete Cosine Transforms: Principles and Applications*, Ph.D. thesis, Georgia Institute of Technology, May 1993.
- [5] A. V. Oppenheim, R. W. Schaffer: *Discrete Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.
- [6] H. V. Sorensen, D. J. Jones, M. T. Heideman, C. S. Burus: "Real-Valued Fast Fourier Transform Algorithms", *IEEE Trans. on ASSP*, Vol. ASSP-35, pp. 849-863, June 1987.

A COMPARISON OF FIR AND IIR FILTER BANKS IN IMAGE SUBBAND CODING APPLICATIONS

M. DOMAŃSKI and R. SWIERCZYŃSKI

POLITECHNIKA POZNAŃSKA, INSTYTUT ELEKTRONIKI I TELEKOMUNIKACJI, UL. PIOTROWO 3A, 60-965 POZNAŃ, POLAND

1. INTRODUCTION

Subband coding of images has shown itself a powerful technique for digital image data compression. Its implementation needs some filter banks which sometimes are the most complicated parts of the system. Therefore a careful choice of the type of the filter bank as well as its optimum design are of great importance.

Most of the papers on subband coding of image and video report the use of the finite impulse response (FIR) filters. The classical linear-phase quadrature-mirror filters (QMFs) [1] are willingly applied because of lack of any stability and phase problems. Nevertheless the references described a lot of nonlinear filter banks which are more efficient from the point of view of their implementations. Unfortunately they are not very willingly used in practice of subband coding of image and video.

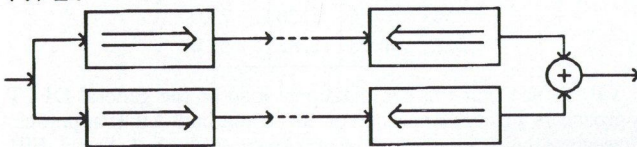
The aim of the paper is to show in the experimental way that subband coding of images does not need linear phase filters. The results of simulation experiments with real images prove that nonlinear-phase recursive filter banks can be used more efficiently than FIR filters as long as a full-frame buffer is available in the system.

2. NONLINEAR-PHASE FILTER BANKS

In order to avoid phase problems the reversible systems have been proposed. The basic idea of reversible SBC system is to use the opposite directions of data processing in the coder and in the decoder, thus compensating for shifts caused by the filters. For the sake of brevity we deal only with maximally decimated separable filter banks organized in a multi-level (tree-like) structures.

A two band analysis/synthesis system is the basic building block of such a system. Therefore we are going to consider only such systems.

TYPE I



TYPE II

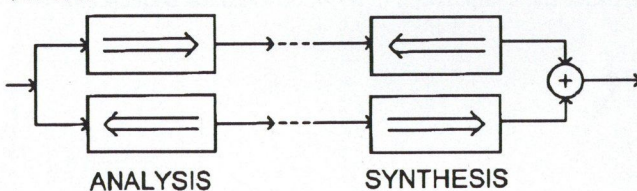


Fig. 1. Two types of reversible systems

For the IIR and FIR filters there exist two basic versions of the reversible structure shown in the figure below. The directions of processing are indicated schematically in Fig. 1.

For IIR filters which are inherently nonlinear-phase, the structure of Type I supports very efficient polyphase implementations. Nevertheless the reversion of signal processing direction needs all the lines and columns to be stored somewhere in the system, i.e., a full-frame buffer is necessary. The nonlinear-phase FIR filters have been considered in the references first of all in the arrangements of Type II where perfect reconstruction is achieved with lattice filters [9].

3. COMPLEXITY CONSIDERATIONS

Before presenting the experimental results of subband coding, we compare the complexities of basic filter bank solutions, similarly as it has been already done in the references (e.g., [9]).

We compare recursive biceiprocal IIR filter banks in perfect-reconstruction reversible systems [5], [6], [10], classic linear-phase quadrature mirror FIR direct-form filter banks [1] and the nonlinear-phase lattice perfect-reconstruction FIR filter banks [3], [4], [9]. Compared are the numbers of arithmetic operations as well as the numbers of delay elements for separable polyphase tree-structured multilevel implementations. As an example, consider a common 10-band system (see the Fig. 2). Similar systems are predominantly used in subband coding of image and video [11].

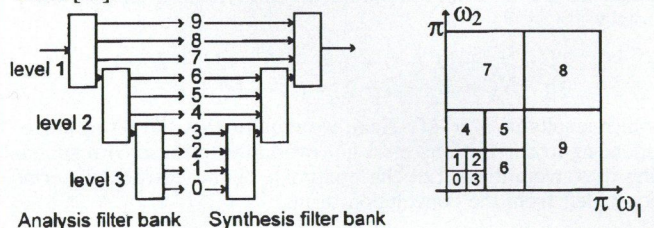


Fig. 2. Tree-structured filter banks and the respective spectrum partition

A two band analysis/synthesis system is the basic building block of such a system taken into further considerations (Table 1 and 2).

The corollary is that IIR filter banks lead to substantially reduced numbers of arithmetic operations as well as delay elements. The reversible systems with IIR filter banks are an interesting solution for the systems where a full-frame buffer is available. FIR filter banks (even nonlinear-phase) exhibit much higher numbers of multiplications, additions and delay elements. In particular the increased number of delay elements is related to high demand of silicon area.

Some of the filters operate in lines while the others operate in columns. Therefore some delays correspond to RAMs storing whole lines or columns of an image. Of course it needs a lot of silicon surface and minimization of the number of delays is very

important. It is unlike many one-dimensional applications.

Table 1. Comparison of the corresponding IIR and FIR filter banks

	IIR filter (direct)			FIR QMF [1]		
	5	5	7	15	23	31
Filter order	5	5	7	15	23	31
Multiplications per input sample	1	1	1.5	8	12	16
Additions per input sample	4	4	5.5	8	12	16
Delay elements	2	2	3	14	22	30
Full-frame buffer necessary	yes	yes	yes	no	no	no

Table 2. Comparison of analysis filter banks corresponding to the filters of the type 32D proposed by Johnston [1]

	FIR		IIR
	QMF [1]	lattice	
Perfect reconstruction	approx.	yes	yes
Linear phase of a filter	yes	no	no
Filter order	31	31	7
Multiplication per sample	16	17	1.5
Additions per sample	16	16	5.5
Delay elements	30	15	3

There are also well-known lattice linear-phase perfect reconstruction filters [7], [9]. We omit them because their complexity is usual higher as nonlinear-phase FIR lattices and we are going to show that linear-phase filters do not guarantee better performance of subband coding.

4. COMPARISON OF CODING EFFICIENCY

Some experiments with test images have been done by the authors. There were compared the coding efficiencies of two systems: the first used FIR linear-phase quadrature-mirror filters while the second was equipped with recursive elliptic filters.

In order to obtain fully comparable results, the same coding scheme has been used in both cases. It was based on partition of the whole two-dimensional spectrum of an image into 10 subbands. The subband 0 was encoded using simple nonadaptive DPCM technique with a first-order predictor. The prediction error signal was compressed using standard Huffman coding. The technique used for high-frequency subbands [2], [12] exploited intersubband correlation. We lay a particular stress on the fact that the coding technique used exploited relations between pixels from different subbands produced by the analysis filtering being a nonlinear-phase IIR filtering. One could expect that such a situation should lead to much better results for linear-phase filters. The experimental results showed that it was not true. The experiments proved that the coding efficiency characteristics (i.e., quality versus compression) were very similar for linear-phase FIR QMFs and IIR filters.

The experiments have been made for two well-known test images: "LENA" and "BOATS". For example we compared 5-order IIR filter and 24-tap FIR QMF with similar amplitude characteristics (Fig. 3).

The plots shown in Fig. 4 and 5 prove that IIR filters behave often even slightly better.

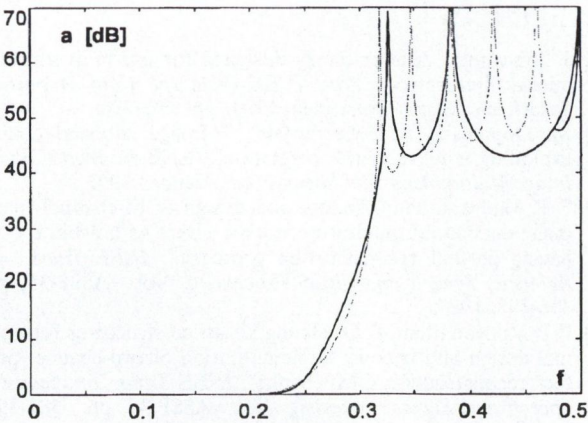


Fig. 3 Attenuation of the 5-th order elliptic filter (continuous line) and the corresponding C24 [1] FIR filter (dashed line)

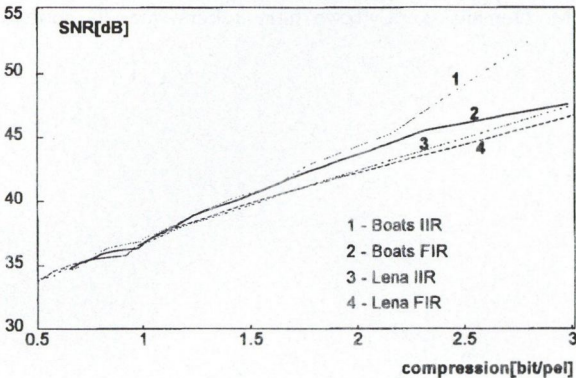


Fig. 4. Signal to noise ratio versus compression for the test images "LENA" and "BOATS"

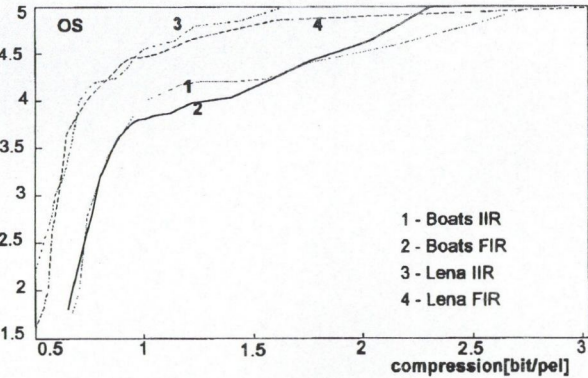


Fig. 5. Opinion score versus compression for the test images "LENA" and "BOATS"

5. CONCLUSIONS

The experimental results prove that linear-phase filter banks offer no advantage from the point of view of the coding efficiency. It is the main result of this paper. This result have been obtained from simulation experiments performed for the two types of filter banks in the same repeatable conditions. The result is valid for still image and intraframe video coding. The authors hope that this clear result will increase interest in application of IIR in subband coding of image and video. Nevertheless the papers does not consider the phenomena which could arise when nonlinear-phase space filters are combined with motion compensated predictive coding.

The work has been supported under the grant no. 8550400206 of Committee of Scientific Researches.

REFERENCES

- [1] J. Johnston: "A filter family designed for use in quadrature mirror filter banks", *Proc. IEEE 1980. Int. Conf. Acoustics, Speech, and Signal Processing*, 1980. pp. 291-294.
- [2] M. Domański, R. Swierczyński: "Image subband coding exploiting inter-subband correlation", *IEEE 8th Workshop on Image Multidimensional Signal Proc.*, Cannes 1993.
- [3] P. P. Vaidyanathan: "Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having perfect reconstruction property.", *IEEE Trans. on Acoustic, Speech and Signal Processing*, Vol. ASSP-35, pp. 476-492, 1987.
- [4] P. P. Vaidyanathan, P. Q. Hoang: "Lattice structures for optimal design and robust implementation of two-channel perfect reconstruction QMF banks", *IEEE Trans. on Acoustic, Speech and Signal Processing*, Vol. ASSP-36, pp. 381-391, March, 1988.
- [5] M. Domański: "Efficient wave filter banks for subband coding of images", *Archiv für Elektronik und Übertragungstechnik*, Vol. 45, pp. 160-167.
- [6] M. Domański: "Cyfrowe filtry rekursywne dla subzakresowego kodowania obrazów", *Krajowe Sympozjum Telekomunikacji*, Bydgoszcz 1988, in Polish.
- [7] T. Q. Nguyen, P. P. Vaidyanathan: "Two-channel perfect reconstruction FIR QMF structures which yield linear phase FIR analysis and synthesis filters", *IEEE Trans. on Acoustic, Speech and Signal Processing*, Vol. ASSP-37, pp. 676-690. May, 1989.
- [8] M. Vetterli, D. Le Gall: "Perfect reconstruction FIR filter banks: some properties and factorizations", *IEEE Trans. on Acoustics, Speech and Signal Proc.*, Vol. ASSP-37, pp. 1057-1071. 1989.
- [9] P. P. Vaidyanathan: *Multirate system and filters banks*, Prentice-Hall, Inc., 1993.
- [10] T. Ramstad: "IIR filterbank for subband coding of images", *Proc. IEEE 1988 Int. Symp. on Circuits and Systems*, Helsinki, 1988.
- [11] J. W. Woods, ed.: *Subband image coding*, Kluwer, 1991.
- [12] M. Domański, R. Swierczyński: Efficient simultaneous subband coding using recursive filter banks, COST Workshop Intelligent Terminals and Source and Channel Coding, Budapest 1993.

RESTAURATION, RECOGNITION

MOMENT-BASED IMAGE RECONSTRUCTION IN THE PRESENCE OF NOISE

R. C. PAPADEMETRIOU

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
UNIVERSITY OF PORTSMOUTH,
PORTSMOUTH, PO1 3DJ, ENGLAND

1. INTRODUCTION

The mathematical concept of moments has been around for many years and has been utilized in many diverse fields ranging from mechanics and statistics to pattern recognition and image understanding. Very early, it was recognized that a truncated set of moment values could offer a more convenient and economical representation of an image segment than a pixel-format representation.

Hu [1] first presented results on how to achieve fundamental image transformations (i.e., translation, rotation, scale change, etc.) with the image moment representation. Dudani et al [2] used moment invariants, defined by Hu, to represent the shape of aircraft images. Later, Teague [3] extended Hu's moment invariants and introduced the concept of orthogonal moment sets to reconstruct the image from moments. Because of the wide applicability of image moments as features there has been a lot of effort recently in designing fast computational algorithms [4], [5], [6] as well as VLSI implementations of moment-generating algorithms for real-time operation [7].

Going back to the basic problem of image compression/reconstruction, it is well known that an image can be fully reconstructed from the infinite number of its moments (since they are related to the coefficients of the power expansion of its characteristic function). Clearly this is of little use, unless it can be shown that a finite, small set of moments usually available, can reconstruct the original image to within an adequate degree of accuracy. The interesting question, then, is how much information does a finite set of moments retain and what method (or methods) can be used to make a best reconstruction of the original image. An answer was given by Papademetriou [8], [9], [10], where the proposed Maximum Entropy (ME) method was compared to the Legendre Moments (LM) method proposed by Teague [3]. The superiority of the first, however, was demonstrated through several simulation examples mainly under noise-free conditions. In this paper the comparative study is extended to include **noisy images** and **noisy moment vectors**.

In Section 2 of this paper the basics of moments are presented, while Sections 3 and 4 explain the philosophies of the two methods. Simulation experiments demonstrating the superiority of the ME method are given in Section 5.

2. BASIC MOMENT CONCEPTS AND NOTATION

The two-dimensional geometric moments (GM) of order $(p + q)$ of the image intensity function $f(x, y)$ are conventionally defined in terms of Riemann integrals as

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (p, q = 0, 1, 2, \dots) \quad (1)$$

If $f(x, y)$ is piecewise continuous with bounded support (i.e., has nonzero values only in the finite part of the $x - y$ plane), then moments of all orders exist. The double moment sequence $\{M_{pq}\}$ is uniquely determined by $f(x, y)$ and conversely $f(x, y)$ is uniquely determined by $\{M_{pq}\}$ [1].

When integrals are replaced by sums, Eq. (1) gives the moments of order $(p + q)$ for a digitized image segment $f(x, y)$,

i.e.

$$M_{pq} = \sum_x \sum_y x^p y^q f(x, y), \quad (2)$$

where $f(x, y)$ expresses the image gray-level.

A **complete moment set (CMS)** of order n consists of the all the moments of order n and lower and can be represented by a triangular matrix of

$$N_{total} = \frac{(n+1)(n+2)}{2} = \frac{1}{2}(n^2 + 3n + 2) \quad (3)$$

moment values, as shown below

$$\begin{matrix} M_{00} & M_{01} & M_{02} & \dots & M_{0n} \\ M_{10} & & & & \\ M_{20} & & \bullet & & \\ - & & & & \\ - & & & & \\ - & & \bullet & & \\ M_{n0} & & & & \end{matrix} \quad (4)$$

Thus, if an $L \times L$ image segment is compressed by retaining only a CMS of order n , the **compression ratio (CR)** is given by

$$CR \triangleq \frac{L^2}{N_{total}} = \frac{2L^2}{(n+1)(n+2)}. \quad (5)$$

3. THE LEGENDRE MOMENTS (LM) METHOD

The definition of the geometric moment, as given by Eq. (1), has the form of the projection of the intensity function $f(x, y)$ onto the monomial $x^p y^q$. However, the basis set $\{x^p y^q\}$, while complete (Weierstrass approximation theorem [11]), is not orthogonal. Orthogonal moment forms [3] may be defined by using Legendre polynomial basis functions $P_n(x)$ rather than conventional monomials.

The Legendre polynomials of degree n , defined [12] by

$$P_n(x) = \sum_{j=0}^n C_{nj} x^j = \sum_{k=0}^{[n/2]} (-1)^k \frac{(2n-2k)!}{2^n k!(n-k)!(n-2k)!} x^{n-2k}, \quad (6)$$

where

$$\left[\frac{n}{2} \right] = \begin{cases} \frac{n}{2}, & \text{for } n \text{ even} \\ \frac{n-1}{2}, & \text{for } n \text{ odd} \end{cases}$$

are a complete orthogonal basis set over the range -1 to $+1$, i.e.,

$$\int_{-1}^1 P_m(x) P_n(x) dx = \frac{2}{2m+1} \delta_{mn}, \quad (7)$$

where δ_{mn} is the Kronecker delta.

The Legendre moments (LM) of order $(p + q)$ are defined by

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P_p(x) P_q(y) f(x, y) dx dy \quad (p, q = 0, 1, 2, \dots) \quad (8)$$

For the moments to be orthogonal, the image must be scaled to be within a 2×2 square centred at the origin.

Legendre moments may be obtained directly from geometric moments by

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \sum_{j=0}^p \sum_{k=0}^q C_{pj} C_{qk} M_{jk}. \quad (9)$$

A given Legendre moment depends only on geometric moments of the same order and lower, and conversely.

By using Legendre rather than geometric moments, an approximate inverse transform — to obtain $f(x, y)$ from $\{L_{pq}\}$ — may be achieved by moment matching [3], i.e.,

$$f_N(x, y) \sim \sum_{j=0}^N \sum_{k=0}^j L_{j-k, k} P_{j-k}(x) P_k(y), \quad (10)$$

which is a truncated series, with N the maximum order of Legendre moments available. In this method, obviously, all unknown moments ($n > N$) are assumed to be zero, which is not very legitimate, indeed.

4. THE MAXIMUM ENTROPY (ME) RECONSTRUCTION

The entropy maximisation approach to the solution of underdetermined inverse problems (e.g., the classical moment problem) has roots in the works of Shannon [13] and Jaynes [14]. The **maximum entropy formalism**, exploiting the concept of the entropy of a random variable, casts the problem of determining a probability density function (*pdf*) into the form of an optimisation problem:

"Suppose a set of constraints on a probability distribution is known, but the constraints do not completely determine the distribution and nothing more about the distribution is known. Then the least prejudiced or biased assignment is the *pdf* of maximal (Shannon) entropy satisfying the given constraints".

This principle can be easily applied to the image reconstruction problem from a finite set of moments. Any irradiance distribution $f(x, y)$, being non-negative, can be considered as a *pdf*, because it can be, also, easily normalized to integrate to unity.

The given moments are the constraints in our optimisation problem, which can be stated mathematically as:

Maximize

$$H = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \log[f(x, y)] dx dy \quad (11)$$

subject to

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy = M_{pq} \quad (p + q = 1, 2, \dots, N) \quad (12)$$

and

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy = 1 = M_{00} \quad (13)$$

(scale normalization). Solution of this standard variational problem yields

$$f_N(x, y) = \exp \left[-\lambda_0 - \sum_{p, q} \lambda_{pq} x^p y^q \right], \quad (14)$$

where the λ 's are Lagrange multipliers determined from the constraint conditions. The notation $f_N(x, y)$ is used to point out that it is an estimate of the unknown intensity function $f(x, y)$ based on moments up to order N (i.e., on a CMS of order n).

The most important difference between the two methods is that, while the LM approach assumes all the unknown (or not given) moments to be zero, the ME method results in an estimate which is maximally non-committal with regard to missing information (i.e., unknown moments).

The ME reconstruction of moment-compressed images may produce more accurate estimates than the LM method using appreciably fewer moments. The need for more computational time in the ME case becomes continuously less restrictive,

because of the steadily increasing computing power available today and the development of fast algorithms [15], [16].

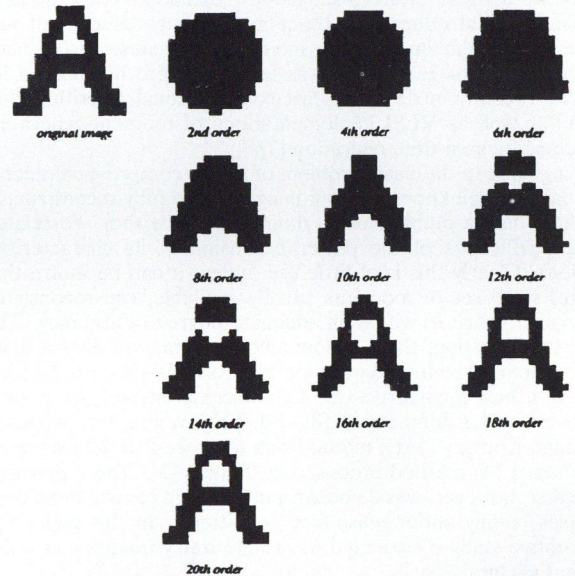
5. COMPUTER SIMULATIONS

The two-dimensional, binary-valued capitalized letter A, defined across a 21×21 pixel array, was chosen as the test image for all simulation experiments. Fig. 1 summarizes the results for image reconstruction under noise-free conditions. Specifically, Figs. 1(a) and 1(b) show examples of reconstruction of the test image from different CMSs using both methods. Although the presented reconstructions are thresholded versions of the continuous ones, all mean square error calculations were developed by comparing the original and the actual reconstructed image (not the threshold image).

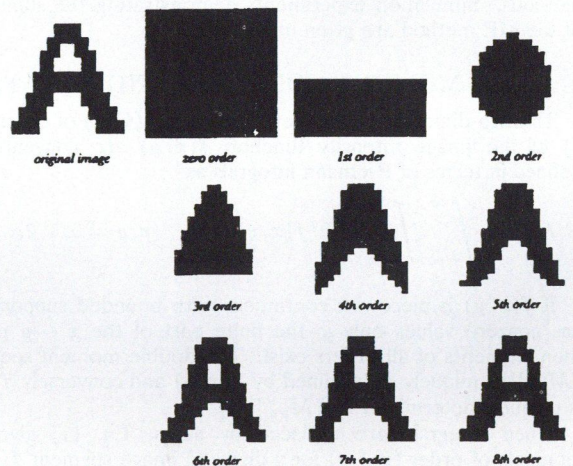
The normalized mean-square reconstruction error (NMSE) between an image $f(x, y)$ — defined over a region D of the xy -plane — and its reconstructed version $\hat{f}(x, y)$ from a finite set of its moments (up to order N), defined by

$$\bar{e}^2(N) = \frac{\int_D \int [f(x, y) - \hat{f}(x, y)]^2 dx dy}{\int_D \int [f(x, y)]^2 dx dy} \quad (15)$$

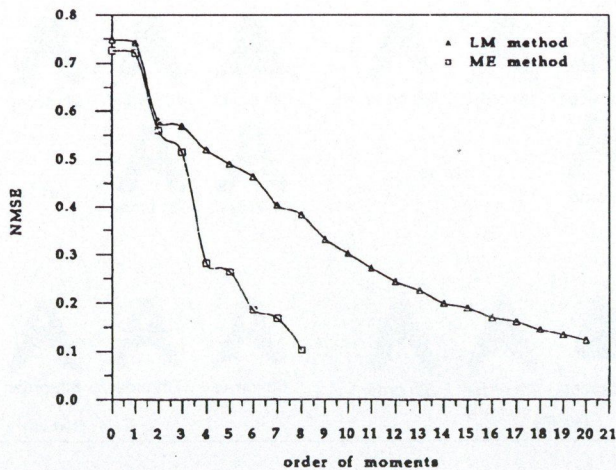
is considered as a good measure of the image reconstruction ability of the moments; so, it has been adopted here for comparing the performance of the two methods. Obviously, for discrete data processing integrations are replaced by summations in Eq. (15).



(a) Noise Free Reconstruction of Letter A using LM Method



(b) Noise Free Reconstruction of Letter A using ME Method



(c) Normalized mean-square error vs. maximum moment order used
Fig. 1. Noise free image reconstruction

Fig. 1(c) presents this error as a function of N . The superiority of the entropy method is obvious; an ME reconstruction even with moments up to only the 18th order is much better than an LM reconstruction based on moments up to the 18th order.

Figs. 2 and 3 show simulation results of the reconstruction of a noisy image (for different normal noise levels) by including increasingly higher-order moments. As measure of the amount of noise present on the image, the signal-to-noise ratio (SNR) is used, which is defined here as the ratio of the image energy per unit area to the noise variance, i.e.,

$$SNR = \frac{\frac{1}{S_D} \int_D \int [f(x, y)]^2 dx dy}{\sigma_n^2}, \quad (16)$$

where S is the area of the region D .

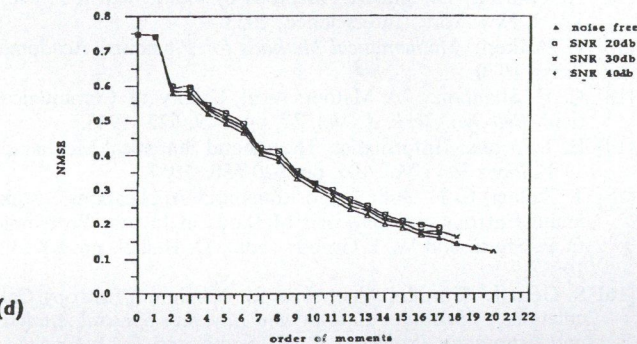
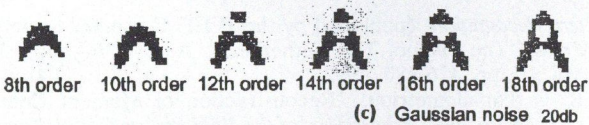
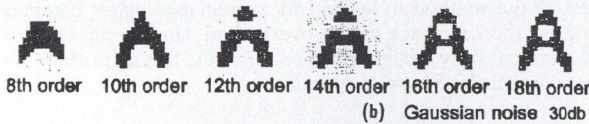
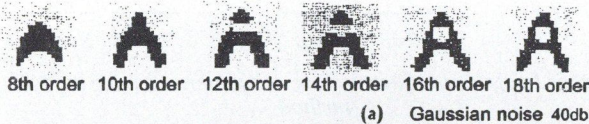


Fig. 2. Reconstruction of a noisy image using the LM method

The normalized reconstruction error, given in Figs. 2(d) and 3(d) (for LM and ME, respectively) as a function of N (the order of the CMS used) with the SNR as a parameter, was calculated by averaging over ten noisy image realizations generated for each SNR value. Comparing these two plots, it turns out that, although the ME method appears to be affected by noise much more than the LM method, for reasonable noise levels the first is still superior to the second; the results from the two methods become very similar only for higher noise levels on the image, which is not of practical interest here.

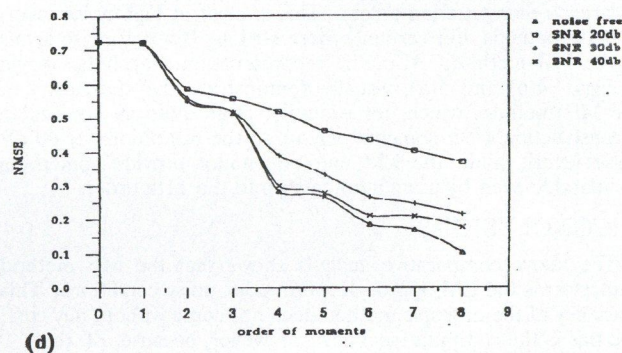
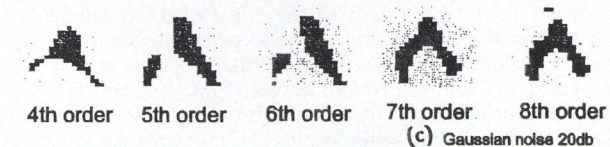
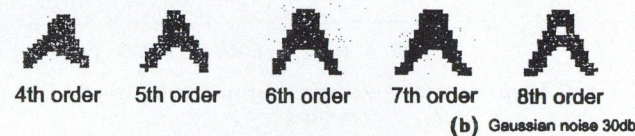


Fig. 3. Reconstruction of a noisy image using the ME method

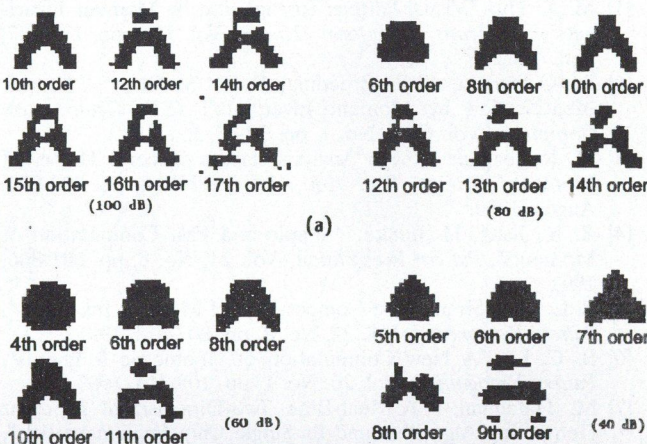


Fig. 4(a)

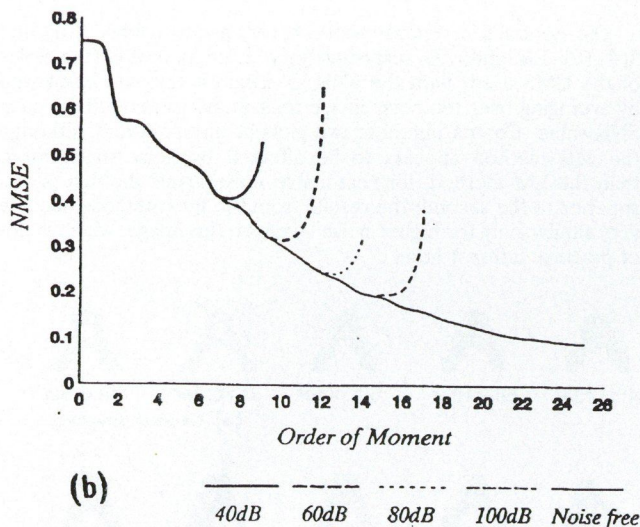


Fig. 4. Image reconstruction from noisy moments using the LM method

The investigation of the effect of noise degradation in the image domain is followed by a study of the effect of noise degradation in the representation domain (i.e., noise on the moment vectors). Results for additive Gaussian noise on the moments are given in Figs. 4 and 5. From the NMSE curves (Figs. 4(b) and 5(b)) we come to a conclusion pertaining to both methods: For each SNR value, there is a certain **optimal** order of moments (different for the two methods), which leads to the best image reconstruction. Using moments of order higher than the optimal order will result in larger reconstruction errors. This is clear at high noise levels in both methods, but certainly more striking (for any noise level) in the LM method. Also, the reconstruction snapshots, shown in Figs. 4(a) and 5(a), visually demonstrate the superiority of the ME method, which, for example, gives quite an acceptable reconstruction with moments only up to the 6th order (for 60 dB noise level), while the LM method cannot provide something comparable even by using moments up to the 11th order.

6. CONCLUSIONS

The above comparative analysis shows that the ME method outperforms the LM approach even under noisy conditions. This efficiency of the entropic method does not come without any cost; and this is the computational cost. However, because of the

REFERENCES

- [1] M. K. Hu: "Visual Pattern Recognition by Moment Invariants", *IRE Trans. on Inform. Theory*, Vol. IT-8, pp. 179-187, Feb. 1962.
- [2] S. A. Dudani, K. J. Breeding, R. B. McGhee: "Aircraft Identification by Moment Invariants", *IEEE Trans. on Computers*, Vol. C-26, No. 1, pp. 39-45, Jan. 1977.
- [3] M. R. Teague: "Image Analysis via the General Theory of Moments", *J. Opt. Soc. Am.*, Vol. 70, No. 8, pp. 920-930, August 1980.
- [4] X. Y. Jiang, H. Bunke: "Simple and Fast Computation of Moments", *Pattern Recognition*, Vol. 24, No. 8, pp. 801-806, 1991.
- [5] B. C. Li, J. Shen: "Fast Computation of Moment Invariants", *Pattern Recognition*, Vol. 24, No. 8, pp. 807-813, 1991.
- [6] B. C. Li: "A New Computation of Geometric Moments", *Pattern Recognition*, Vol. 26, No. 1, pp. 109-113, 1993.
- [7] M. Hatamian: "A Real-Time Two-Dimensional Moment Generating Algorithm and Its Single Chip Implementation", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 3, pp. 546-553, June 1986.
- [8] R. C. Papademetriou: "ERIM - Entropic Reconstruction of Images from their Moments", unpublished manuscript, 1985.
- [9] R. C. Papademetriou: "Reconstructing with Moments", *Proceedings of the 1th LAPR International Conference on Pat-*

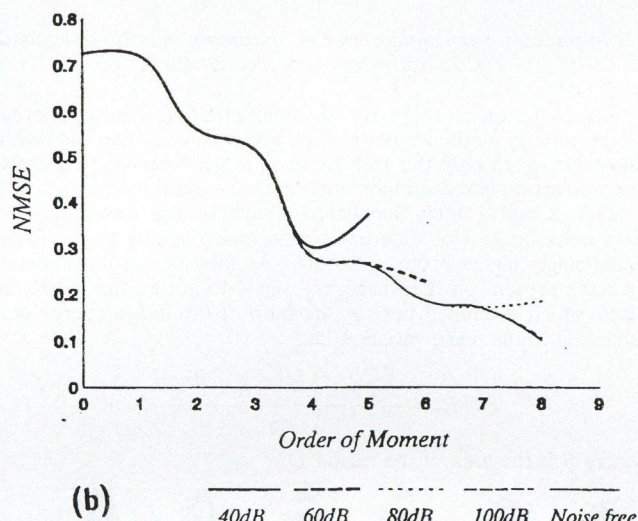
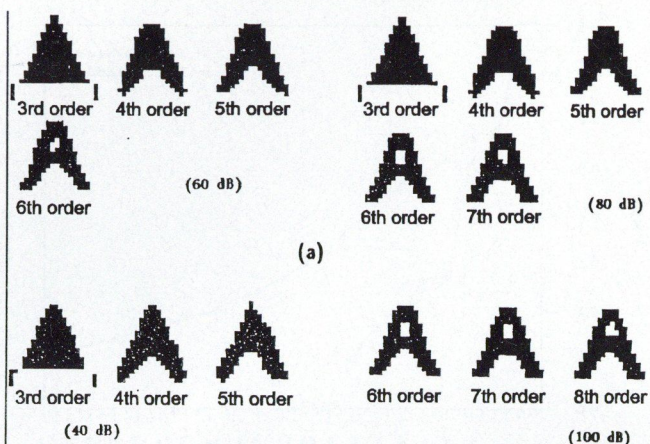


Fig. 5. Image reconstruction from noisy moments using the ME method

success of the method in leading to appreciably higher compression ratios, this drawback can be overlooked, since it can be eventually eliminated by designing more efficient, fast algorithms and special purpose VLSI architectures.

- tern Recognition, (published by the IEEE Computer Society Press), The Hague, The Netherlands, August 30-Sept. 3, Vol. III, pp. 476-480. 1992.
- [10] R. C. Papademetriou: "Reconstruction of Moment Compressed Images", *Proceedings of the 1993 Picture Coding Symposium (PCS '93)*, Swiss Federal Institute of Technology, Lausanne, Switzerland, 17-19 March, pp. 18. 1993.
- [11] R. Courant, D. Hilbert: *Methods of Mathematical Physics*, Vol. I, New York: Interscience, 1953.
- [12] G. Arfken: *Mathematical Methods for Physicists*, Academic Press, 1970.
- [13] C. E. Shannon: "A Mathematical Theory of Communication", *Bell Syst. Tech. J.*, Vol. 27, pp. 379, 623. 1948.
- [14] E. T. Jaynes: "Information Theory and Statistical Mechanics - I", *Phys. Rev.*, Vol. 106, pp. 620-630. 1957.
- [15] J. Skilling, S. F. Gull: "Algorithms and Applications", *Maximum Entropy and Bayesian Methods in Inverse Problems*, C. R. Smith and W. T. Grandy (eds), D. Reidel, pp. 83-132. 1985.
- [16] S. Geman: "Parallel Algorithms for Maximum Entropy Calculations", *Maximum Entropy and Bayesian Spectral Analysis and Estimation Problems*, C. R. Smith and G. J. Erickson (Eds), D. Reidel, pp. 317-318. 1987.

STOCHASTIC AND DETERMINISTIC METHODS IN MOTION PICTURE RESTORATION

R. D. MORRIS and W. J. FITZGERALD

COMMUNICATIONS AND SIGNAL PROCESSING LABORATORY
CAMBRIDGE UNIVERSITY ENGINEERING DEPARTMENT
TRUMPINGTON STREET, CAMBRIDGE, CB2 1PZ
ENGLAND

1. INTRODUCTION

The problem addressed here is that of restoring 'scratches' or 'blotches' which are a major defect in archived motion pictures. These can be formed by abrasion removing some of the emulsion, causing a bright defect, or dirt becoming attached to the film, resulting in a dark defect.

The common feature of these defects is that they can be characterized as a *temporal discontinuity*. In an image sequence, the scratches and blotches are obvious as areas for which there is no match in the previous or subsequent frames, taking motion between the frames into account.

The heuristic for the detection of scratches and blotches is thus — a *scratch is a connected area of a frame which is unpredictable from either the previous or following frames*. The use of Markov Random Field (MRF) theory allows the spatial continuity to be included in the detection algorithm.

The detection is reliant on the estimation of motion between the frames. A hierarchical blockmatching algorithm was used in this work — see [1], [2]. This algorithm has been found to be robust to the degradations and is able to estimate the large displacements often found in motion picture sequences.

Once the scratches have been detected, the interpolation problem is a missing data problem. The observed data at the scratch locations bears no relation to the actual scene information. In Bayesian terminology, the likelihood is uniformly distributed over all the allowed values. Hence the interpolation problem is a signal modelling one. MRF models are used successfully for this problem.

2. SCRATCH DETECTION USING MRFs

Consider two adjacent frames from a sequence. Let S denote the pixel lattice of these two frames taken together, and X be the observed gray-levels at these lattice points. Let $\mathcal{N}_i$ denote the neighbourhood of site i which is within the frame, and $\mathcal{T}_i$ denote the motion-compensated temporal neighbourhood in the other frame of the pair. Define a discontinuity frame, D , between the two image frames, on a lattice S' . Let $d_i \in \{-1, +1\}$, where $d_i = 1$ denotes a temporal discontinuity between the two frames, and where $d_i = -1$ denotes no discontinuity. It is this frame D which is to be estimated.

Bayes theorem states that

$$p(D = d|X = x)\alpha p(X = x|D = d)p(D = d). \quad (1)$$

This consists of two terms: a likelihood model for the observed frames, given the discontinuity locations, and a prior model for the discontinuity locations. It is this prior model which encodes the spatial continuity of the scratches. Using a simple spatio-temporal likelihood function and an Ising prior to organise the discontinuities, we have

$$p(D = d|X = x) = \frac{1}{Z} \exp\left(-\frac{1}{T}U(d)\right) \quad (2)$$

$$U(d) = \sum_{i \in S'} \left[\alpha(1 - d_i) \sum_{j \in \mathcal{T}_i} (x_i - x_j)^2 - \beta_1 \sum_{j \in \mathcal{N}_i} d_i d_j + \beta_2(1 + d_i) \right], \quad (3)$$

where the pixels in the temporal term are at positions in S corresponding to position i in S' and its temporal neighbour, and terms not involving d have been neglected. Optimising this expression solves for the maximum a-posteriori (MAP) configuration of the discontinuity frame. Regions which have discontinuities in both the forwards and backwards directions are consistent with the heuristic for scratches and are classified as such.

2.1. Parameter Estimation

The above model depends on three parameters: α, β_1, β_2 . Estimates for the parameter values may be found as follows.

The parameter β_1 determines the strength of the self-interaction of the discontinuity field. Ripley [3] gives arguments for a value around 1 by considering the conditional probability assigned to a pixel when surrounded by different numbers of pixels in the same state. The parameter β_2 'balances' the increase in probability due to introducing a discontinuity. Thus to balance a difference of e_1 requires

$$\alpha e_1^2 \simeq \beta_2. \quad (4)$$

Also, consider an isolated pixel with error e_2 . For this to be detected requires

$$\exp(-\beta_2) > \exp(-\alpha e_2^2 + 4\beta_1). \quad (5)$$

Thus by *quantifying* the heuristic the values of the parameters of the MRF used to detect them may be chosen in a consistent manner.

The problem of finding the optimum configuration $\hat{d}$ for the discontinuity frame is now addressed.

2.2. Stochastic Simulated Annealing (SSA)

The variable T in Eq. (2) alters the 'peakiness' of the distribution. SSA uses this as a control parameter to enable the MAP configuration to be found. For large values of T the distribution is essentially uniform; for $T \rightarrow 0$ it becomes concentrated at the mode. SSA repeatedly samples from the distribution as the temperature is reduced. A logarithmic cooling schedule will cause convergence to the MAP configuration [4]. Practically the logarithmic cooling schedule is too slow, and the sub-optimal exponential schedule, $T = Ca^k$, $a < 1$, is often used [5].

2.3. First-Order Mean Field Approximation (MFA)

SSA finds the MAP configuration but is slow and computationally intensive. MFA is a deterministic optimisation technique which retains many of the features of SSA. The mean value of the field is

$$\bar{d} = \sum_d dp(d)$$

which is the minimum variance estimate [6]. This summation is over all possible states of the field, and so some simpler approximation must be made.

Approximate the influence of d_j , $j \in \mathcal{N}_i$ on the calculation of $\langle d_i \rangle$ by the influence of $\langle d_j \rangle$. This gives

$$\langle d_i \rangle = \frac{1}{Z_i^{mf}} \sum_{d_i} d_i \exp \left(-\frac{1}{T} \left[\alpha(1-d_i) \sum_{j \in \mathcal{T}_i} (x_i - x_j)^2 - \beta_1 \sum_{j \in \mathcal{N}_i} d_j \langle d_j \rangle + \beta_2(1+d_i) \right] \right). \quad (6)$$

This is an estimate formed by finding the mean value of each pixel, based on the current values of the pixels in its neighbourhood. This approximate mean value of d_i can be computed easily on the basis of the values $\langle d_j \rangle$ in its neighbourhood. For small state-spaces and small neighbourhoods especially, the sum involved in Eq. (6) in calculating $\langle d_i \rangle$ is over a small number of terms, and is not computationally intensive to calculate.

Typically for binary problems, such as the scratch detection problem, a zero initial field is chosen, and Eq. (6) is calculated for all $i \in S'$. The temperature is reduced, and the iteration repeated. A similar cooling schedule as for SSA is usually used, but this mean field approximation often converges much more quickly, with possibly poorer results. The values of $\langle d_i \rangle$ computed from Eq. (6) are continuous valued, and are thresholded to give the final detection field.

2.4. An Improved Mean Field Approximation

In this section the probability distribution of Eq. (2) is approximated by some much simpler distribution, one for which the maximum probability state is trivial to find. For a binary system defined on $\{-1, +1\}$, the distribution

$$p_0(d) = \frac{1}{Z_0} \exp \left(\frac{1}{T} U_0(d) \right), \quad U_0(d) = \sum_i m_i d_i \quad (7)$$

is convenient. Clearly $p_0(d)$ is maximized for $d = m$. The remaining problem is how to choose the parameters m_i so that the distribution $p_0(d)$ best approximates $p(d)$ in Eq. (2) in some sense [7], [8]. Define

$$\langle A \rangle_m = \sum_{d_1 \dots d_N = \pm 1} A p_0(d). \quad (8)$$

Consider

$$Q = -\frac{1}{T}(U - U_0). \quad (9)$$

From the binomial expansion

$$\langle \exp(Q) \rangle_m \geq \exp(\langle Q \rangle_m) \quad (10)$$

giving

$$-T \ln Z \leq -T \ln Z_0 + \langle U - U_0 \rangle_m. \quad (11)$$

In statistical mechanics terms this is the Gibbs-Bogoliubov-Feynman bound [9], and clearly U_0 best approximates U when the right hand side of Eq. (11) is minimized, such that the tightest bound is obtained. That is

$$\nabla_m [-T \ln Z_0 + \langle U - U_0 \rangle_m] = 0.$$

In Eq. (3) define $h_i = \beta_2 - \alpha \sum_{j \in \mathcal{T}_i} (x_i - x_j)^2$, and remove constant terms. The summations involved in forming $-T \ln Z_0 + \langle U - U_0 \rangle_m$ can be performed resulting in

$$\begin{aligned} \frac{\partial}{\partial m_i} (-T \ln Z_0 + \langle U - U_0 \rangle_m) = \\ = \frac{1}{T} [1 - \tanh^2(\frac{m_i}{T})] \left[h_i + m_i - \frac{\beta_1}{2} \sum_{j \in \mathcal{N}_i} \tanh(\frac{m_j}{T}) \right]. \end{aligned} \quad (12)$$

The optimum parameters of the approximation can be found by gradient descent using these gradients. Again Eq. (12)

is used to minimize the bound in Eq. (11) at a series of temperatures associated with an annealing schedule. At each reduced temperature the converged result of the minimisation at the previous temperature is used as the starting point. Again, the final result is thresholded to give the detection field.

3. MISSING REGION INTERPOLATION USING MRFs

As outlined earlier, the problem of interpolating into the detected scratch regions is a missing data problem. If the likelihood function for the frame, given the observed frame and the detection field is examined, it is

$$p(X = x|D = d) = \prod_{\{i: d_i = -1\}} \delta(x_i) \quad (13)$$

so that the undergraded areas of the image are left unmodified, but in the areas marked as scratches, the observed data does not influence the restoration. This is due to the *replacement* nature of the degradation.

An image sequence model, conditioned on the image at locations not marked as scratches must be constructed, and used for the interpolation. In this section a simple model based on a spatio-temporal MRF will be introduced for this problem, and details of its application via the Gibbs sampler and the mean field approximations presented.

Again using a first-order neighbourhood, pair cliques and the quadratic potential function, the local conditional probability distribution may be written as

$$\begin{aligned} p(X_i = x_i) = \\ = \frac{1}{Z_{X_i}} \exp \left(\frac{1}{T} \sum_{j \in \mathcal{N}_i} (x_i - x_j)^2 - \frac{\lambda}{T} \sum_{j \in \mathcal{T}_i} (x_i - x_j)^2 \right), \end{aligned} \quad (14)$$

where the temporal neighbourhood $\mathcal{T}_i$ now includes pixels from frames both sides of the current frame. The restriction to a small neighbourhood and simple cliques, whilst not suitable for general modelling of complex images, is suitable for this purpose as the scratch regions tend to be fairly small, and the temporal neighbourhood provides much information about the true sequence. Thus the energy function for this problem is

$$U(x) = \sum_{\{i: d_i = 1\}} \left[\sum_{j \in \mathcal{N}_i} (x_i - x_j)^2 + \lambda \sum_{j \in \mathcal{T}_i} (x_i - x_j)^2 \right]. \quad (15)$$

The quadratic potential function is used to give a smooth interpolation.

3.1. Interpolation by Gibbs Sampling

The Gibbs sampler can be used to draw a representative sample from a distribution such as that described by the energy function in Eq. (15). This sample is then used as the interpolation. Specifically, samples are drawn from $p(X_k | X_{S \setminus \{k\}})$, where k indexes successive elements from the set $\{d_i = 1\}$. Information from the edges of the scratch and from the temporal neighbourhood will propagate into the scratch regions. The restoration will, however, still be a sample from $P(X = x)$, and as such may show unnecessary variations, for example, texture may appear in a uniform region of the image. To overcome this, annealing may be introduced into the Gibbs sampler, to cause the interpolation to converge to the maximum probability solution.

3.2. Interpolation by First-Order Mean Field Approximation

The mean field approximation may also be used with the energy function of Eq. (15) to provide an interpolation of the missing regions.

As before, the influence of x_i , $j \in \mathcal{N}_i$ in the calculation of $\langle x_i \rangle$ is approximated by the influence of $\langle x_j \rangle$. This results in the mean field energy function becoming

$$U_i^{mf}(x_i) = \sum_{j \in \mathcal{N}_i} (x_i - \langle x_j \rangle)^2 + \lambda \sum_{j \in \mathcal{T}_i} (x_i - x_j)^2, \quad (16)$$

and the estimate of $\langle x_i \rangle$ is given by

$$\langle x_i \rangle = \frac{1}{Z_i^{mf}} \sum_{x_i} x_i \exp \left(-\frac{1}{T} U_i^{mf}(x_i) \right). \quad (17)$$

This is iterated over all $\{i : d_i = 1\}$, and the converged results the interpolation. When x_i takes discrete values, the values generated must be rounded to the nearest discrete value at the end of the iterative procedure.

3.3. An Improved Mean Field Interpolator

As above the energy function of Eq. (15) is approximated by some simpler energy function. For MRFs defined over a large number of gray-levels, which may conveniently be approximated as a continuous state-space, the energy function

$$U_0(x) = \sum_{\{i: d_i=1\}} (x_i - m_i)^2 \quad (18)$$

is useful approximation. Again the problem is to select the parameters m_i to minimize the error caused by using $U_0(x)$ in place of $U(x)$. Again the 'best' approximation is when $\nabla_m [-T \ln Z_0 + \langle U - U_0 \rangle_m] = 0$.

In the calculation of $\langle U - U_0 \rangle_m$ care must be taken when performing the integrals (the summations in Eq. (8) are replaced by integrals now that the variables are continuous) to distinguish between values of $x_{i,j} \in \mathcal{N}_i$ which are *within* the scratch region, and hence variable, and those outside the scratch region which are fixed, known values (see Fig. 1). The symmetry of the neighbourhood structure, and the use of the quadratic potential function enables the integral to be evaluated.

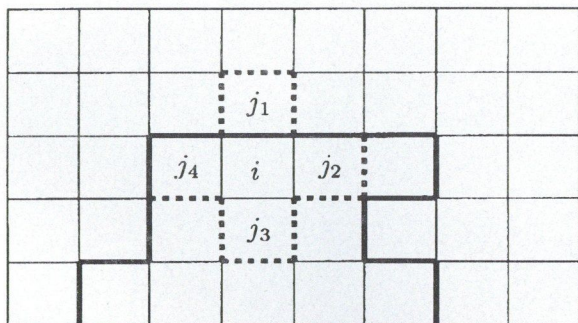


Fig. 1. Some neighbours of x_i may be within the scratch area, denoted by the bold line (i.e. j_2, j_3, j_4) others may not (i.e. j_1).

Performing the integrals gives

$$\begin{aligned} \frac{\partial}{\partial m_i} (-T \ln Z_0 + \langle U - U_0 \rangle_m) &= \sum_{j \in \mathcal{N}_i \cap d_i=1} 4(m_i - m_j) + \\ &+ \sum_{j \in \mathcal{N}_i \cap d_i=-1} 2(m_i - x_j) + \lambda \sum_{j \in \mathcal{T}_i} 2(m_i - x_j). \end{aligned} \quad (19)$$

Again these gradients are used to find the optimum m_i 's to minimize the errors in the approximation. Once the optimum m_i 's have been found, the optimum interpolator is that given by $x_i = m_i$ at the scratch locations.

4. DETECTION RESULTS

To quantify the action of the detector under the three optimisation schemes, the algorithms were applied to an artificially degraded sequence. Fig. 2 shows a frame from the sequence, with artificial scratches added.



Fig. 2. Degraded frame from sequence

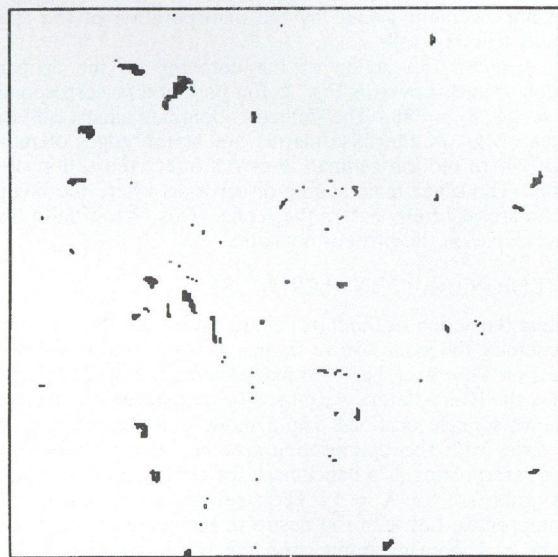


Fig. 3. Detection frame

Fig. 4 shows a plot of Correct Detection rate vs. False Alarm rate for the three detectors, averaged over twelve frames. For this problem the two approximate optimisation schemes provide accurate approximations, their responses being very close to that of the SSA optimisation. The solid curve on the graph indicates the characteristic of the detector which operates by thresholding the motion-compensated frame differences [10].

The new detector produces some improvement, but the improvement is limited. Examination of Eq. (2) reveals why this is the case. If the term $-\beta_1 \sum_{j \in \mathcal{N}_i} d_i d_j$ is neglected then the distribution becomes a simple product of terms, one for each pixel, and depending on the values of α and β_2 , each term will be maximized by a particular value of d_i — this corresponds exactly to thresholding the motion compensated frame difference.

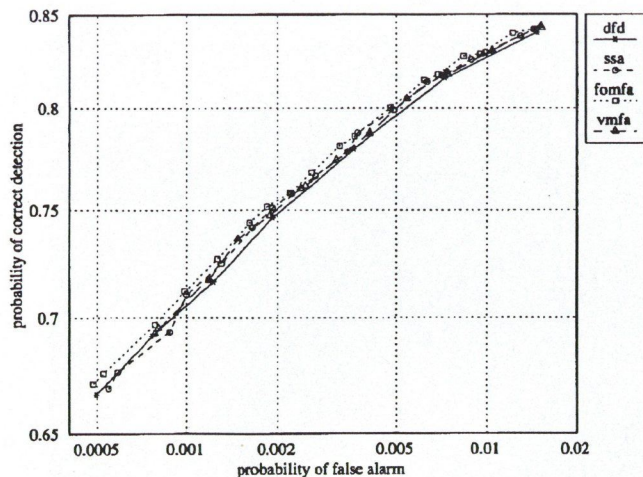


Fig. 4. Detector operating characteristics

The additional term in Eq. (2) will only become important if the other two terms are very closely balanced. This will happen for scratches of small gray-level difference. Clearly these will only make up a small proportion of the total number of scratches, and so the improvement gained by the incorporation of the spatial continuity will be small.

Fig. 3 shows the action of the detector on the artificially degraded frame shown in Fig. 2, for parameters corresponding to $e_1 = 24$, $e_2 = 30$. The detector identifies almost all visible blotches. Most of the false alarms are at the edges of moving objects, where motion estimation errors affect the action of the detector. The other main source of errors is where the lining of the actor's coat briefly enters the scene. This type of false alarm is unavoidable on the present heuristic.

5. INTERPOLATION RESULTS

To test the action of the interpolator under the three optimisation schemes, the same twelve frames as were used in the detection section were used here. To judge the action of the interpolator only, the interpolators were used to reconstruct the frames at the known scratch locations. Fig. 5 shows how the mean squared error varies with the optimisation scheme, with a simple frame average interpolator as a benchmark for comparison. The results in this graph are for $\lambda = 1$. The strength of the temporal link is a compromise between the desire to include good information from the neighbouring frame, and the possibility of introducing erroneous information due to errors in the motion estimation. The results on this diagram need some comment.

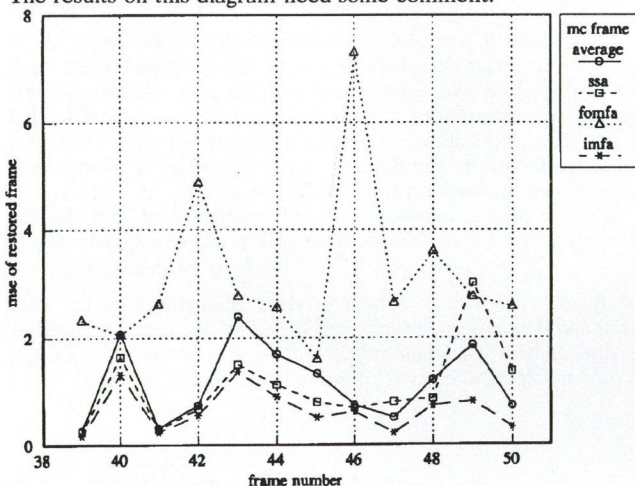


Fig. 5. MSE for each optimisation scheme

5.1. Gibbs sampler interpolation

The Gibbs sampler restoration is a *sample* from the distribution describing the reconstructed frame. This will show some variation each time the sample is drawn. The mean squared error will therefore be different each time the restoration is performed. Visual comparison of the restored frames with those restored by the simple frame averager showed a definite improvement in image quality.

5.2. First-order mean field approximation interpolator

The results for the first-order mean field approximation interpolator were not satisfactory. Their visual quality was poor. This is a result of the approximations not allowing spatial information to propagate into the centre of scratch regions. The large state-space also causes the reconstruction to be badly biased by the initialization.

5.3. Improved mean field interpolator

Because the potential function used was quadratic, the gradients allow the algorithm to converge very closely to the optimum. This is the cause of the very low mean squared errors shown on Fig. 5. The use of gradient based optimisation also allowed this algorithm to converge much more quickly than the other two. The visual quality of the restored frames was also very good.

Fig. 6 is the restored frame from the artificially degraded sequence, where SSA has been used to detect the scratches, and the improved mean field interpolator has been used to interpolate into these regions. The artefacts due to the false alarm regions, for example in the actor's hair, are not distracting and the overall visual quality is very good.



Fig. 6. Interpolated frame, scratch locations from SSA detector

6. CONCLUSIONS

The use of Markov Random Fields in the detection and interpolation of scratches in archived motion pictures has been described. Both stochastic and deterministic optimisation schemes have been discussed. A hybrid algorithm has been found to give the best results for the overall restoration problem. The limiting factor in the accuracy of the restoration was found to be the motion estimates, and work on developing motion estimation algorithms matched to this problem is needed.

REFERENCES

[1] M. Bierling: "Displacement estimation by hierarchical block matching", *SPIE: Visual Communications and Image Processing*, 1001: 942-951, 1988.

[2] J. Boyce: "Noise reduction in image sequences using adaptive motion compensated frame averaging", *Proceedings of IEEE ICASSP'92*, pp. 461-464, 1992.

[3] B. D. Ripley: *Statistical inference for spatial processes*, CUP, 1988.

[4] S. Geman, D. Geman: "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Trans. PAMI*, 6(6): 721-741, November 1984.

[5] J. Konrad, E. Dubois: "Bayesian estimation of motion vector fields", *IEEE Trans. PAMI*, 14(9), September 1992.

[6] D. Geiger, F. Girosi: "Parallel and deterministic algorithms from MRF's: Surface reconstruction", *IEEE Trans. PAMI*, pp. 401-412. 1991.

[7] H. P. Hiriyanaiiah, G. L. Bilbro, W. E. Snyder: "Restoration of piecewise-constant images by mean-field annealing", *J. Opt. Soc. Am.*, pp. 1901-1912, 1989.

[8] G. L. Bilbro, W. E. Snyder, R. C. Mann: "Mean-field approximation minimizes relative entropy", *J. Opt. Soc. Am.*, 8(2): 290-294. February 1991.

[9] D. Chandler: *Introduction to Modern Statistical Mechanics*, OUP, 1987.

[10] A. C. Kokaram: *Motion Picture Restoration*, Ph.D. thesis, Cambridge University, 1993.

FISH SPECIES RECOGNITION USING A BAYESIAN CLASSIFICATION PROCESS

N. CASTIGNOLLES

M. CATTOEN

M. LARINIER

INP-ENSEEIH
LABORATOIRE D'ELECTRONIQUE-GTTSI
2, RUE CHARLES CAMICHEL
31071 TOULOUSE CEDEX, FRANCE

INP-ENSEEIH
LABORATOIRE D'ELECTRONIQUE-GTTSI
2, RUE CHARLES CAMICHEL
31071 TOULOUSE CEDEX, FRANCE

CSP-CEMAGREF INSTITUT
DE MECANIQUE DES FLUIDES-GHAAPPE
AVENUE DU PROFESSEUR CAMILLE SOULA
31400 TOULOUSE, FRANCE

1. INTRODUCTION

A consequence of dam construction in rivers is the gradual disappearance, since the beginning of the century, of migratory fish. To avoid this, devices called fish passes have been constructed to help them to get over obstacles. There, window panes are used to observe and count by species the fish which cross. At the present time, a video camera is placed in front of the window pane and is connected to a video tape recorder to record fish crossings. Then, someone has to view the video tapes to recognize and count the fish. Since it is a very tedious job, our objective is to automate it by using a vision system. In this article, we focus on the automatic recognition problem. Species to be recognized are : shad, eel, barbel, bream, salmonids (salmon and sea trout), and river trout.

2. IMAGE ACQUISITION CONDITIONS

The images used to accomplish the recognition are taken by a video camera fitted with electronic shutter (shuttering time : 2 ms) in a backlit fish pass (Fig. 1) [6]. A watertight caisson is placed at the back of the pass. It contains several neon lightings and is closed by a translucent material which diffuses light. This arrangement provides good contrast and very sharp fish outlines in spite of their rapid movements. Fig. 2 is an example of the images taken under these conditions while Fig. 3 represents its gray-level histogram.

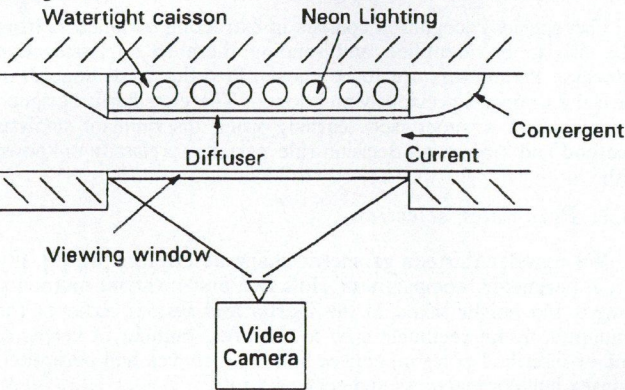


Fig. 1.

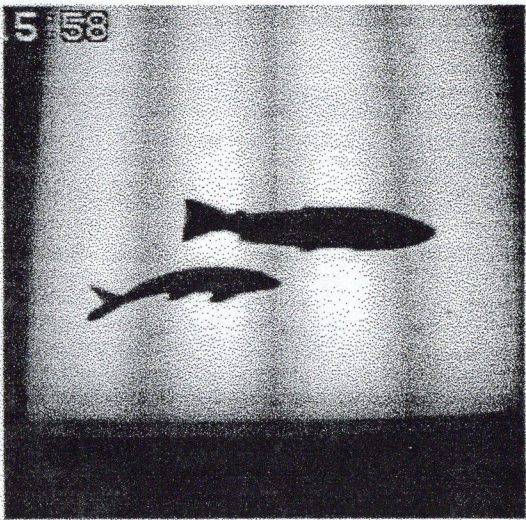


Fig. 2.

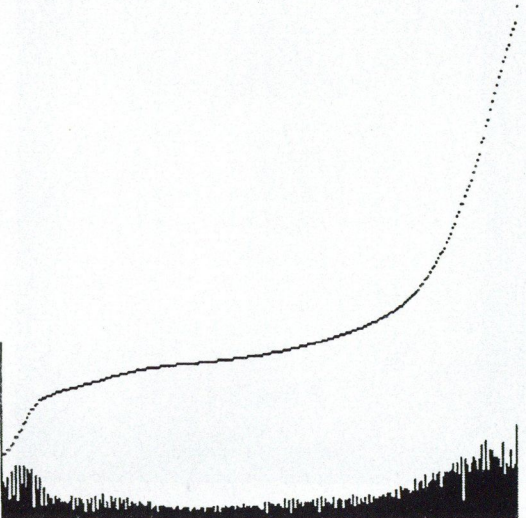


Fig. 3.

Previous histogram is bi-modal and a median threshold permits to accurately isolate the fish from the back of the fish pass (Fig. 4).

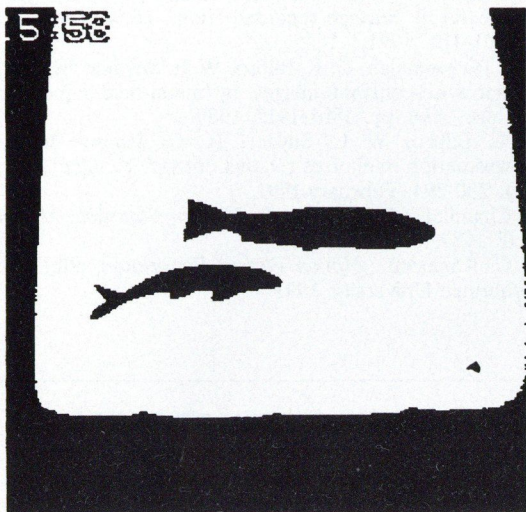


Fig. 4.

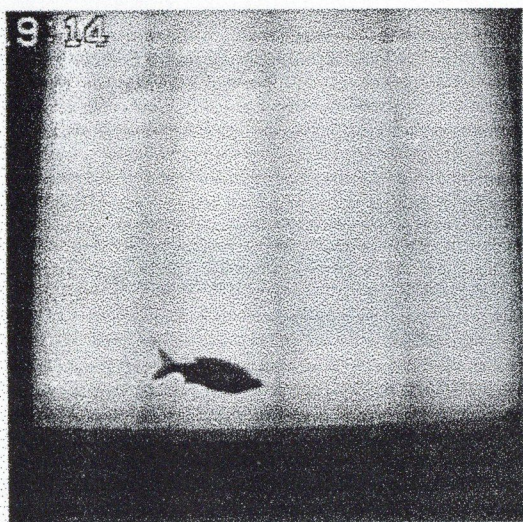


Fig. 5.

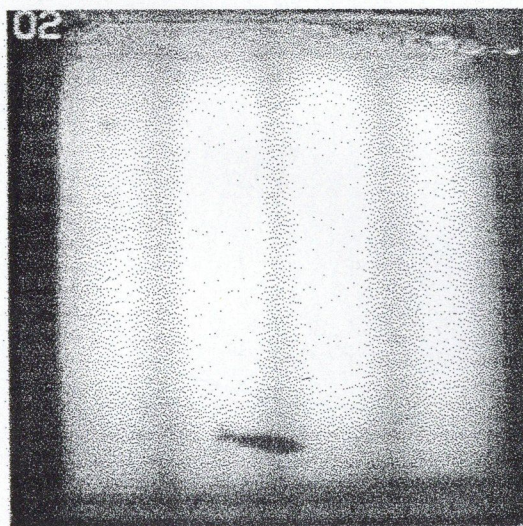


Fig. 6.

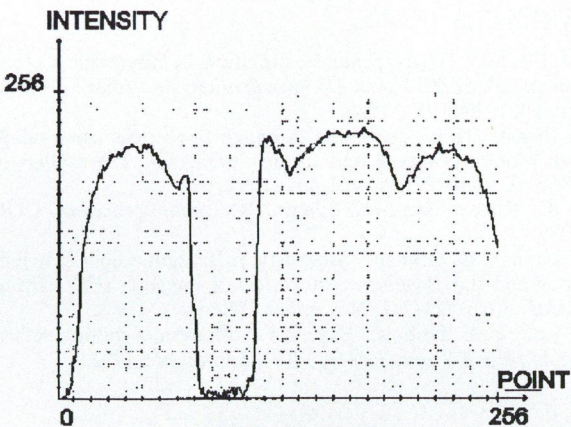


Fig. 7.

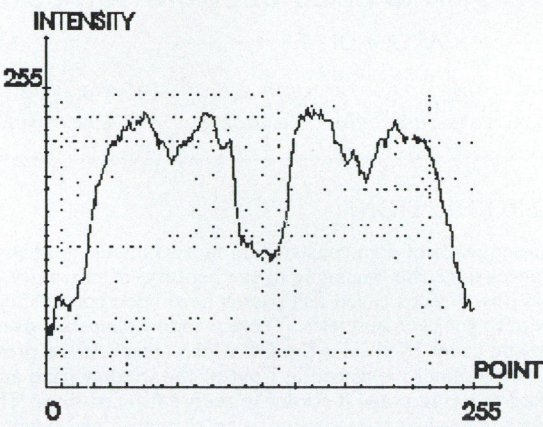


Fig. 8.

However, a problem can appear when water becomes turbid. Image's contrast decreases and then it is not so easier to isolate the fish from the back of the pass. Fig. 5 has been taken while water was clear whereas water was turbid in Fig. 6. Fig. 7 and Fig. 8 show the two corresponding intensity profiles.

The fish is clearly better isolated from the back of the pass in Fig. 5. In Fig. 6 it is more difficult to separate them. If this situation comes to appear frequently, it will be necessary to avoid binarization and use another method to extract fish to be recognized.

3. SPECIES RECOGNITION

Fish species recognition consists in extracting parameters from the fish to be identified and sending them to a classification process. Parameters are to be chosen in order to distinguish at best the 6 species previously mentioned. The classifier is designed by means of a supervised learning using discriminant analysis method and a bayesian decision rule permits to classify unknown fish.

3.1. Parameter selection

We consider thirteen geometric shape descriptors [4], [5], [8]: area, perimeter, compactness, Hu's two first invariant moments, length and height taken in the inertia axis system, ratio of the minimum frame rectangle area to the area, number of vertex of the ex-inscribed polygon, convex hull area, convex hull perimeter, convex hull compactness and concavity rate.

To evaluate parameter performances, each one has been used separately to classify the six species. This work has been done upon a classifier design data base including about 2000 images. Obtained results are presented in Fig. 9.

	Shad	Eel	Barbel	Bream	Salmonids	River trout
Area (p1)	17,29	100	31,67	13,07	72,29	54,97
Perimeter (p2)	10,53	79,52	40	56,97	61,71	48,34
Compactn. (p3)	62,78	100	45,28	44,08	70,65	17,22
Hu's first moment (p4)	73,68	100	10	98,26	26,09	53,64
Hu's second moment (p5)	72,18	100	6,39	99,65	26,32	45,03
Lenght (p6)	24,81	53,01	36,67	59,23	64,36	60,26
Height (p7)	21,41	93,98	20,28	19,34	56,3	67,55
Obj. area /min. Rect. area (p8)	54,89	2,41	35,83	35,89	25,44	79,47
Vertex number (p9)	37,97	100	58,89	29,79	9,45	73,51
Convex hull area (p10)	14,29	100	36,11	17,25	60,96	59,6
Convex hull perimeter (p11)	24,81	63,86	41,39	58,89	63,48	53,64
Convex hull compactn. (p12)	48,12	93,98	38,61	89,9	5,79	1,32
Concavity rate (p13)	69,92	9,64	41,11	30,31	26,32	80,79

Fig. 9.

Best parameters, for each species, are boldfaced in previous array. All of them (p1, p3, p4, p5, p6, p7, p8, p9, p11, p12, p13) could be processed for each fish to recognize but, as species recognition is included in a tracking process [2], it is necessary to reduce its processing time as much as possible. For that, two points are to be considered:

- some parameters are correlated;
- the schedule introduces different levels of constraints: error rates must be lower than 5 % for salmonids, 15 % for shad, eel and river trout and 25 % for bream and barbel.

Fig. 10 represents the parameter correlation matrix.

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13
p1	1	0,96	-0,13	-0,1	-0,22	0,95	0,89	0,04	0,7	0,98	0,96	-0,11	0,02
p2		1	0,03	-0,01	-0,16	0,98	0,9	0,19	0,73	0,98	0,99	-0,05	0,21
p3			1	0,9	0,87	0,06	-0,2	0,51	-0,15	-0,06	0,02	0,81	0,59
p4				1	0,96	0,07	-0,3	0,25	-0,33	-0,08	0,03	0,94	0,32
p5					1	-0,07	-0,39	0,2	-0,4	-0,2	-0,12	0,91	0,26
p6						1	0,83	0,15	0,64	0,95	0,99	0,05	0,15
p7							1	0,3	0,81	0,92	0,86	-0,35	0,22
p8								1	0,23	0,15	0,15	0,04	0,88
p9									1	0,74	0,67	-0,39	0,22
p10										1	0,97	-0,12	0,16
p11											1	0	0,16
p12												1	0,07
p13													1

Fig. 10.

Keeping p1 (area), p3 (compactness), p4 (Hu's first moment), p6 (length), p7 (height), p9 (number of vertex of the ex-inscribed polygon) and p13 (concavity rate) appear to be a good compromise between a good discrimination (included in the schedule) and a reduced processing time.

Let's notice that, even if p6 is very correlated with p1, it is useful to better classify salmonids, which is the highest constraint.

3.2. Classification principle

The learning stage uses the multiple discriminant analysis method. Its principle consist in setting up an axis system as linear combinations of parameters allowing to separate at best the different classes by projections [3]. These so called discriminant axis, are the eigen vectors of the $T^{-1}B$ matrix, where T is the total covariance matrix and B the interclass covariance matrix. One axis discrimination performance is given by the associated eigen value.

Once the learning is accomplished, a new fish is classified using a bayesian decision rule [3]:

Let p be the number of classes, x the fish parameter vector and $P(w_j/x)$ the conditional probability that class w_j is correct knowing x .

Let the cost function c_{ij} to classify a fish in the class w_i whereas it belongs to the class w_j be:

$$c_{ij} = 1 - \delta_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases}$$

Then, a fish is assigned to the class w_j which corresponds to the maximum probability $P(w_j/x)$, where j varies from 1 to p.

Let $p(x/w_j)$ be the probability density function for x given that the state of nature is w_j , $P(w_j)$ be the a priori probability

of class w_j and $p(x) = \sum_{j=1}^p p(x/w_j)P(w_j)$.

Using Bayes' rule:

$$P(w_j/x) = \frac{p(x/w_j)P(w_j)}{p(x)}$$

and supposing our problem is gaussian, the decision rule can be written as follows:

$$f_d(x) = w_i \iff g_i(x) = \max(g_j(x)); \quad j = 1, \dots, p,$$

where f_d is the decision function and $g_i (i = 1, \dots, p)$ are the p discriminant functions.

All classes are supposed equiprobable and, since they have got close covariance matrix, the discriminant functions can be written as:

$$g_i(x) = m_i^t \sum^{-1} x - 1/2(m_i^t \sum^{-1} m_i),$$

where m_i is the w_i class mean vector and $\sum$ is the average covariance matrix.

4. REJECTION CRITERION

It can happen that leaves, tree branches of fish of other species than the six previously mentioned circulate in the pass. To avoid wrong classifications in these cases, a rejection criterion is used [7]. It is based upon the calculation of Mahalanobis distance to the nearest class center:

$$d^2(x, m_i) = (x - m_i)^t \sum^{-1} (x - m_i).$$

A statistical analysis has been carried on to determine a rejection threshold for each class. It has been done from the data base used to accomplish the classifier design. Fig. 11 represents an example of calculation of Mahalanobis distance for each fish of bream class' data base. The rejection threshold is set to 3σ , where σ is the class standard deviation. In Fig. 12 an eel has been introduced among breams and is clearly rejected.

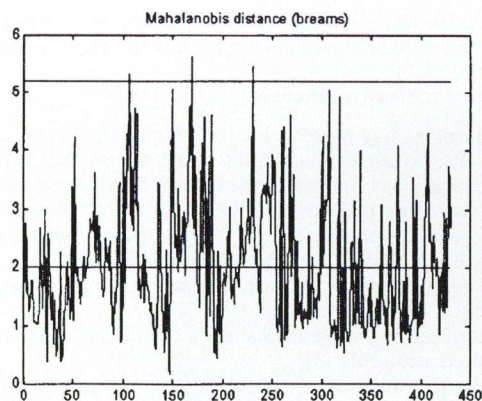


Fig. 11.

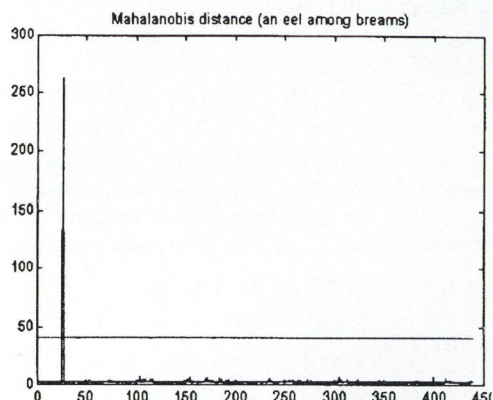


Fig. 12.

REFERENCES

- [1] H. Arnason, K. Bengoetxea, L. F. Pau: "Vision applications in the fishing and fish products industries", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 2, No. 4, pp. 657-671, 1988.
- [2] N. Castignolles, M. Cattoen, M. Larinier: "Identification and counting of live fish by image analysis", in *Image and Video Processing II*, S. A. Rajala, R. L. Stevenson, eds., *Proc. SPIE* 2182, pp. 200-209, 1994.
- [3] R. O. Duda, P. E. Hart: *Pattern classification and scene analysis*, A Wiley-Interscience Publication, 1973.
- [4] M. K. Hu: "Visual pattern recognition by moments invariants", *IRE Transactions on Information Theory*, 1962.

5. RESULTS

The test set includes about 1000 images of fish among the six species to recognize and obtained results are presented in the following array (Fig. 13).

Species	Shad	Eel	Barbel	Bream	Salmonids	River trout
Shad	90,8	0	6,12	0	3,08	0
Eel	0	100	0	0	0	0
Barbel	5,03	0	88,8	0	2,81	3,36
Bream	0	0	0	100	0	0
Salmonids	0	0	0,5	0,25	99	0,25
River trout	0	0	0	0	0	100

Fig. 13.

6. CONCLUSION

The obtained results are satisfactory. They are in accordance with the schedule and only 1 % of salmonids are misrecognized.

Furthermore, this species recognition is a part of a dynamic process of counting which tracks each fish while it is in the observation field [2]. So, a temporal redundancy is generated, which improves previous recognition results.

Now, some species are to be added to the ones yet recognized: pike, lamprey, carp... and the problem of turbid water has to be taken into account.

- [5] A. K. Jain: *Fundamentals of image processing*, Practice Hall International Editions, 1989.
- [6] J. M. Jolion: "Etude de faisabilité d'un système de comptage/reconnaissance de poissons sur le site d'une passe à poissons", Laboratoire d'Informatique Graphique et d'Intelligence Artificielle de Lyon, 1990.
- [7] M. H. Masson: *Contribution à l'élaboration d'une méthode de décision avec rejet par réseaux de neurones. Application au diagnostic de systèmes*, thèse de doctorat de l'Université de Compiègne, 1992.
- [8] N. J. C. Strachan, P. Nesvada, A. R. Allen: "Fish species recognition by shape analysis of images". *Pattern Recognition*, Vol. 23, No. 5, pp. 539-544. 1990.

RECOGNITION OF PRINTED BERBER CHARACTERS USING MODIFIED RAPID TRANSFORM

J. TURÁN

DEPARTMENT OF RADIOELECTRONICS TECHNICAL UNIVERSITY OF KOŠICE
PARK KOMENSKÉHO 13 04021 KOŠICE, SLOVAKIA

1. INTRODUCTION

Optical character recognition can be achieved in the pattern domain [1]. Geometrical or structural features are extracted and processed by different techniques that often involves high dimensionality vectors. The implementation of pattern transformations has been introduced as an alternative for character recognition [2]. They allow the definition of lower dimensionality represen-

tations, but they have to propose significant features in order to reduce the loss of information.

Various transformations have been suggested as a solution of the problem of high dimensionability of the feature vector and long computation time: Fourier, Walsh-Hadamard and Hough Transform [3]. In this paper, we explore the application of rapid and modified rapid transform [4], [6], to extract features used in the recognition of printed Berber characters.

The rapid transform (RT) is a fast shift invariant transform [4]. The RT is useful for pattern recognition, if the position of the pattern is unknown or the pattern is moving [5]. In some applications (recognition of characters), it is required that the transform be non invariant under reflection however, RT is invariant under reflection. The modified rapid transform (MRT) was presented to break undesired invariances of the RT [6], which leads to a loss of information about the original pattern. This is achieved by combining the RT with preprocessing steps using a neighbour operator. If the neighbour operator is asymmetric, the undesired invariance under reflection can be overcome with only one preprocessing step. Thus the MRT can distinguish many more patterns from one another than the original RT can.

A new method of recognition of printed Berber characters using MRT is presented in this paper. We apply the MRT in feature extraction stage of character recognition process. In the following, some properties of the RT and MRT are first reviewed; then the new method of recognition of printed Berber characters is presented. Finally, the experimental results are given in applying of the proposed character recognition method to recognition of printed Berber characters including dependence of recognition efficiency on number of selected features.

2. MODIFIED RAPID TRANSFORM

Transform methods can be used to obtain alternative descriptions of signals. These alternative descriptions have many uses such as classification, redundancy reduction, coding, etc., because some of these tasks can be better performed in the transform domain. However, nonlinear or even noninvertible transforms can be useful for applications such as classification and pattern recognition. As most of these applications are based on the exploitation of the fact that transform is a way for changing of statistical and spectral characteristics of the signal. Transforms which do not change with cyclic shifts in the sequence are called translation invariant. Fast translation invariant transforms are valuable tool for pure shape-specific feature extraction in pattern recognition problems. The transforms may be used to extract features of one- or two-dimensional patterns, which are invariant under cyclic permutations to characterize objects independent of their position. In the field of pattern recognition and also scene analysis is well known the class of fast translation invariant transforms — certain transforms (CT) [7] based on the fast signal graph (Fig. 1) of the original rapid transform (RT) [4] but with choosing of other pairs of simple commutative operators. The RT results from a minor modification of the Walsh-Hadamard transform (WHT) [8].

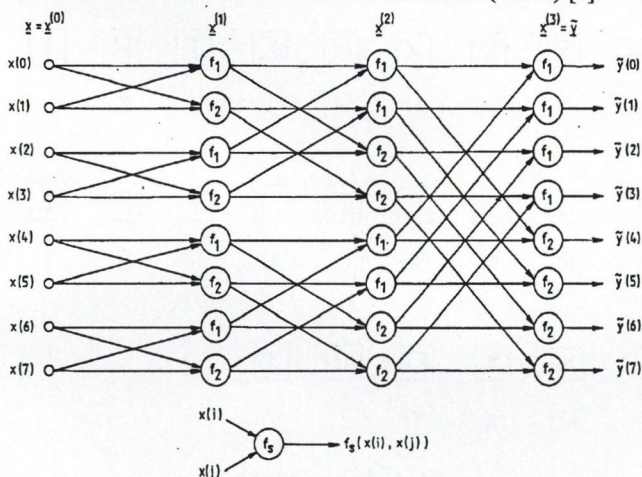


Fig. 1. Signal flow graph of the rapid transform (RT) with eight input components ($f_1(a, b) = a + b$ and $f_2(a, b) = |a - b|$)

The signal flow graph for the RT is identical to that of the WHT, except that the absolute value of the output of each stage of the iteration is taken before feeding it to the next stage. This is not an orthogonal transform, as no inverse exists. With the help of additional data, however, the signal can be recovered from the transform sequence, i.e. inverse rapid transform can be

defined [9], [10], [11]. RT has some interesting properties such as invariance to cyclic shift, reflection of the data sequence, and the slight rotation of a two-dimensional pattern. It is applicable to both binary and analogue inputs and it can be extended to multiple dimensions. Various properties of RT have been developed in [7], [12]. RT was used in the recognition of alphanumeric characters [4], [13], [16], robotics [5] and scene analysis [13], [17].

More recently was introduced the modified rapid transform (MRT) [6] which can distinguish many more patterns from one another than the original RT can. The MRT was presented to break undesired invariances of the RT which leads to a loss of information about the original pattern. This is achieved by combining the RT with preprocessing steps using a asymmetric neighbour operator α . This operator is used to break undesirable invariances but keep the shift invariance of the MRT. Using the symbolic notation we can introduce MRT as follows: Fig. 1 is a eight-point signal flow graph of the RT. The RT requires $N = 2^n$ input pixels, where n is a positive integer. Each columns in Fig. 1 corresponds to a particular computational step; n steps are required. In general the variables $x^{(r)}$ in any column (r) are calculated from variables $x^{(r-1)}$ in the preceding column ($r - 1$) by

$$\begin{aligned} x^{(r)}(i + 2js) &= f_1(x^{(r-1)}(i + 2js), x^{(r-1)}(i + (2j + 1)s)) \\ x^{(r)}(i + (2j + 1)s) &= f_2(x^{(r-1)}(i + 2js), x^{(r-1)}(i + (2j + 1)s)), \end{aligned} \quad (1)$$

where operators f_1, f_2 for RT are

$$f_1(a, b) = a + b; \quad f_2(a, b) = |a - b| \quad (2)$$

and $s = 2^{n-r}$; $t = 2^{r-1}$; $i = 0, \dots, s - 1$; $j = 1, \dots, t - 1$ and $x \equiv x^{(0)}$ are input data (pixels) and $x^{(n)} \equiv \bar{x} = RT\{x\}$ are spectral coefficients of RT. Signal graph of MRT (Fig. 2) results from signal graph of RT with adding in general k processing steps $x' = \alpha x$. This maps the element $x(i)$ of input vector x to element $x'(i)$ of vector x' by working on the elements $x(i)$, $x(i + 1)$ and $x(i + 2)$

$$x'(i) = f_0(x(i), x(i + 1), x(i + 2)) \quad (3)$$

It is important that the operator f_0 be asymmetric because we want to destroy the invariance of RT under reflection. Operator f_0 may be realized in the following simple manner [6]

$$x'(i) = f_0(x(i), x(i + 1), x(i + 2)) = x(i) + |x(i + 1) - x(i + 2)| \quad (4)$$

For 2D patterns one can use either 1D RTs in sequence, for the horizontal (x) and vertical (y) directions, or one 2D RT. Then we can choose the following symmetries of the neighbour operators:

$$f_0^x: \quad x'(i, j) = x(i, j) + |x(i + 1, j) - x(i + 2, j)| \quad (5a)$$

$$f_0^y: \quad x'(i, j) = x(i, j) + |x(i, j + 1) - x(i, j + 2)| \quad (5b)$$

$$f_0^{x+y}: \quad x'(i, j) = x(i, j) + |x(i + 1, j) - x(i + 2, j)| + |x(i, j + 1) - x(i, j + 2)| \quad (5c)$$

$$f_0^{xy}: \quad x'(i, j) = x(i, j) + |x(i + 1, j) - x(i, j + 1)|, \quad (5d)$$

where $x'(i, j)$ and $x(i, j)$, $i, j = 0, \dots, N - 1$ are pixels of the desired preprocessed and input 2D pattern (image). From theoretical considerations and computer simulations the following results was delivered:

1. The MRT utilising f_0^x , resp. f_0^y break the invariance under reflection only in one direction x , resp. y .

2. Operators f_0^{x+y} and f_0^{xy} break the invariance under reflection in both directions. Because of its simplicity the operator f_0^{xy} is superior to the operator f_0^{x+y} . In the following only the operator f_0^{xy} will be used as a neighbour operator for 2D MRT.

3. The concept of modification can be used for all transforms from the class CT [7] based on two commutative operators. The efficiency of the modification may be different for each transform.

4. Selecting a specific symmetry of the neighbour operator one can keep specific invariances and destroy undesired invariances of some transform member of the class CT.

5. MRT can be applied in all areas where the RT (or any transform from class CT) can be used. Some undesired

invariances of RT can be destroyed applying only one preprocessing step.

6. Experiments from use of MRT [6] in character recognition showed, that MRT can distinguish many more patterns from one another than the RT or the Fourier power spectrum.

7. The computational cost of the MRT is not much greater than that of the RT. In fact, one preprocessing step needs N additions and subtractions (Fig. 2), while each layer of the signal flow graph of the RT (Fig. 1) needs $N/2$ additions and subtractions. In total RT needs $N \log_2 N$ additions and subtractions, whereas the MRT with k processing steps need $N/2 [\log_2(N) + 2k]$.

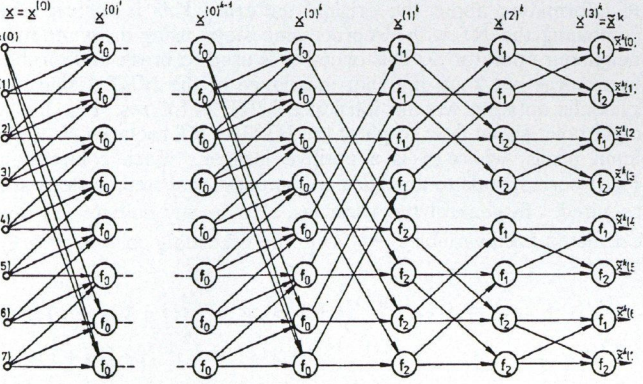


Fig. 2. Signal flow graph of the modified rapid transform (MRT) with eight input components and k preprocessing steps ($f_1(a, b) = a + b$, $f_2(a, b) = |a - b|$ and $f_0(a, b, c) = a + |b - c|$)

3. RECOGNITION SYSTEM MODEL

The recognition system (Fig. 3) is simulated in digital computer. It contains the following sub-systems:

- Original digital picture preprocessing system CSPO-II [18] was used to accept the physical input picture and then transduces it into a measurable matrix. CSPO-II divides a visual pattern into small elements and after suitable preprocessing produces an $N \times N$ matrix over the binary field; the element becomes 1 or 0 depending upon whether it is black or white.
- The MRT processor according to its function may be also called a feature extractor. A 2D RT or MRT of all binary prototypes is taken in this stage. Then feature selection is carried out in the RT or MRT "spectral" domain on various basis (maximum value of spectral coefficients, variance zonal sampling and interclass standard deviation).
- The selected MRT (or RT) features of binary pictures (characters) are in the teaching process fed into the memory. Thus the memory unit learn the a priori knowledge of each class before the system can be used to make any decision. In the recognition process the selected MRT (or RT) features are fed into the classifier, which discriminates each pattern (character) and assigns a category (a class) to it by some decision rule. We use a simple classifier based on cross responses d_{kl} between two different patterns from class k and l defined in the next section.

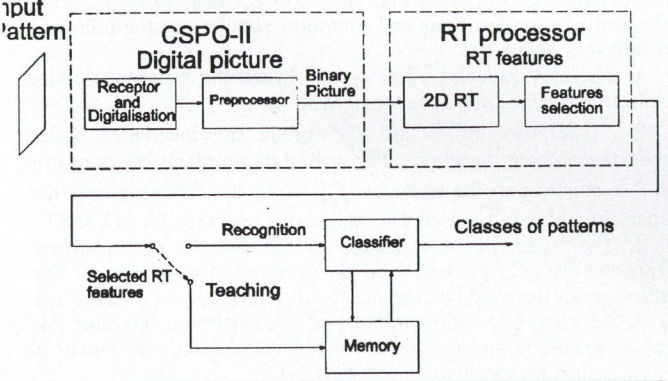


Fig. 3. Basic block scheme of the recognition system

4. RECOGNITION OF PRINTED BERBER CHARACTERS

The modern Berber alphabet [3] consists of 29 independent characters, with 25 consonants and 4 vowels (Fig. 4). Each character is separated from the others. We implemented feature extraction by RT and MRT at the set of 29 Berber characters. In general, the efficiency of feature extraction can be assessed by the system confusion matrix $D = \{d_{kl}; k, l = 1, \dots, M\}$ where d_{kl} are cross responses (or the distances between any two different characters k, l in the feature space) and M is the number of classes or number of different characters. The confusion matrix can be calculated in two steps shown as follows:

1. All $M = 29$ prototypes of Berber characters, each represented by a binary $N \times N$ matrix $(x_k(i, j))$, with $i, j = 1, \dots, N$; $k = 1, \dots, M$ and $N = 16$ are transformed to the RT or MRT transform domain

$$\tilde{x}_k(i, j) = \tau\{x_k(i, j)\}, \tag{6}$$

where $\tau \in \{RT, MRT\}$.

2. The cross response $d_{kl}^{(1)}$ between two different characters from class k and l is defined as follows:

$$d_{kl}^{(1)} = \sum_{ij=1}^N |\tilde{x}_k(i, j) - \tilde{x}_l(i, j)| \tag{7}$$

with $k, l = 1, \dots, 29$. Here, $d_{kl}^{(1)}$ is similar to as a simple distance measure between classes k and l .

To further reduce the computational cost in the classification process we selected only a couple of elements $\tilde{x}_k(i, j)$ of the feature matrix as the feature vector $F_k(p)$ (where $F_k(p) \in \{\tilde{x}_k(i, j)\}$; $p = 1, \dots, P$ and $P \ll M$, in on experiments $P = 6$). The cross response $d_{kl}^{(2)}$ can be now defined as follows:

$$d_{kl}^{(2)} = \sum_{p=1}^P |F_k(p) - F_l(p)| \tag{8}$$

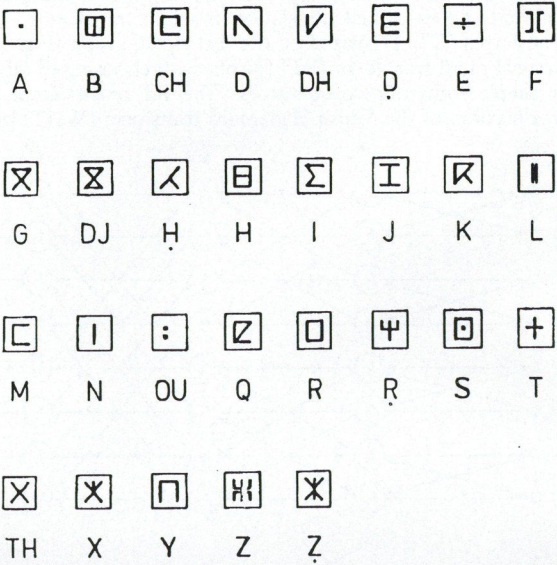


Fig. 4. Berber alphabet

The results of experiments of dependence of recognition efficiency on number of selected features and influence of noise are as follows:

1. The simple RT as feature extractor is not sufficient. There are some groups of Berber characters which cannot be separated even if we use all 16×16 features in the RT spectral domain. Difficulties occurs in separation of characters from the following groups ($\{D, DH\}$; $\{A, OU\}$; $\{CH, M, Y\}$; $\{B, H\}$; $\{L, N\}$ and $\{G, DJ, TH, X, Z\}$).

2. MRT is able to separate all the Berber characters. A real 100 % character recognition can be obtained with selecting only couple ($P = 6$) of features in the MRT spectral domain even in the presentation of some noise (up to 5 %).

5. CONCLUSION

We apply the MRT in feature extraction stage of character recognition system. Only one preprocessing step in MRT signal

REFERENCES

- [1] T. Iijima, et al.: "A theory of character recognition by pattern matching method", *Proc. of the Int. Conf. on Pattern Recognition*, Washington, DC, 1973, pp. 50-56.
- [2] C. Y. Suen: "Distinctive features in automatic recognition of handprinted characters", *Signal processing*, Vol. 4, Nos. 2, 3, 1982, pp. 193-207.
- [3] A. Oulamara, J. Duvernoy: "An application of the Hough transform to automatic recognition of Berber characters", *Signal processing*, Vol. 14, 1988, pp. 79-90.
- [4] H. Reitboeck, T. P. Brody: "A Transformation with Invariance Under Cyclic Permutation for Application in Pattern Recognition", *Inf. and Control*, Vol. 15, 1969, pp. 130-154.
- [5] J. Chmúrny, J. Turán: "Two-dimensional Fast Translation Invariant Transforms and Their Use in Robotics", *Electronic Horizon*, Vol. 15, No. 5, 1984, pp. 211-220 (in Slovak).
- [6] M. Fang, G. Häusler: "Modified Rapid Transform", *Applied Optics*, Vol. 28, No. 6, 1989, pp. 1257-1262.
- [7] M. D. Wagh, S. V. Kanetkar: "A Class of Translation Invariant Transforms", *IEEE Trans. on Acoustic, Speech and Signal Proc.*, Vol. ASSP-25, No. 3, 1977, pp. 203-205.
- [8] N. Ahmed, K. R. Rao: "Orthogonal transforms for digital signal processing", New York, Springer-Verlag, 1975.
- [9] V. Vlasenko, K. R. Rao: "Unified matrix treatment of discrete transforms", *IEEE Trans.*, Vol. C-28, 1979, pp. 934-938.
- [10] J. Turán, J. Chmúrny: "Two-dimensional Inverse R Trans-

form", *Computers and Art. Intelligence*, Vol. 2, No. 5, 1983, pp. 473-477.

- [11] M. Fang: "Class of Invertible Shift Invariant Transforms", *Signal processing*, Vol. 23, No. 4, 1991, pp. 35-44.
- [12] M. D. Wagh, S. V. Kanetkar: "A multiplexing theorem and generalization of R-transform", *Int. J. Comp. Math. Sec. A*, Vol. 5, 1975, 163-171.
- [13] H. Burkhardt: "Transformationen zur lageinvarianter Merkmalsgewinnung", *Fortschrittsbericht*, (Reihe 10, Nr. 7) VDI-Verlages, Düsseldorf, 1979.
- [14] J. Turán, J. Chmúrny: "Processors for technical realization of fast translation invariant transforms", *Computer and Artificial Intelligence*, Vol. 3, No. 6, 1984, pp. 563-572.
- [15] P. O. Wang, R. C. Schiau: "Machine recognition of printed chinese characters via transformation algorithms", *Pattern Recognition*, Vol. 5, 1973, pp. 303-321.
- [16] M. A. Nasarimhan, V. Devarajan, K. R. Rao: "Simulation of alphanumeric machine print recognition", *IEEE Trans.*, Vol. SMC-10, 1980, pp. 270-275.
- [17] H. Schütte, S. Frydrychowicz, J. Schröder: "Scene matching with translation invariant transforms", S-ICPR, Miami, USA, 1980, pp. 195-198.
- [18] J. Turán, J. Mihalík, M. Chudáček: "Digital picture preprocessing system", *Journal of Electrical Eng.*, Vol. 40, No. 9, 1989, pp. 693-698 (in Slovak).

PATTERN RECOGNITION METHOD BASED ON k NEAREST NEIGHBOUR RULE

A. JÓZWIK

INSTITUTE OF BIOCYBERNETICS AND BIOMEDICAL ENGINEERING
02-109 WARSAW, TROJENA 4.

1. THE k NEAREST NEIGHBOUR RULE

The most effective classification methods are based on the k nearest neighbour rules (k -NN). The number k is a fixed natural number not greater than a numerical force of reference set (training set). To classify the new point by the k -NN rule, it is necessary to find k nearest points in the reference set and to assign the class which is most frequently represented among the k "nearest neighbours". The number k should be selected in such a way that minimizes the misclassification probability.

2. FEATURE SELECTION PROBLEM

Some of the features used to describe objects may have no relation with the considered classification problem. They act as a noise and very often decrease the classification quality. Thus, the feature selection procedure should be included to the process of developing the classification rule. From the all features that are in our disposition, the feature set which offers minimum probability of misclassification ought to be selected.

It should be noted that the criterion for determination the optimum number k of nearest neighbours as well as the feature selection criterion are the same. Let us consider the way of it's estimation.

3. MISCLASSIFICATION RATE ESTIMATION

To estimate a probability of wrong classification from the reference set, we will use the "leave one out" method. In this method, each point from the reference set is classified by a decision rule derived from the reference set decreased by this classified point. The number of misclassified points divided by the number of points in the reference set estimates the probability of misclassification (error rate).

4. FINDING THE OPTIMUM NUMBER OF NEAREST NEIGHBOURS

The number " k " of nearest neighbours can assume the values not greater than the number " m " of objects in the reference set. In the case when $k = m$ all objects would be assigned to the class that has a maximum number of objects in the reference set. Ties can be broken by k -NN rule with $k = m - 1$ and sequentially by $k = m - 2$ and so on until $k = 1$. By use of the "leave one out" method the error rates may be calculated for all possible k -NN rules, i.e. for $k = 1, 2, \dots, m$. The value of k that corresponds to the smallest misclassification rate is the selected number k . The optimum value of " k " very rarely is equal to " m " or is close to " m ".

In that case all or nearly all objects are assigned to the class is the largest in the reference set. Thus, the classification is of poor quality. Therefore, it is sufficient to review some smaller numbers of "k". Our experience implies that it may be sufficient to take into account the values of "k" between 1 and an integer part of square root of "m".

5. FEATURE SELECTION STRATEGY

It was very easy to find the optimum number of nearest neighbours. The full review of all possible feature sets is possible only when the number of features is very small, say less than 10. For ten features, it would be necessary to review 1023 feature combinations (i.e. $2^{10} - 1$). The number of possible feature combinations grows rapidly with number of features. Because of this, the so called backward strategy is recommended. It operates in the following way. In the first step, an error rate for all "n" features is computed. Then "n" combinations by (n - 1) features is reviewed and the set of (n - 1) features is selected. In the third step, the (n - 1) combinations by (n - 2) features are reviewed and the best set of (n - 2) features is found. We stop when all combinations by single features will be evaluated. From all combinations, reviewed in the above way, we select combination which offers minimum value of error rate. Error rates computed for each reviewed feature combination refers to the optimum value "k" of nearest neighbours. When some combinations give minimum error rate, we select the one which corresponds to smaller optimum value "k".

6. DATA STANDARDIZATION PROBLEM

All features used to describe an object can be expressed in different units. These units may be of the same type or different types. The units problem could be very easily removed, for instance, by dividing each feature by its mean value. However, the different feature value units do not disturb us. The problem is whether the classification quality depends on units.

Let us assume that we use two features only. When we use the first one, we obtain certain misclassification rate greater than zero. The second feature is used to improve the quality. Furthermore, let us assume that the values of the first feature varies between 0 and 10000 while the values of the second feature varies between 0.000001 and 0.00001. The second feature will have no influence on classification quality because all distances calculated for k-NN rule will depend on the first feature even the second feature used as a single could be sufficient for perfect classification. This disadvantage can be removed by data standardization done by the following formula: $f[i, j] = (f[i, j] - f[j]) / s[j]$, where $f[i, j]$ is the value of j-th feature, $f[j]$ and $s[j]$ are its mean value and standard deviation respectively. Mean values of all features will be then equal to zero and standard deviation equal to 1. So, all features will have the same "chance". The data standardization can force the bad as well the good features. If it will force the bad feature, then such a feature will be removed during the feature selection process.

7. THE PARALLEL k-NN RULE

We can imagine the parallel net of two-decision k-NN classifiers. Thus, the classified point is simultaneously send to each of these component classifiers. The final decision may be obtained by voting of the component classifiers. The classification rule that is realized by the presented above parallel net of k-NN classifiers we will call the parallel k-NN rule. For each of the component classifier, a separate feature selection as well as the optimum number of nearest neighbours can be found. The presented work will show, on artificial data, that the parallel network of two-decision classifiers offers usually significantly better classification quality. It was noticed that this improvement is obtained mainly because a separate feature selection for each component classifier. We decided to use the artificial data set since only for such data the Bayes' risk can be known a priori and it is possible to establish which classification rule is closer to that one realized by Bayes' classifier.

8. EXPERIMENTS WITH AN ARTIFICIAL DATA SET

We consider the three class decision problem in two dimensional feature space. The first class occupies the square with vertices: [0, 0], [100, 0], [100, 100] and [0, 100]; second class: [0, 100], [100, 100], [100, 200] and [0, 200] and the third class: [0, 100], [200, 0], [200, 100] and [100, 100]. The random numbers are generated by function $rn = random(10000)$ which exists in standard version of Turbo Pascal v.6 package. The function *random* (10000) returns a random natural number from the interval [0, 10000]. Thus, the first class contains points: $x1 = rn1/100$, $x2 = rn2/100$, second class: $x1 = rn3/100$, $x2 = 100 + rn4/100$ and the third class points: $x = 100 + rn5/100$, $x2 = rn6/100$. As a distance the Euclidean measure is assumed.

The tables given below show the obtained results for two data sizes and different classifiers. The three types of classifiers were considered:

- Simple = k-NN rule (it is a priori known that feature selection is useless, so two features were used)
- Par. no f.s. = parallel k-NN rule without feature selection for all the three component classifiers.
- Par. with f.s. = parallel k-NN rule without feature selection

Table 1. Results for data set size equal to 3000 points. Each class is represented by 1000 points.

Experiment	Simple	Par., no f.s.	Par. with f.s.
1.	0.0067	0.0057	0.0000
2.	0.0030	0.0027	0.0000
3.	0.0047	0.0043	0.0000
4.	0.0017	0.0000	0.0000
5.	0.0053	0.0027	0.0000
6.	0.0043	0.0000	0.0000
7.	0.0057	0.0000	0.0000
8.	0.0030	0.0000	0.0000
9.	0.0027	0.0007	0.0003
10.	0.0043	0.0000	0.0000

Table 2. Results for data set size equal too 300 points. Each class is represented by 100 points.

Experiment	Simple	Par., no f.s.	Par. with f.s.
1.	0.0033	0.0043	0.0000
2.	0.0100	0.0033	0.0000
3.	0.0133	0.0100	0.0000
4.	0.0167	0.0100	0.0000
5.	0.0100	0.0100	0.0000
6.	0.0033	0.0000	0.0000
7.	0.0000	0.0000	0.0000
8.	0.0167	0.0133	0.0000
9.	0.0100	0.0043	0.0000
10.	0.0167	0.0100	0.0000

The optimum number k of NN for the simple classifier varied between 1900 and 2100 for the data size contained 3000 points and between 9 and 30 if each of the classes were represented by 100 points. In the case of the parallel k-NN rule the optimum numbers k of NN varied between 1 and 230 for the larger data size and between 1 and 20 for the smaller of the two considered data sizes.

9. CONCLUDING REMARKS

The results presented in the above tables show that the parallel k-NN rule can offer better performance than the simple rule. Furthermore, this difference is more significant for smaller data sets.

The parallel k-NN rule do not ensure the better result because features selection made for the component classifier does not always improve their performances.

The examples of data sets (when the parallel k -NN rule do not give better results) can be obtained from the same data sets by rotating all points in these sets by 45 degrees.

It may happen that the parallel k -NN offers a somewhat worse result than the simple k -NN rule (see table 2, experiment 1), but this does not happen frequently. Furthermore, the parallel classifier does not require more computations. The optimum

numbers k of NN are much more smaller and, because of feature selection, the distances are calculated in smaller dimensions since for pairs of classes less features are usually selected than in the case of full reference set.

The considerations presented above prove that the parallel k -NN rule is worth of recommendations.

IMAGE ANALYSIS

SEGMENTATION AND COUNTING OF OVISACS WITH MORPHOLOGICAL FILTERING IN ULTRASOUND IMAGES

I. E. PRATIKAKIS

IRIS RESEARCH GROUP
VRIJE UNIVERSITEIT BRUSSEL
PLEINLAAN 2, B-1050 BRUSSELS, BELGIUM

B. G. MERTZIOS

DEPARTMENT OF ELECTRICAL ENGINEERING
DEMOCRITUS UNIVERSITY OF THRACE
67 100 XANTHI, GREECE

J. CORNELIS

IRIS RESEARCH GROUP
VRIJE UNIVERSITEIT BRUSSEL
PLEINLAAN 2, B-1050 BRUSSELS, BELGIUM

1. INTRODUCTION

The objective of this paper is to measure the effect of oothecae stimulation in order to characterize their response to various hormones, which are given in order to provoke ovulation and enhance fertilization. These measurements are usually taken in ultrasound images by manual delineation of the ootheca and ovisacs or a simple estimation based on experience [2]. This approach suffers from subjectivity, which limits its clinical usefulness for quantitative analysis. In this paper we describe a semiautomatic technique using morphological filters for isolating the oothecae in an uterus ultrasound image, in order to take the appropriate measurements (histogram, diameters, counting). There exist related works by Thomas et al [5] to automate the measurement of femur length in fetal ultrasound images and by Klinger et al [6] to automate the segmentation of echocardiographic images. Specifically, the evaluation in the proposed technique is based on three factors:

- The amount of liquid and solid parts,
- The number of ovisacs and
- The max diameter of the ovisacs.

The overall procedure consists of three separate stages: (i) Formation, digitalization and preprocessing of images, (ii) Segmentation and (iii) Measurement.

2. THEORY: MORPHOLOGICAL OPERATORS

2.1. Binary case

Dilation, Erosion, Opening, Closing

In the theory of mathematical morphology, each signal is viewed as a set, and its geometrical features such as peaks and valleys are modified by morphologically probing the signal X with a structuring element A . The fundamental operations are [1]:

The dilation of X by A is defined as

$$X \oplus A^s = \bigcup_a X_{-a}$$

where $\bigcup$ stands for the set union with respect to all $a \in A$, $A^s = [-a : a \in A]$ is the symmetric set of A and $X_{-a} = [x - a : x \in X, a \in A]$ the translated set of X by a .

The erosion of X by A is defined as

$$X \ominus A^s = \bigcap_a X_{-a}$$

where $\bigcap$ stands for the set intersection with respect to all $a \in A$.

In addition to dilation and erosion, there are two transformations, the *opening* and *closing*. Often treated as fundamental transformations, they can be constructed directly from dilation and erosion. Mathematically, we use the following expressions

$$\text{Opening: } X \circ A = (X \ominus A^s) \oplus A$$

$$\text{Closing: } X \bullet A = (X \oplus A^s) \ominus A$$

Hit-Miss Transformation, Thinning, Pruning

The Hit-Miss (HM) transformation of X by a disjoint pair (A, B) of a structuring element is defined [1] as the set transformation $X \otimes (A, B) = (X \ominus A) \cap (X^c \ominus B)$ where A is the "hit" structuring element and B is the "miss" structuring element. By "hits" we mean intersection with the foreground of X , whereas

the "misses" will refer to intersection with the background of X , namely the set complement X^c . Thus the HM transformation is the intersection of a foreground erosion and a background erosion. It should be also noted that the elements A, B are both defined with repeat to the same center position. By definition, we thin X by (A, B) when we subtract $X \otimes (A, B)$ from X , and we thicken X by (A, B) when we add $X \otimes (A, B)$ to X : denoting these transformations by $X \square (A, B)$ and $X \blacksquare (A, B)$ respectively, we can write

$$\text{Thinning: } X \square (A, B) = X / X \otimes (A, B)$$

$$\text{Thickening: } X \blacksquare (A, B) = X \cup X \otimes (A, B)$$

Following a thinning operation we usually apply an operation which cuts the undesired parasitic branches when they appear. In our discussed problem we need the extraction of the end points and we implement this extraction by a pruning-like operation, which is a thinning operation with the sequential probing of the S-E structuring element (Fig. 1).

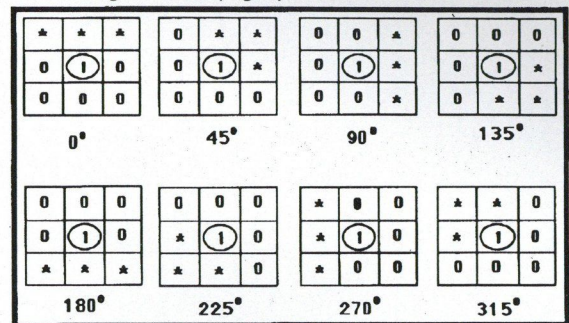


Fig. 1. S-E element rotated by 45°

2.2. Gray-level case

Morphological transformations have also been extended from binary to grayscale images by Sternberg [8] and Serra [1]. If $f(x, y)$ is a finite support graytone image function on Z^2 , and $g(x, y)$ is a fixed graytone pattern of size one that represents the structuring element.

Then the erosion of f by g is the function:

$$(f \ominus g)(x, y) = \min_{i,j} \{f(x+i, y+j) - g(i, j)\}$$

while the dilation of f by g is the function:

$$(f \oplus g)(x, y) = \max_{i,j} \{f(x-i, y-j) + g(i, j)\}$$

The opening and the closing of f by g are constructed by combining the functions of erosion and dilation, as follows:

$$\text{Opening: } X \circ g = (X \ominus g^s) \oplus g$$

$$\text{Closing: } X \bullet g = (X \oplus g^s) \ominus g$$

3. EXPERIMENTAL METHODS

3.1. Procedure of formation, digitalization and preprocessing of the images

Ultrasound images are acquired with a commercially available curved linear array, real time ultrasound system. A preprocessing

of the digital images is done before the application of the main morphological processing, in order to eliminate the degradation associated with video-tape recording of the ultrasound signal. Two main types of degradation are observed: (i) high frequency impulses (maximum gray-level) and (ii) abrupt movements of local parts of the image.

To eliminate it, we propose a type of filtering with the opening-closing (OC) morphological filter by a rhombus 3×3 structuring element to clean and smooth the blurred image [3].

3.2. The proposed segmentation algorithm

An opening operation consists of two successive operations (erosion-dilation) with a symmetric structuring element, having a particular size. In the considered case we cannot estimate the size of the structuring element. In order to face this difficulty, we propose a classical Opening Transformation with a user guided approach, which will be called *Interactive Gray-level Opening* (IGLO). This IGLO may be implemented by applying a manual erosion operation, which is used to erode the image until we are satisfied by the kind of information which has disappeared. The end of the erosion operation assigns the size of the structuring element and the beginning of the dilation, as the last part of the opening operation. The basic morphological operations are defined in the Theory part.

After the IGLO, we apply a thresholding operation by which every gray-level above zero goes to white. Although we have succeeded acceptable edge definition, the resulting contours are incomplete in some regions. Therefore, a method is required to produce a closed contour. This is achieved by applying iteratively the binary dilation, until the connection occurs among independent regions. After we have obtained the connected regions by the binary dilation, we continue the process by applying to the image a thinning operation, in order to reconstruct the image after IGLO. Then we obtain an image with connected regions. The thinning operation is applied by an S-L structuring element (Fig. 2), which is rotated each time by 45° and each rotation cycle is repeated a number of times equal to the number of iterations in the binary dilation.

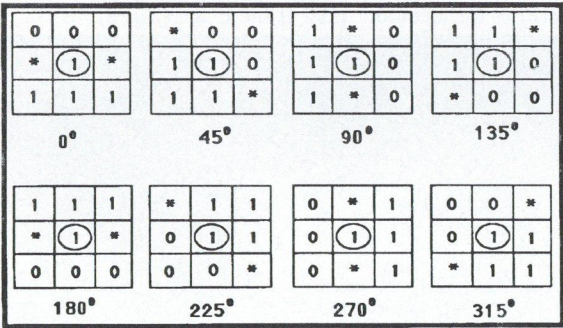


Fig. 2. S-L element rotated by 45°

The pruning operation follows the thinning operation in order to discard the parasites dendrites, which are produced by the thinning operation. We use an S-E structuring element (Fig. 1), which is rotated by 45° as many times as the thinning operation is applied.

The result from the pruning operation can be seen as a body with a cavity inside. This cavity is the *area of interest* (AROI) and we need to extract the boundary of this area. The boundary extraction consists of two sequential phases:

- In the first phase we dilate the pruned image and subtract the pruned image from its dilation. This results in two contours. Using the contour-following algorithm, which is represented in the sequel, the external contour will disappear.
- In the second phase we dilate the internal contour which remains and subtract the internal contour from its dilation. Again, an outer and inner contour result. The external contour corresponds to the boundary of the AROI. Therefore, the inner contour is removed. The contour which remains is the outline of the segmented region.

3.3. Contour-following algorithm

The underlying idea of this algorithm is based on the probing of a structuring element S-F, which operates as a controller of an area of the same size. Note that, the width of the contour is one pixel. The S-F checks the area by a logical ANDing operation between the structuring element and the elements of the respective neighborhood, which is under control. A graphical representation of our contour-following approach is shown in Fig. 3.

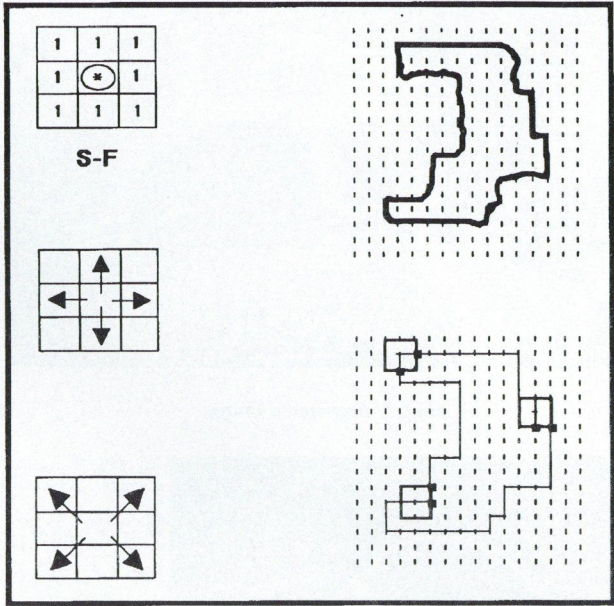


Fig. 3. Contour-following algorithm

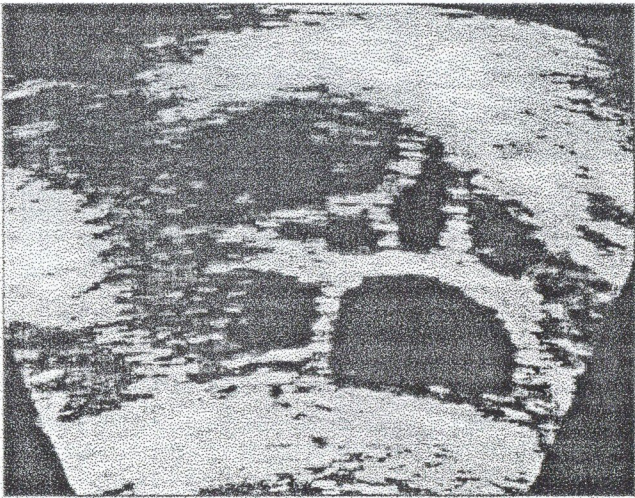


Fig. 4. Original image

4. MEASUREMENT STAGE

After we have obtained the segmented area, we are able to take the appropriate measurements (histogram, diameters, counting).

4.1. Histogram measurement to determine the amount of the liquid and the solid parts

The first measurement under consideration determines the amount of the liquid and solid parts of the ootheca. The liquid part is considered as the minimum gray-level (absolute black), while the solid parts are represented by the rest of the gray-levels. To derive an histogram we make use of simple image processing

not affect the validation of our results because in practice the physicians consider the ovisac as a convex body and for example the measurement of the maximum diameter is limited to the borders of the convex area.

REFERENCES

- [1] J. Serra: *Image Analysis and Mathematical Morphology*, Vol. 1, New York Academic Press, 1982.
- [2] E. Asimakopoulos, S. Mantalenakis: *Ultrasonography in Obstetrics and Gynecology*, University Studio Press, Thessaloniki, 1989.
- [3] I. Pitas, A. N. Venetsanopoulos: *Nonlinear Digital Filters, Principles and Applications*, Kluwer Academic Publishers, 1990.
- [4] L. Vincent, S. Beucher: *The morphological approach to segmentation: An introduction*, School of Mines, Paris, France, Internal Rep. CMM, July 1989.
- [5] J. G. Thomas, R. A. Peters, P. Jeanty: "Automatic segmentation of ultrasound images using morphological operators",

Further investigation of the present methodology is required to lead to a fully automatic implementation in order to be applied in the daily clinical routine. Also, this method can be useful as a low-level processing stage in an expert system for gynaecologists.

- IEEE Transactions on Medical Imaging*, Vol. MI-10, No. 2, pp. 180-186, June 1991.
- [6] J. W. Klinger, C. L. Vaughan, T. D. Fraker, L. T. Andrews: "Segmentation of echocardiographic images using mathematical morphology", *IEEE Transactions on Biomedical Engineering*, Vol. 35, No. 11, pp. 925-934, Nov. 1988.
- [7] R. M. Haralick, S. R. Sternberg, X. Zhuang: "Image analysis using mathematical morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 4, pp. 532-550, July 1987.
- [8] S. R. Sternberg: "Biological image processing", *Computer*, Vol. 16, No. 1, Jan. 1983.

ESTIMATION OF EDGE PARAMETERS AND IMAGE BLUR FROM LOCAL DERIVATIVES

V. KAYARGADDE and J.-B. MARTENS

INSTITUTE FOR PERCEPTION RESEARCH
P. O. BOX. 513, 5600 MB EINDHOVEN, THE NETHERLANDS

1. INTRODUCTION

An edge is completely characterized by its position, foreground value, background value, orientation and edge profile. Reliable estimates of edge parameters are essential for many applications in computer vision. Estimates of edge parameters also have applications in image coding, such as coding using oriented edges [1], coding using the local dimensionality of the image [7], etc. Many authors have developed methods to estimate the parameters of step edges [2], [3]. However, edges that occur in real images are blurred. There are many applications that need an estimate of edge blur or image blur parameters: image deblurring, depth estimation from camera defocus [4]. We developed an estimation algorithm as part of a study into objective measures for image quality. Previous work on estimation of edge blur parameters has been mainly aimed at depth estimation [4], [5]. The main disadvantage of those methods is that they require an accurate and separate estimate of at least another edge parameter, either the edge orientation or the position.

To be able to effectively describe and use edge features it is essential to estimate all parameters of the edge. In this paper, a method is presented for simultaneously estimating all parameters of a step edge that is blurred by a Gaussian kernel. The algorithm uses polynomial transforms. A polynomial transform is a local image description technique [6] that incorporates many properties of the early stages of the human visual system.

In this paper, a brief review of polynomial transforms is presented and the polynomial coefficients of a blurred edge are derived. We show how the parameters of a blurred edge can be estimated from the polynomial coefficients. The problem of blur parameter estimation is studied further. An algorithm to estimate the spread of a Gaussian image blurring kernel is presented. This algorithm combines the estimates of edge blur parameters at different edge locations in the image. Finally, some results of image blur estimation are presented.

2. POLYNOMIAL TRANSFORMS

Image analysis using polynomial transforms involves two steps. First, the image is localized by multiplying with a window function $w(z, y)$. This windowing takes place at several positions over the entire image. Second, the image within every window is described as a sum of weighted polynomials. The polynomials that are orthogonal with respect to the window function are used as the

basis functions for the polynomial expansion [6]. For example, when the Gaussian window is used, the Hermite polynomials are used for the expansion. The mapping from the input image to the coefficients of the polynomials, referred to as polynomial coefficients, is called a forward polynomial transform.

The forward polynomial transform of an image can be interpreted as a multirate filterbank with filters $a_{m,n-m}(-x, -y)$ [6]. The analysis functions are given by $a_{m,n-m}(x, y) = \varphi_{m,n-m}(x, y) \cdot w^2(x, y)$, where $\varphi_{m,n-m}(x, y)$ are the bases polynomials orthogonal with respect to $w^2(x, y)$. We use a Gaussian window function for polynomial analysis, mainly because the resulting analysis functions have many properties in common with the receptive field profiles found in the human visual system. An important property being that the analysis functions are equal to derivatives of a Gaussian [6]. Hence the n th order polynomial coefficient is equal to the n th order local derivative. A Gaussian window $w(x, y) = (1/\sqrt{\pi}\sigma) \exp(-(x^2 + y^2)/2\sigma^2)$ is specified by its spread parameter σ .

3. POLYNOMIAL COEFFICIENTS OF AN EDGE

Gaussian blur is one the most commonly encountered descriptions of blur; for example, optical blur is approximately Gaussian [4]. We use the following (Gaussian) edge model

$$\begin{aligned} \text{edge}(x, y; L_e, \Delta L, \sigma_b, d, \theta) = \\ = L_e + \frac{\Delta L}{2} \text{erf} \left[\frac{x \cos \theta + y \sin \theta - d}{\sigma_b} \right], \end{aligned} \quad (1)$$

for a blurred edge with mean value L_e , height ΔL , blur parameter σ_b , distance from the origin (i.e., the center of the window $w(x, y)$) d , and orientation θ . In the limit $\sigma_b \rightarrow 0$, the above model reduces to a steps edge.

The relation between the polynomial coefficients of a blurred and unblurred image can be easily derived using the expressions for the Fourier transform of the analysis functions. Using that relation, the polynomial coefficients of a blurred edge are derived from those of the unblurred (step) edge [8]. The polynomial coefficient $f_{m,n-m}$ of order n (order m in x and order $n-m$ in y) of a blurred edge is given by

$$f_{m,n-m} = f_n \sqrt{\frac{n!}{m!(n-m)!}} \cos^m \theta \sin^{n-m} \theta, \quad (2)$$

where f_n is a function of L_e , ΔL , d , σ_b and the spread of the Gaussian window σ . The f_n can be obtained from the polynomial coefficients of order n . The coefficients f_n up to order 3 for an edge of $\Delta L = 100$, $\sigma_b = 0.5$ as a function of d are shown in Fig. 1.

4. ESTIMATION OF EDGE PARAMETERS

Polynomial coefficients (i.e. derivatives) up to order three are required to solve for all the five edge parameters: L_e , ΔL , d , θ and σ_b . Using the expressions for the polynomial coefficients of the blurred edge given in Eq. (2), the following estimates for the edge parameters are obtained

$$\left(\frac{\sigma_b}{\sigma}\right)^2 = \left(\frac{2f_2^2}{f_1^2} - \frac{\sqrt{6}f_3}{f_1}\right)^{-1} - 1 \quad (3)$$

$$\frac{d}{\sigma} = \frac{f_2}{f_1} \left(\frac{2f_2^2}{f_1^2} - \frac{\sqrt{6}f_3}{f_1}\right)^{-1} \quad (4)$$

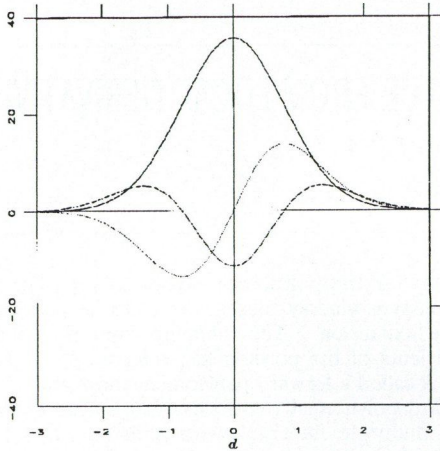


Fig. 1. The coefficients f_1 (solid), f_2 (dotted) and f_3 (dashed) of an edge of $\Delta L = 100$ and $\sigma_b = 0.5$. The spread of analysis window $\sigma = 1.0$

$$\Delta L = f_1 \sqrt{2\pi} \left(\frac{2f_2^2}{f_1^2} - \frac{\sqrt{6}f_3}{f_1}\right)^{-(1/2)} \exp \left[\frac{f_2^2}{f_1^2} \left(\frac{2f_2^2}{f_1^2} - \frac{\sqrt{6}f_3}{f_1}\right)^{-1} \right] \quad (5)$$

The mean value L_e is obtained by substituting the above parameter values in $f_{0,0}$. The edge orientation θ can be obtained from the ratio $f_{0,1}/f_{1,0} = \tan \theta$.

The effect of additive noise on the estimates of edge parameters has been studied using simulations. The estimate of θ is unbiased. Simulations show that the estimates of ΔL and d do not have much bias even at low SNR. The estimate of σ_b is unbiased at high SNRs but becomes increasingly biased at low SNRs. It was found that the bias in the σ_b^2 estimate is proportional to $(\sigma_n/f_1)^2$, where σ_n is the input noise variance [8]. The bias can be corrected if the noise variance is known.

5. BLUR ESTIMATION ALGORITHM

The blur estimation algorithm contains two steps: 1) detection of locally one-dimensional (1D) edge regions and 2) estimation of σ_b at each of the detected locations. Locally 1D edge regions are detected by: 1) locating local extrema in the first derivative image, 2) discarding edge points that are not locally 1D, by thresholding the 2D energy, and 3) thresholding f_1 . The 2D energy indicates the extent to which a pattern is locally two dimensional.

For example e_{2D} up to order two is given by $e_{2D} = 0.5([(f_{2,0} - f_{0,2})^2 + 2f_{1,1}^2]^{0.5} - |f_{2,0} + f_{0,2}|)$. Since reliable estimates of σ_b are provided by edges with high f_1 , reliable edges are selected from the image by thresholding f_1 . At each detected edge location σ_b is estimated using Eq. (3).

When the blur is uniform over the image (i.e., σ_b is space invariant), the estimates of σ_b at all edge locations are combined to obtain an estimate of the image blur. We assume that *there are at least locations in the undistorted image where the image is locally a one-dimensional edge*. This assumption is true for most natural scenes. If the noise variance is constant over the image, then the estimate $\hat{\sigma}_b^2$ and the bias factor K can be derived simultaneously by minimizing the weighted error

$$\min_{\hat{\sigma}_b^2, K} \sum_i \left[\hat{\sigma}_b^2 - \left(r_i + \frac{K}{f_1^2} \right) \right]^2 w_i, \quad (6)$$

over all edge locations, where r_i is the raw estimate of σ_b^2 at edge location i and w_i expresses the confidence in this estimate. The current implementation has $w_i = (f_1)_i$.

When using a single-scale algorithm described above, the window size σ has to be chosen *a priori*. Using the single-scale algorithm (at one value of σ), reliable estimates of σ_b can be made only within a limited range for the parameter σ_b . To increase this range, the image can be analyzed with multiple window sizes and the appropriate window size can be chosen *a posteriori*. The single-scale algorithm can be easily extended to a multi-scale algorithm [8].

6. RESULTS

The blur estimation algorithm has been tested by applying it to natural and synthetic images with different amounts of blur. Images with different amounts of blur were generated from the 'original image' by convolving with filters of known σ_b . Results of applying the algorithm on two different scenes (a synthetic image: RAND, an image of randomly positioned rectangular and triangular blocks of different gray-levels, and a natural image: LENA or the lady with a hat) are shown in Fig. 2. In both the plots, the first 6 estimates were obtained using a window of $\sigma = 2.0$ and the next 7 estimates using a window of $\sigma = 2.83$.

The dotted line in Fig. 2a corresponds to $\hat{\sigma}_b = \sigma$. The estimates show that the original image LENA is already blurred and has σ_b of about 0.87 pixels. The dotted line in Fig. 2b corresponds to the expected total blur of the image, assuming that the blur in the original image is Gaussian with $\sigma_b = 0.87$.

The total blur is given by $\sigma_b = \sqrt{\sigma_{b0}^2 + \sigma_{bi}^2}$ where σ_{b0} is the blur of the original image and σ_{bi} is the additional input blur. The length of the errorbars equals twice the standard error of the mean. The standard errors of the estimates for LENA are higher than those for the synthetic image, mainly because the natural image contains edges that have undergone different amounts of blur. The results in Fig. 2 show that the blur estimation algorithm is sensitive to even small amounts of blur (i.e., subpixel accuracy) and gives a reliable estimate of blur over a wide range.

The blur estimation algorithm has also been tested by applying it to images with noise. Images with different amounts of blur and noise were generated from the blurred images by adding zero-mean Gaussian distributed noise of known σ_n . Results of applying the algorithm on the same two scenes are shown in Fig. 3. All estimates are obtained by using a polynomial transform with window spread $\sigma = 2$. The length of the errorbars equals twice the standard error of the mean. Points joined by a line correspond to images that have the same amount of blur, but contain different amounts of input noise. The parallel curves demonstrate that the algorithm gives reliable estimates of blur parameter even at low SNRs.

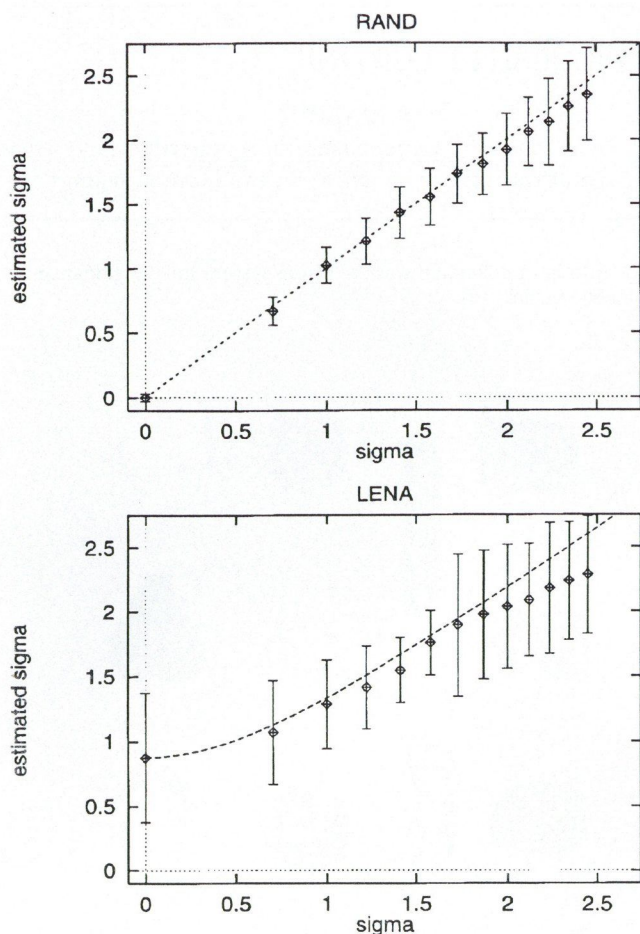


Fig. 2. Results of image blur estimation for a) a synthetic image RAND and b) a natural image LENA.

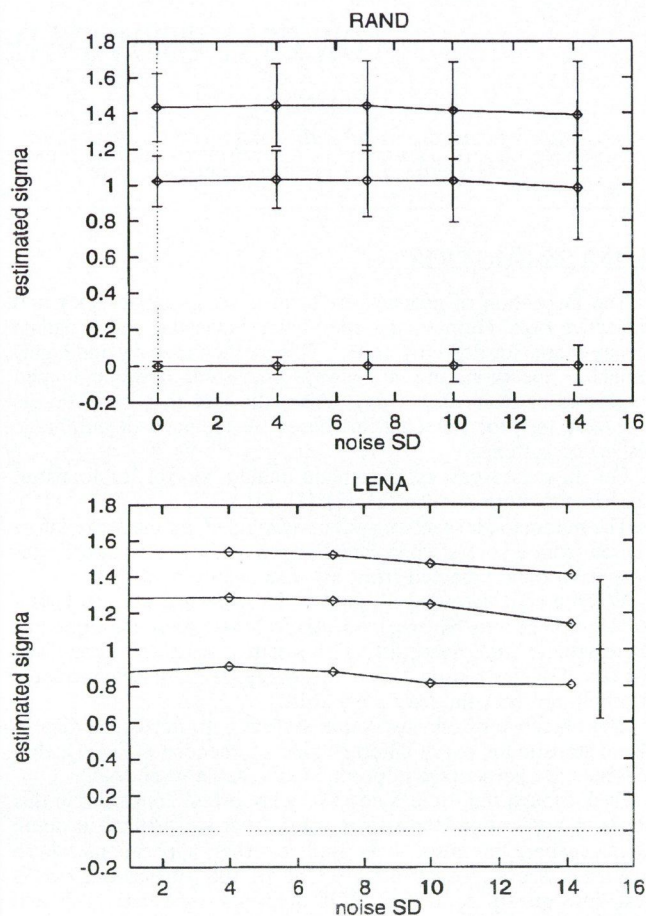


Fig. 3. Results of image blur estimation in presence of noise

REFERENCES

- [1] M. Kunt, A. Ikonopoulou, M. Kocher: "Second-generation image-coding techniques", *Proc. IEEE* 73, 1985. pp. 549-574.
- [2] E. Lyvers, O. Mitchell, M. Akey, A. Reeves: "Subpixel measurements using a moment-based edge operator", *IEEE Trans. Pattern Anal. Machine Intell.* 11, 1989. pp. 1293-1308.
- [3] J. Chen, G. Medioni: "Detection, localization and estimation of edges", *IEEE Trans. Pattern Anal. Machine Intell.* 11, 1989. pp. 191-198.
- [4] A. Pentland: "A new sense for depth of field", *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-9, 1987. pp. 523-531.
- [5] S. Lai, C. Fu, S. Chang: "A generalized depth estimation

7. CONCLUSIONS

A new method for estimating important parameters of a blurred edge has been presented. The estimation method is based on the local derivatives of the image. It has been shown that derivatives up to order 3 are needed to estimate all parameters of a blurred edge: the position, height, background value, orientation and spread of the blurring kernel. An algorithm to estimate image blur based on the edge blur estimates has also been presented. The algorithm provides reliable estimates of the spread of the Gaussian blurring kernel even at low SNRs when it is applied to natural images with different amounts of blur and noise.

- algorithm with a single image", *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-14, 1992. pp. 405-411.
- [6] J. B. Martens: "The Hermite transform-theory", *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-38, 1990, 1595-1606.
- [7] J. B. Martens: "The Hermite transform-applications", *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-38, 1990. pp. 1607-1618.
- [8] V. Kayargadde, J. B. Martens: "Estimation of edge parameters and image blur using polynomial transforms", Submitted to: *CVGIP: Graphical Models and Image Processing*, 1993.

LOW COST VISION SYSTEM FOR QUALITY CONTROL

J. CLIMENT

AUTOMATIC CONTROL AND COMPUTER ENGINEERING
DEPARTEMENT (ESAIL), UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)
PAU GARGALLO, 5 08028, BARCELONA, SPAIN

S. ALVAREZ

COMPUTER ENGINEERING DEPARTMENT
UNIVERSITAT ROVIRA I VIRGILI
CARRETERA DE SALOU, S/N 43006, TARRAGONA, SPAIN

1. INTRODUCTION

The inspection of products by hand is an intensive labor and subjective task. Human inspection involves viewing and assigning quality scores for defective terms. This method is costly and highly variable. Decisions are not always consistent between human inspectors or from day to day. Also, the fact that inspection is a tedious task for personal, increases the demand of automatic inspection systems.

For all these reasons, automatic quality control for foodstuff has become more popular [1], [2], [3], [4].

The automatic inspection system developed, extracts size information from a sort of chocolates. A study about standard deviations from their expected sizes, has also been carried out.

When a chocolate doesn't present its expected size attributes, it does not fit into his reserved plastic hole. As a consequence, the customer finds loose-fitting chocolates, scattered about into the box. For this reason, the company rejects all these chocolates which do not pass the fixed standards.

The objective of our inspection system is to develop a dimensional statistic for every different kind of chocolate (Fig. 1), that indicates the percentage of production suitable to be sold.

Even though real-time is not a very important constraint in this concrete application, the vision system that we present in detail in this paper, has also been used in other applications where real time was a critical factor. One of the applications was a real time quality control of RGB television screens. The aim of the application was to detect defects on the color phosphorus emulsions, due to malformations during the fabrication process

[5]. Another application was the vision system for the guidance of a mobile vehicle [6].

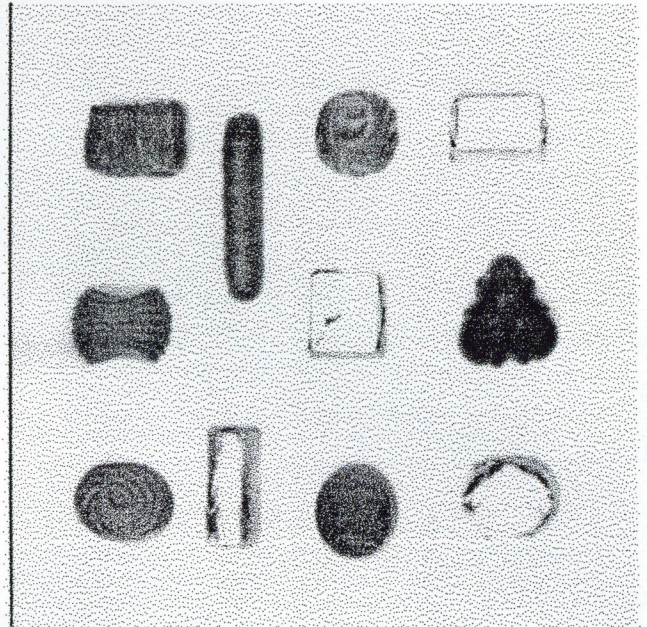


Fig. 1. Shapes of different types of chocolates

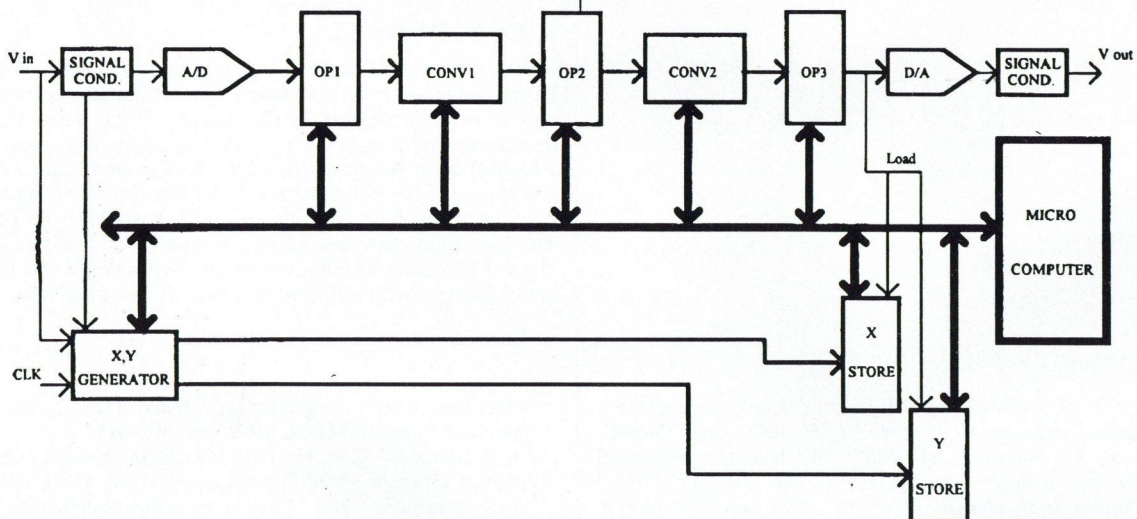


Fig. 2. Vision System Architecture

2. ARCHITECTURE OF THE VISION SYSTEM

The vision system used consists of the following parts (Fig. 2):

- Signal conditioning and digitalization of input video signal.
- Two cascaded stages of digital signal processors. They perform multilevel convolutions with 22 bits of resolution. These processors work in real time.
- Three operation modules.
- Conversion and conditioning of output video signal.

- Generation of X, Y coordinates.
- Storage of X, Y coordinates.
- Microcomputer.

Next, we make a brief description of this pipe-lined architecture:

Input video signal is conditioned to be digitized with 8 bits of resolution. Digital video is re-histogramed using OP1. Next, the first convolution stage (CONV1) performs the programmed con-

volutions on the data coming from OP1, using the programmed operators. With the results so obtained, more complex operations can be performed in OP2. This result is convoluted again by the second stage (CONV2), this means that a second convolution is performed on the results coming from the first one. This output can be also transformed using the OP3 stage. All these operations are performed at video-rate.

If the data coming out from CONV2 satisfies a desired condition, the module OP3 is programmed to set the Load signal. This means that an expected local feature is present, so, the coordinates of the current pixel are stored in the X and Y STORES.

All this process is developed inside an attention window. The size and situation of this window are programmable. Every time that the result of a new pixel is obtained from the pipe-line, its coordinates are actualized.

A description of different subsystems is now presented:

INPUT STAGE: It consists on the signal conditioning, the A/D conversion, and the extraction of Vsync and Hsync. It is useful to condition the input signal to achieve the highest resolution when it is digitized. A flash A/D converter is used.

OP1: Re-histograms the digitized image to enhance the system's performance. The gray-levels of the input data are modified according to the programmed transformation. It has been implemented with a fast dual-ported memory.

FIRST CONVOLUTION: Performs the extraction of the edges of the image. Module OP2 is used to operate the two components of the gradient obtained by CONV1. The input is the multilevel image of the current chocolate. The output is the binary contour image [7], [8].

SECOND CONVOLUTION: Performs the thinning of the edges obtained by preceding stage. The use of CONV2 together with the module OP3, allows to implement morphologic operations, such as skeletonization. Load signal is activated when a contour pixel is present.

X, Y STORE: These are high speed 9-bit wide memories. Each memory is organized such that the data is read in the same sequential order that it was written (FIFO). Stores the edge coordinates. The rate at which the microprocessor reads these coordinates is 2000 pairs of coordinates per frame.

X, Y GENERATOR: This module continually gives the x, y coordinates of the pixel being inspected. Each coordinate is represented by two words of 9 bits. These coordinates are always relative to the attention window. The attention window is the part of the image where the inspection will take place. The size and location of this window are programmable. This window is generated from the synchronisms obtained in the input stage, and the pixel clock.

Out of the working window, all operations in the pipe-line are inhibited. At the end of the window, an interruption signal is generated indicating that all the stored information can be read.

OUTPUT STAGE: It consists of the conversion to analog signal and the conditioning to its monitorization. The Load signal can also be overlapped on the outgoing image.

MICRO COMPUTER: Programs the different convolvers and operation memories, and controls the data path. When all the contour coordinates are stored in the X-Y memories, the computer reads all this data and empties the FIFOs. Then, using the developed algorithms, coordinates are examined for extraction of shape characteristics.

3. INSPECTION PROCEDURE

The complete inspection scene consists of the figure of the chocolate, and some calibration marks (Fig. 3). These marks are required for the periodical calibration of the system. Calibration marks do not affect the vision procedure, thus the attention window is positioned just over the chocolate. So, the vision system only processes the image of the product, but changes the size and position of the attention window every time it needs feedback for calibration.

The first restriction we must look at, is that we can register 2000 pairs of coordinates per frame as a maximum rate. This means that the amount of information to be stored has to be

optimized. We want to reduce the number of pixel coordinates to be processed, without losing the main characteristics of the chocolate. For this reason the edges of the image are extracted and thinned afterwards.

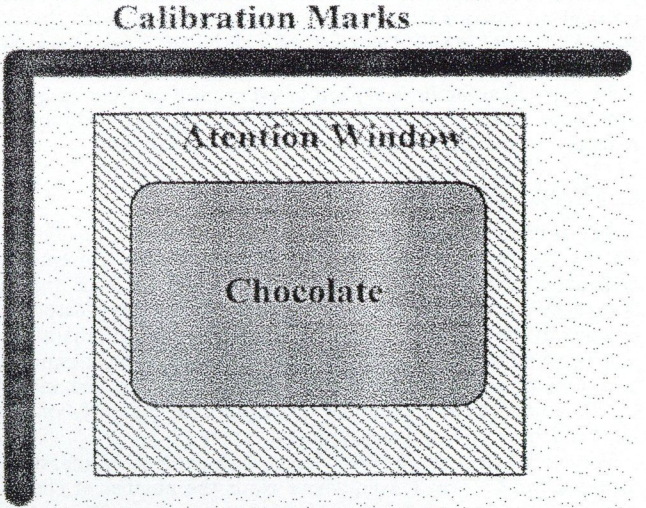


Fig. 3. Inspection Scene

The edges are obtained using a Sobel classic algorithm. Other algorithms could also be used [9], [10], but their use would involve a very costly implementation. Furthermore, the chocolate is so contrasted, that using a simpler method [11] the results of edge extraction are optimum.

Once the edges have been obtained, they must be thinned, because each edge pixel means a new storage of coordinates in the X, Y STORAGE module. The thinning process reduces the amount of information to be stored. Besides, in order to apply further algorithms for extracting perimeters and relevant information about the product shapes, one-pixel contours are indispensable.

The deletion or retention of a pixel p depends on the configuration of the pixels in a local neighborhood containing p . To ensure connectedness, the pixel p is deleted if its removal does not change the 8-connectivity of the contour image [12]. Image points are examined for deletion in a raster scan order. It is understood that a pixel p examined for deletion is a white pixel, and the pixels in its 3×3 window are labeled as shown in Fig. 4.

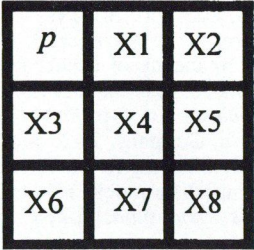


Fig. 4. Local neighborhood containing p

A sequence of points $P_1, P_2, \dots, P_N$ is called an 8-path if P_{i+1} is an 8-neighbor of P_i , $i = 1, 2, \dots, N - 1$ [13]. The pixel p considered for deletion is cleared (set to black) if:

1. p is a white pixel.
2. There exists a white 8-path from one of the rest of the pixels of the west column (X_3, X_6) to one of the rest of the pixels to the north row (X_1, X_2).

When the inspection process reaches the end of the attention window, the microprocessor reads the coordinates of the contour of the chocolate, and empties the X, Y STORAGE module. With this amount of data, calculates the following features : area, perimeter, area/perimeter², radius of the inscript circumference, radius of the circumscript circumference, and the difference between these two radii. These six parameters update the data base corresponding to the concrete type of chocolate.

The algorithm developed to extract desired features from the contour coordinates is based on a Chain Code structure [14]. Coordinates are examined in a contour following order. The perimeter is incremented by one every time that a new coordinate is examined. Area is incremented or decremented by the value of the y coordinate, depending on if the contour is followed east or west direction. Even though we are working with square pixels, a width correction factor is applied to every pixel in order to avoid distortions. As our grid is 8-connected, another correction factor is applied to the pixels belonging to a diagonal segment.

4. FINAL RESULTS

Let us denote the measurement vector as $C = (C_1, C_2, C_3, C_4, C_5, C_6)$, where $C_1..C_6$ are the six mentioned measurements of desired features stored in the database. Every time that a new chocolate of a concrete type is measured, a new vector C_j is added to the data base. Since our hardware works in real time, time between two consecutive inspections is large enough to make several measures of the same piece of chocolate. So, the vector C_j is obtained from an average of 10 different measures of the same chocolate to avoid measurement errors.

For every different type of chocolate, we get a set of measurement vectors $C_j (j = 1..N_i)$, where N_i is the number of chocolates inspected of type $i (i = 1..11)$. A mean vector $\bar{X}(i)$ and a deviation vector $\sigma(i)$ are obtained for every kind of chocolate.

The mean resolution the system works with is 0.25 mm/pixel. Fig. 5 shows probability density functions of two different parameters for two classes of chocolate. The results are shown for the circle shaped and the clover shaped chocolates. The parameters

shown are $4 \cdot \pi \cdot \text{area}/\text{perimeter}^2$, and the difference between maximum and minimum radii. The first parameter gives us an idea about the circularity of the chocolate, and the second one gives an idea about the lobes in its contour.

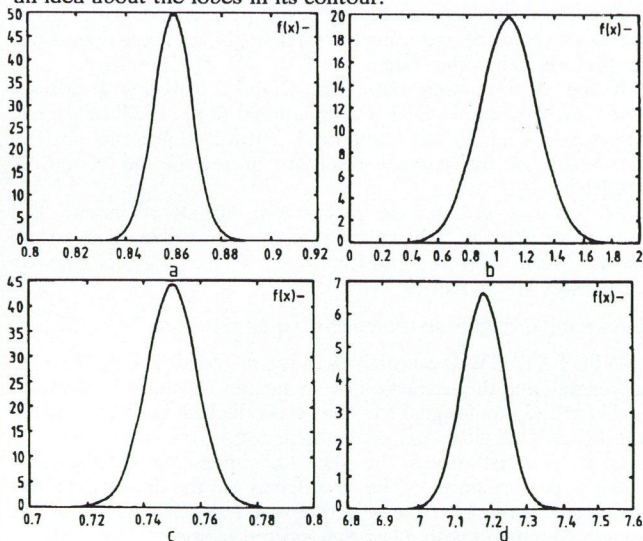


Fig. 5 a, b circularity, and radii difference of the circular shaped chocolate. c, d show the same features for the clover shaped

REFERENCES

- [1] F. E. Dowell: "Automatic Inspection of peanut grade samples", *Food Processing Automation, Proc. of the 1990 Conference*, St Joseph, Michigan, USA. pp. 275-281.
- [2] C. T. Morrow, P. H. Heinemann, H. J. Sommer, Yang Tao, Z. Varghese: "Automated Inspection of potatoes, apples and mushrooms", *Proc. of the 1st. International workshop on robotics in agriculture & the food industry*, Avignon, 14-15. June, 1990.
- [3] G. E. Rehkugler, J. A. Throop: "Apple sorting with machine vision", *Trans. of the ASAE*, Vol. 29. 1985.
- [4] M. J. Delwiche, S. Tang, J. F. Thompson: "A high speed multiprocessor inspection system for dried prunes", *Food Processing Automation, Proc. of the 1990 Conference*, St Joseph, Michigan, USA. pp. 281-289.
- [5] Ll. Vila, C. Sierra, A. B. Martinez, J. Climent: "Intelligent Process Control by means of expert systems and machine vision", *Proc. of the 5th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems*, Paderborn, Germany, 9-12. June, 1992.
- [6] A. B. Martinez, J. Climent, R. M. Planas, J. M. Asensio: "Vision Based Compass for mobile robots", *Proc. of the IEEE-Intelligent Vehicles Symposium*, Tokyo, Japan, 14-16. July, 1993.
- [7] J. S. Wezskas: "A Survey of threshold selection techniques", *Computer Graphics and Image Processing*, Vol. 7. 1978. pp. 259-265.
- [8] Y. Nakagawa, A. Rosenfeld: "Some experiments on variable thresholding", *Pattern Recognition*, Vol. 11. 1979. pp. 191-204.
- [9] J. F. Canny: "A Computational Approach to Edge Detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8. No. 6. Nov. 1986. pp. 679-698.
- [10] D. C. Marr, Hildreth: "Theory of Edge Detection", *Proc. of the Royal Society of London*, series B, Vol. 207. 1980. pp. 187-217.
- [11] J. S. Lim: *Two-Dimensional Signal and Image Processing*, Prentice-Hall, Inc. 1990.
- [12] A. Rosenfeld: "Connectivity in digital pictures", *J. ACM*, Vol. 17. No. 1. pp. 146-160. 1970.
- [13] L. Lam, S. W. Lee, C. Y. Suen: "Thinning Methodologies-A Comprehensive Survey", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-14. No. 9. Sept., 1992. pp. 869-885.
- [14] H. Freeman: "Computer processing of line drawing images", *Computer Surveys*, March, 1974. pp. 57-98.

CROWDING EVALUATION BY MEANS OF MULTI-HYPOTHESIS MODELLING

C. S. REGAZZONI and A. TESEI

DEPARTMENT OF BIOPHYSICAL AND ELECTRONIC ENGINEERING (DIBE), UNIVERSITY OF GENOA
VIA ALL'OPERA PIA 11A, I-16145 GENOVA, ITALY

1. INTRODUCTION

Crowding evaluation in complex environments is a problem currently addressed in the research field of surveillance systems; it is useful for two main purposes:

a) detection of overcrowded situations;

b) statistical temporal evaluation of the number of people to plan traffic activities.

The specific application considered in this paper is crowding estimation in an underground station.

The system has the following characteristics:

1) *Multisensor input*: images are acquired by using several physical optical sensors, and then processed by different virtual sensors (i.e., nodes that apply feature-extraction algorithms).

The following features have been selected [1] in order to minimize computational load and to obtain estimation accuracy :

- number of edge points;
- number of edge-point histogram maxima;
- sum of the amplitudes of edge-point histogram maxima;
- number of edge points / rectangle y-dim;
- rectangle x-dim.

Feature values are extracted from each rectangular area occupied by people and localized in the image plane by means of change detection and focus of attention algorithms.

2) *Distributed Extended Kalman Filter (DEKF) system*: crowding estimation is performed according to a distributed philosophy by a hierarchical network of nodes (virtual sensors), each of which is locally devoted to people evaluation on the basis of static (current) and temporal (previous) information. Each node is implemented as a Kalman Filter having the number of people as the local status variable to estimate and predict.

3) *Multiple-model approach*: different models of the virtual sensors are locally used in accordance with the positions and dimensions of the rectangular areas selected by the focus-of-attention algorithm.

2. CROWDING-ESTIMATION SYSTEM DESCRIPTION

From experimental statistical analysis [1], [5] it is possible to deduce that in public environments, such as railway or underground stations, stadiums, supermarkets, etc., the number of people present in a specific monitored area is temporally correlated (at average, it significantly changes only after about 20 seconds). In order to employ this temporal correlation in estimating the number of people present in an image of the monitored zone, an inference system based on Kalman Filtering has been designed.

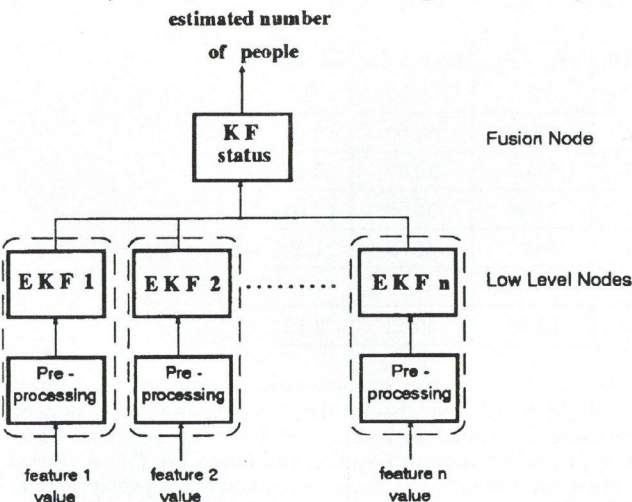


Fig. 1. Fragment of DEKF architecture for a single camera

Each network node is ruled by the following dynamic system

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k)) + \mathbf{v}(k)$$

$$y(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{w}(k),$$

where $\mathbf{x}$ is the status vector, y is the observed variable, $\mathbf{v}$ and $\mathbf{w}$ are Gaussian white noise having null mean value and known variance, $\mathbf{h}(\cdot)$ is a linear function and $\mathbf{f}(\cdot)$ is the transition function..

The basic scheme of the system architecture for a single camera is presented in Fig. 1. The lowest hierarchical level presents n Extended Kalman Filters ($n = 5$ in the present application), one for each feature type extracted from an acquired image. At this level y is the number of people associated to the extracted feature value by the pre-processing block; $\mathbf{f}(\cdot)$ is nonlinear (so that the filters are EKF), having the following

expression in order to model the behaviour of the number of people p (increasing, decreasing and being steady), as time varies:

$$f(x) = p^*(a - b \cdot \exp(-c \cdot p)), \text{ where } x = [p, a, b, c].$$

The EKF local estimates are propagated towards the upper level where a fusion node provides the resulting estimate of the people number present in the scene area monitored by the correspondent camera. Since at this level $f(\cdot)$ is linear, then the node is a standard KF and the status variable is scalar ($x = p$). The multi-model approach is at the basis of the pre-processing mechanism for mapping the feature values with the correlated number of people.

3. VIRTUAL SENSOR MODELLING

Virtual sensor modelling consists in creating a mapping between the feature values and the number of people present in a given rectangle.

The innovative idea presented in this paper is to stress the importance of the position and dimensions of the rectangle considered. A training set of images are processed by Low-Level algorithms (based on the Sobel filter) in order to create edge-images. A Focus-of-Attention algorithm (based on the application of a change-detection algorithm to the current image versus the background image) detects minimum bounding rectangles of occupied areas. Each rectangle is characterized by four parameters: the horizontal and vertical dimensions and the two co-ordinates of the center of gravity. The selected features are extracted from each rectangular area of the edge-images, and their measures are saved during a training phase.

A simple clustering (i.e., ISODATA [2]) of the rectangular areas is performed by using the four parameters previously computed and saved. At the end of this phase, we can classify each rectangle by associating it with a cluster.

Finally, a model for each cluster of rectangles is computed by means of Dynamic Programming (Bellman Principle [3]). Model building is considered as the problem of searching for the optimal trajectory between two fixed points (initial and final points of the model), in a piece-wise linear way, in order to minimize a suitable quadratic cost [4]. The cost function is defined in such a way that it penalizes the distance of the optimal current tracking point from the current measures and from the optimal successive tracking point. Such a cost allows for model continuity and favours the presence of at most one mode. A further modelling phase involves using a simple interpolation algorithm that is able to smooth peaks and fill holes, and that gives the model a monotone or, at least, unimodal shape.

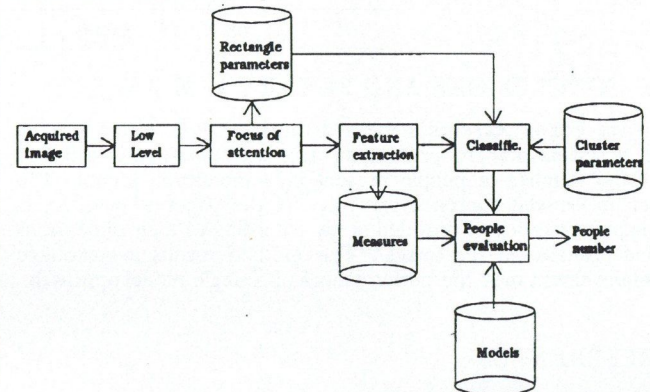


Fig. 2. Inference process on-line phase for crowding evaluation.

The on-line phase is illustrated in Fig. 2.

Each image acquired with a camera is processed by the Low-Level and Focus-of-Attention algorithms to extract measures of the selected features. Each rectangle extracted from an image is classified according to the off line clustering and the corresponding model is used to determine the number of people. The number of people present in an image is simply computed by summing up the crowding values of all rectangles.

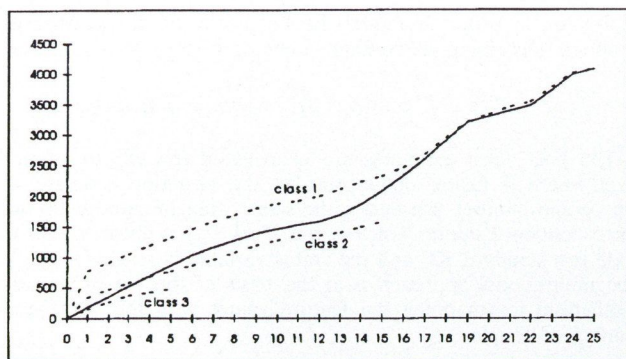


Fig. 3. Single model (continuous line) and multiple models (dashed lines) of the feature: number of edge points (the shapes are in the plane "number of people vs. feature value")

In Fig. 3 one can compare the present approach with the single-

model one, which computes only one model for all rectangles, without performing the clustering step. One can notice that the multiple-model approach is more accurate than the one based on a single model. For particular ranges of feature values, the number of people is underestimated, whereas for others it is overestimated.

4. RESULTS

By means of the multiple-model approach, we can improve the performance of a crowding-evaluation system.

We used the models themselves as evaluators of the number of people present in an area; this number corresponded to the number associated with the feature values obtained by the Low- and Middle-Level algorithms. As shown in Table 1, the multiple-model approach makes it possible to obtain percent increases in the performances of the selected features.

In Table 2 percentage results referring to some test image sequences are presented at the Low and fusion levels. Fusion mechanism allows to largely increase the number of cases in which estimation error is very low.

Table 1. Note: Hit = 0 – 20% error, Grazed = 20 – 30% error, Missed => 30% error (data in %)

Feature	Single Model			Multiple Models		
	Hit	Grazed	Missed	Hit	Grazed	Missed
num. of edge points	32.91	15.40	51.70	46.93	14.30	38.77
num. of hist.-maxima	31.62	13.02	55.36	40.05	11.73	48.21
sum. amplit. hist.-maxima	30.80	10.45	58.75	40.88	12.01	47.11
num. of edge points/rect y_dim	51.51	17.14	31.35	53.16	16.22	30.61
rectangle x_dim	31.81	10.08	58.11	39.69	11.73	48.58

Table 2. Only the Hit percentage has been reported here (0 – 20% error)

Sequence #	EKF 1-Hit %	EKF 2-Hit %	EKF 3-Hit %	EKF 4-Hit %	EKF 5-Hit %	KF-Hit %
1	65.00	69.00	67.00	65.00	57.00	90.00
2	67.11	53.95	56.58	73.68	39.47	80.26
3	54.00	46.00	54.00	72.00	36.00	66.00
4	70.30	55.45	71.29	78.22	46.53	71.29
5	60.58	59.62	58.65	63.46	45.19	72.12
6	68.75	64.06	62.50	62.50	40.62	78.12

5. CONCLUSIONS AND FUTURE WORKS

The paper presents a method for modelling virtual sensors in a system able to provide an accurate quantitative estimate of the number of people present in a monitored scene. The multiple-model approach presented in the paper is based on a simple clustering of crowded areas according to their dimensions and positions in the image. This method results in a notable improvement over the performances of a single-model approach.

REFERENCES

- [1] C. S. Regazzoni, A. Tesei, V. Murino: "A real-time vision system for crowding monitoring", *Proc. of IECON 93*, Hawaii, November 1993, pp. 1860-1864.
- [2] J. T. Tou, R. C. Gonzales: *Pattern Recognition Principles*, Addison-Wesley, Massachusetts, 1974
- [3] R. Bellman, S. E. Dreyfus: *Applied Dynamic Programming*, Princeton University Press, New Jersey, 1969.
- [4] C. S. Regazzoni, A. Tesei, G. Vernazza: "A Bayesian network

As future work, the multiple-model approach will replace the single-model one also in the crowding-evaluation system, previously developed at DIBE, based on the Bayesian Belief Theory of information propagation and fusion [4], [5]. A second future trend is oriented to track the position of the people present in the monitored scene in time, by means of their rectangular occupancy areas extracted from the image. This procedure is useful to provide the people localization in the 3D monitored environment.

- for automatic visual crowding estimation in underground stations", submitted to *Machine Vision and Applications: An International Journal*, 1993.
- [5] R. Bozzano, C. S. Regazzoni, A. Tesei, G. Vernazza: "A Bayesian network for crowding estimation in underground stations", in: *Proc. of 7th International Conference on Image Analysis and Processing* (in press).

TEXTURE ANALYSIS

IMAGE PROCESSING SYSTEM FOR DETECTION OF WEAVING DEFECTS

T THOMAS and M. CATTOEN

ECOLE NATIONALE SUPERIEURE D'ELECTROTECHNIQUE, D'ELECTRONIQUE,
D'INFORMATIQUE ET D'HYDRAULIQUE DE TOULOUSE
2, RUE CAMICHEL, 31071 TOULOUSE, CEDEX, FRANCE

1. INTRODUCTION

Quality control is a great part of the manufacturing process used to improve the quality of the finished product and to ensure the efficiency of the manufacturing process. The inspection of products for quality control is an important example of costly process that, if successfully completely automated, will improve efficiency and competitiveness.

In many industries, quality inspection is carried out manually, but the inspection of their products relies on the subjective judgement of individual inspectors. So the automation of this process is the subject of more research. In the textile industry, the automatic inspection allows the continuity of the quality of material manufacturing, and the optimization of the pinking according to the situation of the defects.

Our work follows a study led in the laboratory [2], which permits in a given experimental environment the defect detection on material.

In this paper, our method of defect detection [7] is explained. Then the system configuration and some results are presented. In this part, the difference between two kinds of filter for the periodic texture suppression, is accentuated. On the other hand a discussion is made on the contribution of the geometrical analysis and the characterization of the defects.

2. DETECTION METHOD

It is difficult to memorize the characteristics of every fault because there is a great diversity of defects. The detection must be only made from the information contained in the material image analysed.

2.1. Method description

Simply patterned material is made up by a network of chain and frame threads, which explains that the densitometrical profile of a defectless material (in row and column) is periodic [5]. The detection method exploits the periodicity breaking of the densitometrical profile when a defect appears. The periodic characteristic of the material is deleted by filtering, so the defects — characterized by an aperiodic component — are more easily detected. The processing line is shown on the Fig. 1.

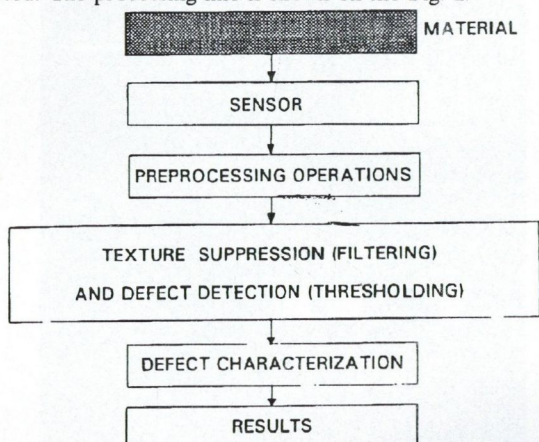


Fig. 1. Treatment chain

2.2. Method analysis

2.2.1. Texture suppression

The texture periodicity deletion was initially performed by a moving average filter. We have endeavoured to improve the filtering by a raised cosine filter (Turkey filter).

2.2.2. Profile thresholding

The row and column filtered profile of a defectless material image, present some amplitude variations caused by different factors. We have to take this variations into account by introducing two thresholds defined by $T_- = M - k\sigma$ and $T_+ = M + k\sigma$ (M and σ^2 are the mean and the variance of the profile and k a constant experimentally established). Above them, the material is regarded as defective.

2.2.3. Image block size

Due to the fact that the profile is an integration operation, the variation — brought about a small defect in comparison with the of the images blocks — is trivial. So we need to fit the size of the image blocks to the size of the small defect.

2.2.4. Geometrical analysis and characterization

The detection allows the obtaining of an image having zones where both densitometrical profiles (in rows and columns) exceed the tolerance threshold. Such zones are called "objects". The detection is not made in a continuous way. In most cases, one defect is constituted by several objects. The last steps of the processing are

- the geometrical analysis of the objects which goal is their joining (defect reconstruction). This analysis is divided in three steps
 - object labelling,
 - object merging,
 - small object deletion;
- the defect characterization.

3. METHOD IMPLEMENTATION

3.1. Learning procedure

The last step of the learning phase is made on a defectless sample of material. It is composed by some measurements for the preprocessing operations and the computation of the material modelling parameters (row period, column period).

3.2. Continuous stage

The continuous stage corresponds to the stationary response of the system where the detection is made at the same speed as the image lines are acquired. This stage uses the parameters previously determined.

Four operations are made in sequence

- image line acquisition and correction,
- profile computation,
- filtering,
- profile analysis.

3.3. System configuration

The development structure used is based on a PC compatible computer. It is built around INTEL® 386DX microprocessor clocked at 33 MHz. The computer is connected to the image

acquisition and visualization system by an input-output board built around two parallel interface adapters. The system is made up

- an image acquisition board (B/W or color),
- an image visualization board.

4. PERFORMANCES

4.1. Definitions

Before the characterization, to evaluate the detection viability, the no detection rate is defined as the ratio of the number of blocks where the defect has not been detected, to the number of blocks containing a defect, and the false detection rate as the ratio of the number of blocks where the defect has been detected, to the number of blocks which does not contain a defect.

4.2. Example of detection

The Fig. 2 is an example of defect detection. A defect is perceptible on the line densitometrical profile: In the column densitometrical profile before the treatments, the defect cannot be seen. However after the treatments, the defect is detected.

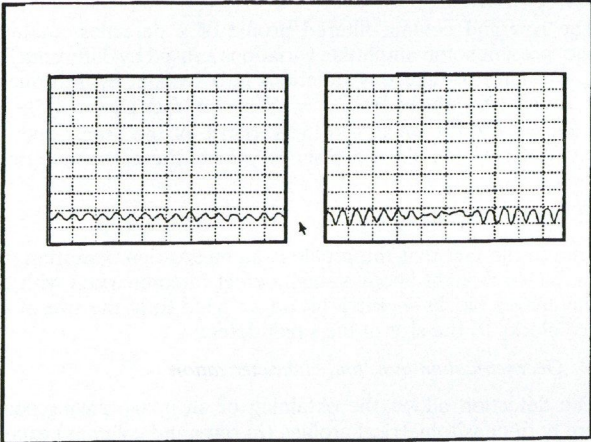


Fig. 2. Example of profile of the Fig. 3a

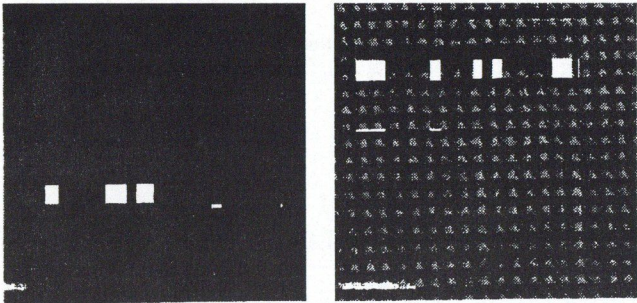


Fig. 3. Example of defect detection

In Fig. 3a, we can notice a shift between the position of the defect and the detection is noticed on the row and the column detection. This shift results from the filtering computed by

$$m_l(y)_{filter} = \frac{1}{N} \sum_{k=1}^N h(k)m_l(y+k)$$

(the n^{th} sample of the filtered profile is computed from the N samples following n of the profile)

4.3. Detection rate

The two detection rates according to k are shown on the Fig. 4. The experimental results have confirmed that $k = 3$ is the best compromise between no detections and false detections.

By using the profile of gray-level mean and a raised cosine filter, a no detection rate of 2 % and a false detection rate of 6 % were obtained.

Detection criterion

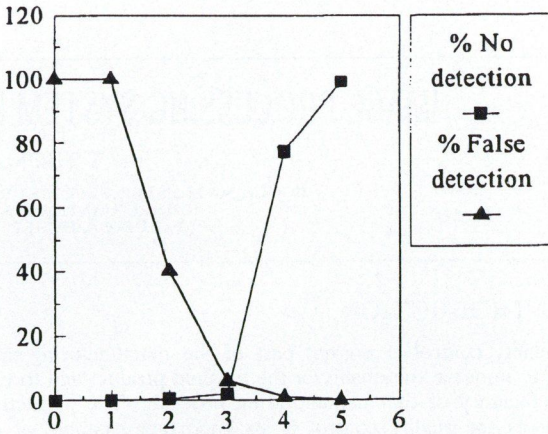


Fig. 4. Detection criterion

4.4. Filter influence

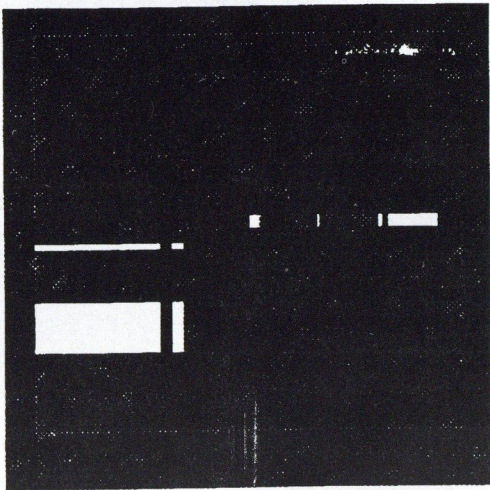


Fig. 5. Detection with moving average filter

In the Fig. 5, there are some false detections. They are due to the fact that with the moving average filter, the high frequencies are not well filtered. So there are residual components, and the detection is harder. The raised cosine filter leads to a more reliable detection (Fig. 6). In the two previous figures, the detection parameters are the same, but only different filters are used.

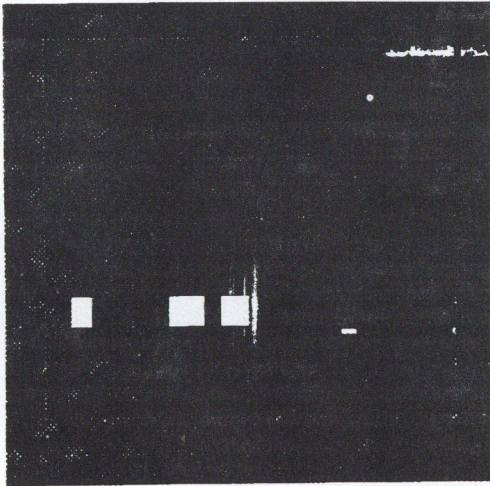


Fig. 6. Detection with raised cosine filter

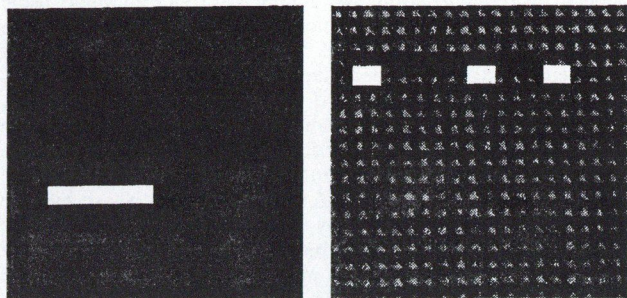


Fig. 7. Results of geometrical analysis

Fig. 7 shows the results of the geometrical analysis in the same conditions as in Fig. 3.

To evaluate the contribution of the characterization, the global no detection rate is defined as the ratio of the number of the present but not detected defects, to the number of present defects in the image. The global false detection rate is the ratio of the number of the detected but not present defects in the image, to the number of present defects in the image. In the same conditions as in the first evaluation, a global false detection rate of 1 % and a global no detection rate of 0.5 % were obtained.

The decrease of the false detection rate is the consequence of the characterization.

REFERENCES

- [1] M. Cattoen, B. Esclassan: "Image processing system for real time detection of weaving defects", Conference International Signals and Systems — RABAT (Maroc), 9-11 Avril 1990.
- [2] B. Esclassan: *Contribution a l'étude et a la réalisation d'un système de détection en temps réel de défauts de tissage*. Thèse de doctorat, INP, Toulouse, 1990.
- [3] O. D. Faugeras: "Vision par ordinateur en robotique. Etat de l'art". Premier colloque image, Biarritz, Mai 1984.
- [4] P. Peltie: "Ou en est le visitage automatique?", *L'Industrie textile*, Janvier 1988.
- [5] T. Thomas, M. Cattoen: "Method for defect detection on material using a 2D periodic model", Deuxième conférence internationale IMAGE'COM, Bordeaux, March 23-25, 1993.

- [6] T. Thomas, M. Cattoen: "System for defect detection on material", 7th Annual European Computer Conference, COMPEURO 93, Paris/Evry, France, May 24-27, 1993.
- [7] T. Thomas, M. Cattoen: "Automatic inspection of simply patterned material in the textile industry", IS&T/SPIE's Electronic Symposium on Electronic Imaging: *Science & Technology*, San Jose California, USA, February 6-10, 1994.
- [8] F. M. Vilnrotter, R. Nevatia, K. Price: "Structural analysis of natural textures", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI8, Janvier 1986.
- [9] P. Vollet: *Analyse et synthèse d'images de textures structurées*, Thèse de doctorat, EPFL, Lausanne, 1987.

An explanation of the decrease of the no detection rate is the following: when a defect appears in two adjacent image blocks but not both are detected, it is regarded as one no detection. After the characterization, the detection rates are computed on the complete image, this case can not appear.

5. CONCLUSION

The objective of our study is the optimization of an existing algorithm [1] of defect detection. The texture suppression has been improved by the use of the raised cosine filter. On the other hand, the false detection rate has been decreased by the defect characterization.

However we have limited our investigations to plain or simple patterned materials.

The extension of this work is the defect classification. In this case, the number of periods will be an important parameter. When a faithful description of the defect is needed, a small image block size will be chosen. So a multiple step detection will be considered

- defect detection with an image block size about 7 periods,
- defect classification with an image block size about 2 or 3 periods.

The other extension of this work is the detection on complexly patterned material. Here a new work will be made on pattern recognition and the modelling will be more difficult (several periods characterize the material) and the existing algorithm will have to be fitted or new one will be developed to allow the operations of detection.

ANALYZING TEXTURE ANISOTROPY VIA GRAY-LEVEL DIFFERENCE FEATURES

D. CHETVERIKOV

COMPUTER AND AUTOMATION INSTITUTE, HUNGARIAN ACADEMY OF SCIENCES
BUDAPEST, P.O. BOX 63, H-1518 HUNGARY

1. INTRODUCTION

Since the publication of the recent stimulating paper by Kass and Witkin [1], there has been growing interest in the investigation of *oriented patterns* such as texture images originating from flow-like processes. Directionality has become a popular topic of texture research (see e.g. [2]-[5]). Directionality can be viewed as local *anisotropy* that stems from dominating orientation of elongated texture elements. Computer analysis of this textural property usually involves orientation-sensitive filtering followed by local orientation coherence evaluation [1], [2]. This can be done at variable scale.

Julesz' pioneering work on *preattentive* (spontaneous) human texture perception (e.g. [6]) convinced the image analysis community that the second order statistics of texture images play a dominant role in spontaneous texture discrimination. This conjecture was supported by the impressive performance of the co-occurrence features in the computer analysis of texture patterns

[7]. In many cases, similar features based on a simplified and faster version of the co-occurrence probability matrix (CPM) — the gray-level difference histogram (GLDH) — were found [8], [9] to yield as good results as the co-occurrence based features.

Later the Julesz' conjecture was criticised as being applicable to limited classes of patterns. The interest of the researches has started to gradually shift towards *attentive* perception which is responsible for evaluation of such fundamental properties as texture symmetry, directionality, regularity and structural complexity. In a recent paper: [4], Rao and Lohse reported on the results of a study of human texture perception aimed at identifying those high level texture features that account for most of the attentive texture discrimination capability of the human vision system. They conclude that directionality and regularity are among those very few high level texture features that guide the process of perceptual grouping (taxonomy) of textural patterns.

Directionality is a special although perceptually important case

of anisotropy. While this special case received considerable attention, anisotropy in general was studied in just a few early works on texture analysis. Davis [10] introduced the notion of co-occurrence based *polarogram*. We used a more general term of *anisotropy indicatrix* (directional polar diagram) [11] and studied texture anisotropy via indicatrices depicting linear edge density and edge orientation distribution of texture edge map. Later, the anisotropy features introduced in [11] were successfully applied to rotation-invariant texture discrimination [12].

As it was pointed out in [11], the relevance of anisotropy analysis is to a large extent related to the crucial role played by *symmetry* in natural sciences in general and in human and computer vision in particular. Basic conservation laws of physics follow from the symmetry properties of the space-time. Analysis of a physical phenomenon (flow, field, etc.) is considerably simplified if a proper coordinate system is selected that complies with the symmetry (and anisotropy) of the phenomenon. The same observation applies to texture patterns as well. Julesz [6] concluded that the presence of symmetry facilitates human perception of texture. (For a recent short survey on the role of symmetry in vision, see [13].) Kass and Witkin [1] correctly emphasise that directionality evaluation of oriented patterns is indispensable to properly set up the coordinate system for further detailed analysis. Unfortunately, in many works on texture the patterns studied are manually pre-oriented so as to simplify the task in question. A more realistic case of arbitrary orientation and the problem of orientation sensitivity (compare to edge detection!) are rarely addressed. To approach these problems, one has to define *axes of anisotropy* (or, locally, axes of prevailing directionality [1]). We will show how the symmetry analysis of an anisotropy indicatrix yields axes that can be used for this purpose.

The increasing number of studies on symmetry of planar and 3D shapes and local gray-value patterns (see e.g. [14], [15]) indicate the recognition of the role of symmetry in vision. Recently, a local symmetry operator has been applied to texture discrimination [16]. Motivated by this recognition as well as by the discovery of the importance of directionality for high level texture perception, we reconsidered our previous research on anisotropy [11], [12] in an attempt to use co-occurrence for detailed anisotropy analysis. In this paper we present the initial results of our study aimed at texture anisotropy and symmetry analysis via the GLDH extended to arbitrary angles. (We preferred GLDH to CPM because of its simplicity and comparable performance. The same analysis can be done via CPM features in the same way.)

In section 2, we introduce the notion of extended GLDH and define the GLDH features used to indicate anisotropy. Examples of GLDH based anisotropy indicatrices for random and regular textures are demonstrated and their stability under rotation is shown. Section 3 is devoted to symmetry analysis of texture via anisotropy indicatrix and to definition of anisotropy axes. Also, it is experimentally shown how indicatrices of a regular pattern vary with spacing magnitude. In section 4, we comment on image resolution aspects of anisotropy. Finally, the computational efficiency of the approach, its limits and directions of further research are discussed in section 5.

2. CO-OCCURRENCE FOR ARBITRARY SPACING

It is generally believed (see, for example, [4]) that co-occurrence features are not suitable for anisotropy analysis because in digital images the magnitude and the angle of the spacing vector are interrelated. The Davis' co-occurrence based polarogram [10] was only computed for the multiples of $\pi/4$. At small interpixel distances, a very limited number of angles can be investigated which is not sufficient for anisotropy analysis. On the other hand, small spacings were shown to be very useful for texture discrimination and segmentation [9]. They carry important information on density of texture elements, or texture coarseness. Small spacings are better suited for segmentation as well since for them the features can be evaluated in a small window while large spacings require larger windows.

The CPM and GLDH can easily be extended to arbitrary spacing. One can estimate the gray-value in an arbitrary (non-integer) location by interpolating the values of the neighbouring pixels. Such interpolation is a standard operation in image

processing (see e.g. [17]). By using the linear interpolation followed by requantisation, we obtain the required value using five integer multiplications/divisions per pixel. (We estimate the CPM/GLDH in each digital raster position of the origin of the spacing vector; the operation can be extended to arbitrary locations of the origin as well.)

The interpolated gray-value and the gray-value of the origin are to increment the corresponding entry of the co-occurrence matrix. In our initial experiments described in this paper, we used the *symmetric* gray-level difference histogram whose bits are the frequencies of absolute gray-value differences. This means that the histogram needs only be computed in the angle range $\alpha \in [0, \pi)$. In some cases it may be desirable to distinguish between α and $\alpha + \pi$ and use the non-symmetric GLDH, but we do not do this now.

For a given spacing magnitude d , GLDH is evaluated for a sequence of angle values. A set of normalized features are computed for each "oriented" histogram and depicted in polar co-ordinates as anisotropy diagrams indicating texture anisotropy with respect to these features. In this study, standard GLDH features [7] were applied.

A GLDH based anisotropy indicatrix is a polar diagram representing one of the above GLDH features computed for a series of angle values from $[0, 2\pi]$. Fig. 1 shows examples of indicatrices for a variety of patterns from the Brodatz' album [18]. All Brodatz images in this study, including rotated versions of the same texture, were acquired using a CCD camera and a frame grabber and digitized to $256 \times 256 \times 8$ bits. The indicatrices were computed for angles $\alpha_i = 0, 5, 10, \dots, 175$ degrees which results in reasonably smooth curves. The curves were then extended to $[180, 355]$, scaled to separate them for better visibility and overlaid on the corresponding texture images. Computation of the extended GLDH features for a 256×256 pixel size image takes about 0.5 sec per angle on a PC AT (486) computer. At the angle resolution used in this study, this means 18 sec per image which is sufficiently fast for extensive testing of the proposed approach. The GLDH based indicatrices shown in Fig. 1 reflect the anisotropy of the texture patterns.

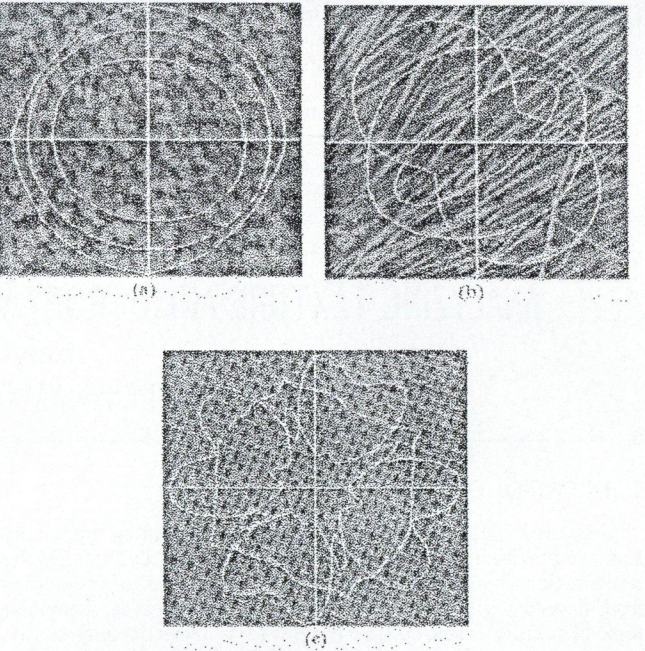


Fig. 1. Examples of anisotropy indicatrices for textures of different regularity: (a) sand, $d = 1$; (b) straw, $d = 1$; (c) woven wire, $d = 13$ (close to the period). The indicatrices are zoomed in the descending order: MEAN, DEV, ASM.

An important problem of anisotropy analysis is the stability of an indicatrix shape under image rotation. This general problem is related to the intrinsic anisotropy of algorithms operating on

a digital raster which has its own anisotropy. A similar problem is addressed in edge detection when one tries to develop an orientation insensitive operator. Fig. 2 demonstrates the stability of the GLDH based indicatrices computed for a number of pattern orientations.

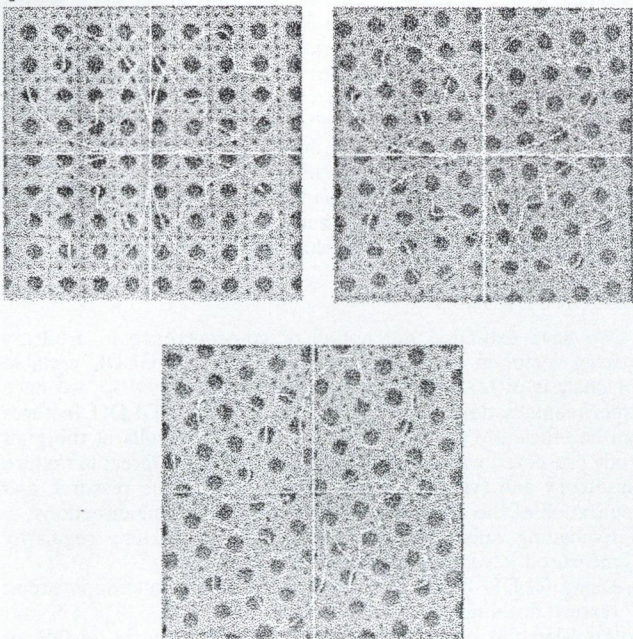


Fig. 2. The MEAN and DEV indicatrices of a regular pattern (cane) computed for different image rotations with $d = 28$ (close to the period)

3. TEXTURE SYMMETRY

It was already mentioned that regularity had been also found to be one of the basic high level features of texture perception. At present, texture regularity can be evaluated by a limited number of methods (see e.g. [19].) The regularity manifests itself in the periodicity of a CPM/GLDH feature computed as a function of spacing magnitude. The degree of regularity depends on direction within the pattern. We have made initial experiments to study how the indicatrix of a regular texture varies with spacing magnitude. Since regularity is related to symmetry, let us first introduce a measure of reflectional texture symmetry.

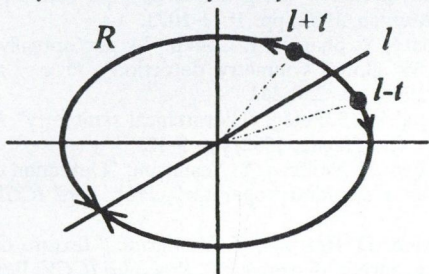


Fig. 3. Computing symmetry measure $REFSYM(l)$ of an elliptical indicatrix R

Given an anisotropy indicatrix, we evaluate texture symmetry in direction α as the reflectional symmetry of the indicatrix with respect to the axis having orientation α and passing through the indicatrix origin. For each angle α_1 match the two halves of the indicatrix that are defined by the axis oriented α_1 (see Fig. 3.). The matching rate is used as a measure of reflectional texture symmetry in a given direction. This measure is defined as follows:

$$REFSYM(l) = \left(\frac{2}{M} \sum_{t=1}^{M/2} \left[1 - \left| \frac{R(l+t) - R(l-t)}{R(l+t) + R(l-t)} \right| \right] \right)^{\gamma} \quad (3)$$

In (3), $t, l = 0, 1, \dots, M/2$, where M is the number of angles. $R(s)$ is the value of the indicatrix at angle α_s and s is defined modulo M . Power γ is used to make the symmetry measure more sensitive.

The value of the symmetry measure varies between 0 and 1. Large $REFSYM(l)$ indicates the presence of symmetry in direction α_1 . $REFSYM(l)$ can be viewed as an anisotropy indicatrix for texture symmetry. Since we consider indicatrices with two-fold rotational symmetry, it is easy to see that such a symmetry indicatrix will have two times more symmetry axes than the original indicatrix.

In our early work [11], we defined the *main axis of anisotropy* as the major axis of inertia of the indicatrix. The orientation of the major axis of a planar star-shaped polygon can be easily computed via the second order moments of the polygon [11]. The two rotation-invariant combinations of the moments were shown to be useful in anisotropy evaluation and rotation-invariant texture discrimination [12]. However, for highly symmetric polygons (e.g. for highly regular textures) the major axis of inertia cannot be used. A more general definition of an axis of anisotropy we will use here is that of an axis corresponding to a high, prominent peak of a symmetry indicatrix. We call a peak P of a symmetry indicatrix an *axis of anisotropy* if the following two conditions are satisfied:

$$P > HEIGHT \quad \text{and} \quad \frac{\max\{V_1, V_2\}}{P} < FLATNESS,$$

where V_1, V_2 are the heights of the valleys surrounding the peak P , $HEIGHT$ and $FLATNESS$ the two threshold parameters characterizing the peaks. In this study, we used $HEIGHT = 0.5$ and $FLATNESS = 0.75$. Other parameters were $\gamma = 5$ and $M = 72$.

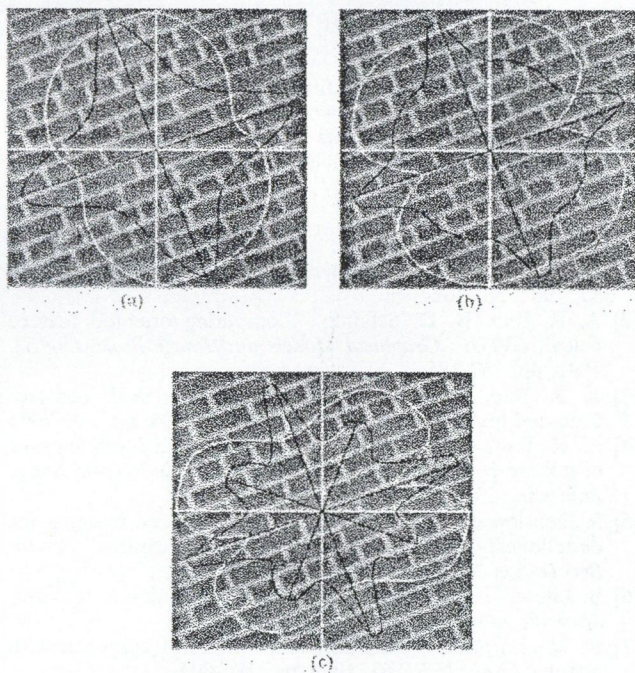


Fig. 4. MEAN indicatrix of a regular pattern (brick wall) computed for different spacing magnitudes: (a) $d = 1$; (b) $d = 9$; (c) $d = 17$.

The symmetry indicatrix and the detected axes of anisotropy are shown in dark.

Fig. 4 shows examples of the MEAN indicatrix and its symmetry counterpart for the same regular pattern and varying spacing. As the spacing approaches the characteristic value — the texture period — the indicatrix becomes more symmetric. In addition to the two basic axis of anisotropy, two weaker axes appear that indicate other specific directions within the pattern.

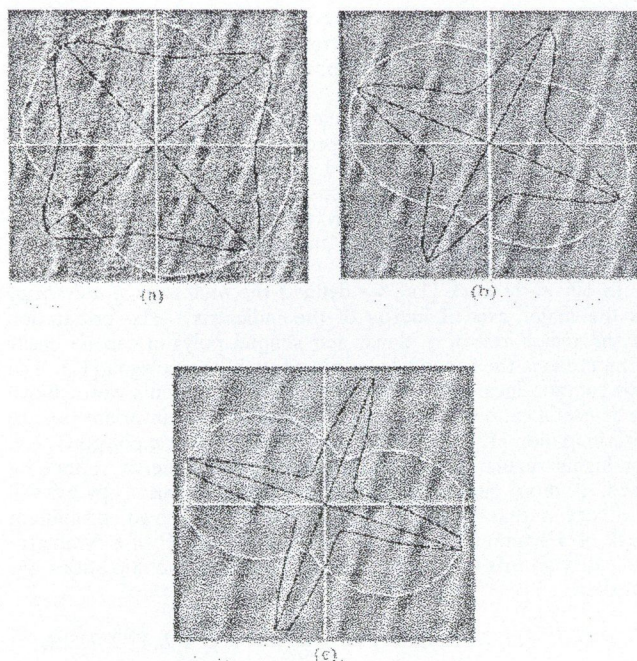


Fig. 5. MEAN indicatrix of a hierarchical texture (woven matting) computed for $d = 1$ at three resolution levels. The symmetry indicatrix and the detected axes of anisotropy are shown in dark.

4. TEXTURE ANISOTROPY AND IMAGE RESOLUTION

Although the need for multiresolution analysis of textures was recognized in pioneering works on multiresolution image processing, research in multiscale texture processing was in fact limited to the fractal model which is not applicable to regular

REFERENCES

- [1] M. Kass, A. Witkin: "Analyzing oriented patterns", *CVGIP* 37, 1987, pp. 362-385.
- [2] A. R. Rao, B. G. Schunk: "Computing oriented texture fields", *CVGIP: Graphical Models and Image Processing* 53, 1991, pp. 157-185.
- [3] A. R. Rao, R. Jain: "Computerized flow fields analysis: Oriented texture fields", *IEEE PAMI* 14, 1992, pp. 693-709.
- [4] A. R. Rao, G. L. Lohse: "Identifying high level features of texture perception", *CVGIP: Graphical Models and Image Processing* 55, 1993, pp. 218-233.
- [5] S. Denslow et. al.: "Statistically characterized features for directionality quantitation in patterns and textures", *Pattern Recognition* 26, 1993, pp. 1193-1205.
- [6] B. Julesz: "Experiments in the visual perception of texture", *Scientific American* 232, 1975, pp. 34-43.
- [7] R. M. Haralick: "Statistical and structural approaches to texture", *Proc. IEEE* 67, 1979, pp. 786-804.
- [8] R. W. Connors, C. A. Harlow: "A theoretical comparison of texture algorithms", *IEEE Trans. PAMI* 2, 1980, pp. 204-222.
- [9] J. Weszka et. al.: "A comparative study of texture measures for terrain classification", *IEEE Trans. SMC* 4, 1976, pp. 269-285.
- [10] L. S. Davis: "Polarograms: a new tool for image texture analysis", *Pattern Recognition* 13, 1981, pp. 219-223.

patterns. It is clear that the fundamental properties of textures — anisotropy, regularity, symmetry, complexity — may depend on image resolution. A systematic study of the behaviour of these properties at multiple resolutions or varying scale would be desirable and instructive. Here we will only show some results indicating that this research area is worth putting efforts in.

Fig. 5 shows MEAN indicatrices and their symmetry curves obtained at three consecutive levels of resolution. A Gaussian resolution pyramid was built for a hierarchical texture using the procedure described in [22]. The two higher levels of the pyramid were expanded to the original image size as proposed in [22] producing interpolated images with less details. As the fine texture is smoothed out, the indicatrix changes, its symmetry peaks become more prominent and the anisotropy axes rotate to approach the visually perceived characteristic directions.

5. CONCLUSION

We have extended the notion of co-occurrence to arbitrary spacing vector in order to make the CPM and GLDH suitable for analysis of texture anisotropy and symmetry. Also, we have experimentally demonstrated that the extended GLDH features can be efficiently used for this purpose. The results of the pilot study presented will hopefully stimulate further interest in texture anisotropy and symmetry. We intend to continue research and application of the proposed method in the following directions:

- Evaluating anisotropy via the degree of texture regularity measured in varying direction.
- Using GLDH based anisotropy features for rotation-invariant texture discrimination and segmentation.
- Applying the proposed method to non-symmetric GLDH or CPM in order to estimate orientation of tilted textured surfaces via texture gradient (skewed symmetry of indicatrix).

ACKNOWLEDGEMENTS

This research was supported in part by the grants OTKA 2579 and PHARE-ACCORD H9112-11266. The author thanks Tamás Réti for the idea of symmetry detection in planar shapes and Peter Meer for a valuable discussion.

- [11] D. Chetverikov: "Textural anisotropy features for texture analysis", *Proc. IEEE Conf. on PRIP*, Dallas, 1981, pp. 583-588.
- [12] D. Chetverikov: "Experiments in the rotation-invariant texture discrimination using anisotropy features", *Proc. 6th ICPR*, Munich, 1982, pp. 1071-1073.
- [13] F. Labonté, Y. Shapira, P. Cohen: "A perceptually plausible model for global symmetry detection", *Proc. 4th ICCV*, Berlin, 1993, pp. 258-263.
- [14] H. Zabrodsky, S. Peleg: "Hierarchical symmetry", *Proc. 11th ICPR C*, The Hague, 1992, pp. 9-12.
- [15] D. Reisfeld, H. Wolfson, Y. Yeshurun: "Detection of interest points by a symmetry operator", *Proc. 3rd ICCV*, Osaka, 1990.
- [16] Y. Bonnef, D. Reisfeld, Y. Yeshurun: "Texture discrimination by generalized symmetry", *Proc. 4th ICCV*, Berlin, 1993, pp. 261-265.
- [17] B. Jähne: "Digital Image Processing", Springer-Verlag, Berlin, 1991.
- [18] P. Brodatz, Textures, New York, Dover, 1966.
- [19] D. Chetverikov: "Generating contrast curves for texture regularity analysis", *Pattern Recognition Letters* 12, 1991, pp. 437-444.

TEXTURE ANALYSIS BASED ON WAVELET DECOMPOSITION

CS. KISS

TECHNICAL UNIVERSITY OF BUDAPEST
DEPT. OF MICROWAVE TELECOMMUNICATIONS
GOLDMANN TÉR 3, H-1111 BUDAPEST, HUNGARY

P. CSILLAG

KFKI RESEARCH INSTITUTE FOR MEASUREMENT
AND COMPUTING TECHNIQUES
P.O. BOX 49, H-1525 BUDAPEST, HUNGARY

G. NÉMETH

KFKI RESEARCH INSTITUTE FOR MEASUREMENT
AND COMPUTING TECHNIQUES
P.O. BOX 49, H-1525 BUDAPEST, HUNGARY

L. BÖRÖCZKY

KFKI RESEARCH INSTITUTE FOR MEASUREMENT
AND COMPUTING TECHNIQUES
P.O. BOX 49, H-1525 BUDAPEST, HUNGARY

1. INTRODUCTION

Textures provide important characteristics for surface and object identification from aerial or satellite images, biomedical imagery and many other types of images. Their analysis is fundamental to many applications such as quality control in different fields of industry and medical diagnosis.

The process of texture discrimination can be divided into three phases, such as feature extraction, feature selection and classification/segmentation. Extraction of texture features deals with the computation of features from the image data, which completely embody information on the spatial distribution of gray-level variation in the texture. Generally, a set of features is used for texture discrimination, however, there is no definite conclusion concerning the set of features, which has the best overall performance.

The subject of feature selection in texture analysis is concerned with mathematical tools to create an optimal feature set sufficiently characterizing the distinguishing properties of the different texture classes.

For each texture classification/segmentation problem feature extraction is a crucial task. Most existing texture features and the texture analysis itself can be divided into two categories, namely structural and statistical one. The former approach is based on the view that textures are made up of primitives appearing in more or less regular repetitive spatial arrangements. Statistical feature extraction techniques are mainly of three types: spatial gray-level dependence methods [1], [2], stochastic model-based features [3], [4], and transform/filtering methods [5]-[7].

Recent developments in multiresolution signal analysis such as Gabor and wavelet transforms helped to overcome one of the main difficulties of traditional texture analysis, that is the lack of appropriate tool to characterize effectively different resolutions of texture. Theoretical and experimental investigation showed, that multichannel texture analysis is an efficient tool and has several advantages concerning the traditional feature extraction methods. In this paper an idea is proposed for wavelet based texture discrimination. The theory of wavelet transform is discussed in Section 2. In Section 3 the transform-based methods for texture analysis are presented. The new feature quadruplet method is discussed in Section 4. Finally, the conclusion and the main topics of the future research are given.

2. PRINCIPLES OF WAVELET TRANSFORM

The Fourier transform has been the most useful technique for the frequency analysis of signals for a long time. Due to the fact that sinusoids have an infinite support, such an approach has undesirable effects if one deals with signals which are localized in time and/or space. The wavelet representation

$$f(x) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} c_n^m \Psi_{m,n}(x),$$

where

$$\Psi_{m,n}(x) = 2^{-m/2} \Psi(2^{-m}x - n)$$

has some important applications in image processing.

To construct function Ψ , first we determine a scaling function $\Phi(x)$ which satisfies

$$\Phi(x) = \sqrt{2} \cdot \sum_k h_k \Phi(2x - k).$$

Then, function $\Psi(x)$ will be

$$\Psi(x) = \sqrt{2} \cdot \sum_k g_k \Phi(2x - k),$$

where

$$g_k = (-1)^k h_{1-k}.$$

The forms of $\Phi(x)$ and $\Psi(x)$ are not required to perform the wavelet transform, which depends only on h_k . A J level decomposition

$$f(x) = \sum_k [c_{j+1,k} \Phi_{J+1,k}(x) + \sum_{j=0}^J d_{j,k} \Psi_{j+1,k}(x)]$$

can be given recursively.

The coefficients $c_{0,k}$ are given and for coefficients $c_{j+1,n}$ and $d_{j+1,n}$ the following relations are held:

$$c_{j+1,n} = \sum_k c_{j,k} h_{k-2n}$$

$$d_{j+1,n} = \sum_k c_{j,k} g_{k-2n}.$$

The coefficients h_k can be found in [8].

This multiresolution wavelet transform results in a "compact" nonredundant image representation in contrast to the traditional methods, such as low-pass filtering and Laplace pyramid transform [9].

For example, image decomposition by a 2D wavelet transform can be done as follows. The image is split into a low resolution part and the difference signal which describes the difference between the low resolution image and the actual one. Due to the correlation which exists in the original image, the difference signal will have a histogram which is peaked around zero. The low resolution image still contains spatial correlation. Therefore, this decomposition can be repeated several times, so that a pyramidal image decomposition is created [10].

The size of a low level image S^{LL} is a quarter of the size of the original image. There are three different signals: S^{HL} , S^{LH} and S^{HH} . The S^{HL} indicates scale variations in the x-direction, and its high value indicates the presence of a vertical edge. Large values of S^{LH} and S^{HH} indicate the presence of a horizontal edge and a corner point, respectively (see Fig. 1).

The traditional pyramid-type wavelet transform recursively decomposes subsignals in the low frequency channels. However, the most significant information of a texture often appears in the middle frequency channels. Further decomposition only in the lower frequency region, may not help much for the purpose of classification. This concept can be illustrated in Fig. 2 and Fig. 3 where the pyramid-structured wavelet transform is applied to two different kinds of images.

S^{LL}	S^{HL}
S^{LH}	S^{HH}

Fig. 1. Traditional one level wavelet decomposition

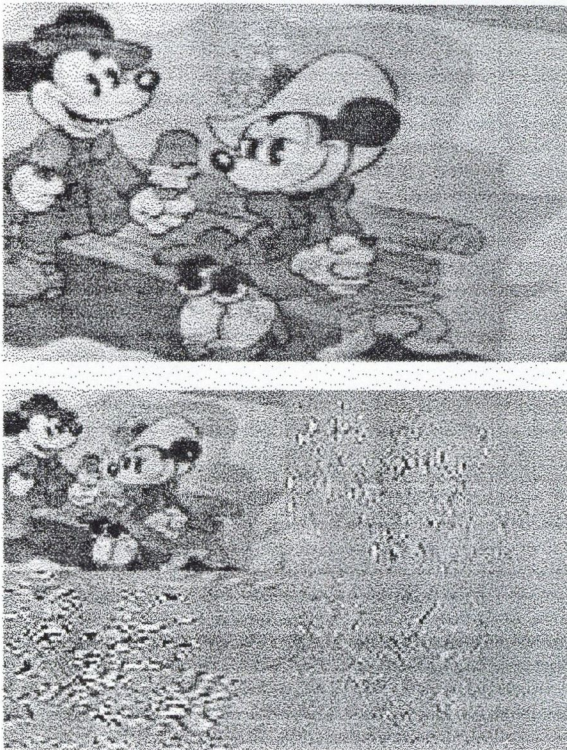


Fig. 2. Mickey image and its wavelet decomposition

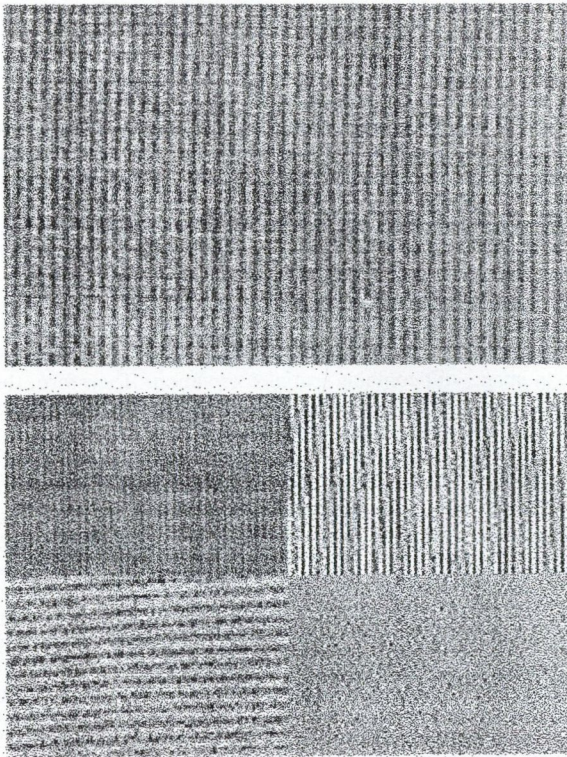


Fig. 3. Sackcloth texture and its wavelet decomposition

The Mickey image as an ordinary image and its pyramid-structured wavelet transform is shown in Fig. 2. We can recognize the Mickey image clearly from its low frequency channel.

In contrast, we are not able to recognize a similar texture pattern in the low frequency channel for Fig. 3. Instead, we observe some vertical and horizontal patterns in the middle frequency region.

3. FEATURE EXTRACTION BASED ON WAVELET DECOMPOSITION

Feature extraction based on wavelet transform has been studied by several authors. For texture analysis S. G. Mallat [5] proposed a texture discrimination scheme based on discrete wavelet decomposition of textured images in order to obtain the fractal dimension of the particular textures. However, it is well known, that the single fractal dimension is not sufficient to unique classification of different textures.

Another approach to feature extraction was developed by A. Kundu [7]. In this algorithm a QMF filter bank was used to decompose the texture into several subbands, and special features e.g. "zero-crossing" features were calculated for the high-subbands.

T. Chang and C.-C. J. Kuo proposed an efficient method recently [6]. Its main principle is presented below. The idea of this approach leads to a new type of wavelet decomposition called tree-structured wavelet transform. The conventional multiresolution image representation based on wavelet transform decomposes subimages of the low frequency channels recursively. It can be seen that, this decomposition is not very useful for a large class of natural textures, because their most significant information appears in the middle frequency channels. The key difference between this algorithm and the traditional pyramid wavelet representation is that the decomposition is no longer applied to the low frequency subsignals recursively. Instead, it can be applied to the different signals of each pyramid level.

At first, a given texture image is decomposed into 4 subimages by a 2D wavelet transform. For all subimages an energy measure is calculated and compared with each other. If the energy of a subimage is significantly smaller than the others, we stop the decomposition in this region since it contains less information. The subimage containing higher energy will be decomposed further. This recursive and adaptive procedure can be represented by a quadtree structure or energy map. For texture classification the feature set will be chosen from the energy map as the most dominant channel-energy values (see Fig. 4).

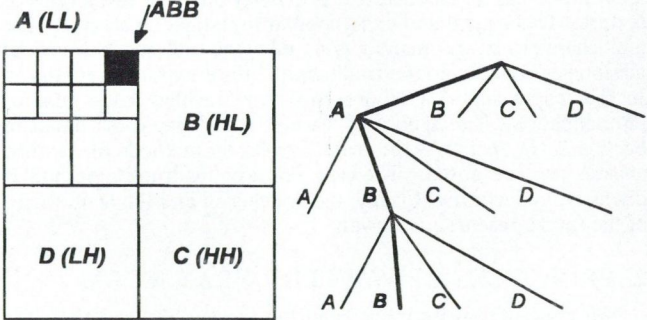


Fig. 4. Quadtree structure, or energy map

It is important to note, that the tree-structured wavelet transform is effective for textures which have dominant middle frequency channels.

Our attempt is to bridge the gap between the traditional feature extraction and feature extraction based on wavelet transform. According to the above information the one-level wavelet transform decomposes the original picture into four subimages. The transformation produces four texture patterns, which are not similar to the original image however they characterize it. Therefore, features that can identify the texture have to be found and extracted from the subimages.

If we compute the one-level wavelet transform, one feature

can be extracted from each subimage, thus we can get a feature quadruplet. These four features define a point in the four-dimensional field, typical for adequate texture

In the learning phase of texture classification this method results in a point for every typical texture. In the classification phase the decomposed unknown texture gives its feature quadruplet, which also corresponds to a point in the above field. These vectors are then compared, and the unknown texture can be classified based on an appropriate distance measure in 4D space.

4. TEXTURE ANALYSIS USING FEATURE QUADRUPLET

Steps of feature extraction

1. Decomposition of texture into 4 subimages using 2D wavelet transform.
2. Calculation of the traditional features (e. g. run-length matrix) for every subimage.
3. These four features E_{LL} , E_{LH} , E_{HL} , E_{HH} , define a point

REFERENCES

- [1] R. M. Haralick, K. Shanmugam, I. Dinstein: "Textural features for image classification", *IEEE Trans. Syst. Man Cybernet.*, Vol. 3, Nov., 1973, pp. 610-621
- [2] M. H. Galloway: "Texture analysis using gray-level run lengths", *Comput. Graphics. Image Process.*, Vol. 4, 1975, pp. 172-179
- [3] Ch.-M. Wu, Y.-Ch. Chen: "Statistical Feature Matrix for Texture Analysis", *Graphical Models and Image Processing*, Vol. 54, No. 5, Sept., 1992, pp. 407-419
- [4] R. M. Haralick: "Statistical and structural approaches to texture", *Proc. IEEE* 67, 1979, pp. 786-804
- [5] S. G. Mallat: "A theory for multiresolution signal decomposition: The wavelet representation", *IEEE Trans. Pattern Anal. Mach. Intelligence*, Vol. 11(7), July, 1989, pp. 674-693
- [6] T. Chang, C.-C. J. Kuo: "Texture Analysis and Classification and Tree-Structured Wavelet Transform", *IEEE Transactions*

in the four-dimensional field. This four-dimensional feature vector can be used for e.g. texture classification.

5. CONCLUSIONS AND FUTURE RESEARCH

We have attempted to summarize the methods of transform-based feature extraction. Recent publications on this field show, that multichannel texture analysis is an efficient tool and has several advantages concerning the traditional feature extraction methods

In the future we are going to focus our investigation on the wavelet transform, and feature quadruplet in texture analysis more thoroughly, theoretically as well as experimentally. Furthermore, application of other types of multiresolution approaches (biorthogonal wavelets, Gabor wavelets, etc.) for feature extraction are also planned. Design of complete texture classification/segmentation schemes for specific applications, e.g. for industrial quality control and biomedical imagery is also a topic of our future research.

- on Image Processing*, Vol. 2, Oct., 1993, pp. 429-441
- [7] A. Kundu, J.-L. Chen: "Texture Classification Using QMF Bank-Based Subband Decomposition", *Graphical Models and Image Processing*, Vol. 54, Sept., 1992, pp. 369-384
- [8] I. Daubechies: "Orthonormal bases of compactly supported wavelets", *Commun. Pure Appl. Math.*, Vol. 41, Nov., 1988, pp. 909-996
- [9] P. J. Burt, E. H. Adelson: "The Laplacian pyramid as a compact image code", *IEEE Trans. Commun.*, Vol. Com-31, Apr., 1983, pp. 532-540
- [10] Cs. I. Kiss, G. Németh, D. Csetverikov, L. Böröczky: "On feature extraction for texture analysis", submitted to *Periodica Polytechnica*, 1994.
- [11] T. R. Read: "A Review of Recent Texture Segmentation and Feature Extraction Techniques", *CVGIP: Image understanding*, Vol. 57, No. 3, May, pp. 359-372, 1993.

MULTI-DIMENSIONAL PROCESSING

MAPPING STRATEGIES FOR SIGNAL AND IMAGE PROCESSING ALGORITHM PARALLELIZATION

I. ERÉNYI and I. VASSÁNYI

KFKI RESEARCH INSTITUTE FOR MEASUREMENT AND COMPUTING TECHNIQUES
P.O. BOX 49, H-1525 BUDAPEST, HUNGARY

1. INTRODUCTION

Most of the algorithms of image and signal processing are very computationally intensive, i.e. they require vast processing power. It applies also to other computations where massive data-handling is involved. To provide highly efficient solutions for these areas:

- processing architectures and algorithms must be designed and selected in accordance to their mutual interdependence, i.e. they must conform to each other,
- architectures should be adaptive and their configuration dynamic,
- the processing requirements can only be met by the integrated use of special purpose processors and migrating appropriate tasks to these processors.

Development of efficient massively parallel systems requires methodologies for matching parallelized algorithms to processor array based architectures. On the other hand, the deeper understanding of this two-way matching process requires methodologies for the design of parallel algorithms, for the modification of existing algorithms and their mapping onto parallel processing architectures, and also for the adaptive (and dynamic) reorganization of the processor arrangement. For these methodologies, software tools are required to support implementation. This involves work in development environments, notations for expressing algorithms and their transformations.

2. PROBLEM STATEMENT

The 'Mapping Problem' is commonly referred to as finding the best suited processor architecture or, in the case of a given architecture, the optimal load distribution and scheduling scheme for the parallelized subtasks of the initial problem, using an objective function based on minimal communication overhead and maximal processor utilization. More globally, the mapping process can be considered as a generalized technique, extendible for the coverage of configurable architectures as well. From the mapping efficiency point of view, algorithms can be classified as shown in Fig. 1.

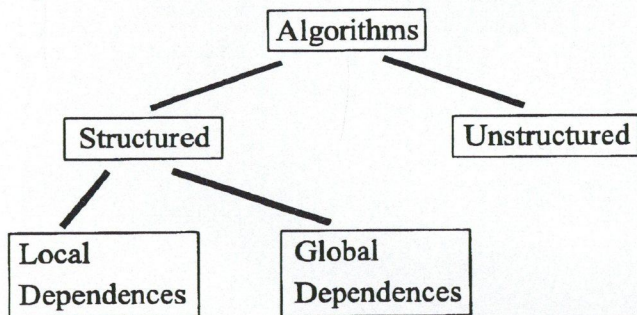


Fig. 1. Algorithm classes

The branch of *unstructured algorithms* stands for ordinary programs. Some parts of such problems may efficiently be parallelized at a fine granularity level (e.g. nested loops operating on non-interdependent variables) but the overall structure of the parallel code remains largely unstructured i.e. non-repetitive. This implies that the mapping is to be solved at the task level

(load balancing, task scheduling and migration algorithms) on an existing host architecture rather than on an algorithm-specialized one.

Most pixel level image processing algorithms possess three common properties: regularity, recursiveness, and locality [1]. They belong to the class of *structured algorithms* with local data dependencies which is characterized by a repetitive structure, and they can easily be decomposed into uniform sub-structures. The local nature of an algorithm results in localized communication among the processing elements (PEs) in the VLSI array processor realization. Some examples of the many important digital signal and image processing algorithms falling into this category are convolution, matrix multiplication and decomposition algorithms, IIR filters, image enhancement, edge detection, etc.

On the contrary, a well-known example for global dependencies is the FFT or the Viterbi decoding algorithm which requires global data exchange after each iteration step.

3. CONCEPTS OF FORMAL MAPPING METHODS FOR STRUCTURED AND LOCALIZED ALGORITHMS

The basic mapping methodology involves the application of various transforms on some representations of the algorithm and the derivation of a formal description for the proper PE array. The original dimensionality of the algorithm is usually reduced during the mapping process to decrease the number of necessary PEs. Among many formal approaches the most widely used is perhaps the method of projections [2], which employs a graph representation of data dependencies. The data dependence graph (DG) is an acyclic graph, which can be derived either from the algorithm's single assignment code form or its recursive equations. Then a sequence of projections is applied defined by projection vectors d_i over the index space of the algorithm. This leads to a signal flow graph (SFG) version from which the processor array can be designed directly. The projections reduce the number of graph nodes and, consequently, the number of PEs in the resulting array. This is equivalent to the transposition of the projected index dimension into the time domain. For example, the $N \times N$ matrix-vector multiplication algorithm has a 2D DG, each node representing one multiplication and one addition, see Fig. 2. When this graph is projected onto a 1D SFG along the projection vector $[1 \ 0]^T$, one array processor will execute one row multiplication, lasting N times longer.

The scheduling of the PEs in the SFG is determined by a suitably chosen scheduling vector s , somewhat constrained by the previous choice on d . Then the I/O timing and the design verification can be performed automatically.

The projection is therefore defined by the two selected vectors s and d . The optimal values of s and d have already been derived for the most important signal processing algorithms in a number of case studies [3], [4]. The aim of the research on formal methods is, nevertheless, the full automation of the mapping process in a future CAD tool. The automated solution could perhaps use an optimized algorithm-architecture library as a knowledge base, start the mapping from the optimized architecture of a similar algorithm and fulfill the optimization with the help of the formal methods aided with some heuristics.

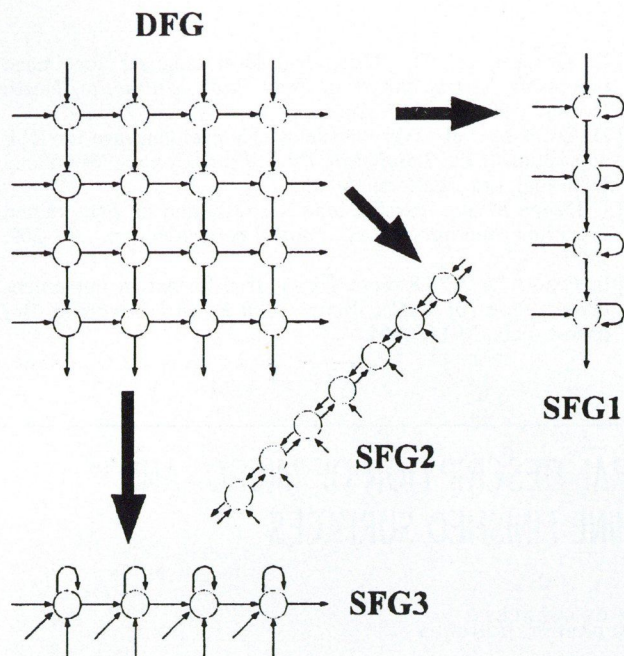


Fig. 2. The projection mapping method

4. THE TARGET ARCHITECTURE

As it was emphasized above, the hardware characteristics of the target architecture should be kept in mind during the whole of the mapping process. The design of the target architecture depends largely on the applied VLSI technology.

4.1. Hardware implementation background

Recent developments in VLSI technology and processor arrays have considerably increased the processing capability at various hardware levels.

Parallel computers using arrays of processors. Inexpensive massively parallel computers are becoming a reality. Their effective use requires parallel algorithms and methods for mapping them onto arrays of processors. This is not an easy task. Massively parallel computers may be classified into a number of various categories based on the granularity (size) of assigned tasks or operations. Some of the typical massively parallel systems are realized as networks of transputers, network of DSPs, RISCs, etc. [5]-[7]. Here the operation of the system is fundamentally affected by the design of the parallel software.

Algorithmically dedicated components. Characteristic applications, algorithms, or protocols may be implemented by the use of off-the-shelf VLSI processing components, e.g. convolvers, contour tracers, dedicated transform chips, etc. Though these circuits are not flexible, the system design methodology must be flexible enough to allow their use in a convenient way.

Application Specific Integrated Circuits (ASICs). Current VLSI technology offers high-level design aids and methodologies as full custom components to implement ASICs for selected applications.

Programmable Logic Devices (PLDs). PLDs provide hardware parts for the semi-custom design philosophy. For special applications requiring unique systems they may provide an economically feasible hardware solution. Their use however may impose a further mapping step in the realization of the individual PEs at 'deep logic' level (e.g. bit-level) [8]. The possibility to build re-programmable 'coprocessors' and modules using PLDs (e.g. Xilinx LogicCells) attached to high-speed hosts opens new vistas to this design approach. This means to design 'soft' architectures that can be redefined and reconfigured dynamically (as the need arises) and adaptively during execution time, thus providing adaptive architectures.

4.2. The optimal strategy

The projection technique and the other formal methods successfully find the optimal array architecture and reduce the DG dimensionality. However, the enlarged grain size (amount of computation between two synchronizations of adjacent PEs) means more algorithm-specific operation, which in many cases may endanger the versatile use of the array architecture. It is probably more advantageous to choose a 'slightly configurable' or 'soft' processor array as the target architecture even at the expense of sub-optimality. In other words, it would be desirable to find a *minimal set of hardware features* that enable a PE array to execute a *maximal set of similar algorithms* in a nearly optimal way, see Fig. 3.

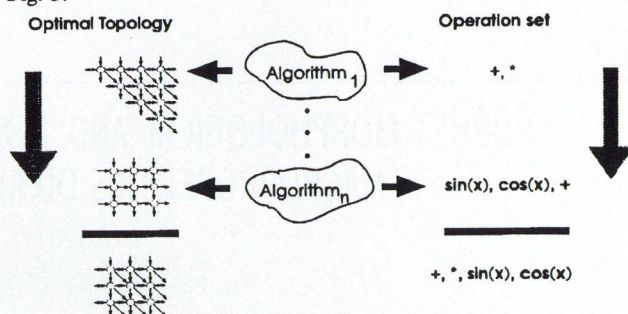


Fig. 3. A feasible mapping strategy

The most important hardware characteristics of a configurable PE array can be summarized as follows:

- order of magnitude of the number of connected PEs, subject to grow as technology advances
- overall topology of the array i.e. the number and organization of communication links
- organization of communication in terms of wavefront vs. systolic scheduling
- PE hardware features: operation set, size of communication link buffers and register bank
- control of the array: the methods for programming local computations and I/O scheduling for the individual PEs.

The complexity of the PEs and the interconnection network should be kept as low as possible to allow feasible VLSI implementation of reasonably sized PE arrays. For example, the following relatively small operation set satisfies most image processing needs: addition, multiplication, division, Euclidean norm, circular and hyperbolic functions. A realization proposal for a similar operation set can be found in [4]. As for the supported topology, a bi-directional mesh-based topology seems to cover the majority of optimally projected array structures.

The 'minimal set' of the necessary hardware features can be determined by simulating the processor array for different image processing algorithms and in each case comparing the performance with the best reached so far on a dedicated array optimized for the respective algorithm. The resulting array should be a good trade-off between the algorithm scope covered (configurability) and the performance of the array. Then, the standard mapping method must be modified to take into consideration both the algorithm's properties and the configurability of the array — this means that the principles of the task allocation methods and the formal (regular) mapping methods must be composed.

5. SUMMARY

The paper discusses some aspects of the fine grain size formal mapping methods aided with some heuristics. The use of a 'slightly configurable' PE array is proposed that seems to suffice for a wide range of image processing problems and simultaneously offers a feasible VLSI realization. Further research is needed on the desirable hardware characteristics of such arrays. Also the available VLSI components' impact on the mapping process should be thoroughly investigated.

REFERENCES

[1] S. Y. Kung: *VLSI Array Processors*, Prentice Hall, NJ, 1988.

[2] W. Stewart et al.: "Mapping Recursive Algorithms onto Array Architectures", *Proc. of the Int. Workshop on Algorithms and Parallel VLSI Architectures*, Pont-a-Mousson, June 10-16, 1990.

[3] L. Chisci, G. Zappa: "A Systolic Array for Computing the LQ Optimal Gain", *Proc. of the Int. Workshop on Algorithms and Parallel VLSI Architectures*, pp. 115-124, Pont-a-Mousson, June 10-16, 1990.

[4] J. Duprat et al.: "Fast VLSI Implementation of CORDIC Using Redundancy", *Proc. of the Int. Workshop on Algorithms and Parallel VLSI Architectures*, Pont-a-Mousson, June 10-16, 1990.

[5] D. Crookes et al.: "Using high-level language for image processing on transputers in *From Pixels to Features*", North Holland, 1991, pp. 313-326.

[6] D. Crookes et al.: "Very efficient high-level language for DSP applications". In *VLSI Systems for DSP and Control*, Woodhead Publishing Ltd., 1991, pp. 99-103.

[7] F. Dehne et al.: "Parallel branch and bound on fine-grained hypercube multiprocessors", *Parallel computing*, pp. 201-209, 1990.

[8] B. Fehér: "A 2D Discrete Cosine Transformation Implementation", *Proc. of CERN Workshop on Parallel Processing*, Budapest, Feb. 10-11, 1994.

ROBUST MORPHOLOGICAL AND TEXTURAL DESCRIPTION OF MICRO- AND MACROPROPERTIES ON MACHINE FINISHED SURFACES

I. LOVÁNYI and Á. NAGY
TECHNICAL UNIVERSITY OF BUDAPEST
MUEGYETEM RKP. 9., H-1111 BUDAPEST, HUNGARY

1. INTRODUCTION

The economical situation of nowadays justifies all efforts of improving the efficiency, flexibility, quality and reliability of production. In this context the growing importance of flexible cells is evident, as this conception provides the highly automated production of different products in a varying number due to the ever changing demands. There are estimations that in leading industrialized countries the biggest part of new machines will be installed in flexible cells by the end of this decade [1].

On the other hand highly automated new approaches of Computer Integrated Manufacturing call attention on new problems which are partly not foreseen by users:

CIM installations often imply too complex machines of diversified technologies packed in the working area in a very condensed way resulting in after all a long and expensive realization and finally a reduced rentability and reliability. Paradoxically the high level of automation demands the presence of specially skilled operators to survey the system and handle all kinds of generally very delicate perturbations. Due to this fact some authors tend to reconsider a little bit the optimal balance between automation and human presence, concluding in the surprising fact that man regains some domain against robots [2].

One aspect of this problem is the lack of appropriate models for the interpretation of breakdowns and defective production including the problem of intervention and preventive maintenance. One part customers and vendors of new flexible cell equipment equally deny to give detailed statistics about their breakdowns and defects considered to be too confidential information.

On the other hand the initial troubles in new flexible cell installations are often regarded as infantile disorders and not as consequences of basic design mistakes.

Though there are many factors of this question, the following paper will discuss only two of them:

- Flexible cell needs flexible instrumentation to interact efficiently with the environment during operation. Integration of vision into the flexible cells seemed to be extremely useful to solve various tasks related to increased flexibility. The paper will briefly present the results of our research in this field.
- Installation of vision into industrial environment claims for modular and standard system design, which can reduce the cost, time and risk of such realizations and provides easy update or replacement in the future [3]. In fact industrial people would have more confidence toward this approach rather general in conventional instrumentation.

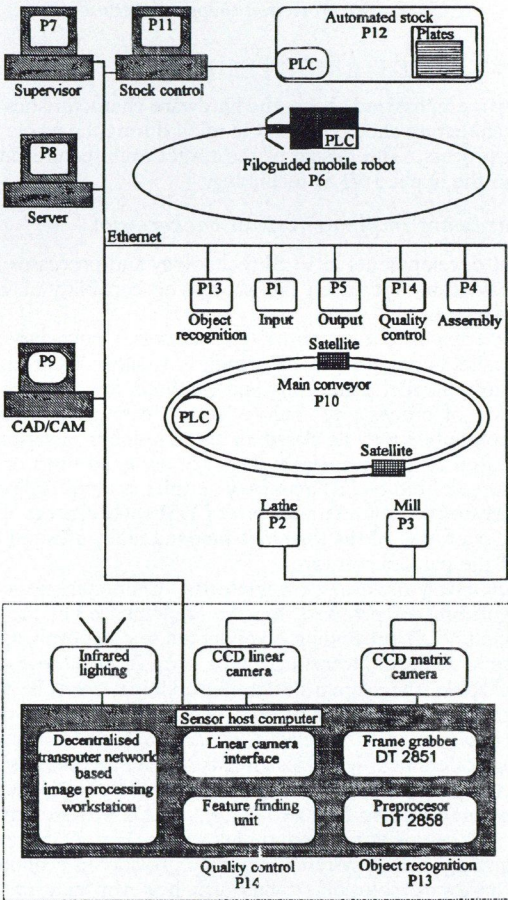


Fig. 1.

The basic configuration of the flexible cell we used is a commercialized one and was installed at INSA de Rennes in 1992. This part gives only a general summary of this flexible cell, further details can be found in [4]. We examine here rather the extension possibilities of the cell with intelligent sensors aiming to increase overall functionality. Fig. 1. shows the general structure of the system indicating only the key elements.

A vision based input *object recognition post* (P13) was developed for real time localization and classification purposes. Near to the output post a high resolution linear camera based real-time vision system realizes 100 % on-line quality control of macro properties measured on the output products thus creating a separate *quality control post* (P14).

Now we summarize some new possibilities provided by the integration of vision into the flexible cell:

1.1. 2D localization and classification of input pieces

A matrix camera based vision system localizes and classifies input objects on a plate, in real time with mechanics (Fig. 2.). Robust algorithms provide functioning under varying ambient light conditions without using additional light source or modification of background-object contrast. The recognition of industrial parts is based on parallel implementation of statistical test (Kolmogorov-Smirnov) type region growing and gray-scale morphological transformations by means of a transputer network and a low level DT2858 preprocessor utilising the inherent data parallelism of these algorithms.

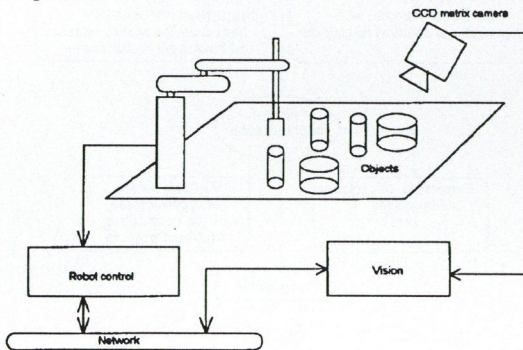


Fig. 2.

1.2. On-line quality control of macro properties

A linear camera based vision system completed by infrared lighting is the heart of the on-line quality control post (Fig. 3.). The measurement of critical dimensions on more axes would demand special positioning mechanics. In our case the robot of the output post makes this task. The precision of robot movement doesn't affect the accuracy of measurement as it is based on the interpretation of the whole object. Owing to this fact the resolution of measurement is relatively small, but 100 % video inspection has evident advantages opposed to conventional contact methods for dimensional measurement in industrial environment.

Resolution: 50 µm / pixel

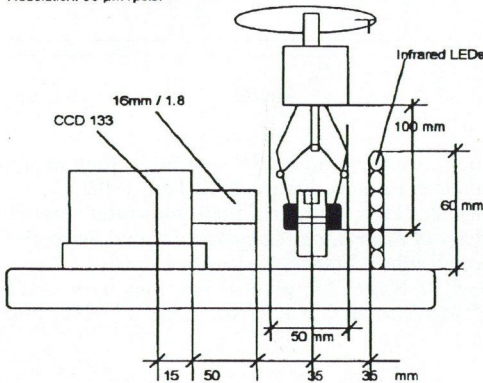


Fig. 3.

1.3. Off-line quality control of micro properties

Inspecting surface quality has a growing interest in manufacturing. Measurement of engineered surfaces by noncontact video methods has many advantages compared with conventional me-

chanical or optoelectronic setups. With complex evaluation of surface textures it is possible to detect defects as well as to localize the source of them throughout the process. We made preliminary experiments using elements of the our vision system. The method is based on the texture evaluation of laser illuminated surfaces using second order statistics, fractal description and global geometrical properties. Fig. 4. shows the principle of measurement and the sensor head layout of an industrial application.

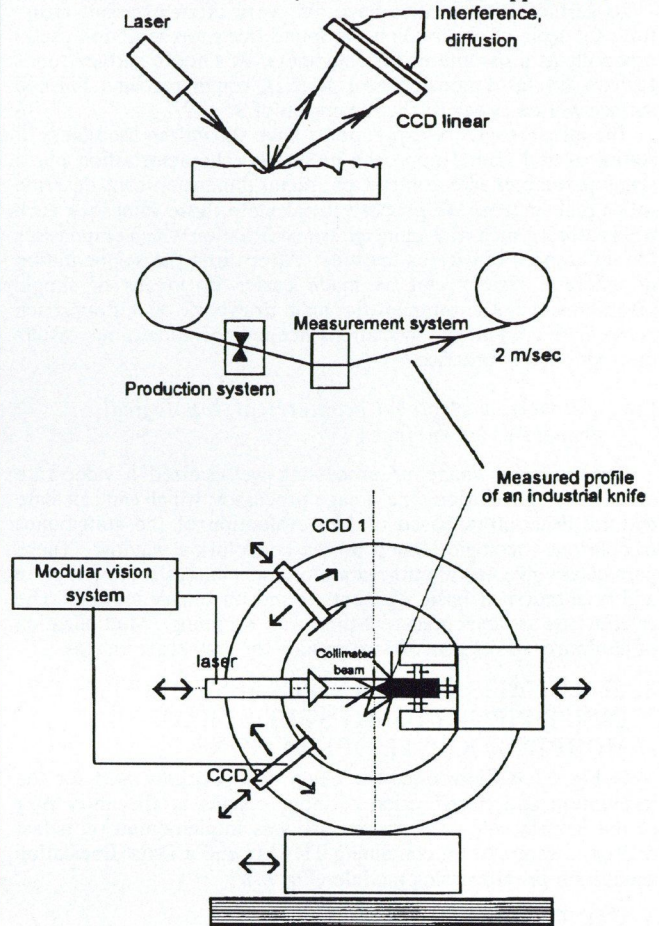


Fig. 4.

2. TEXTURE ANALYSIS OF MACHINE FINISHED SURFACES

The texture of a surface can be characterized by the repetition of basic elements which consist of several pixels arranged in a periodic, quasi-periodic or random way. During this analysis one can distinguish the following general steps:

- defining a set of basic texture elements which can characterize different type of surfaces
- execution of an efficient image preprocessing
- finding correlation between image properties and basic texture elements
- finding correlation between texture properties and the quality of machining

The realization of these steps might change case by case. For instance in the production the textures of machine finished surfaces to be analysed are rather random without repetition or orientation significative or — contrary — quite periodical. In other applications we may seek mainly the size, distribution, orientation and distance of regions of different textures.

2.1. Representation with probability methods

For the digitized image of surfaces we calculated a concurrence matrix based on second order statistics. A set of well known useful parameters like entropy, contrast, variance measures of

correlation, etc. were derived from these calculations. Experiments have shown that homogeneous textures can be well described by second order statistics and this approach is still valid for slightly structured textures. A practical problem is the need of big computational power to execute these calculations in real-time.

2.2. Representation with fractal dimensions

As surface irregularities have also very often invariant structures for scale of observation, we found this representation useful especially as a parameter for roughness. A smooth surface tends to have fractal dimension near to 2, a rough machine finished surface will be closer to the dimension of 3.

The initial results of experiments have shown the usefulness of statistical and fractal approach in surface characterization but a single parameter like contrast or fractal dimension can't describe well a real surface. We propose to calculate these values for each pixel realising such way an image transformation which emphasises the differences of various textures. Afterwards the segmentation of different regions can be made easier by means of simple algorithms. For the moment the main drawback of this approach is the huge amount of calculations needed to validate our results of experience in practice.

2.3. Extraction of global geometrical-topological properties of surfaces

For real-time image preprocessing we realized a video-rate hardwired "convolution type" image processor, which can calculate textural descriptors based on the evaluation of the statistics of different topologies found by the convolution window. These parameters give useful information on the shape, size, distribution and orientation of basic elements found on binary images. The calculations are executed real-time with scanning. Multiplication of hardware makes possible extensions for gray-scale images.

3. 2D LOCALISATION AND CLASSIFICATION OF INPUT PIECES USING STATISTICAL MORPHOLOGICAL OPERATIONS

In Fig. 5. we introduce the chain of algorithms used for the localization and classification of input objects at the entry post of the flexible cell. The vision task was implemented on a fast multiprocessor system containing TRAMs and a Data Translation acquisition-preprocessing module (Fig. 6.).

4. CONCLUSIONS

The paper shortly presented a modular vision system used in a flexible cell. Evaluation of operation has shown us that efficiency and flexibility of production lines might be increased substantially with additional external sensors, especially with visual ones. We enumerated and classified several possibilities of integration vision into the flexible cell pointing out the immediate results as well as some future benefits. We developed a modular industrial vision system capable to cover all examined tasks. General hardware

REFERENCES

- [1] M. Widmer: "Modèles mathématiques pour une gestion efficace des ateliers flexibles", Presses polytechniques et universitaires romandes, 1991.
- [2] M. Alberganti: "Automobile, l'homme remet", *Industries et Techniques*, pp. 36-40, 06/09/1991.
- [3] J. L. Davies, K. F. Gill: "Machine vision and automated assembly", *Mechatronics*, Vol. 3, No. 4, pp. 493-501, 1993.
- [4] R. Prat et al.: "Automatization des systèmes et gestion de

and software integration methods of these vision sensors has been found providing remarkable extensions and preserving at the same time 100 % compatibility with the original system, which demand was for us — as in most cases of "real" industrial applications — a must.

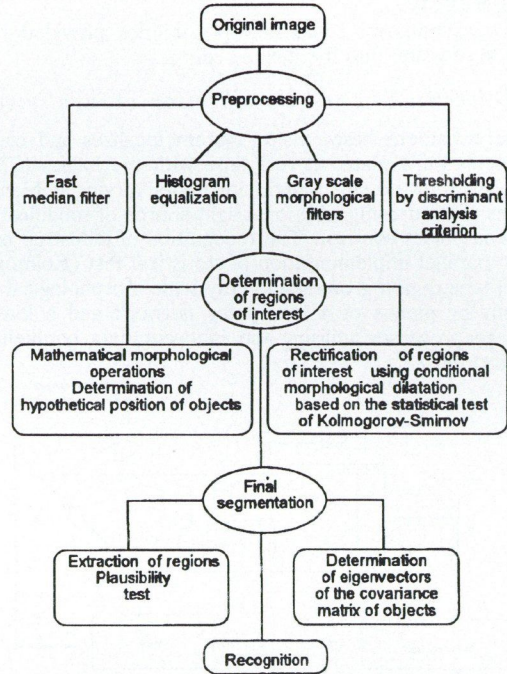


Fig. 5.

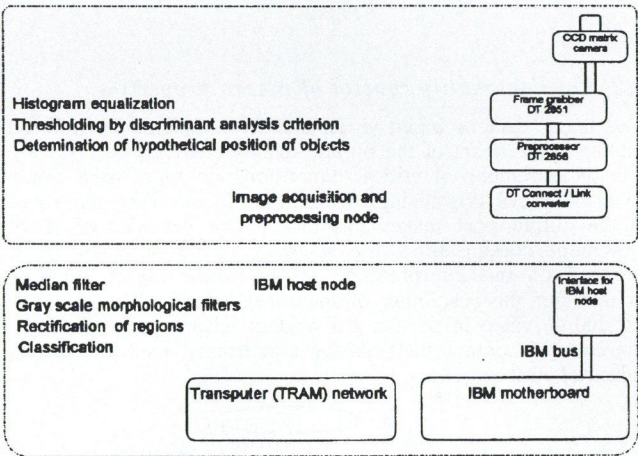


Fig. 6.

- production", *Cellule flexible CIM — Class. Stage de formation transnationale*, Rennes, France, 9-12 Juin, 1992.
- [5] I. Loványi, A. Nagy: "Industrial real-time vision": Applications, *Proceedings of Sister Cities Computer Science Research Centres Workshop*, Rennes, France, 17-19 April, 1991.
- [6] I. Loványi, A. Nagy: "3D robot vision using laser based active lighting", *Mechatronics*, Vol. 3, No. 2, pp. 173-180, 1993.

A DATA-FLOW APPROACH TO DEVELOP IMAGE PROCESSING APPLICATIONS

B. VÁRKONYI

TECHNICAL UNIVERSITY OF BUDAPEST, DEPARTMENT OF PROCESS CONTROL
MUEGYETEM RKP. 9., H-1111 BUDAPEST, HUNGARY

1. INTRODUCTION

In the last decades the progress in computers and imaging devices lead to the emerging need for image processing equipments. Every project in this field faces a very difficult problem at the start, how to develop the necessary software. Most of the time only general purpose systems are available at the first stages, and thus a lot of effort is wasted for creating a convenient software development environment specially suited for the image processing tasks. Some companies offer solutions, but these programs are rather expensive, many times not portable, stuck to a certain hardware (like [1], [2], [4], [6]), or they cover just partially the problems to be solved, and unfortunately they cannot meet all the requirements.

What we would really like to get is a special image processing oriented development frame. The pictures should be handled independently from their size, format and the imaging devices. We would like to port our source code to different machines without modifications! The application programmer wants to think in high-level abstractions (it is not efficient if he must know about all the details). As he has to manage very big data sets, he needs also some computer assistance for their administration. It would be also nice to get a "backplane" for combining simple tools into more sophisticated ones in a fast and reliable way. These requirements are very natural, but none of the commercially available systems can satisfy these conditions, it is almost impossible to build up such an environment from different already existing sources.

In the following sections we will introduce the data-flow approach of a software development frame called "RAster Picture Laboratory" (RAPIL). The RAPIL is not a complete solution for all the above problems, but it fits far better to the needs of a research setup than any other available product, and it is running on our existing equipment (not just a faint promise). As it supports modular and incremental specification, design and implementation, the future extension paths are rather straightforward (for more details see [15] and [10]).

We hope that other projects can get some benefits from the methodologies discussed here. The data-flow support of RAPIL is motivated by the basic philosophy of providing easy access to image processing systems in a system independent way. It is intended to serve as the basic building block for advanced graphical user interaction to image processing systems and system independent optimizing compilers. The choice of the data-flow paradigm was guided by the observation that image processing as a discipline in signal processing relates very well to functional formulations of the processing task. Functional languages are known to have no artificial dependencies of control besides pure data dependencies. Such languages suit very well for compiler optimizations of various resources, from memory space on single processor machines to communication paths on multi-processors. Even complex systems composed of different specialized processors can be dealt with.

The present stage supports an intermediate representation of flow graphs based on the OPE-object of the MSS package (see also [16]) and an interpreter for the processing of such flow graphs on the currently supported RAPIL systems. Several processing modules in the IMP library have been equipped with data-flow interfaces and can be used with this interpreter and the number of such modules will steadily increase in the future.

2. THE BASIC PRINCIPLE OF DATA-FLOW PROGRAMMING

The data-flow paradigm lives from the observation that every functional expression directly relates to a directed graph in which the nodes represent the function operator and the arcs stand

for the flow of results. Each of these results is produced by the application of a functional operator on the actual operands where the operands are represented by inbound, or incident, arcs and the result is shown by the outbound, or adjacent, ones. In the case of non-recursive expressions these rules lead to an acyclic directed graph as the representation of a corresponding functional expression. For our current purpose this restriction to non-recursive functional expressions is powerful enough to denote most of the common image processing tasks. The extension to some kind of recursive functions is possible and can be learned from literature.

Graphical representations of functional expressions can easily be interpreted as the machine language of a data processing system. The nodes translate to processing instructions and the arcs denote the data dependencies of the individual instructions. The problem how to do the interpretation of these graphs in a certain data processing system has been subject to twenty years of research.

A data-flow program is represented as a directed graph consisting of named *nodes*, which represent instructions, and arcs which represent data dependencies among nodes. Operands are propagated along the arcs in the form of data packets, called *tokens*. The execution of an instruction is called the firing of a node. The basic instruction firing rule common to all data-flow systems is as follows: *A node is enabled as soon as tokens are present at all its input arcs.*

When a node fires, a token from each input arc is removed and a result token is placed on each of its output arcs. At the machine level, a data-flow graph is represented as a collection of activity templates, each containing the following components:

- opcode of the represented instruction,
- operand slots for holding operand values, and
- destination address fields, referring to the operand slots of subsequent activity templates that need to receive the result value.

3. DATA-FLOW IMAGE PROCESSING SUPPORT PACKAGE IN RAPIL

The Data-Flow Image Processing Support Package in RAPIL consists of three major components, the program representation by means of MSS chains, the data-flow interpreter, and the data-flow interface modules of the processing libraries IMP, FEE and SAN. (For details on RAPIL parts you should refer first to [15] and [16]).

The application programmer generates, by means of some kind of translator program, an MSS representation of the data-flow graph. In this MSS representation the nodes are given by OPE objects whereas the arcs are denoted by some other MSS object (an SAP in most cases) which describes the data-token on the arc. The interpreter is a program module which controls the execution of this data-flow program representation and does the memory allocation for the created data objects.

Data-flow interfaces to the processing library modules allow for the execution of these module functions by the data-flow interpreter. These interfaces have to be supplied by the system programmer who developed the module as the semantical knowledge about the function is to be coded herein.

3.1. Program Representation and Semantics

Data-flow programs are formed by a MSS chain consisting of OPE objects and data objects (SAPs in general). The OPEs represent the individual nodes of the program and the data objects contain the results/operands they exchange during execution and, therefore, represent the arcs.

Table 1. MSS link identifiers for data-flow programs

/* link id field values */			
/* hiword==0 is reserved for RAPIL for the following purposes: */			
/* bytei==1 : data-flow graphs */			
#define MSSL_OPE	0x00000101L	/* object processing elements	*/
#define MSSL_PER	0x00000102L	/* permanent objects	*/
#define MSSL_TMP	0x00000103L	/* temporary objects	*/
#define MSSL_SRC	0x00000104L	/* source objects	*/
#define MSSL_DST	0x00000105L	/* destination objects	*/

The objects of the MSS chain are grouped by four relational links. MSS has four predefined link identifiers reserved for that purpose (Table 1). MSSL-OPE links all OPE objects of the chain and, thereby, grouping all nodes of the data-flow program for traversal by the interpreter. MSSL-PER comprises all data objects that will hold valid data at the termination of the data-flow program. MSSL-TMP links all data objects that are only valid for a short period of time during the execution. MSSL-SRC groups the data objects that have to be provided before the start of the program, e.g. the input data of the program graph. MSSL-DST, on the other hand, comprises all data objects that represent the output of the program.

3.1.1. The Object Processing Element Class

In the OPE class four slots of the main extension field *mext* contain control information for the data flow interpreter (see Table 2). The extension field *ext* is used to pass parameters and pointers to the data objects, a certain operation has to work on.

Table 2. OPE main extension field

```

struct ope_mext
{
    long id;          /* 8: tool identifier */
    long stat;        /* 9: tool status */
    long ioctl;       /* A: I/O control mask */
    long iosta;       /* B: I/O status */
    long extC;        /* C: extension slot */
    long extD;        /* D: extension slot */
    long extE;        /* E: extension slot */
    long extF;        /* F: extension slot */
};

```

- *tool.id* is an identifier that gives a reference to the executable function
- *tool.stat* controls the execution and represents the status of the OPE
 - 0 means no execution
 - *pos* means the number of repeated execution
 - *neg* means the termination of the function is controlled externally
- *tool.ioctl* represents the I/O control mask. It consists of 16 fields of two bits corresponding to the 16 extension slots. Refer to Table 3.
 - 0 in a field means that the extension slot carries no dynamically created data object (it might be used for parameter passing)
 - 1 means that this slot refers to an input object
 - 2 means that this slot refers to an output object
- *tool.iosta* contains also 16 fields of two bits which code for the status of the data object. The data flow interpreter marks the corresponding field with a 1 if it found the object core to be present (see Table 4). This saves considerable time. The presence of object cores need not be tested twice to enable the execution of the processing function

Table 3. OPE I/O control masks

/* I/O control mask - 16 channels x 2 bits */			
#define OPE_IO_NONE	0	/* channel is not used	*/
#define OPE_IO_INP	1	/* input channel	*/
#define OPE_IO_OUT	2	/* output channel	*/

Each OPE object contains slots to pass up to 16 parameters whose meaning is completely left to the data-flow interface of the specified function. The slots can contain operands and results in form of pointers to data objects in the program MSS chain as well as literals in form of long integers. The literals may be used in any way the system programmer writing the corresponding data-flow interface might choose. It might even contain packed information as the data-flow interpreter does no manipulations on the parameter slots.

Table 4. OPE I/O status values

/* I/O status - 16 channels x 2 bits */			
#define OPE_IO_EMPTY	0	/* data object is empty */	*/
#define OPE_IO_RDY	1	/* data object is ready */	*/

Corresponding with the 16 slots are 16 descriptors and status flags in the OPE slots *mext.tool.ioctl* and *mext.tool.iosta*, the descriptor codes for the type of the operand slot. They must always have the order of input (operand), output (result) and static, where static comprises literals and unused slots as well. The status flag is set when a certain operand has been detected by the interpreter but the node is not yet enabled. The role of these slots, therefore, is analogous to the tokens in a classical data-flow machine. As the data objects in the MSS chain stand for the arcs of the data-flow program the input object slots code for the incidence relation and the output object slots give the adjacency relation of the graph.

3.1.2. Data Objects

Data objects in the program chain reserve space for the results/operands that are created and consumed during execution of the program. The type of the object and its parametrization is determined by the processing function and must be set before execution. Especially the size of the object core must be known in advance. The current version of the Data-Flow Support Package will detect an error if the size of the object does not match the need of the specified function. It is left to the compiler to derive the proper parameters from the given program and its input data.

The data-flow program format in RAPIL uses data objects to represent arcs in the graph. However, for sake of efficiency this implementation stretches the term arc and implements *multi-arcs* with one source and multiple destinations. This makes the copying of data tokens obsolete in case of multiple use of a result in more than one node. The data objects must reflect this change by carrying a *reference counter* in the upper word of slot *chain.st*. A negative entry is used to code for permanent objects. A non-zero number stands for the number of adjacent nodes referring to the data of this object.

Input objects are given before execution and are the only objects to contain actual data at that moment. All other objects are empty which is denoted by a NULL in the core pointer slot *main.cmema*. At the end of the execution only the permanent objects (e.g. the destination objects) contain core data and may be handed out in a result MSS chain. Some programmers might see the need for permanently available data objects not being destination type result data but some kind of constants (not literals), for instance reference pictures or convolution kernels. The OPE class links, therefore, make a distinction between permanent and destination objects.

4. THE RAPIL DATA-FLOW INTERPRETER

The data-flow interpreter DFI is designed to provide a memory efficient implementation on classical single processor machines. Some restrictions of current operating systems make it not very suitable for systems without demand-paging virtual memory. The problem arises from the lack of dynamical binding of function calls in a portable way. All functions accessible to the interpreter have to be permanently linked to the executable image leading to a quite big program image.

A data-flow operation is always associated with the library func-

tion implementing it, although a library function may implement more than one data-flow operation. The distinction is then made by literals controlling the function in the library module. The transfer is made through a function pointer in a descriptor table composed of *df_function_struct* descriptors. Table 5 shows the structure of such a descriptor and an example of a definition for the *sobel* library module is given in Table 6.

The interpreter starts by fetching the root object of the MSSL-OPE link and then repeatedly traversing the chain of this relational link. The input channels are checked for tokens and in the negative case the input object is checked for the presence of a data core which then is marked by a token flag (see Section Data Objects above). If all input objects are found to be present the function is enabled and executed by transferring control to the function in **DF_FUNCTIONS[open→mext.tool.id].df_function*. The reference counters of the input objects are decremented and in case of zero (all tokens consumed) the core memory space is freed. After execution of a fired node the OPE is marked as processed by setting *mext.tool.iosta* to -1. This OPE will be discarded in the next scan. The scanning ends if no node fired in the last scan. In this case a MSS chain containing the permanent objects is returned.

Table 5. Data-Flow Operator Descriptor

struct df_function_struct				
{				
char	*df_function_name;	/* Name of the data flow function	*/	
OPE_FUNC_PTR	df_function;	/* Pointer to the data flow function	*/	
int	number_of_inputs;	/* Number of inputs of the function	*/	
char	*coded_input;	/* String with coded input types	*/	
int	number_of_outputs;	/* Number of outputs of the function	*/	
char	*coded_output;	/* String with coded output types	*/	
};				

Table 6. SOBEL Function Example

{ "sobel",	im0_sobel,	3,	"DF_ DF_PAR_LONG/FIX:1 DF_PAR_LONG/DEF:16",
		2,	"DF_ DF_SAP" }
{ "sobel_sqr",	im0_sobel,	3,	"DF_ DF_PAR_LONG/FIX:2 DF_PAR_LONG/DEF:16",
		2,	"DF_ DF_SAP" }
{ "prewitt",	im0_sobel,	3,	"DF_ DF_PAR_LONG/FIX:3 DF_PAR_LONG/DEF:16",
		2,	"DF_ DF_SAP" }
{ "prewitt_sqr",	im0_sobel,	3,	"DF_ DF_PAR_LONG/FIX:4 DF_PAR_LONG/DEF:16",
		2,	"DF_ DF_SAP" }

5. DATA-FLOW FUNCTIONS AND IMAGE PROCESSING MODULES

The library modules in the libraries IMP, FEE and SAN must be prepared to be used by the data-flow support modules. Every processing function must provide a data-flow interface which is contained in module *im0function.c*, *fe0function.c*, or *sa0function.c* respectively. Contrary to library core functions they only get a pointer to the OPE object describing the node they shall implement. The operands and results as well as the literals must be checked or at least the relevant information must be passed to the core function after creation of the destination core and the setup of the destination descriptor fields (according to the object type of the result). The result must then be checked and the output must be discarded in case of error. In case of success the input data tokens must be consumed by decrementing the reference count of the input data objects.

The following source code fragment demonstrates the interfacing principle:

```
int im0_convolve(ope_poi)
mssobj_poi ope_poi;
{
mssobj_poi s_poi,d_poi,aux_poi;
int flag,ind;
```

```
char *adr;
/* init */
s_poi = (mssobj_poi)ope_poi->ext.mss.ext0;
d_poi = (mssobj_poi)ope_poi->ext.mss.ext1;
/* if the destination's core does not exist, then create it */
if (d_poi->main.cmema==(long)NULL)
{
adr = mss_crco(d_poi);
if (adr==NULL) return(-1);
}
/* setup destination fields */
memcpy(&d_poi->mext.pic,
&s_poi->mext.pic,sizeof(struct sap_mext));
memcpy(&d_poi->ext.trf,
&s_poi->ext.trf,sizeof(struct sap_ext));
/* invoke the appropriate processing routine */
switch((int)s_poi->mext.pic.psize)
{
case 1
flag = imb_convolve((char *)s_poi->main.cmema,
(char *)d_poi->main.cmema,
(int)s_poi->mext.pic.xsize,(int)s_poi->mext.pic.ysize,
ope_poi->ext.mss.ext2,ope_poi->ext.mss.ext3);
break;
```



```

default:
    return(-1);
}
/* if not successful, then empty destination */
if (flag < 0) mss_dico(d_poi);
/* if successful, decrement reference count */
else if (hiword(s_poi->chain.st) > 0)
    (hiword(s_poi->chain.st))--;
/* done */
return(flag);
}

```

Note that the handling of different image formats must be done also here. If there are no core functions for a certain image format the interface module should give an appropriate error exit code. If there is a core function for the given format it is called with the proper parameters. In cases when the output format is different, the interface module is responsible for the necessary transformations and preparation of destination objects. The intent is that the core functions might be completely independent from RAPIL (e.g. external, third party libraries), and thus they might not be able to handle RAPIL related data structures. But even core functions written especially for RAPIL have usually limited flexibility in data handling not to compromise efficient execution. Typically, for different pixel sizes the programmer might give separate core function implementations. The selection of such separate implementations is a task for the data-flow interface module.

A core function can implement multiple data-flow operators. In some cases the data-flow interface function must reflect this by different handling of operands and results.

6. THE DATA-FLOW WORKBENCH

It is easy to recognize the trend of spreading graphical programming paradigms. In the middle of the eighties the idea of an image processing workbench was born [8]. But the implementation came to reality only when the development of RAPIL was started. The first demonstration of the concept was created in the DOS/DR-GEM environment in 1988, later it was ported and extended into a X-Window (see [3] and [5]) and a MS Windows version. Unfortunately, because of the lack of the necessary human resources, these versions are at the prototype level only at the moment, but the first full implementation for MOTIF might be available in this year.

The workbench concept for programming was tested very early in the development of RAPIL. We found it very promising for our exploratory and experimental programming. The static data-flow graphs can be formulated in a graphical editor. The nodes of the graph can be selected from tool libraries. To support the programmer the tools are equipped with history and annotation records. Sometimes graphical representations are not satisfactory, so the workbench provides a text window, where we can edit the same graph described with a textual data-flow language. The modifications are updated in the other window automatically.

We can also initiate the compiling of the program into an MSS database. After successful compilation we can call the data-flow interpreter to execute the program.

The workbench should contain debugging aids as well. The workbench can be dynamically configured to support different personal requirements. New tools can be integrated through a guided procedure.

7. APPLICATIONS OF THE DATA-FLOW PARADIGM IN IMAGE PROCESSING

The first applications of the data-flow paradigm was introduced in [14], where it was demonstrated how to compose intensity and range data fusion algorithms into a data-flow graph represented in MSS. Later on in various student projects we tested the possibilities of creating example applications. The data-flow interpreter worked properly on quite different MS-DOS, UNIX and VMS configurations. Real applications will be developed when the next release of our Workbench will be available this year.

To develop meaningful and usable applications we need a library of OPEs. The details of the existing image processing and feature extraction libraries (providing data-flow interface modules) can be found in [12] and [13]. Here we give only a short listing of the modules:

- Image Processing OPEs: interpolation operator, normal vector operators, convolutions operators, covering curve smoothing, extremum operator, histogram modification, median filtering, normal scalar product operator, Roberts operator, Robinson operator, symmetrical hysteresis smoothing, Sobel operator, binarization, labelling, Tsuruoka's thinning, Deutsch's thinning, Hilditch's thinning, Tamura's thinning etc.
- Feature Extraction OPEs: component graph creation, geometrical element addition, point graph creation, histogram operator, Wei's edge detection operator, Martelli's edge tracking operator etc.

We have running developments for more early processing routines. One of our main focus in recent months is to provide frequency domain analysis tools as well. We want to achieve the richness of the well-known SPIDER package in about two years. Anyhow, we will include a lot of new algorithms, especially in range image processing, not included in any commercial package.

8. CONCLUSION

In this article we described a data-flow approach to develop image processing applications. The RAPIL tools supporting data-flow programming initiated a lot of research efforts and student projects (not possible to list here), and proved to be a valuable means in education as well. Some example image processing applications were already developed using the data-flow support, but this work was limited by the features of the first workbench prototypes.

The data-flow support package will be an important base for developing intelligent sensors for the IRENE project, too (details can be found for example in [11]). In that case the data-flow paradigm helps to develop blackboard architectures.

The details of data-flow programming in RAPIL is described in [9]. In recent times our team is just designing and implementing the newest version of the data-flow workbench. The next release of the RAPIL Data-Flow Support is scheduled in September 1994.

Next year we plan to start the migration from plain C to C++ as a programming language for the project. This was not possible previously, since there were no good quality C++ implementation on all of our target systems. We would also like to solve the dynamic linking problem, but in this case we have to do more research to find good portability techniques to remain consistent with our original goals.

ACKNOWLEDGEMENTS

This work was started as part of a longer project supported by the Deutsche Forschungsgemeinschaft. The first phase was done at the Technical Expert Systems and Robotics Department of the Forschungszentrum Informatik (Informatics Research Center), Karlsruhe, West-Germany in the frame of the cooperation with the Process Control Department of the Technical University of Budapest. Many thanks to Prof. Dr.-Ing. Ulrich Rembold (head of the department in the FZI Karlsruhe), to Prof. Dr. Andor Frigyes (former head of the department in the TII Budapest), and to the project leader Prof. Dr. rer. nat. Paul Levi for providing the necessary background. Special thanks to Peter Nitezki, who's many ideas about data-flow image processing were built into the RAPIL. His help was crucial to the success of this project.

This work was later continued at the Process Control Department of the Technical University of Budapest. The second phase is strongly connected to the OTKA (National Scientific Research Fund) project grant no.: 756/OTKA/-5-404/BME/. Many thanks to Prof. Dr. Péter Arató (head of the department) for providing a good background. Special thanks to Assoc. Prof. Dr. Béla Lantos (OTKA project leader) for his consultation and important comments.

I would also like to express my thanks to the French, German and Hungarian students implementing parts of RAPIL.

REFERENCES

- [1] Canon Inc., Information Systems Research Center. *VIEW-Station Software System, An Overview*, second edition, May 1990. (published through Internet).
- [2] Data Translation, Inc., Marlboro, Massachusetts, USA. *DT-IRIS SP0231 User Manual*, second edition, 17 April, 1987. Preliminary, Version V01.01.
- [3] U. Fiebrich: *RAPIL Workbench for X-Window*. Master's thesis, Budapesti MHuszaki Egyetem (Technical University of Budapest), 1993. (in Hungarian)
- [4] Grinnell Systems Corporation, San Jose, California, USA. *Grinnell Software Package User's Manual*, release 2.2 edition, 19 June, 1981. (from distribution tape)
- [5] W. Hermann: *Nutzerschnittstelle für die Bildverarbeitung*. Master's thesis, University of Karlsruhe, 1989. (in German)
- [6] Imaging Technology Inc., Woburn, Massachusetts, USA. *ITEX 100 Programmer's Manual*, first edition, 1986.
- [7] MATROX Electronic Systems Ltd., Dorval, Quebec, Canada. *PIP-EZ M DOS Software Library*, rev. 2 edition, 27 August, 1986.
- [8] P. Nitezki: "Spezifikation einer Demonstration des Werkbank-Konzepts". *Familie schneller Bildverarbeitung*, Arbeitsbericht 7, October 1986.
- [9] P. Nitezki, B. Várkonyi: "Data-Flow Image Processing", *Introduction and Programmers' Guide*. Forschungszentrum Informatik, Karlsruhe (Technical University of Budapest), release 2.0 edition, 1991.
- [10] B. Várkonyi: *Some problems of 3D vision systems based on range finding*. PhD thesis, Technical University of Budapest, August 1993. in Hungarian under judgement.
- [11] B. Várkonyi: "Design plan for the Intelligent Robot Environment: IRENE. Techn. Univ. of Budapest, Process Control Dept., 756/OTKA/-5-404/BME/ (in English), December 1991.
- [12] B. Várkonyi: "RAPIL Feature Extraction Library, Programmers' Guide". Technical University of Budapest, release 1.0 edition, 1992.
- [13] B. Várkonyi: "RAPIL Image Processing Library, Programmers' Guide". Technical University of Budapest, release 1.0 edition, 1992.
- [14] B. Várkonyi, P. Levi: "Practical range image processing". Forschungszentrum Informatik, Karlsruhe, West-Germany, May 1989.
- [14] B. Várkonyi, P. Nitezki: *An object oriented software development frame for image processing: The Raster Picture Laboratory*. Forschungszentrum Informatik, Karlsruhe, West-Germany, May 1989.
- [15] B. Várkonyi, P. Nitezki: "RAPIL Mass Storage Subsystem". Forschungszentrum Informatik, Karlsruhe, Technical University of Budapest, release 1.2 edition, 1990.

ANATOMIC SEGMENTATION, FEATURE EXTRACTION AND PSEUDOCOLORING FOR BIOMEDICAL IMAGE MEASUREMENTS

A. VLAICU, S. LUNGU, E. MECIU, and S. IACOB

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA
26 BARITIU STR, 3400 CLUJ-NAPOCA, ROMANIA

1. INTRODUCTION

Mankind's principal means of interacting with his environment is visual. As age diminishes the acuity of our senses, we use eyeglasses commonly, hearing aids occasionally and practically never any prosthetic aids for any of the remaining senses (an ancient Chinese proverb says that a picture is worth 1000 words). Many scientific instruments directly produce pictorial images or have some type of graphics display (to present a spectrum, for instance), expecting the operator to be able to extract the usable information more rapidly than from a list of numbers.

Recognising, counting and measuring the size, shape, position, density and other similar properties of particular objects in an image can be done by computers relatively quickly and with excellent reproducibility.

A large variety of instruments produce images in suitable forms for acquisition and analysis and computerized measurement is used to extract specific information from images much more accurately and reproducibly than a human can. In normal situations humans rarely need to measure an object in an image, but they can interact with their environment to bring a comparison object or ruler into play. Computer image measurement is less easily distracted from what is important by trivia in images and is better than a human observer at paying attention to all the details. On the other hand, humans are better at recognizing objects, often based on very incomplete or unconventional images, capability that is much harder to program in the computer. The process of image measurement involves a huge reduction in the amount of data, by selecting from the original image the objects or features that are important. An original image may represent a million separate points stored in the computer, but the desired information may be as simple as, for example, the number of white blood cells on a slide, the size (area of a surface, volume, etc.) or the presence of a tumour in an X-ray image. This selection and

reduction of the data amount by ignoring irrelevant information is at the heart of image analysis and measurement. The human eye has sensors with peak sensitivities for red, green and blue light and the spectral response is frequency dependent. Combinations of those colors are interpreted as a continuous range of color and experiments stated that the human eye can distinguish only a limited number of gray-levels (20-30 levels in monochrome images under good conditions) and that, at the same time, the number of different colors available for display is much more important [6]. That is why, an important aim for the processing of biomedical images is to enlarge the visual perception of the human eye by pseudocoloring the gray-level images obtained from most of the instruments used for biomedical visualization.

2. ANATOMIC SEGMENTATION AND FEATURE EXTRACTION OF BIOMEDICAL IMAGES

Further, a set of representative biomedical images will be considered. Fig. 1 shows the image of an animal tadpole obtained with an optical microscope, Fig. 2 presents an echograph image, Fig. 3 a X-ray radiography and Fig. 4 a CT reconstructed 2D image. The information content of each image is slightly different and so are also the relevant features that we want to extract. That's why the processing of these images has to be appropriate to the proposed goal and to the features in the image.

Anyway, some common features of these images make that a series of algorithms for the digital processing of medical images as those presented in Figs. 1-4 are similar.



Fig. 1.

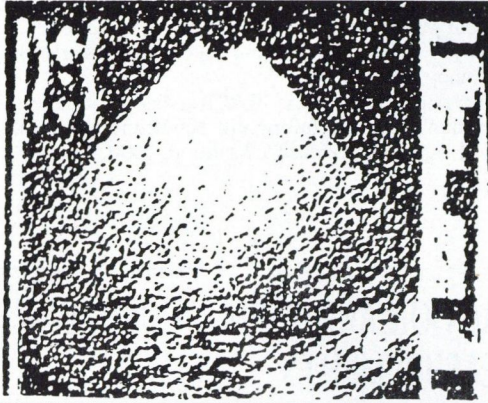


Fig. 2.



Fig. 3.

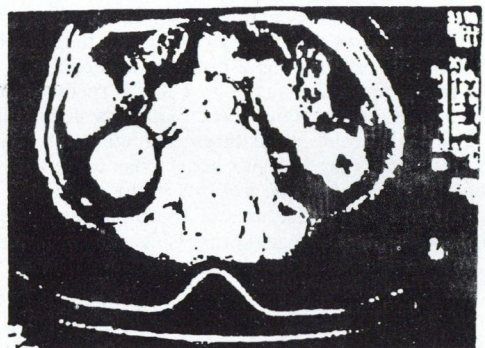


Fig. 4.

In order to recognize or measure objects in images, it is necessary to distinguish them from their surroundings, that is to separate a region of contiguous pixels that share the same property. The human eye has within the retina a certain number of cells, known as amacrine cells, which connect to several retinian cells in order to extract higher level information, for instance to detect an edge oriented in a particular direction. The responses of several cells are combined, some positively and some negatively: when a uniform region is viewed, there is no output and only when an edge lies in a proper alignment, the cells produce a significant signal for the brain. The most important feature of the method is inhibition: the positive and negative connections on the logic cells allow neighbouring pixels to be considered [2].

The same technique is used in spatial operations upon images, performed for the enhancement of the contours or for the edge extraction. For example the Laplacian kernel operator presented below, has negative coefficients corresponding to all the surrounding pixels, representing an inhibition:

$$H = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

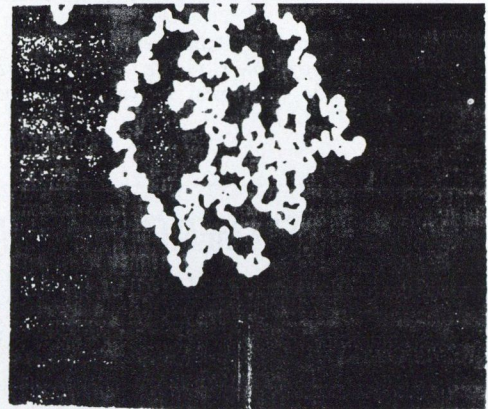


Fig. 5.

For extracting the edges, several different algorithms can be used. In Table 1, some of them are presented, along with the value of the coefficients used in experimental studies. Combination of these algorithms are used in order to extract the contour of an object in the image [1], [3]. Fig. 5 presents the contour obtained from the image presented in Fig. 1 by applying the horizontal Sobel operator and, after that, the vertical Sobel operator on the image. By combining the images in Fig. 1 and 5 a composite image is obtained (Fig. 6) featuring particular regions of interest in the original image.

There are also other edge-detecting operators besides Sobel and Kirsch. Marr (1982) found that the eye uses smoothing techniques in which the smoothing distance varies (the distance is roughly equivalent to the maximum size of features which are to be ignored). By forming different smoothed images using different scales, edges of different size structures can be extracted from the image, by taking the difference of two successive images. There is an alternative way to get virtually the same result and this is the so-called LOG procedure, that is the Laplacian of a Gaussian. If the smoothed image is subjected to the Laplacian operator, the edges are selected, as the smoothing operation has suppressed the points and lines. The weights of the two operators can be combined into a single operation, in order to obtain the LOG image and the zero-crossings of the resulting image mark the edges of features [2].

Table 1.

Operator	a ₁₁	a ₁₂	a ₁₃	a ₂₁	a ₂₂	a ₂₃	a ₃₁	a ₃₂	a ₃₃
Directional gradient E	1	1	1	1	-2	1	-1	-1	-1
Directional gradient N-E	1	1	1	1	-2	-1	1	-1	-1
Directional gradient S-W	1	1	-1	1	-2	-1	1	1	-1
High-pass filter 1	0	-1	0	-1	5	-1	0	-1	0
High-pass filter 2	0	-1	0	-1	4	-1	0	-1	0
Laplacian	-1	-1	-1	-1	9	-1	-1	-1	-1
Diagonal Laplacian	-1	0	-1	0	4	0	-1	0	-1
Horizontal Laplacian	0	-1	0	0	2	0	0	-1	0
Vertical Laplacian	0	0	0	-1	2	-1	0	0	0
Horizontal Prewitt	-1	-1	-1	0	0	0	1	1	1
Vertical Prewitt	1	0	-1	1	0	-1	1	0	-1
Horizontal Sobel	1	2	1	0	0	0	-1	-2	-1
Vertical Sobel	1	0	-1	2	0	-2	1	0	-1
Horizontal Kirsch	-3	-3	3	-3	0	3	-3	3	5
Vertical Kirsch	5	5	5	-3	0	-3	-3	-3	-3

Table 2.

Grey level	Corresponding color	Color coefficients			Grey level	Corresponding color	Color coefficients		
		R	G	B			R	G	B
0	black	0	0	0	32	light brown	63	31	16
1	dark blue	0	0	1	33	dark orange	63	31	0
2		0	0	2	34		63	33	0
3		0	0	3	35		63	35	0
4		0	0	4	36		63	37	0
5		0	0	5	37		63	39	0
6	dark brown	15	2	0	38		63	41	0
7		17	3	0	39		63	43	0
8		19	4	0	40		63	45	0
9		21	5	0	41		63	47	0
10		23	6	0	42		63	49	0
11		25	7	0	43	light orange	63	51	0
12		27	8	1	44	dark yellow	63	53	0
13		29	9	1	45		63	63	0
14	dark red	26	0	0	46		63	63	2
15		29	5	5	47		63	63	4
16		32	5	5	48		63	63	6
17		35	5	5	49		63	63	8
18		38	5	5	50		63	63	10
19		41	5	5	51		63	63	12
20		44	5	5	52		63	63	14
21		47	5	5	53		63	63	16
22		50	5	5	54		63	63	18
23		53	5	5	55		63	63	20
24		56	5	5	56		63	63	22
25	red	59	5	5	57		63	63	24
26	light brown	48	24	9	58		63	63	26
27		53	26	11	59		63	63	28
28		55	27	12	60	light yellow	63	63	30
29		57	28	13	61		61	61	61
30		59	29	14	62		62	62	62
31		61	30	15	63	white	63	63	63

With most of the derivative edge-finding methods, a problem that we must deal with is the continuity of the edges. Due to the fact that the methods discussed use purely local comparisons between pixels, there is no guarantee that the points will merge into connected lines. This is particularly the case when there is side lightening or nonuniform illumination, or if the image is very noisy.

There are several approaches to connecting incomplete lines in continuous contours, including methods that belong to scene understanding. These are computationally very demanding and this paper will not deal with them. The images we usually wish to measure are simpler in organization and we will assume that boundaries revealed in the edge image lie in the same image plane and should be connected to form a simple tessellation or

set of feature outlines. The connection of gaps between line segments can be made by curve fitting with the simplest case of straight lines or the extension of the method to quadratic curves. The zero-crossing of the LOG method is guaranteed to have a continuous path. Another method is to generate contour lines. It is helpful to imagine the image as having an elevation proportional to the pixel brightness. Then, the image is equivalent to a conventional map and contour lines of selected brightness are analogous to isoelevation contours marked on topographic maps. If there are two regions at different elevations, there must be somewhere between them a location along the contour line at any intermediate elevation. The boundary line can pass between pixels and may either be located by interpolation or may be arbitrary assigned to the nearest pixel.

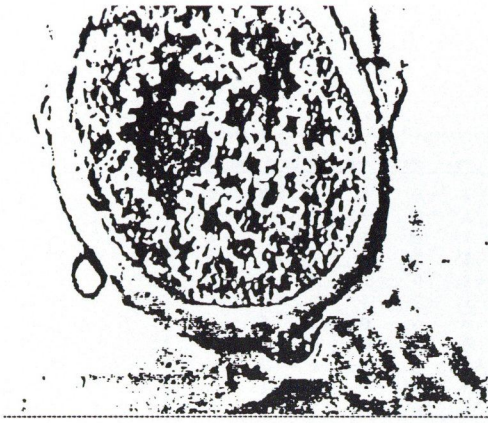


Fig. 6.

Region growing is a method similar, in a way, to the edge following, in spite of the fact that it deals rather with threshold methods. Starting with any interior pixel in the image, each neighbouring pixel is examined to decide whether it is part of the same feature. The criteria used are quite varied but, more often, the brightness value of a pixel is used. If the difference between the new pixel and the neighbour is less than an arbitrary limit, the pixel is considered to belong to the feature [4], [7].

The use of brightness discrimination to select pixels belonging to features of interest is a convenient method of converting a gray-scale image to a binary one. Discrimination by thresholding is much more efficient than any edge detection or region growing method because it works on the entire image at once and the resulting binary image is a pixel-based representation of features of interest that are interesting for most measurement operations. Thresholding on original gray-scale image in which the features of interest are different in brightness than others or background is an important step for the preparation for further measurements (surface, perimeter, etc.) of features. This needs not to be performed on original gray-level images. Smoothing operations for the reduction of noise or background leveling methods, subtracting one image from another, produce derived images from the original one. These can be discriminated to produce better results than those obtained with the original [5]. Background leveling by subtracting or dividing an image by another is intended to modify the brightness values of pixels specifically, so that similar features in different locations would have the same brightness.

Due to the fact that most of the biomedical images are low-contrast ones, the histogram modelling is a powerful technique for image enhancement. The problem is to modify the image so that its histogram has a desired shape. This is useful in stretching the low-contrast levels of images with narrow histograms [1], [3].

In histogram equalization, the goal is to obtain a uniform histogram for the output image. Considering a pixel's value $v > 0$ as a random variable with a continuous probability density function $p_v(x)$ and a cumulative probability distribution:

$$F_v(x) = P[v \leq x]. \quad (1)$$

Then, the random variable

$$v' = F_v(x) = \int_0^x p_v(x) dx \quad (2)$$

will be uniformly distributed over $(0, 1)$. Supposing the input with N gray-levels $x_i = 0, 1, \dots, N-1$ with probabilities $p_v(x_i)$. These probabilities can be determined from the histogram of the image giving the number of pixels $h(x_i)$ having a gray-level value x_i [1]. In this case:

$$p_v(x_i) = \frac{h(x_i)}{\sum_{i=0}^{N-1} h(x_i)} \quad i = 0, 1, \dots, N-1. \quad (3)$$

The output image v' , assumed to have also N gray-levels, is given by:

$$W = \sum_{x_i=0}^v p_v(x_i) \quad (4)$$

$$v' = \text{Int} \left[\frac{W - W_{\min}}{1 - W_{\min}} (N - 1) + 0.5 \right], \quad (5)$$

where $W_{\min}$, is the smallest positive value for W obtained from (4). The Eq. (5) simply requantizes uniformly the set of values $\{W_k\}$ into $\{W'_k\}$. The results obtained by histogram equalization on biomedical images are very useful from the point of view of the other processing methods (thresholding and edge detection) used for image measurements, as well as for the better understanding and interpretation of the output image. Figs. 7, 8, 9 present the result of histogram equalization applied to the images in Figs. 1-3.



Fig. 7.

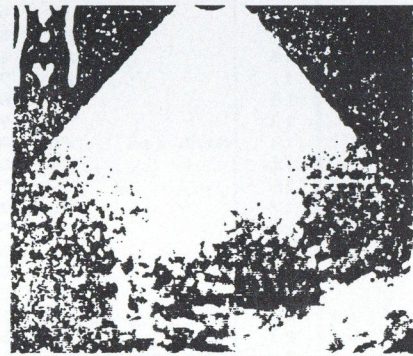


Fig. 8.

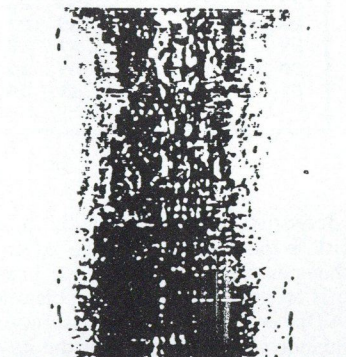


Fig. 9.

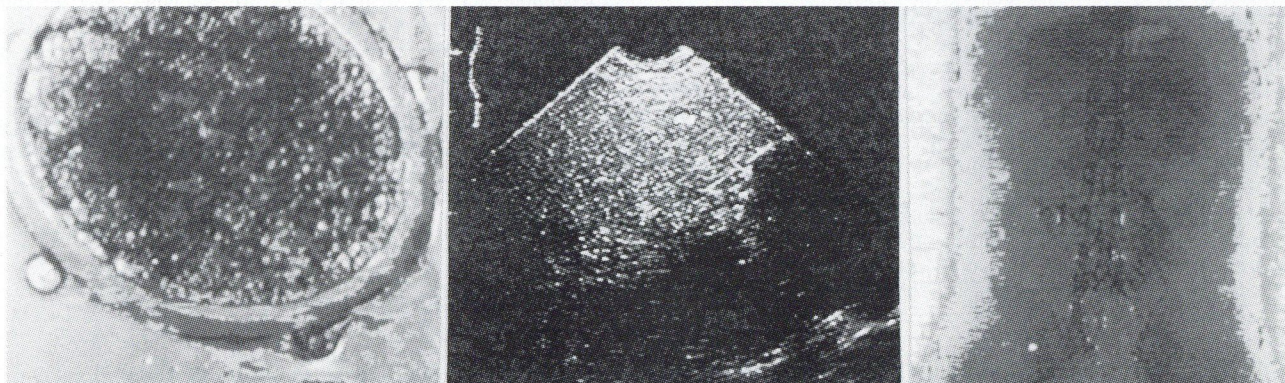
3. PSEUDOCOLORING THE BIOMEDICAL IMAGES

The human eye is capable of distinguishing a rather small number of gray-levels (25-30, under good conditions) but can easily distinguish hundreds or thousands of colors [6]. False-color or pseudo-color display of gray-level images is therefore a powerful way to accentuate small brightness variations [5]. If the original image is digitised and stored with 256 brightness levels, it is common to use 8 bit DAC's for each colour. The number of different colours obtained in this way, hence, effect the intelligibility of the image. That is why the use of 6 bit DAC's, allowing us to obtain a number of 64 different colours for the corresponding 64 gray-levels in the original image is quite satisfactory.

False or pseudo color display is used especially for astronomical images. They can emphasize gradients and communicate a great deal of information. When misused, however, false color or

pseudocolor can seriously confuse the eye, hiding the original information present in the image [8]. It is appropriate to use a smooth variation of color assignments versus brightness, in order to minimize this problem. This can be done by selecting a series of colors that follow some smooth line on the color chart, technically known as the chromaticity diagram. Some often-used color sequences are: (a) a rainbow, proceeding from deep red through orange, yellow, green, cyan, blue and violet, all of them with the same brightness value; (b) a spiral starting at dark violet and proceeding around the diagram to red, while simultaneously increasing the white component of the color; (c) a color temperature variation from dark red through orange and yellow to white, and beyond to blue-white [2], [9], [10].

We propose a different color palette for the representation of gray-level images. The gray-levels and the values for the R, G and B values as corresponding color assignment is presented in table of pseudo-coloring the images in Figs. 7, 8, 9 is Figs. 10, 11, 12.



Figs. 10, 11, 12.

4. CONCLUSIONS

Anatomic segmentation and feature extraction of biomedical images is a combination of different algorithms that modify the original content of data in the image in order to make it more appropriate for visualization or measurement. The algorithms and their implementation is different for different type of medical images, depending on its specific features and on the proposed goal.

By choosing an appropriate color palette for pseudo coloring

gray-level images, the enhancement of human eye perception is obtained. The palette, proceeding from black and dark blue, through dark brown, deep red, light brown, orange, yellow to white is better discernible by the human eye than the corresponding 64 gray-level range but, on the other hand, has certain similarities with it from the point of view of the sensation given to the eye. The smooth variation in color assignments, close to the gray-level transitions in the original image allow a better perception of the low-contrast details without affecting the intelligibility of the image.

REFERENCES

- [1.] A. K. Jain: *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, N.J., 1989.
- [2.] J. C. Russ: *Computer-Assisted Microscopy. The Measurement and Analysis of Images*, Plenum Press, New York, 1990.
- [3.] J. C. Russ: *The Image Processing Handbook*, CRC Press, Boca Raton, Florida, 1992.
- [4.] J. S. Lim: *Two-Dimensional Signal and Image Processing*, Prentice Hall, Englewood Cliffs, N.J., 1990.
- [5.] A. E. James, J. H. Anderson, C. B. Higgins: *Digital Image Processing in Radiology*, Williams & Wilkins, Baltimore, 1988.
- [6.] A. E. Burgess, R. F. Wagner, R. J. Jennings, H. B. Barlow: "Efficiency of human visual signal discrimination", *Science* 214, 1981.
- [7.] K. Assmann, K. H. Hohne: "Investigation of structures and operations for medical image data bases", *Proc. SPIE* 418, 1983.
- [8.] A. K. Jain, A. Nassir, D. Nelson: "Multispectral Feature Display via Pseudo Coloring", *Proc. SPSE*, 1982.
- [9.] G. Brandaway: "A procedure for optimum choice of a small number of colors from a large palette for color imaging", *Electronic Imaging 87*, San Francisco CA, 1987.
- [10.] O. C. Bouman, M. Orchard: "Color image display with a limited palette size", *Proc. SPIE. Conf. Visual Commun. Image Processing*, Philadelphia, Pa, Nov. 8-10, 1989.

SURFACE INTERPOLATION TECHNIQUE AND 3D SEGMENTATION FOR RECONSTRUCTING 3D OBJECTS FROM SERIAL SECTIONS

A. VLAICU, S. LUNGU, and C. RUSU

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA
26 BARITIU STR, 3400 CLUJ-NAPOCA, ROMANIA

1. INTRODUCTION

Due to the progress in imaging instruments and graphic workstations, the access to the third dimension in medical imaging has already provided valuable information in various fields such as reparative surgery, radiotherapy treatment planning, neurosurgery, the study of structural and morphological characteristics of organs in biomedical sciences, etc. This has made three-dimensional (3D) medical image processing and display a rapidly emerging field of research [2], [3].

The main task in a computer-aided reconstruction process consists of extracting the regions of interest (ROI) or their contours in all sectional images and in forming surfaces and volumes between the contours of adjacent slices. If the interslice distances between successive contours are small, the 3D object can be displayed by stacking up the successive contours in the correct order. If the interslice resolution is larger than the resolution within the slice, the anisotropy in the voxel dimensions requires carefully designed interpolation schemes.

Since the information provided for reconstructing a 3D object is contained in the serial sections only, the smoothness of the reconstructed slices is a major concern [2], [3]. Various surface interpolation algorithms have been proposed to satisfy this criterion. The technique of lofting utilizes parametric curves for surface representation. In this approach, sectional curves are represented by uniform B-splines manually and then surface is interpolated between sections by cardinal splines. Other methods approximate the closed bounding surfaces between two contours extracted from consecutive slices, by a collection of triangular patches or reconstruct the surface of an object by a 3D volumetric scene description [5]. The scene description is obtained from a volume-segment structure which is characterized by a collection of planar slices. In order to extract the information from the given planes, a set of associated contours in consecutive slices is identified first. A triangulation process, producing local bounding patches between every pair of consecutive contours follows. The reconstructed object is formed by coalescing the adjacent triangular patches into polygonal facets such that the orientation of constituent triangular patches are preserved [6].

The segmentation process has been approached in two different ways. In the first, one tries to find the contours of the object in the section, while in the other one looks for isodensity objects. The first approach is limited by the large number of contours with respect to the signal-noise ratio. In the second one, in the 3D case some degree of operator intervention is required and voxel anisotropy may create artificial links between slices, that are difficult to eliminate [6], [7].

In the paper an interpolation method and segmentation algorithms implemented on an IBM-PC 486 DX computer are presented.

The goal of the computation was to obtain a 3D isotropic data volume for the display of different views according to the ROI and, on the other hand, to realize a 3D representation of an object by segmentation and volume rendering.

2. MORPHOLOGICAL INTERPOLATION

Interpolation methods based on morphological properties of an object is preferable to gray level interpolation methods. The goal is to create an intermediate slice by operating on two adjacent slices and trying to avoid artifacts that appear mostly at boundaries, due to the anisotropy of voxels [6], [11]. Suppose we have n axial slices with 640×480 pixels each, with pixel dimensions of 0.25 mm. The distance between slices being 1 mm, in order

to obtain the isotropy of the voxels we have to reconstruct three intermediate slices and to insert them between the i and $i+1$ slices in the 3D volume of data.

The algorithm proposed has two successive steps: first we reconstruct by interpolation an intermediate slice j and then, using slices i and j and, respectively j and $i+1$, we reconstruct slices $j-1$ and $j+1$. Figs. 1 and 2 illustrate the algorithm.

Assuming that the n axial slices are available, each of them having k lines and m columns, the reconstruction of the intermediate ones can be made either by reconstructing sagittal or coronal slices, using successively lines or columns from the n axial slices. Considering the first line from the i and $i + 1$ slices, the reconstruction of the intermediate slice line j is made by linear interpolation for the regions containing pixels within a certain gray-level domain Δ_q .

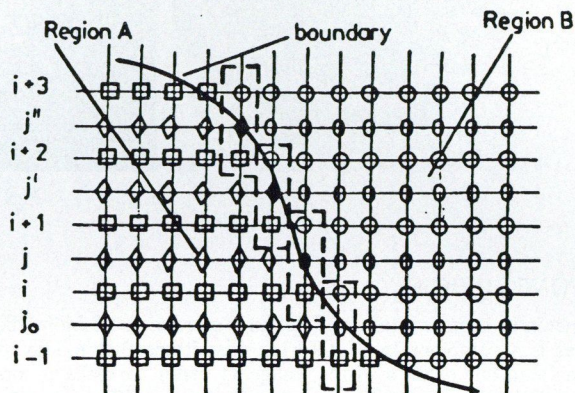


Fig. 1. Interpolation for different gray-level regions and boundary — first step

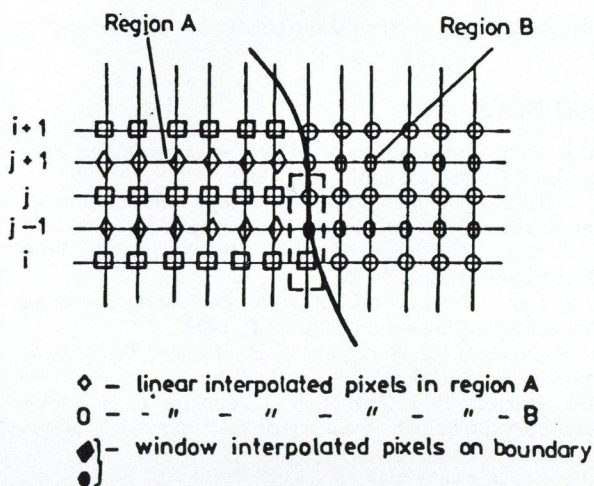


Fig. 2. Interpolation of different gray-level regions and boundary — second step

In Fig. 1 two regions A and B with different gray-levels and the boundary between them are considered. First, a linear interpolation of the pixels inside the regions A and B is performed, resulting the pixels from lines j_0, j, j', j'' , except for the pixels near the boundary. The algorithm for the vertical interpolation is applied only for the pixels having the gray-level

$$g_p = g_A \pm \Delta g \text{ or } g'_p = g_B \pm \Delta g.$$

Near the boundary, the linear interpolation is not useful because the contour of the two regions has to be preserved in the intermediate slices too. That is why, for the boundary, a window of 3×3 pixels is taken into account and the number of pixels n_A and n_B having the gray-level g_p and g'_p respectively is computed. If $n_A > n_B$, the value g_p will be allocated to the boundary pixel and if $n_B > n_A$, the value g'_p will be allocated. If $n_A = n_B$ (it is possible that within the window, the value of a pixel to be uncertain) a 5×5 pixel window is taken into account and the procedure continues in the same way.

Thus, the pixels belonging to a certain region in the reconstructed slices will have gray-levels obtained by linear interpolation and the pixels in the boundary regions will have a gray-level computed as the mean value of the majority of the neighbouring pixels, having gray-levels within a certain domain $\pm \Delta g$. An illustration of the boundary pixels interpolation is made in Fig. 3. If we assume that interpolation is done on line j from left to right, we can see that pixels 5 and 6 from the intermediate line cannot be linearly interpolated, because the gray-levels of the corresponding pixels in lines i and $i+1$ are different. Consequently a window of 3×3 pixels around pixel 5 is considered and, within this window, 5 pixels have the gray-level close to g_A and 2 have g_B (pixel 5 and 6 on line j have yet uncertain gray-level). The mean gray value of the 5 pixels having gray values close to g_A will be allocated to the 5-th pixel in line j . The same rule is followed for the 6-th pixel in line j and for the 7-th and 8-th pixel in line j_0 .

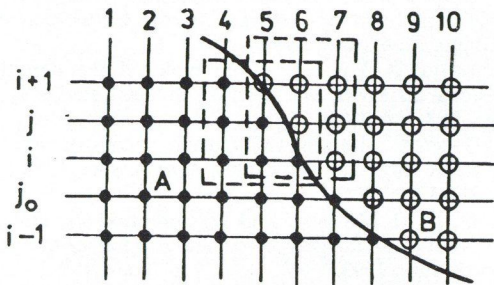


Fig. 3. Boundary pixels interpolation



Fig. 4.



Fig. 5.



Fig. 6.

The first interpolation algorithm applied produces the image in Fig. 4, containing 61 lines (the 31 original and 30 intermediate lines). The second interpolation, for the 61 lines in Fig. 4 produce the image in Fig. 5, containing 121 lines. Thus, a sagittal slice is obtained. The interpolation procedure can be followed even further (Fig. 6, with 241 lines) but the result is a slice with different vertical and horizontal resolutions (the vertical resolution is 0,125 mm and the horizontal one, equal to the slices resolution, is 0,25 mm).

3. BOUNDARY-BASED SEGMENTATION

Using the morphological interpolation algorithm presented in section 2 a 3D isotropic data volume is obtained. This information is useful for surface rendering, that is for the visualization of either axial or sagittal and coronal slices. For volume rendering, the separation of certain regions of interest is, however, necessary.

If various tissue regions within the part of the body scanned can clearly be distinguished, the boundary of the region of interest with the surrounding tissue region can be obtained by thresholding algorithms. At such a boundary, it is reasonable to expect a discontinuity in the distribution of the density. The main idea in boundary-based segmentation is to detect where these discontinuities occur and to locate the boundary.

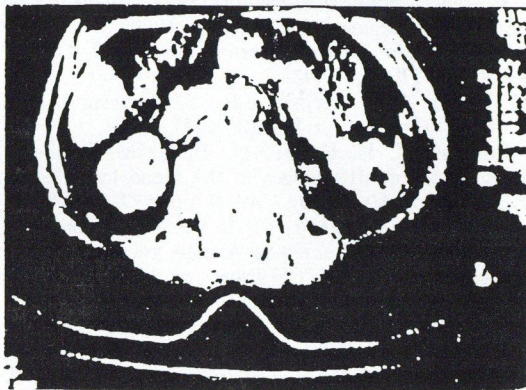


Fig. 7.



Fig. 8.

Segmentation methods use edge detection operators in order to obtain the close boundary of the ROI. Either Sobel or Kirsch operators, or the LOR (Laplacian of a Gaussian) operator can be used. Further, once the ROI is bordered, an algorithm for filling the closed contour of the ROI with pixels having a nonexistent value in the rest of the processed image is used. In this way, a thresholding procedure in order to separate the ROI from the rest of the image is easy to carry on. Finally, by combining the original slice with the thresholded image, the segmented image of the ROI, having the original values for the inside pixels, is obtained. Fig. 7 presents an axial slice in the abdominal region. A vertical Sobel operator followed by a horizontal Sobel operator is applied, in order to detect the contours in the image (Fig. 8). Furthermore, the ROI are filled with white-level pixels and an algorithm preserving the values of the pixels inside the contours of the ROI and erasing the others is applied (Fig. 9). For volume

rendering of the ROI, the slices obtained in this way can be used or they can be pseudo coloured according to anatomic atlases, for a better resemblance with real body structures.

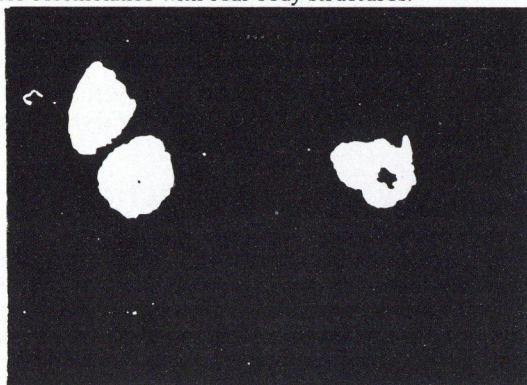


Fig. 9.

REFERENCES

- [1] A. K. Jain: *Foundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, N.J., 1989.
- [2.] J. C. Russ: *The Image Processing Handbook*, CRC Press, Boca Raton, Florida, 1992.
- [3] J. K. Udupa, G. T. Herman, eds.: *3D Imaging in Medicine*, CRC Press, Boca Raton, Florida, 1991.
- [4] M. Bomans, K. H. Hohne, U. Tiede, M. Riemer: "3D Segmentation of MR Images of the Head for 3D Display", *IEEE Trans. on Med. Imag.*, Vol. 9, pp. 177-183, June 1990.
- [5] W. C. Lin, C. C. Liang, C. T. Chen: "Dynamic elastic interpolation for 3D medical image reconstruction from serial cross sections", *IEEE Trans. on Med. Imag.*, Vol. 7, pp. 225-232, Sept. 1988.
- [6] W. C. Lin, C. C. Liang, C. T. Chen: "Improvement on dynamic elastic interpolation for 3D medical image reconstruction from serial cross sections", *IEEE Trans. on Med. Imag.*, Vol. 9, pp. 71-84, March 1990.
- [7] W. C. Lin, C. C. Liang, C. T. Chen: "A new surface interpolation technique for reconstructing 3D objects from serial cross sections", *Comput. Vision, Graphics, Imag. Proc.*, Vol. 48, pp. 124-143, 1989.
- [8] K. D. Tounies, J. K. Udupa, G. T. Herman: "Registration of 3D objects and surfaces", *IEEE Comput. Graph. and Appl.*, Vol. 10, pp. 52-62, May 1990.
- [9] D. N. Levin, X. Hu, a.o.: "The brain: Integrated three-dimensional display of MR and PET images", *Radiology*, Vol. 172, pp. 783-789, 1989.
- [10] J. G. Colsher: "Iterative 3D Image reconstruction from tomographic projections", *Comput. Graph. Image Proc.*, No. 6, pp. 513-520, 1977.
- [11.] M. Joliot, B. M. Mazoyer: "3D segmentation and interpolation of MRI brain images", *IEEE Trans. on Med. Imag.*, Vol. 12, No. 2, June 1993.

4. CONCLUSIONS

The morphological interpolation algorithm and the boundary-based segmentation method were implemented on a IBM-PC 486 computer. The axial slices used for implementation were 640x480 images. The results of both the interpolation and segmentation were quite satisfactory both from the point of view of the computational time required and the precise definition of contours and surfaces. As such, we actually think of them as a preliminary step toward an automatic analysis and measurement of three dimensional images.

COLOR IMAGE ARCHIVIZATION FOR MEDICAL PURPOSES

M. DOMAŃSKI and M. BARTKOWIAK

POLITECHNIKA POZNAŃSKA, INSTYTUT ELEKTRONIKI I TELEKOMUNIKACJI
UL. PIOTROWO 3A, 90-965 POZNAŃ, POLAND

1. INTRODUCTION

The paper deals with the problem of the palette representation (8 bit/pixel) of full-color (24 bit/pixel) images. This very common problem [1-3] is related to color image visualization using popular and cheap Super Video Graphic Adapter. Nevertheless the principal motivation of this research is related to medical applications. In fact, we deal here with two image compression tasks:

- palette design (weakly lossy),
- lossless compression of an image with a palette.

We deal primarily with microscopic color images of slides of some human tissues. The pictures are produced by an electronic camera (usually a CCD device) and then grabbed into a computer by a kind of a frame grabber. The problem of storing of such images is related to their substantial size of at least 750 kB causing severe problems in large hospitals where many of the data are retrieved very rarely or never. Therefore, from the technical and economic point of view, efficient programs to compress such images are very important. Nevertheless the medical applications require lossless techniques, i.e., techniques that remove only redundant information preserving any other information.

The very important issue is related to simplicity of the algo-

rithm which should be easily implementable on simple and cheap personal computers.

2. PALETTE DESIGN

The initial palette of 16 millions of colors is to be reduced to an image-dependent palette of 256 colors. The input image has the RGB components because the color cameras, both CCD and tube, produce such signals. Similarly, the color cathode ray tubes, being still the most popular color displays, also need the RGB signals. It seems to be quite lucky situation because probably our visual system is also based on the sensors of red, green and blue. The problem is that some distance between two points in the RGB color space corresponds to a unnoticeable color difference while the same distance in the other part of this space corresponds to a quite significant difference in color sensation. Moreover the RGB system is somewhat difficult to handle because the values of RGB are related to color sensation in a rather complicated way. Therefore the RGB colour system is definitely improper for image processing and compression tasks.

After some experiments with sample images we decided to use the IHS (Intensity-Hue-Saturation) coordinates [4], [5]. As

the desired reduction of the palette is very high the process is performed in two steps:

- Vector quantization of the IHS coordinates.
- Recursive reduction of the number of palette entries.

The first step consists of quantization in the IHS color space, i.e., the RGB vectors corresponding to all pixels have to be converted to the IHS-values. The intensity component is uniformly quantized but the quantization step is set adaptively with respect to the actual dynamic range of intensity in the processed picture. Nonuniform vector quantization is performed on the hue-saturation plane, where hue is finer quantized as saturation grows.

Then a palette consisting of all quantized colors existing in a given image is created and simultaneously the respective histogram is calculated. The obtained palette has usually much more than 256 entries, i.e., a further reduction in the second step is needed.

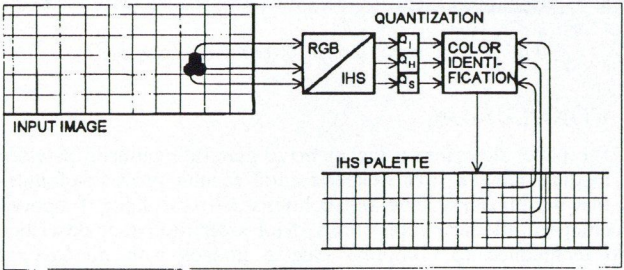


Fig. 1. The palette creation

A color is removed from the palette if it is close to another one. At first the colors corresponding to very small entries to the histogram are removed, i.e., a color vanishes if it is close enough to another more frequent color from the palette. The color differences are measured according to the taxi-cab norm in the IHS space. A color is reduced from the palette if the color difference is smaller than some predefined error. The respective pixels are labelled with the color which is the closest member of the palette. The process begins with the lowest histogram values and is interrupted when satisfactory reduction of the palette is obtained. If such a reduction is not achieved, the previous step is repeated with some higher threshold and starting with the lowest values of the histogram.

3. LOSSLESS COMPRESSION OF AN IMAGE WITH PALETTE

Using the technique described above, we get an 8-bit/pixel-image where the color of each pixel is defined as one from the 256-entry palette. Such an image is very weakly correlated and therefore the direct application of classic techniques of entropy coding, like Huffman or Lempel-Ziv, or predictive coding give poor results.

Here we describe two lossless techniques proposed by the authors.

3.1. Predictive coding of components

The first technique is based on predictive coding of the intensity component which is highly correlated. In the cases where the intensity difference does not define the palette entry uniquely, the difference in hue or even in saturation are additionally coded. A special codebook is constructed. It consists of symbols denoting single numbers (corresponding to an intensity difference), couples of numbers (corresponding to the difference in both intensity and hue) or even three numbers each (corresponding to differences of I, H and S). All of them are coded in the same way using image-dependent Huffman codes.

3.2. Compression using palette ordering

An alternative lossless compression technique for palette images arises from the well-known idea of lossless differential predictive coding (DPCM) augmented by Huffman coding. This standard technique can be adopted for images with appropriately

sorted palettes.

The problem of the optimum sorting of the palette is surely quite difficult. Here, we propose a relatively simple but sub-optimum algorithm.

We aim at a such an ordering that the changes in color number between adjacent pixels exhibit small values with higher probability than large values. The more spiky is such a two-dimensional probability density function the higher is the lossless compression obtained using DPCM and Huffman coding. Therefore the palette should be ordered in such a way that "short" transitions are significantly more probably than "long" transitions.

In order to obtain such an ordering we use the weighted graph representation of the image statistics, where the nodes represent palette entries, and the edges weights are the numbers of respective color transitions. The algorithm consists of one stage graph preparation and eight iterations of node reduction which correspond to ordering of single entries to couples, then couples to the sets of four nodes and so on.

The algorithm:

Graph preparation:

- Create a directed weighted graph with 256 modes. Remove all the self-loops. Iterations for $n = 1 \dots 8$
- Find 2^{8-n} nonincident edges with maximal weights. Denote the beginning node as n_i and the ending node as k_i ($i = 1 \dots 2^{8-n}$). The edge weight is $d(n_i, k_i)$.
- Increase all the edges ending at n_i of $d(n_i, k_i)$. Increase all the edges starting from k_i of $d(n_i, k_i)$.
- Remove all the edges starting from n_i . Remove all the edges ending at k_i . Remove the edge from k_i to n_i .
- Fix the (k_i, n_i) order in the output palette. Join the nodes k_i and n_i .

Repeat steps 2...5 for next n until 8.

4. EXPERIMENTAL RESULTS

The computer programs based on the techniques described are successfully running on some computers in Medical Academy of Poznań.

The first step (palette reduction) gives the compression of 1:3. The algorithm to compress palette-images using predictive component coding gives an additional compression of range of 1:2. The alternative compression technique using differential and Huffman coding of an image with the appropriate ordered palette gives the data reduction of 1:4.

Some results for the medical images (cf. Fig. 2 and 3) are given in the Table 1.

Table 1. Experimental results for some typical images

Image	Opinion score [range 1..5]	SNR [dB]	Error δ_I [%]	Error δ_H [%]	Error δ_S [%]	Data reduction using predictive component coding [ratio]	Data reduction using palette ordering [ratio]
test1	4	26.9	2.3	10.7	1.4	1:2.12	1:4.31
test3	5	23.4	4.8	7.2	3.7	1:1.79	1:3.19
test4	5	29.1	2.9	10.5	1.3	1:1.91	1:3.31
test5	5	29.6	2.6	7.0	1.4	1:2.15	1:4.60

Where the following definitions have been used

$$SNR = -10 \log \frac{1}{N} \sum_{n=1}^N \frac{||\hat{x}_n||^2}{\max_n ||x_n||^2},$$

N — number of samples (pixels) in the image,
 $|| \cdot ||$ — Euclidean norm in the RGB space,
 $x_n, \hat{x}_n$ — RGB-vectors representing image samples before and after palette definition.

The errors are defined as follows:

$$\delta_I = \sqrt{\frac{1}{N} \sum_{n=1}^n \frac{(I_n - \hat{I}_n)^2}{\max_n(I_n)^2}}, \quad \delta_H = \sqrt{\frac{1}{N} \sum_{n=1}^n \frac{(H_n - \hat{H}_n)^2}{\max_n(H_n)^2}},$$

$$\delta_S = \sqrt{\frac{1}{N} \sum_{n=1}^n \frac{(S_n - \hat{S}_n)^2}{\max_n(S_n)^2}}.$$

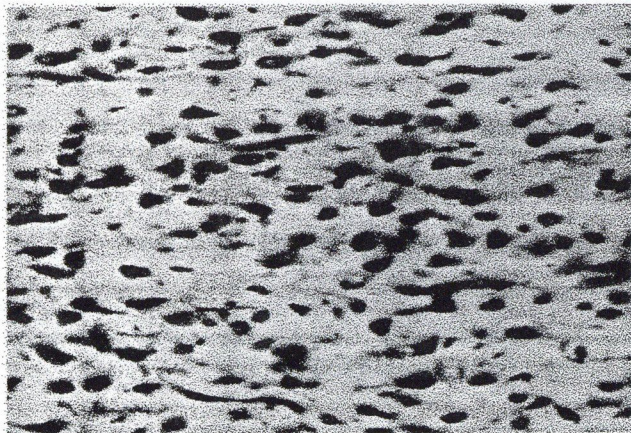


Fig. 2. The "test1" image (intensity component)

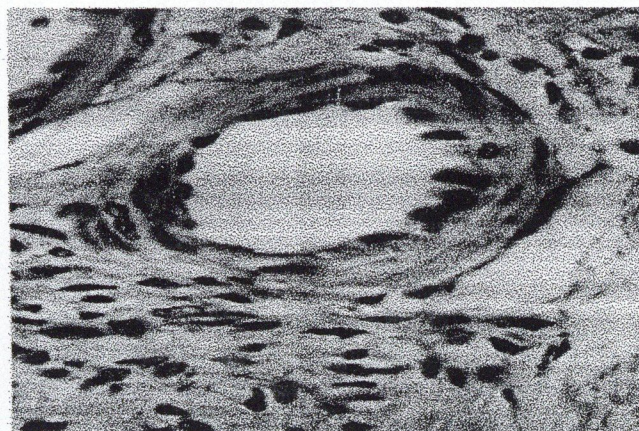


Fig. 3. The "test5" image (intensity component)

5. CONCLUSION

The paper describes a technique to generate reduced palettes of medical images. The experimental results prove usefulness of this weakly lossy (see the columns 3–6 in Table 1 above) technique in medical applications. Moreover the paper describes two techniques to compress palette images with no loss of information. The results are aimed at applications in medical image archivization systems.

The research has been under to the project BW – DPB of Poznań University of Technology.

REFERENCES

- [1] M. T. Orchard, C. A. Bouman: "Color quantization of images", *IEEE Trans. Signal Processing*, Vol. 39, pp. 2677-2690, Dec. 1991.
- [2] R. S. Gentile, E. Walowit, J. P. Allebach: "Quantization and multilevel halftoning of color images for near original image quality", *J. Opt. Soc. Amer. (A)*, Vol. 7, No. 6, pp. 1019-1026, 1990.
- [3] A. Zaccarin, B. Liu: "A novel approach for coding color quantized images", *IEEE Trans. Image Processing*, Vol. 2, pp. 442-453, Oct. 1993.
- [4] G. Wyszecki, W. S. Stiles: *Color science*, 2nd Edition; John Wiley & Sons 1982.
- [5] W. Pratt: *Digital image processing*, 2nd Edition; John Wiley & Sons 1991.
- [6] A. Gersho, R. Gray: *Vector quantization and signal compression*; Kluwer 1992.
- [7] M. Domański: "Vector quantization of color images, Scientific report"; Politechnika Poznańska, Instytut Elektroniki i Telekomunikacji, Poznań 1992, in Polish.
- [8] Przetwarzanie i archiwizacja obrazów kolorowych w systemach cyfrowych. Sprawozdanie merytoryczne BW-44-364; Politechnika Poznańska, Instytut Elektroniki i Telekomunikacji, Poznań 1993.

POSTERS I.

QUANTIZING ACCURACY FOR HIGH QUALITY COLOR IMAGE PROCESSING

Q. GAN, K. KOTANI and M. MIYAHARA

SCHOOL OF INFORMATION SCIENCE, JAIST:
JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY
ISHIKAWA, 923-12 JAPAN

1. INTRODUCTION

1.1. Background

Most of the color image processing terminals and workstations available today do not support full color quality due to limitations imposed by the analog-to-digital conversion. It is widely assumed that 8 bits per (R,G,B) signal provides sufficient color variation in electronic imaging. As our demands for high quality of color reproduction increase higher and higher, it becomes insufficient to provide good color reproduction as required for a perceptually transparent representation and coding.

1.2. Objective

In this paper, we attempt to analyze the discrete colors in a perceptual level. We use a knowledge of uniform color space with a view of examining the distribution of the discrete colors and measuring the perceptual color differences between the adjacent colors. Then we discuss the minimum numbers of quantizing levels of (R,G,B) signals when the color differences between adjacently discrete colors are always satisfying the "just perceptible" criterion.

2. UNIFORM COLOR SPACE AND COLOR DIFFERENCE

In our study, we prefer a perceptually uniform color space (Munsell color space) [1] to RGB color space, because it specifies colors by cylindrical three orthogonal axes (H,V,C) showing the three attributes of color perception (Value, Hue and Chroma) respectively (see Fig. 1). We use a mathematical transform method MTM [2] which transforms (R,G,B) signals to (H,V,C) signals more accurately than conventional methods based on L*a*b* color space. Moreover, in this study, MTM is modified as MTM2 and then combined with a look-up table, the transform accuracy is further improved to transform error (NBS color difference) as max: 0.59, mean: 0.28 when deals with 791 Munsell samples (see Table 1). We call the color space achieved by MTM2 + Table as HVC color space below. In HVC color space, the color difference is calculated by Godlove's formula (Eq. (1)) [3]. The relation of Godlove's color difference and NBS color difference is prescribed as Eq. (2).

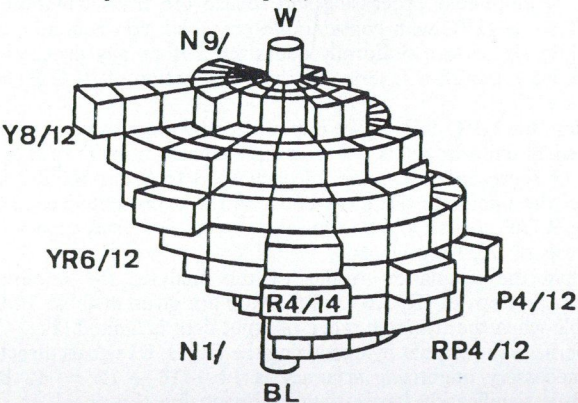


Fig. 1. Munsell Color Space

Table 1. Comparison of Approximation Errors

Method	Max.	Min.	Ave.	Var.
L*a*b*	8.05	0.24	1.73	1.52
MTM	4.53	0.23	1.13	0.46
MTM2	3.90	0.05	0.92	0.36
MTM2+Table	0.59	0.02	0.24	0.015

$$\Delta E_{God} = \quad (1)$$

$$= \sqrt{2C_1C_2 \times \{1 - \cos(\frac{2\pi}{100} \Delta H)\} + (\Delta C)^2 + (4 \times \Delta V)^2},$$

where

$$\Delta H = |H_1 - H_2|, \Delta V = |V_1 - V_2|, \Delta C = |C_1 - C_2|$$

$$\Delta E_{NBS} \approx 1.2 \times \Delta E_{God} \quad (2)$$

3. DISTRIBUTION OF DISCRETE COLORS

We expect to know whether the intuitive impression of a unit change in one of the (R,G,B) color signals is equivalent everywhere within the whole range of color space while the color signals are quantized into discrete values.

3.1. Simulation

A computer simulation is executed to observe the distribution of the (R,G,B) discrete colors in HVC color space. At first, RGB color space is sampled with linear quantization of 256 levels (8 bits) in each axis, giving 256*256*256 points distributed uniformly throughout the color space. For each point, the (R,G,B) signals are transformed to (H,V,C) signals by MTM. Then, the distribution of these separated points in HVC color space is considered.

3.2. Results

In Fig. 2, the photograph shows these discrete colors plotted on several different V-C planes with constant Hue as black points.

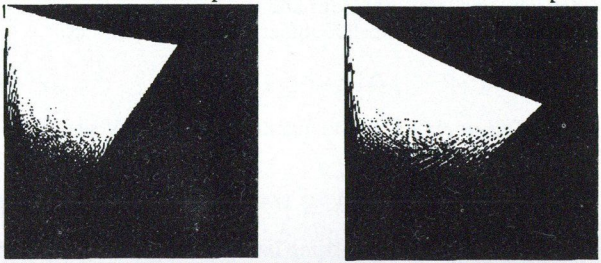


Fig. 2. Distribution of Discrete Colors in V-C Planes (H: constant).
(a) H=5R; (b) H=5G

It is clear that the discrete colors represented by 8 bits (R,G,B) data has an irregular distribution in HVC color space and color quantized in steps are coarser in the dark areas than in the bright areas.

3.3. Color Differences of Quantizing Errors

To make out the visual impact of quantizing errors, the color differences of the discrete colors are calculated. Three kinds of linear quantizing methods are under discussion:

- (R,G,B) signals are quantized linearly,
- γ -precorrected (R,G,B) signals are quantized linearly,
- (R,G,B) signals are linearly quantized after a non-linear transformation (for $\gamma = 3.0$).

Table 2 shows the maximum values of color differences between each discrete color and its most adjacent color in several V-C planes when (R,G,B) signals are linearly quantized by 8 bits respectively [4].

Table 2. Color Differences of Adjacent Colors in V-C Planes (NBS value)

Hue	$\Delta E_{NBS} \text{ MAX}$		
	(1)	(2)	(3)
5R	2.45	1.98	2.11
5YR	3.78	3.12	1.81
5Y	4.26	3.26	2.69
5GY	3.18	2.73	2.71
5G	4.18	3.17	2.98
5BG	5.25	4.23	3.43
5B	1.60	1.63	1.45
5PB	2.79	2.50	2.29
5P	3.86	2.84	2.42
5RP	2.48	1.90	1.92

The results are as the following:

- 1) In each case the maximum color differences of adjacent colors are larger than the "just perceptible color difference" ($\Delta E_{NBS} = 1$).
- 2) It is competent to decrease the perceptual color differences of quantizing errors by transforming the (R,G,B) signals with non-linear functions such as the γ -precorrection, the technique well known in television, before linear quantization. But the quantizing errors of 8 bits quantization are still larger than the threshold difference. Same results are confirmed on H-C planes [5].

4. OPTIMAL QUANTIZATION OF HVC COLOR SPACE

The goal then is to furnish the minimum number of quantizing steps of (R,G,B) signals when the color differences of quantizing errors are always smaller than the "just perceptible color difference". First, we discuss that how many quantizing levels are necessary to quantize (H,V,C) signals with the color difference of $\Delta E_{NBS} = 1$ as the maximum distortion.

4.1. Quantizing Steps of (V,C) Signals

According to Eq. (1) and Eq. (2), the (V,C) signals should be divided into small uniform steps and the color differences between neighbours should be satisfied under the next constraint:

$$\Delta E_{God} \leq 1/1.2. \quad (3)$$

When $\Delta H = \Delta C = 0$, the quantizing levels of V signal can be easily derived from Eq. (1):

$$\Delta V \leq 1/4.8. \quad (3)$$

It indicates that the quantizing interval should be smaller than 1/4.8 within the range of V signal, that means, the number of quantizing steps should be taken more than 48 (i.e. 6 bits). But in real case, the V signal performs a strong impact on color perception, it is asked two bits more, 8 bits quantizing accuracy is necessary in our study [5], [6].

In the case of C signal, as the analysis of V signal, we can obtain the number of quantizing steps should be taken more than 43.2 (i.e. 6 bits).

4.2. Quantizing Steps of H Signal

It can be noticed that the perceptual difference caused by changing H signal depends on the level of Chroma. Therefore the quantizing steps of H should be changed based on the Chroma level of the colors.

We assume that ΔH_C is the necessary quantizing interval of H signal on circle C. So ΔH_1 can be calculated when $\Delta V = \Delta C = 0$ and $C_1 = C_2 = 1$ from Eq. (3):

$$\Delta H_1 \leq 13.68. \quad (5)$$

It indicates that the quantizing interval should be smaller than 15.54 within the range of H signal when $C=1$, that is, the number of quantizing steps should be taken more than 8 (i.e. 3 bits). Further, it is found that the quantizing interval of H signal changes with the variable C as the following relation:

$$\Delta H_C \leq \Delta H_1/C. \quad (6)$$

Hence the quantizing levels should be taken more than 8C steps counting on the level of C, the maximum value of C is 30 at $V = 8.0$, $H = 56.1$ (near 5G).

In accordance with the previous analysis, the minimum number of quantizing steps of (H,V,C) signals can be determined as the following (see. Fig. 3):

- V:256 levels (8 bits)
- C:64 levels (6 bits)
- H:8C levels (2 bits ~ 8 bits)

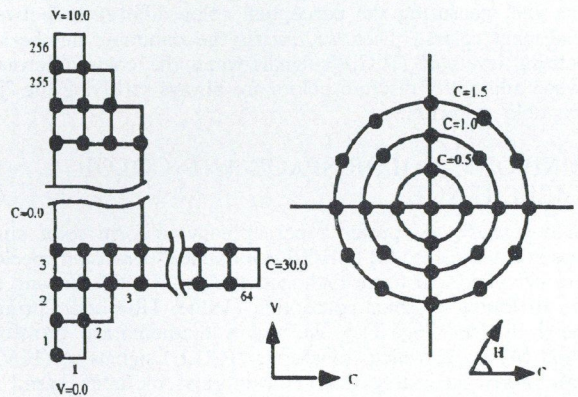


Fig. 3. Quantization of HVC Color Space when $\Delta E_{NBS} \leq 1$

5. QUANTIZATION OF RGB COLOR SPACE

It is desired that the quantization can be performed in HVC color space. But it is hard to maintain stability of the hardware of non-linear processing and realize the transformation of $RGB \leftrightarrow HVC$ with considerable precision, consequently, the (R,G,B) signals are uniformly quantized before any processing. Then, we establish a scheme for the quantization of (R,G,B) signals.

First, the HVC color space is quantized by the previous method into small uniform solids. After reverse transforming the (H,V,C) data of representative colors of each solid to 8 bits RGB color space, the minimum distances between the representative colors along R,G,B axes should be equivalent to the necessary quantizing intervals of (R,G,B) signals.

From the simulation results of the analysis, the minimum quantizing steps of (R,G,B) and (H,V,C) are given in Table 3. It is flexible when the dynamic range of input data is limited (Fig. 4).

We have shown that by quantizing the (R, G, B) signals directly, the necessary quantizing accuracy is $14 + 16 + 12 = 42$ bits which is significantly larger than by quantizing the singals in the HVC color space, because the RGB color space is not uniform to visual perception. In order to satisfy the condition: $\Delta E_{NBS} \leq 1$, the linear quantization should use the minimum interval as the

quantizing interval in the all range of RGB color space. In order to reduce the necessary quantizing accuracy of (R, G, B) signals, a simple non-linear transformation ($e^{1/\gamma}$, $\gamma = 3.0$) before the linear quantization is an advantageous procedure to reduce the necessary quantization steps.

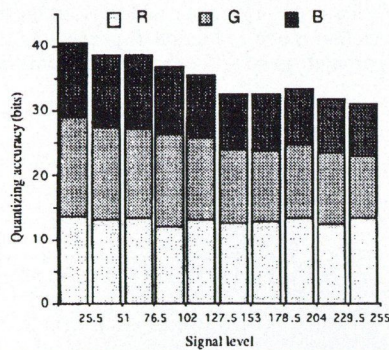


Fig. 4. Quantizing Accuracy in each range of signals (linear case divided into 10 steps)

REFERENCES

[1] S. M. Newhall, D. Nickerson, D. B. Judd: "Final Report of the O.S.A Subcommittee on the Spacing of the Munsell Colors", *J. Opt. Soc. Am.*, Vol. 33, pp. 385-418 (1943)

[2] M. Miyahara, Y. Yoshida: "Mathematical Transform of (R,G,B) Color Data to Munsell (H,V,C) color Data", *Trans of ITE of Japan*, Vol. 43, No. 10, pp. 1129-1136 (1989)

[3] I. H. Godlove: "Improved Color-Difference Formula, with Applications to the Perceptibility and Acceptability of Fadings", *J. Opt. Soc. Am.*, Vol. 41, No. 11, pp. 760-772 (Nov. 1951)

[4] MM. Miyahara: "Quality Assessments for Visual Service", *IEEE COM. Mag.*, Vol. 26, No. 10, pp. 51-60 (Oct. 1988)

Table 3. Quantizing Accuracy under the Condition $\Delta E_{NBS} \leq 1^*$

	Linear Input	Non-linear Input ($\gamma = 3$)
H	8C levels/pixel	
V	8 bits/pixel	
C	6 bits/pixel	
R	14 bits/pixel	10 bits/pixel
G	16 bits/pixel	12 bits/pixel
B	12 bits/pixel	9 bits/pixel

* Calculated by Godlove's formula.

6. CONCLUSION

From the simulations included in this paper, we know

1) The colors in the dark areas in 8 bits digital RGB color space are quantized in steps several times coarser than 1 NBS unit.

2) The quantizing accuracy of R, G and B signals should be not less than 14, 16 and 12 bits for linear quantization and 10, 12 and 9 bits for non-linear quantization ($\gamma = 3.0$) respectively in order to keep the color difference of quantizing errors smaller than 1 NBS unit.

3) Non-linear quantization of (R,G,B) signals can decrease the necessary quantizing steps of (R,G,B) signals.

[5] Q. Gan, K. Kotani, M. Miyahara: "Characteristic Analysis of Color Information Based on (R,G,B) $\leftrightarrow$ (H,V,C) Color Space Transformation", *SPIE VCIP'91*, 1605, pp. 374-381 (1991)

[6] R. C. Zerrer, H. Hemmendinger: "Evaluation of Color Difference Equation - A New Approach", *Color Research and Applications*, Vol. 4, No. 2, pp. 71-77 (1979)

[7] Q. Gan, K. Kotani, M. Miyahara: "Quantizing Accuracy for High Quality Color Image Processing", *SID'93*, Vol. 11.2, pp. 129-132 (May 1993).

ACCELERATED IMAGE CODING SCHEME BASED ON ITERATED FUNCTION SYSTEMS

K. TANAKA, M. IWAKIRI and K. MATSUI

DEPT. OF COMPUTER SCIENCE
THE NATIONAL DEFENSE ACADEMY
1-10-20 HASHIRIMIZU, YOKOSUKA, 239 JAPAN

1. INTRODUCTION

Several image coding schemes have been developed using IFS (Iterated Function Systems) based upon fractal geometry, recently⁽¹⁾⁻⁽⁴⁾. These schemes estimate optimally a set of contractive transformations for the prescribed local region on an original image. Then the sets of transformations are coded and transmitted to the recipient. The image is regenerated by repeating the contractive transformations to each local region at the recipient, recursively. However, because it requires a large amount of processing to estimate the optimal IFS parameters, the reduction of processing time is desirable.

From this point of view, this paper presents a new scheme which accelerates the estimation of IFS parameters. If we represent the characteristic of each region in binary, which are digitized by the average level of the region, the optimal region is obtained by XOR operation and it brings us much reduction of

the processing time of IFS parameter estimation. The details on our scheme are presented below.

2. DEFINITIONS AND BASIC CONCEPT

Some conventional schemes use the following steps in general:

Step1: Let I_0 be the original image in $N_0 \times N_0$ pixels.

Step2: Make disjoint sub-blocks ($R = [r_{ij}]$) of I_0 , where $i, j = 1, 2, \dots, n$.

Step3: Get the average level (α_R) and make $r'_{i,j} \leftarrow r_{i,j} - \alpha_R$.

Step4: Let $D = [d_{ij}]$ be blocks consisting $n' \times n'$ pels on I_0 , where $i, j = 1, 2, \dots, n'$, and the corresponding regions may be overlapped.

Step5: Make the matrix (D) the contractive transformation of $\phi = (n/n')^2$ so as to become the same to R in size. Let the results be $D' = [d'_{ij}]$.

- Step6: Get the average level ($\alpha_{D'}$) in D' , and make $d''_{ij} \leftarrow d'_{ij} - \alpha_{D'}$.
- Step7: Rotate D' by θ and make it the contractive transformation of ρ ($-1 < \rho < 1$). We have $d'''_{ij} \leftarrow \theta \cdot \rho \cdot d''_{ij}$, where $i, j = 1, 2, \dots, n$.
- Step8: Calculate the sum of square error between R and D , i.e.,

$$e = \sum_{i=1}^n \sum_{j=1}^n (r'_{ij} - d'''_{ij})^2. \quad (1)$$

Step7 to Step8 are done with different θ and ρ . Step4 to Step8 are repeated for all D on I_0 , then, the minimum (e_{min}) is obtained. We decide the region with e_{min} to be optimal and store its offset position x and y from $(0,0)$, α_R , ϕ , θ and ρ as IFS parameter (P) (see Fig. 1). These procedures are done for all R in the whole image.

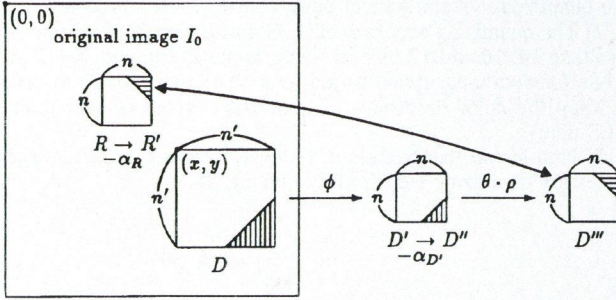


Fig. 1. The concept of image coding scheme based on Iterated Function Systems

Because it requires a lot of processing time to find e_{min} for a pair of possible (R, D) , the conventional methods save the processing time by only the pairs satisfying $e \leq \beta$ in this process, where β is previously set as the threshold of e . Next, a set of P for all R on I_0 are coded and transmitted.

On the other side, the image is decoded by the following steps at the recipient.

- Step1: An initial image J_0 is randomly generated in $N_0 \times N_0$ pixels.
- Step2: Make disjoint sub-blocks ($R = [r_{ij}]$) of J_0 , where $i, j = 1, 2, \dots, n$.
- Step3: Extract the region $D = [d_{ij}]$ ($i, j = 1, 2, \dots, n'$) positioned (x, y) from $(0,0)$ on J_0 , where (x, y) is included in IFS parameter P corresponding to R .
- Step4: Make the matrix (D) the contractive transformation of $\phi = (n/n')^2$ so as to become the same to R in size. Let the results be $D' = [d'_{ij}]$.
- Step5: Get the average level ($\alpha_{D'}$) in D' , and make $d''_{ij} \leftarrow d'_{ij} - \alpha_{D'}$.
- Step6: Rotate D' by θ and make it the contractive transformation of ρ ($-1 < \rho < 1$). We have $d'''_{ij} \leftarrow \theta \cdot \rho \cdot d''_{ij}$, where $i, j = 1, 2, \dots, n$.
- Step7: Replace R by R' after making $r'_{ij} \leftarrow d'''_{ij} + \alpha_R$.

The individual contractive transformation of D based on P (Step3 to Step7) is applied for each R on J_0 , and is recursively repeated as

$$J_k = J_{k-1} \cup P \quad (k = 1, 2, \dots). \quad (2)$$

Finally, we have the regenerated image.

3. PRESENTED SCHEME

In order to accelerate the estimation of IFS parameters, the above concept is revised in this paper. Let us denote the total possible blocks of D on the image as N_D . Because $N_D = (N_0 - 2^m \cdot n + 1)^2$ on I_0 when $\phi = 1/2^{2m}$ ($m = 1, 2, \dots$), N_D becomes 62,001 in case of $N_0 = 256$, $n = 4$ and $m = 2$, for example. The reduction of N_D results in saving the processing time. Therefore, we try to estimate IFS parameters on the image reduced from the original image. First, let I_0 be reduced by

$1/2^L$ ($L = 1, 2, \dots$) both in horizontal and vertical directions, and then, the reduced images be I_L ($L = 1, 2, \dots$), respectively. Next, let the size D on I_L ($L = 1, 2, \dots$) and ϕ be set to $n \times n$ and $1/2^{2L}$, respectively. Because the size (N_L) of I_L is $N_0/2^L$ ($L = 1, 2, \dots$), $N_D = (N_L - n + 1)^2$ on I_L . Table 1 shows N_D on I_0 for $m = 1, 2, 3, 4$ and N_D on I_L for $L = 1, 2, 3, 4$. It tells us that much reduction of processing time may be attained if IFS parameters could be estimated from the reduced images.

Table 1. Possible numbers of D on the image

(a) N_D on I_0 for $m = 1, 2, 3, 4$				
m	1	2	3	4
N_D	62,001	58,081	50,625	37,249

(b) N_D on I_L for $L = 1, 2, 3, 4$				
L	1	2	3	4
N_D	15,625	3,721	841	169

Let us denote the numbers of rotations and amplitude-reductions of pixels as N_θ and N_ρ , respectively, and also define the number of calculations of e for each R as N_c . Under the assumption that IFS estimation is achieved for all possible D , we have $N_c = N_\theta \cdot N_\rho \cdot N_D$. For example, when 4 kinds of rotations ($\theta = 0, \pi/2, \pi, 3\pi/2$) and 8 kinds of amplitude-reductions are specified to the estimation, N_c and SN ratio (dB) of the regenerated image are shown in Table 2 together with the results in case of $\theta = 0$ and $\rho = 1$, where the sample image is Girl (256×256) in SIDBA and $n = 4$. Although the image quality is improved $1 \sim 3$ dB by introducing both rotations and amplitude-reductions, it increases N_c by $N_\theta \cdot N_\rho$ times. Because our object is to accelerate the IFS estimation, θ and ρ are fixed to 0 and 1 in this paper, respectively.

Table 2. N_c and SN ratio ($n = 4$)

(a) with 4 kinds of θ and 8 kinds of ρ				
L	1	2	3	4
N_c	500,000	119,072	26,912	5,408
SN (dB)	35.83	35.44	34.04	32.20

(b) with $\theta = 0$ and $\rho = 1$				
L	1	2	3	4
N_c	15,625	3,721	841	169
SN (dB)	34.07	33.21	31.21	29.00

A new scheme which can effectively classify D and moreover accelerate the parameter estimation is presented in this paper. It requires $N_c = N_D$ times of calculation under above preparations. If some optimal candidates (D) are estimated before calculating the square error between R and D , the number of calculations (N_c) may be moreover reduced. When a pair of R and D with smaller e is found, not that their region characteristics are similar to each other as shown in Fig. 2. Therefore, let us first prepare a pair of R and D as shown in Fig. 3. and digitize them to B_R and B_D in binary by their average levels α_R and α_D , respectively. If two region characteristics are similar each other, it is estimated that the binary matrices B_R and B_D may have almost the same patterns. Next, exclusive-OR (XOR) operation between B_R and B_D is achieved and bits colored "1" are counted (b). If $b \leq \delta$, then obtain the error e between R and D , otherwise ignore the region. This approach can effectively reduce the number of calculations under less deterioration of image quality as shown in Table 3 compared with results in Table 2(b), where $n = 4$ and $\delta = 6$.

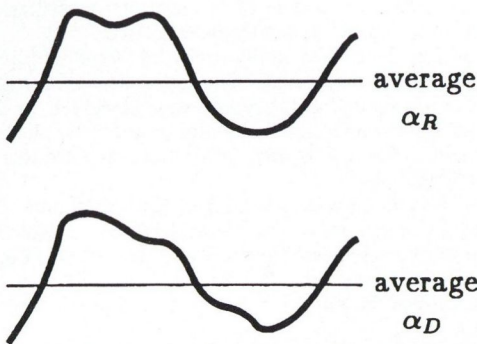


Fig. 2. Illustration of region characteristics. (a) signal form of R ; (b) signal form of D

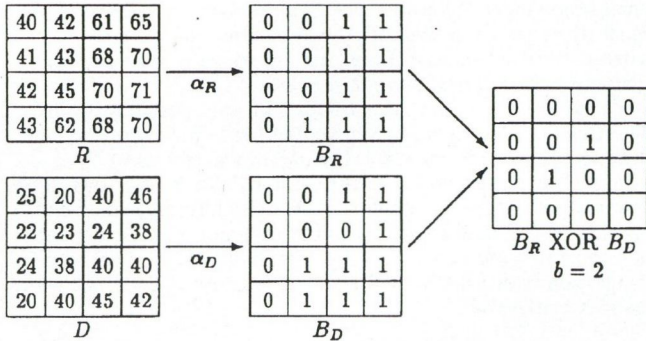


Fig. 3. Illustration of binary matching

Table 3. N_C and SN ratio with binary matching ($n = 4$)

L	1	2	3	4
N_C	4,919.27	1,154.26	249.29	48.78
$SN(dB)$	34.06	33.21	31.20	28.45

In order to attain both fast estimation of IFS parameters and high quality of regenerated image at the same time, it is preferable that the size of R is adaptively changed depending on e_{min} .

REFERENCES

- [1] M. F. Barnsley: "Fractal Everywhere", Academic Press, San Diego, CA (1988).
- [2] M. F. Barnsley, A. E. Jacquin: "Application of Recurrent Iterated Function Systems to Images", *SPIE Visual Commun. and Image Process.* 88, Vol. 1001, pp. 122-131 (1988).
- [3] A. E. Jacquin: "Image Coding Based on a Fractal Theory

If $e_{min} > \Delta$, then the sizes length of R and D are reduced to $n/2$ and the same procedures are achieved again, where Δ is previously set as a threshold for e_{min} . Because the same procedure are done in such the region of $e_{min} > \Delta$, it is possible to attain fast estimation and higher image quality. When $n = 4$, $\Delta = 64$ and $\delta = 6$, N_c , SN ratio (dB), block numbers of both $n = 4(B_4)$ and $n = 2(B_2)$, and volume of IFS parameters (p ; no compressed, bit/pel) are shown in Table 4. It is understood that less of calculations high SN ratio are attained by our scheme. Examples of regenerated images are also shown in Fig. 4 with the original image (IFS parameters are determined from $I_4(16 \times 16)$).

Table 4. Experimental results ($n = 4$, $\Delta = 64$ and $\delta = 6$)

L	1	2	3	4
N_C	9,510.06	2,725.59	777.23	231.35
$SN(dB)$	38.28	38.63	38.18	35.99
B_4	3,269	2,949	2,484	1,688
B_2	3,308	4,588	6,448	9,632
p	2.27	2.36	2.52	2.83

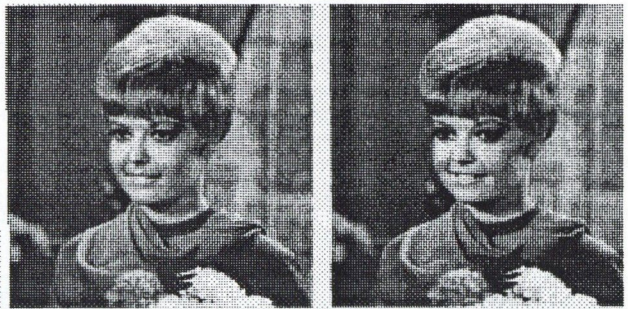


Fig. 4. Examples of output image (Girl). (a) original image; (b) regenerated image

4. CONCLUSION

It is concluded that our scheme attains to accelerate IFS parameter estimation and high image quality at the same time. It will be more investigated how to compress IFS parameters and to estimate the optimum reduced image and thresholds for the IFS estimation in desirable bit-rate and image quality, automatically.

- of Iterated Contractive Image Transformations", *IEEE Trans. Image Process*, Vol. 1, No. 1, pp. 18-30 (Jan. 1992).
- [4] E. W. Jacobs, Y. Fischer, R. D. Boss: "Image Compression: A Study of the Iterated Transform Method", *Signal Processing*, Vol. 29, pp. 251-263 (1992).

ONE APPLICATION OF HOMOMORPHIC FILTERING

M. HUDECOVÁ

EUROTEL S.R.O., BRATISLAVA, SLOVAKIA

J. POLEC

SLOVAK TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF TELECOMMUNICATIONS
ILKOVIČOVA 3, 812 19 BRATISLAVA, SLOVAKIA

R. VARGIC

SLOVAK TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF TELECOMMUNICATIONS
ILKOVIČOVA 3, 812 19 BRATISLAVA, SLOVAKIA

J. PAVLOVIČOVÁ

SLOVAK TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF TELECOMMUNICATIONS
ILKOVIČOVA 3, 812 19 BRATISLAVA, SLOVAKIA

An image signal transmitted by a transmission channel, is often multiplied by noise. Because of this it is necessary to solve the

problem of removing of the multiplicative data-noise. One of possibilities to remove it is homomorphic filtering.

The degraded image due to multiplicative noise, $g(n_1, n_2)$, can be expressed as [1]

$$g(n_1, n_2) = f(n_1, n_2) \cdot v(n_1, n_2), \quad (1)$$

where $v(n_1, n_2)$ is random noise that is not a function of $f(n_1, n_2)$ represent image intensities and are therefore nonnegative, $v(n_1, n_2)$ is also nonnegative. By applying the logarithmic operation, we obtain

$$\begin{aligned} T[g(n_1, n_2)] &= \log[g(n_1, n_2)] = \\ &= \log[f(n_1, n_2)] + \log[v(n_1, n_2)]. \end{aligned} \quad (2)$$

If we denote $\log[g(n_1, n_2)]$ by $g'(n_1, n_2)$ and denote $\log[f(n_1, n_2)]$ and $\log[v(n_1, n_2)]$ similarly, becomes

$$g'(n_1, n_2) = f'(n_1, n_2) + v'(n_1, n_2). \quad (3)$$

The multiplicative noise $v(n_1, n_2)$ has now been transformed to additive noise $v'(n_1, n_2)$ and image restoration algorithms developed for reducing additive signal-independent noise may be applied to reduce $v'(n_1, n_2)$. The resulting image is exponentiated to compensate for logarithmic operation.

For the separation of the original image signal and multiplicative noise it is very important [7]:

- possibility of short signal approximation by suitable orthogonal functions,
- possibility of noise approximation by sequences they are in the sufficient distance from image sequences.

This is the optimization of homomorphic system approximation by using the transform coding of discrete orthogonal transform, according to two conditions introduced above.

This contribution deals with applicability of two-dimensional homomorphic filter for image signal filtering using compression method [6]. For this purpose the signal was processed by zig-zag scanning after two-dimensional discrete transforms. Instead of transmission function of low-pass filter there quantizator for transform coding was used.

Data signals have infinite length Fourier approximation. We disturb the image signal by multiplicative data-noise which is better approximable by functions from Walsh base than Fourier base. Then for homomorphic filtering it is possible to use compression methods based on the two-dimensional discrete Walsh transform.

The fact, that Walsh noise has short sequency spectrum, was the main reason for using the transfer function of filter removing the higher sequences instead of the usual method for two-dimensional digital filtering. For this purpose we processed the signal by zig-zag scanning after two-dimensional discrete Paley-ordered Walsh-Hadamard transform [5] which was applied on homomorphically preprocessed image.

Fig. 1 illustrates the performance of the image restoration algorithm in reducing multiplicative data-noise for discrete cosine, Haar and Paley-ordered Walsh-Hadamard transforms.

From the results of the experiment it follows, that our filter processing is suitable for multiplicative noise filtering in data channel. Using this method it is possible to acquire image enhancement and this method uses simpler computational algorithm in comparison with usually used process, because it does not need complex mathematics.

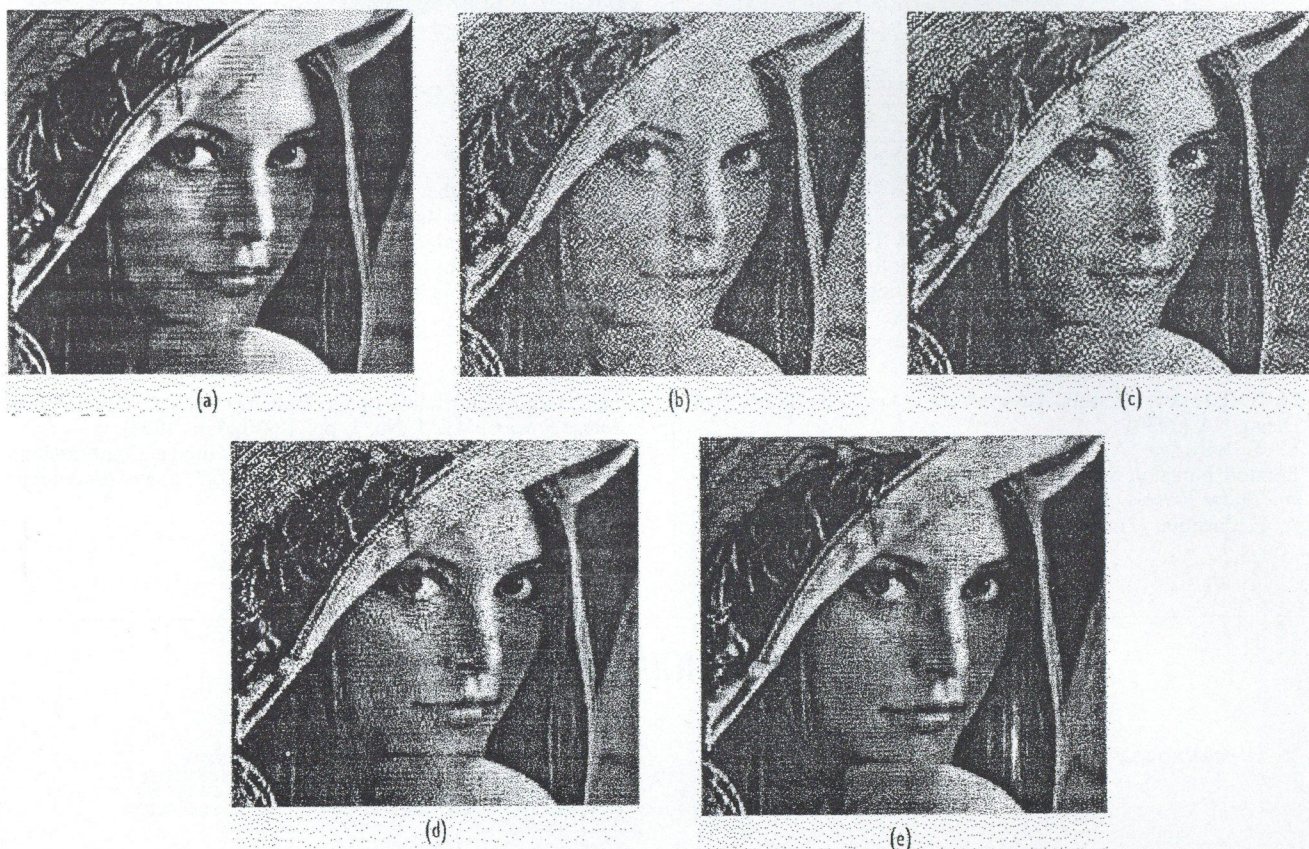


Fig. 1. Performance illustration of the multiplicative data-noise reduction system. (a) Original image of 256×256 pixels; (b) image degraded by multiplicative high-frequency data-noise; image processed by: (c) discrete cosine transform; (d) discrete Haar transform; (e) discrete Paley-ordered Walsh-Hadamard transform

REFERENCES

- [1] J. S. Lim: *Two-dimensional Signal and Image Processing*, Prentice Hall International, Inc., Englewood Cliffs, New Jersey, 1990.
- [2] A. Rosenfeld, A. C. Kak: *Digital Picture Processing*, Academic Press, Inc., 1982.
- [3] F. M. Wahl: *Digitale Bildsignalverarbeitung*, Springer, Heidelberg, 1984.
- [4] V. A. Oppenheim, R. W. Schaffer: *Digital Signal Processing*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1975.
- [5] P. B. Besslich, T. Lu: *Diskrete Orthogonaltransformationen*, Springer, Heidelberg, 1990.
- [6] J. Polec: *Nonstandard Methods in Application of Discrete Orthogonal Transforms for Data Compression*, Dissertation, EF STU, Bratislava, 1994.
- [7] M. Hudecova, J. Polec, R. Vargic: "Discrete Orthogonal Transforms in Homomorphic Filtering of Image", International Conference on Digital Signal Processing, L. Mikulas, 1993.

NEW ORDERING OF SEQUENCES AND BASES FUNCTIONS OF DISCRETE FOURIER AND DISCRETE HARTLEY TRANSFORMS FOR TRANSFORM CODERS

J. POLEC, J. PAVLOVIČOVA, and R. VARGIC

SLOVAK TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF TELECOMMUNICATIONS
ILKOVIČOVA 3, 812 19 BRATISLAVA, SLOVAKIA

In the literature the wide interest has been offered to the DFT for many years. The DFT spectrum has the different digital and geometric shape (symmetry around the center) than transformations with real response.

To unify block transformation outputs from the JPEG-transform coder we can modify 1-dimensional and 2-dimensional DFT and Discrete Hartley transform (DHYT). We use new ordered queues of sequence (with DFT we can speak about frequency) elements, they have no need of special organized quantizator and zonal filter.

1. NEW ORDERED RESULT OF DFT

Let us have the result of 1-dimensional DFT of length $N = 4$:

$$\begin{matrix} \text{Re}\{X(0)\}, & \text{Re}\{X(1)\}, & \text{Re}\{X(2)\}, & \text{Re}\{X(3)\} \\ \text{Im}\{X(0)\}, & \text{Im}\{X(1)\}, & \text{Im}\{X(2)\}, & \text{Im}\{X(3)\} \end{matrix}$$

From DFT properties follows for the complex vector form:

$$\begin{matrix} \text{Re}\{X(0)\} & , & \text{Re}\{X(1)\} & , & \text{Re}\{X(2)\} & , & \text{Re}\{X(1)\} \\ 0 & , & \text{Im}\{X(1)\} & , & 0 & , & -\text{Im}\{X(1)\} \end{matrix}$$

This form clearly shows, that to keep full information from DFT it is enough to save left half part and the reflection-element of the vector. Because of the real character of DC-coefficient and reflection elements, we need N places for real numbers to save DFT result without any lost. That is the same amount as for standard real discrete orthogonal transform. The geometric shape of spectrum in left half part is than very similar to the DCT spectrum shape. Only it is of bigger slope, what is caused by the fact, that it is situated in two planes (real and imaginary). So that, after suitable interchanging of real and imaginary components, the same coding structure as for DCT can be applied to it.

New order will be:

$$X(0), \text{Re}\{X(1)\}, \text{Im}\{X(1)\}, X(2).$$

In this way we get new ordered DFT spectrum for larger vectors, too. Two-dimensional form of spectrum with its properties is suitable for similar new ordering. We shall describe reflection of 2D spectrum on example:

Let's have the 2D discrete Fourier spectrum of dimension $N \times M = 4 \times 4$:

$$\begin{matrix} \text{Re}\{X(0,0)\}, & \text{Re}\{X(0,1)\}, & \text{Re}\{X(0,2)\}, & \text{Re}\{X(0,3)\} \\ \text{Re}\{X(1,0)\}, & \text{Re}\{X(1,1)\}, & \text{Re}\{X(1,2)\}, & \text{Re}\{X(1,3)\} \\ \text{Re}\{X(2,0)\}, & \text{Re}\{X(2,1)\}, & \text{Re}\{X(2,2)\}, & \text{Re}\{X(2,3)\} \\ \text{Re}\{X(3,0)\}, & \text{Re}\{X(3,1)\}, & \text{Re}\{X(3,2)\}, & \text{Re}\{X(3,3)\} \end{matrix}$$

$$\begin{matrix} \text{Im}\{X(0,0)\}, & \text{Im}\{X(0,1)\}, & \text{Im}\{X(0,2)\}, & \text{Im}\{X(0,3)\} \\ \text{Im}\{X(1,0)\}, & \text{Im}\{X(1,1)\}, & \text{Im}\{X(1,2)\}, & \text{Im}\{X(1,3)\} \\ \text{Im}\{X(2,0)\}, & \text{Im}\{X(2,1)\}, & \text{Im}\{X(2,2)\}, & \text{Im}\{X(2,3)\} \\ \text{Im}\{X(3,0)\}, & \text{Im}\{X(3,1)\}, & \text{Im}\{X(3,2)\}, & \text{Im}\{X(3,3)\} \end{matrix}$$

using its properties we can write equivalent characteristic [5]:

$$\begin{matrix} \text{Re}\{X(0,0)\}, & \text{Re}\{X(0,1)\}, & \text{Re}\{X(0,2)\}, & \text{Re}\{X(0,1)\} \\ \text{Re}\{X(1,0)\}, & \text{Re}\{X(1,1)\}, & \text{Re}\{X(1,2)\}, & \text{Re}\{X(1,3)\} \\ \text{Re}\{X(2,0)\}, & \text{Re}\{X(2,1)\}, & \text{Re}\{X(2,2)\}, & \text{Re}\{X(2,1)\} \\ \text{Re}\{X(1,0)\}, & \text{Re}\{X(1,3)\}, & \text{Re}\{X(1,2)\}, & \text{Re}\{X(1,1)\} \end{matrix}$$

$$\begin{matrix} 0 & , & \text{Im}\{X(0,1)\} & , & 0 & , & -\text{Im}\{X(0,1)\} \\ \text{Im}\{X(1,0)\} & , & \text{Im}\{X(1,1)\} & , & \text{Im}\{X(1,2)\} & , & \text{Im}\{X(1,3)\} \\ 0 & , & \text{Im}\{X(2,1)\} & , & 0 & , & -\text{Im}\{X(2,1)\} \\ -\text{Im}\{X(1,0)\} & , & -\text{Im}\{X(1,3)\} & , & -\text{Im}\{X(1,2)\} & , & -\text{Im}\{X(1,1)\} \end{matrix}$$

Result are $N \times M$ real numbers, what is the same amount as for real transformations.

Let's transform previous 2 fields into one matrix:

$$\begin{matrix} \text{Re}\{X(0,0)\}, & \text{Re}\{X(0,1)\}, & \text{Im}\{X(0,1)\}, & \text{Re}\{X(0,2)\} \\ \text{Re}\{X(1,0)\}, & \text{Re}\{X(1,1)\}, & \text{Re}\{X(1,3)\}, & \text{Re}\{X(1,2)\} \\ \text{Im}\{X(1,0)\}, & \text{Im}\{X(1,1)\}, & \text{Im}\{X(1,3)\}, & \text{Im}\{X(1,2)\} \\ \text{Re}\{X(2,0)\}, & \text{Re}\{X(2,1)\}, & \text{Im}\{X(2,1)\}, & \text{Re}\{X(2,2)\} \end{matrix}$$

The similar method for new ordering of complex spectrum to real matrix can be applied to the larger blocks of complex spectrum. Spectrum of DFT, new ordered spectrum of DFT and spectrum of DCT are shown in Fig. 1 for comparison.

Results are obtained from picture fragment of 'LENA' of size 256×256 processed in one 2-dimensional batch. Here we can see that for new ordered DFT spectrum the similar processing methods can be used as for sequence spectrums of other transforms. And its character is so similar to DCT spectrum that nearly the same results of this transformation for signal approximation can be expected.

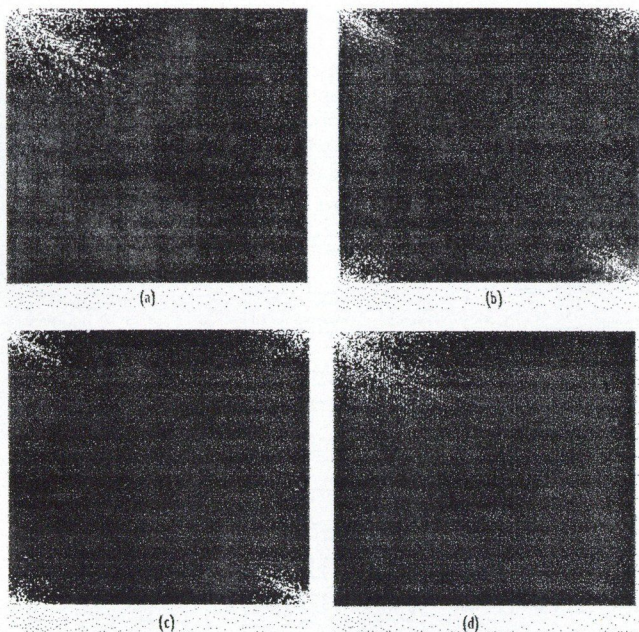


Fig. 1. Logarithmic magnitude spectrum of the picture fragment 256×256 of LENA obtained by 2D DCT (a), by 2D DFT Re (b) and Im (c), and by new ordered result of 2D DFT (d)

2. NEW ORDERED RESULT AND BASE OF DHYT

Because of energy concentration in borders of sequence spectrum it looks to be advantageous to new order it in the similar way as DFT spectrum.

For DHYT this new order corresponds with some changes in order of base functions. In this case we create the new order of complete number of sequence components, because the spectrum is not reflected in values. A feeling of reflection is caused just by similarity in values around the middle of the spectrum. For the vector of DHYT sequence components of length $N = 8$ [4]:

$$X(0), X(1), X(2), X(3), X(4), X(5), X(6), X(7)$$

new ordered set will be:

$$X(0), X(1), X(7), X(2), X(6), X(3), X(5), X(4).$$

The base of the length $N = 16$ obtained in the way described above is in Fig. 2. From shapes of single base functions the same results as for DCT could be expected. We shall call this base with new order: base of DHYT2.

2D sequence spectrum will be new ordered in two steps:

1. to new order the spectrum by rows in the same way as 1D spectrum:

$$u_1(k) \quad \text{for } k = 0 \\ u_2(k) = \begin{cases} u_1(2 \cdot k - 1) & \text{for } k = 1, \dots, N/2 \\ u_1(2 \cdot (N - k)) & \text{for } k = N/2 + 1, \dots, N - 1; \end{cases}$$

2. to new order the result from the first step by columns in the same way.

The result in transform coding is sequence spectrum which does not need any additional processing. The spectrum is in

REFERENCES

- [1] J. S. Lim: *Two-dimensional Signal and Image Processing*, Prentice Hall International, Inc., Englewood Cliffs, New Jersey, 1990.
- [2] A. Rosenfeld, A. C. Kak: *Digital Picture Processing*, Academic Press, Inc., 1982.
- [3] F. M. Wahl: *Digitale Bildsignalverarbeitung*, Springer, Heidelberg, 1984.

Fig. 3. If we compare it with DCT spectrum in Fig. 1 we can say that it is possible to use this new ordered spectrum in the same structures as for DCT spectrum coding are used. We can expect that in some applications (compression, homomorphic filtering, spectral analysis of one- and more-dimensional signals) DCT will be equivalently replaced by these transforms.

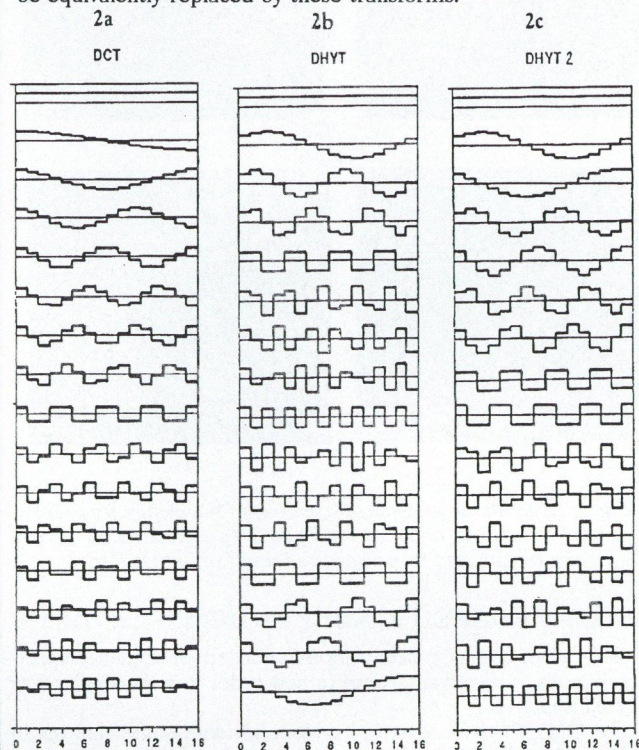


Fig. 2. Bases of DCT (a), DHYT (b), DHYT2 (c) with length $N = 16$

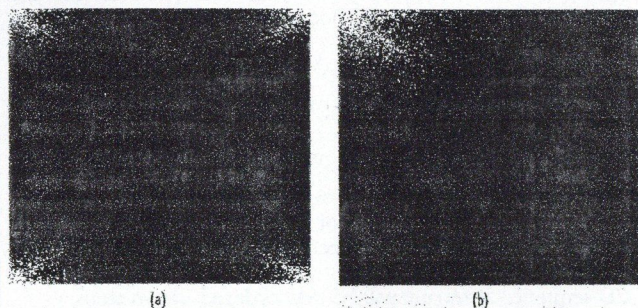


Fig. 3. Logarithmic magnitude spectrum of picture fragment 256×256 of LENA obtained by 2D Hartley transform (a) by 2D Hartley transform with new ordered base (2)

- [4] P. B. Besslich, T. Lu: *Diskrete Orthogonaltransformationen*, Springer, Heidelberg, 1990.
- [5] J. Polec: *Nonstandard Methods in Application of Discrete Orthogonal Transforms for Data Compression*, Dissertation, EF STU, Bratislava, 1994.
- [6] M. Oravec: "Two types of neural networks for image compression", Digital signal processing, Kosice, Sept. 1993.

KOHONEN AND GROSSBERG LEARNING IN NEURAL NETWORKS FOR IMAGE COMPRESSION

M. ORAVEC

SLOVAK TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF TELECOMMUNICATIONS
ILKOVIČOVA 3, 812 19 BRATISLAVA, SLOVAKIA

1. INTRODUCTION

Image compression is essential for applications such as TV transmission, video conferencing, facsimile transmission of printed material, graphics images, etc. Another area for the applications of efficient coding is where pictures are stored in a database, such as archiving medical images, multispectral images and drawings [1]. Numerous bandwidth compression techniques have been developed such as vector quantization [1], differential pulse code modulations, transform coding, hybrid coding, and versions of these techniques (e.g. [6]).

Approach based on neural networks is also very interesting. One of the well-known methods is Cottrell-Munro-Zipser technique [2] using backpropagation algorithm [3]. The self-organizing map introduced by Kohonen (SOM) [4] can be used as one of the possible techniques for designing codebooks for the vector quantization of images. This technique exhibits rapid learning for the vector quantization problem. The counterpropagation network (CPN) [5], [2] can also be considered as a data compression system based upon a concept of vector quantization. But this network functions directly as a nearest-match lookup table. Besides Kohonen learning in this network also Grossberg learning is used.

2. SELF-ORGANIZING MAP

The self-organizing map [4] consists of cells depicted in Fig. 1. Their arrangement can be hexagonal, rectangular, etc. Let $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ be the input vector, which is assumed to be connected in parallel to all the neurons i in this network. The weight vector of cell i is $\mathbf{m}_i = [m_{i1}, m_{i2}, \dots, m_{in}]^T \in \mathbb{R}^n$. The matching criterion for the match of $\mathbf{x}$ with $\mathbf{m}_i$ is based on Euclidean distances. The minimum distance defines the "winner" $\mathbf{m}_c$. The cells doing learning are not affected independently of each other, but as topologically related subsets. The neighbourhood set N_c around cell c is defined. At each learning step all the cell within N_c are updated, whereas cells outside N_c are left intact. The neighbourhood set is centered around that cell for which the best match with input $\mathbf{x}$ is found:

$$\|\mathbf{x} - \mathbf{m}_c\| = \min_i \{\|\mathbf{x} - \mathbf{m}_i\|\}. \quad (1)$$

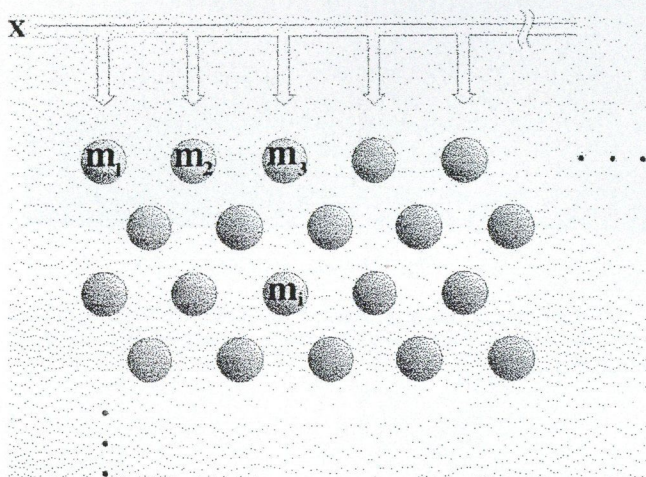


Fig. 1. Cell arrangement for the step

The width or radius of N_c is time-variable, for good global ordering it is very wide in the beginning and shrink monotonically with time (Fig. 2). It is possible to end the process with $N_c = \{c\}$

updating only winner. The updating process (in discrete time notation) is

$$\mathbf{m}_i(t+1) = \begin{cases} \mathbf{m}_i(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] & \text{if } i \in N_c(t) \\ \mathbf{m}_i(t) & \text{if } i \notin N_c(t), \end{cases} \quad (2)$$

where adaptation gain $0 < \alpha(t) < 1$ should decrease with time.

Some practical hints for the application of the algorithm (number of learning steps, rule for $\alpha(t)$, radius of $N_c(t)$) can be found in [4].

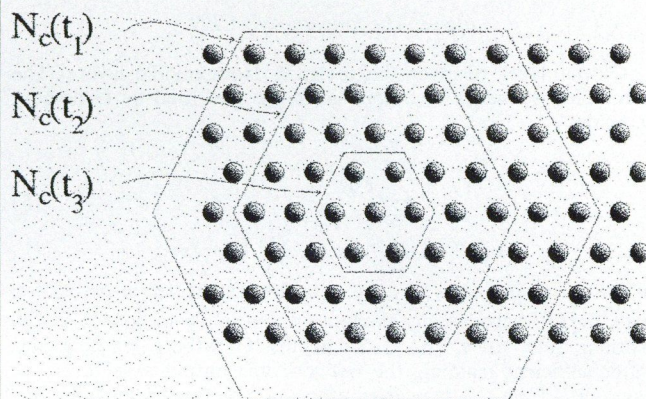


Fig. 2. Examples of topological neighbourhood $N_c(t)$, where $t_1 < t_2 < t_3$

3. COUNTERPROPAGATION NETWORK

The counterpropagation network was proposed by Hecht-Nielsen [2], [5]. The architecture of forward-only CPN is shown in Fig. 3. It consists of 3 layers: an input layer (bottom) containing n units that multiplex the input signals $x_1, x_2, \dots, x_n$ (and m units that supply the "correct" output signal values $y_1, y_2, \dots, y_m$ to the output layer (Grossberg layer — top)) and middle layer (Kohonen layer) with N processing elements that have output signals $z_1, z_2, \dots, z_N$. The outputs of layer 3 represent approximations to the components $y_1, y_2, \dots, y_m$. During training these correct values are supplied to the units of final layer. During training, the transfer function equations for middle layer are:

$$z_i = \begin{cases} 1 & \text{if } i \text{ is the smallest integer for which} \\ & D(\mathbf{w}_i^{\text{old}}, \mathbf{x}) \leq D(\mathbf{w}_j^{\text{old}}, \mathbf{x}) \text{ for all } j \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where D is Euclidean distance. Following the completion of competition process, the next step is weight adjustment using Kohonen learning. Only the processing element that wins adjusts its weight vector in accordance with the equation

$$\mathbf{w}_i^{\text{new}} = (1 - \alpha(t))\mathbf{w}_i^{\text{old}} + \alpha(t)\mathbf{x}(t) \quad (4)$$

($\alpha(t)$ is function of time, it starts at high value and gradually decreases towards zero).

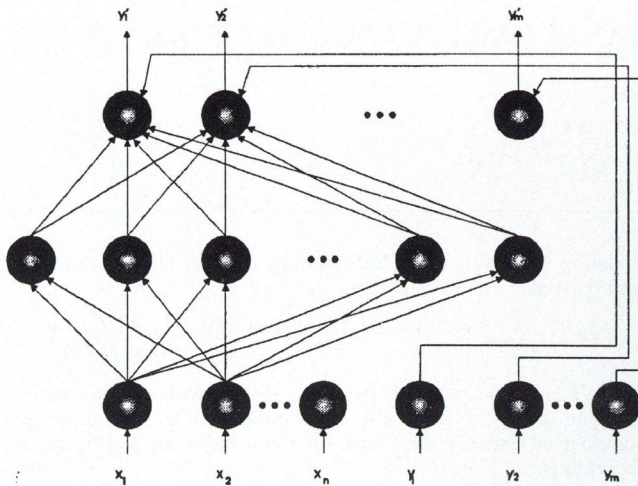


Fig. 3. The forward-only CPN

Output layer receives the z signals from middle layer. The processing elements of output layer are governed by Grossberg learning:

$$y'_j = \sum_{i=1}^N u_{ji}^{\text{old}} z_i \quad (5)$$

$$u_{ji}^{\text{new}} = u_{ji}^{\text{old}} + a(-u_{ji}^{\text{old}} + y_j) z_i,$$

where $\mathbf{u}_j = [u_{j1}, u_{j2}, \dots, u_{jN}]$ is the weight vector associated with the j^{th} processing element of output layer and a is the learning rate ($0 < a < 1$) of the Grossberg learning law. After sufficient training, the network will output a vector $\mathbf{v}_i = [v_{i1}, v_{i2}, \dots, v_{im}]$, whenever processing element i wins the final Kohonen layer competition.

Following the completion of training the network functions exactly as the lookup table shown in Fig. 4. The input vector $\mathbf{x}$ is compared with Kohonen layer weight vectors to find the closest match $\mathbf{w}_i$. Grossberg layer output vector $\mathbf{v}_i$ associated with $\mathbf{w}_i$ is then emitted by the network.

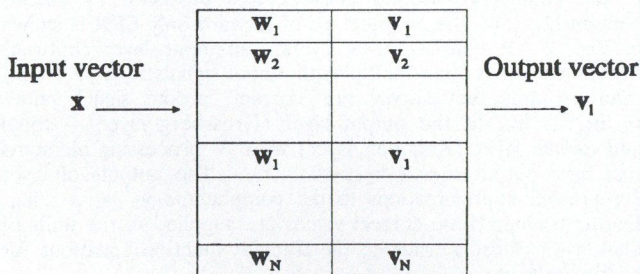


Fig. 4. The CPN as an adaptive lookup table

4. MODIFIED COUNTERPROPAGATION NETWORK

From herein mentioned facts we can see fundamental difference between Kohonen learning in SOM and in CPN. In former network the cells doing learning are arranged 2-dimensionally and they are not affected independently of each other, but as topologically related subsets by defining neighbourhood set. Result of this sort of spatially correlated learning is that the weight vectors tend to attain values that are ordered along the axes of the network [4]. For good global ordering the width of the radius N_c is very wide in the beginning and shrink monotonically with time. The process of learning can be finished with $N_c = \{c\}$, in which case the process is reduced to simple competitive learning. The latter network performs whole learning process as simple competitive learning.

This is why we propose following modified architecture of forward-only CPN:

Input layer contains n units that multiplex the input signals $x_1, x_2, \dots, x_n$ (and m units that supply the correct output signal values $y_1, y_2, \dots, y_m$ to the Grossberg layer) and Kohonen layer with N processing elements that have output signals $z_1, z_2, \dots, z_N$. But N elements of Kohonen layer are arranged in 2 dimensions (supposing rectangular arrangement of the cells and the most simple case when N is power 2 of integer, we have the layer of type $\sqrt{N} * \sqrt{N}$ cells). During training of Kohonen layer (which is now the same as SOM) all attributes of learning in map are maintained (the radius of neighbourhood set N_c shrinks monotonically with time, α is time-variable). Grossberg layer receives the z signals from Kohonen layer (SOM). The processing elements of output layer are governed by Grossberg learning [5].

5. SIMULATION RESULTS

SOM, CPN and modified CPN were used for simulation. SOM had 4096 cells arranged rectangularly as 64×64 cells, CPN had 16 cells in input and Grossberg layer and 4096 cells in Kohonen layer. Modified CPN with 16 cells in input and Grossberg layer and 4096 cells in Kohonen layer was used. Kohonen layer was arranged rectangularly as 64×64 cells. As training set, 4 different 8 bit gray scale images were used, they were divided into 4×4 blocks. Other images, LENA, Girl and Baboon were used for testing, these images were also divided into 4×4 blocks. This configuration results to the bit rate 0.75 bit/pixel.

Image degradation caused by compression is evaluated by means of mean square error MSE and peak S/N ratio defined by $\text{PSNR} = 10 \log(255^2 / \text{MSE}^2)$ [dB].

Visualized SOM (64×64 codewords 4×4) is shown in Fig. 5.

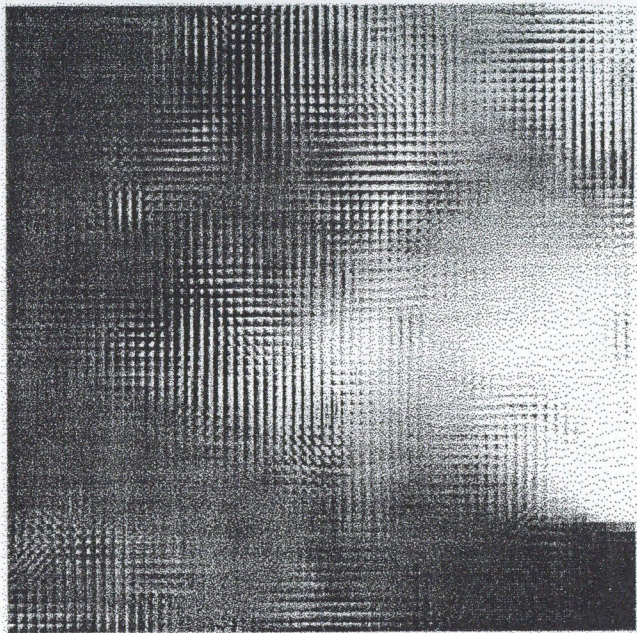


Fig. 5. Visualized SOM

Original image, image processed by SOM, CPN and modified CPN for LENA, Girl and Baboon are shown in Figs. 6, 7, 8, respectively.



Fig. 6. Image LENA: original (top left), processed by SOM (top right), CPN (bottom left), modified CPN (bottom right)



Fig. 7. Image Girl: original (top left), processed by SOM (top right), CPN (bottom left), modified CPN (bottom right)

REFERENCES

[1] N. M. Nasrabadi, R. A. King: "Image Coding Using Vector Quantization: A Review", *IEEE Trans. Commun.*, Vol. 36, pp. 957-971, Aug. 1988.

[2] R. Hecht-Nielsen: *Neurocomputing*, Addison-Wesley Publ. Comp., 1990.

[3] B. Widrow: "30 Years of Adaptive Neural Networks: Perceptron, Adaline and Backpropagation", *Proc. of the IEEE*, Vol. 78, No. 9, pp. 1415-1442, Sept. 1990.

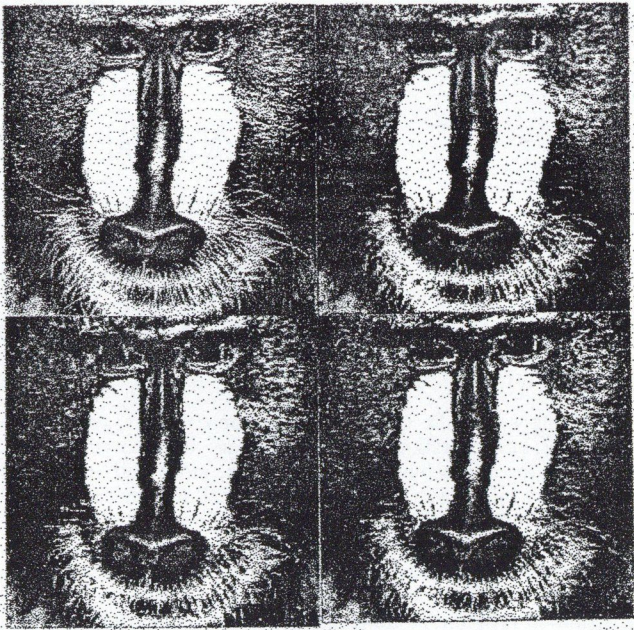


Fig. 8. Image Baboon: original (top left), processed by SOM (top right), CPN (bottom left), modified CPN (bottom right)

Summary of simulation results is given in following table:

	SOM		CPN		modified CPN	
LENA	73.2;	29.5 dB	127.7	27.1 dB	91.9	28.5 dB
Girl	33.6;	32.9 dB	70.5	29.7 dB	41.8	31.9 dB
Baboon	347.4;	22.7 dB	419.2	21.9 dB	369.4	22.5 dB

[4] T. Kohonen: "The Self-Organizing Map", *Proc. of the IEEE*, Vol. 78, No. 9, Sept. 1990.

[5] R. Hecht-Nielsen: "Counterpropagation Networks", *Applied Optics*, Vol. 26, pp. 4979-4984, No. 23, Dec. 1987.

[6] J. Polec, R. Vargic: "Decreasing of Information Redundance of n -dimensional Signal with $(n + 1)$ -dimensional Discrete Orthogonal Transform", *International Conference on Digital Signal Processing '93*, Kosice, 1993.

POSTERS II.

TWO-DIMENSIONAL LEAST SQUARE SVD ALGORITHM

U. BURNIK and J. TASIČ

UNIVERSITY OF LJUBLJANA
"JOZEF STEFAN" INSTITUTE, JAMOVA 39
FACULTY FOR ELECTRICAL AND COMPUTER ENGN., TRŽAŠKA 25
SI-61000 LJUBLJANA, SLOVENIA

G. CAIN

SCHOOL OF ELECTRONIC AND MANUFACTURING SYSTEMS ENGN.
UNIVERSITY OF WESTMINSTER
115 NEW CAVENDISH STREET, LONDON, W1M 8JS, UK

1. INTRODUCTION

Observing a specific object visually, there always appears a problem of interpretation of the necessary information as an image. As the utility provided to capture, store and transport the necessary information never is an optimal one, there exists a necessity to improve the resulting image. Several methods are proposed by the literature; approaches may vary according to the selected image representation, type and level of degradation to be recovered from and to the selected observation domain. The common problem is, however, the same. A degraded image should be affected by a reconstruction function with a resulting image as close to a desired image as possible.

Although the observed object itself is likely to appear analog, the natural form in which it is formed and acquired is a discrete or digital form. This is already at the beginning conditioned by the nature of electronic image sensors. The discrete nature of the image makes the use of digital computers in the reconstruction even more convenient. The efficient tools to use can be acquired from the linear algebra theory.

We have decided to represent the degradation model for our imaging system in a form of discrete linear point-spread degradation functions. For discrete image F degraded to image G and subjected to additive noise N , we may write

$$g(x, y) = \sum_{u=1}^N \sum_{v=1}^N h(x, y, u, v) f(u, v) + n(x, y)$$

or, alternatively, in tensor notation

$$[G] = [[H]]\{[F]\} + [N],$$

with two-dimensional matrices G, F and N and using the four-index operator H [5].

For many cases, it could be assumed that the operator H (or the degradation function itself) is position-independent. In this special case, this is a *space-invariant point-spread function*, and the above tensor product degrades to a two-dimensional convolution:

$$G = H' * F + N.$$

The objective of restoration is to find an inverse to the degradation function. We shall assume, that in limited cases the inverse function appears in a similar convolution form. Even for this special case the solution often remains an ill-conditioned problem.

2. CONVOLUTION FILTER

Consider a situation where it is expected a limited finite-impulse response restoration filter would be an appropriate solution to the problem. Suppose also that all images are real-valued functions. The solution may then be represented in a form of

$$\hat{F} = W * G,$$

with symbol $**$ standing for 2D convolution. We have to point out that the solution in such a formulation exists only for linear, space-invariant distortion functions with finite (space-limited) response.

The general proposed adaptive filter structure for this case is illustrated in Fig. 1.

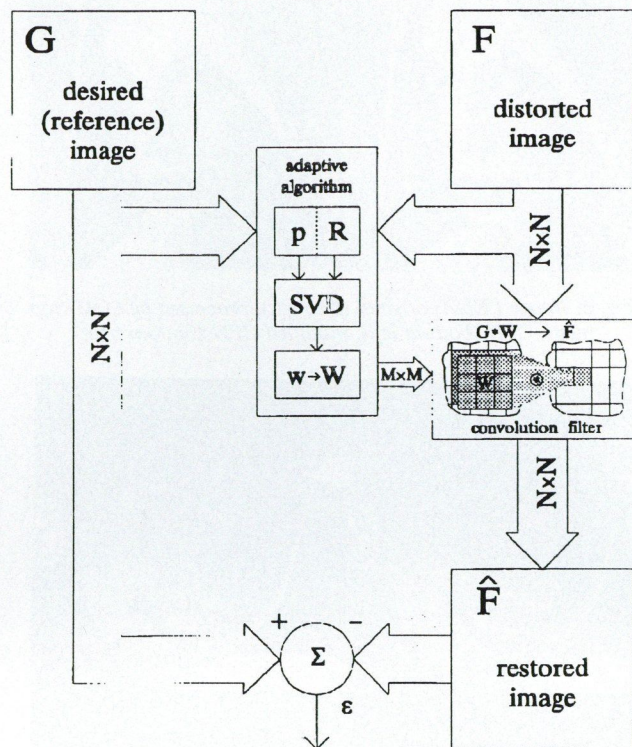


Fig. 1. SVD based 2D adaptive filter

The filter operates on a real image (signal matrix) X that is corrupted with noise. The desired signal (reference image) is also provided. The filtering parameters can be represented in form of an N -by- N matrix W , and the filtering process may be represented by convolving the image input X with the matrix W . During the adaptation, the filtering weights may be changed in order to obtain optimal solution. The filtering result is given by the following equation:

$$\tilde{f}(x, y) = \sum_{i=0}^M \sum_{j=0}^M w(i, j) g(x + i - k, y + i - k).$$

The difference between the desired and the resulting image

$$e(x, y) = f(x, y) - \tilde{f}(x, y)$$

is called the estimation error. From Wiener filter theory, optimal filtering coefficients W are defined by the minimum mean-square error criteria. The objective function $J(W) = E[e^2(x, y)]$ should be minimized for W to obtain the optimum filter.

3. WIENER OPTIMAL SOLUTION

The idea is well known from 1D adaptive filtering, where instantaneous estimates of gradient of the error surface $J(W)$ are used to approach the optimum solution iteratively. The algorithm is popularly called LMS algorithm. It is possible to extend the algorithm to be used in both x and y image dimensions,

iteratively searching for the solution either columnwise or rowwise. The procedure is numerically convenient due to low storage and computing requirements. The problem of this approach is that the instantaneous estimates of the error surface have relatively large variances. The estimate of their gradient vectors may then not always be pointing to a global optimum; the fact could cause unstable performance of the algorithm. The stability may be improved using smaller adaptation step-size, however this seriously affects the convergence rate of the procedure.

Another, in the article described approach captures the information from the entire image, determining the shape and orientation of a global error surface and calculate best possible weight matrix in a single operation. The error surface function for the two-dimensional problem needs to be defined first. In order to keep mathematical notation clear, we decided to restack partial images as well as filtering matrix to a vector form with horizontal and vertical dependency kept in mind.

A single point of the reconstructed image could be defined by the inner vector product

$$\begin{aligned}\tilde{f}(x, y) &= \mathbf{w}^T \mathbf{x}(x, y) \\ \mathbf{w} &= [w(1, 1), w(1, 2), \dots, w(M, M)] \\ \mathbf{x}(x, y) &= [g(x - k, y - k), g(x - k + 1, y - k), \dots \\ &\dots, g(x, y), \dots, g(x + k, y + k)]^T \\ M &= 2k + 1.\end{aligned}$$

Based upon this interpretation, the error surface function is denoted by

$$\begin{aligned}J(\mathbf{W}) &= E[e^2(x, y)] \\ J(\mathbf{W}) &= E[(f(x, y) - \mathbf{w}^T \mathbf{x}(x, y))^2] \\ J(\mathbf{W}) &= E[f(x, y)^2] - 2\mathbf{w}^T E[\mathbf{x}(x, y)f(x, y)] + \\ &\quad + \mathbf{w}^T E[\mathbf{x}(x, y)\mathbf{x}^T(x, y)]\mathbf{w}.\end{aligned}$$

The expectation $E[(f(x, y))^2] = c$ is constant scalar value.

The expectation $E[\mathbf{x}(x, y)f(x, y)]$ is equal to the $M^2 \times 1$ cross-correlation vector between the convolution-input area $\mathbf{x}(x, y)$ of distorted image and desired value $f(x, y)$,

$$\mathbf{p} = E[\mathbf{x}(x, y)f(x, y)].$$

Finally, the expectation $E[\mathbf{x}(x, y)\mathbf{x}^T(x, y)]$ equals the $M^2 \times M^2$ correlation matrix

$$\mathbf{R} = E[\mathbf{x}(x, y)\mathbf{x}^T(x, y)]$$

of the convolution-input area $\mathbf{x}(x, y)$.

Following the presented notation we may express the mean-squared error function as

$$J(\mathbf{W}) = c - 2\mathbf{w}^T \mathbf{p} + \mathbf{w}^T \mathbf{R} \mathbf{w}.$$

For stationary images, the mean-squared error $J(\mathbf{W})$ is exactly a second-order function of correlation coefficients $\mathbf{W}$. The function can be visualized as a parabolic surface having a unique minimum $\mathbf{W}_0$, called *optimum in the minimum mean-squared sense* [2]. We may assume that the gradient of the mean-squared error function at the optimal weights equals zero:

$$\begin{aligned}\nabla &= 0 \\ &= \frac{dJ(\mathbf{W})}{d\mathbf{W}} \\ &= -2\mathbf{p} + 2\mathbf{R}\mathbf{w}\end{aligned}$$

$$\mathbf{R}\mathbf{w}_0 = \mathbf{p}.$$

The equation $\mathbf{R}\mathbf{w}_0 = \mathbf{p}$ is a so-called *normal equation* that defines optimum solution for the convolution weight matrix $\mathbf{W}$. We may notice that for a solution a matrix inversion will be required. Although matrix $\mathbf{R}$ is generally of a smaller size than matrix $\mathbf{H}$ the inversion may still remain an ill-conditioned problem and the solution for $\mathbf{W}$ unstable. That is the reason why special inversion techniques should be employed to solve the mentioned problem.

4. SIGNAL MATRIX AND CORRELATION ESTIMATES

To approach the optimal filtering problem we will first need a good estimate of the correlation matrix $\mathbf{R}$ and cross-correlation vector $\mathbf{p}$. Suppose our knowledge of a specific distortion process is limited to a single original and a single degraded image. With convolution filter matrix of $M \times M$, $M = 2k + 1$ elements only inner $(N - 2k) \times (N - 2k)$ elements of the reconstructed image will be properly defined. Stacking all existing input filter values together we may define an $(N - 2k)^2 \times M$ input signal matrix as:

$$\mathbf{X} = [\mathbf{x}(k, k), \mathbf{x}(k + 1, k), \dots, \mathbf{x}(k, k + 1), \dots, \mathbf{x}(N - k, N - k)]^T$$

with already defined input signal vector

$$\begin{aligned}\mathbf{x}(x, y) &= [g(x - k, y - k), g(x - k + 1, y - k), \dots \\ &\dots, g(x, y), \dots, g(x + k, y + k)]^T.\end{aligned}$$

A good estimate of the correlation matrix is a product

$$\hat{\mathbf{R}} = \frac{1}{N - M + 1} \mathbf{X}^T \cdot \mathbf{X}.$$

Similarly, an estimate for the cross-correlation vector as required for the normal equation can be expressed as

$$\hat{\mathbf{p}} = \frac{1}{N - M + 1} \mathbf{X}^T \cdot \mathbf{d}$$

with

$$\mathbf{d} = [f(2, 2) \ f(2, 3) \dots f(2, N - 1) \dots f(3, 1) \dots f(n - 1, n - 1)]^T.$$

Ignoring constant scalar premultiplications, the normal equation is finally expressed as

$$\begin{aligned}\mathbf{X}^T \mathbf{X} \mathbf{w}_0 &= \mathbf{X}^T \mathbf{d} \\ \mathbf{w}_0 &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{d}.\end{aligned}$$

5. SINGULAR VALUE DECOMPOSITION

The next necessary step is to define a stable inversion process for the matrix $\mathbf{R} = \mathbf{X}^T \mathbf{X}$. There are some well known procedures with stable behaviour even for nonfull rank correlation matrices. In general, those methods attempt to perform inversion for the problem up to its singularity but avoid instability beyond that [5]. One of the possibility is a so-called SDV pseudo-inversion.

Furthermore, it is not recommended to calculate a straight inverse of $\mathbf{R}^{-1} = (\mathbf{X}^T \mathbf{X})^{-1}$. In the inverse, the $(N - M + 1)^2 \times M$ signal matrix $\mathbf{X}$ is premultiplied by its transpose $\mathbf{X}^T$, and later in the expression, $\mathbf{X}^T \mathbf{d}$ is premultiplied by the mentioned inverse. The algorithm may possess more sensitivity than already inherited in the problem. For signal matrix $\mathbf{X}$ with linearly dependent columns, we may have an infinite number of solutions for optimal $\mathbf{W}$ [1].

It is best to simply formulate the solution for $\mathbf{W}$ as

$$\mathbf{w}_0 = \mathbf{X}^+ \mathbf{d},$$

where pseudo-inverse $\mathbf{X}^+$ is defined in terms of the products of the singular-value decomposition of $\mathbf{X}$. The procedure is numerically stable and its solution is unique in that its vector norm is minimum [1].

With singular-value decomposition

$$\begin{aligned}\mathbf{U}^T \mathbf{X} \mathbf{V} &= \Sigma \\ \Sigma &= \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \mathbf{S} &= \text{diag}(\sigma_1, \dots, \sigma_{M^2})\end{aligned}$$

the pseudo-inverse $\mathbf{X}^+$ is defined as

$$\begin{aligned}\mathbf{X}^+ &= \mathbf{V} \Sigma^+ \mathbf{U}^T \\ \Sigma^+ &= \begin{bmatrix} \mathbf{S}^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\end{aligned}$$

$$S = \text{diag}(\sigma_1, \dots, \sigma_{M^2})$$

$$\sigma_i^+ = \begin{cases} 1/\sigma_i, & \sigma_i \neq 0 \\ 0, & \sigma_i = 0 \end{cases}$$

Finally, convolution operator $\mathbf{W}$ can be created by restacking the values of the vector

$$\mathbf{w} = \sum_{i=1}^{M^2} \frac{\mathbf{u}_i^T \mathbf{d}}{\sigma_i} \mathbf{v}_i$$

back to the $M \times M$ matrix form:

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}(1) & \dots & \mathbf{w}(M) \\ \vdots & \ddots & \vdots \\ \mathbf{w}(M(M-1)+1) & \dots & \mathbf{w}(M^2) \end{bmatrix}$$

The non-iteratively calculated filtering parameters are optimal for the specific image/distorted image combination. They may be directly applied in a classical two-dimensional convolution filter.

6. IMPLEMENTATION OF THE PROCEDURE

The final intention of the paper is to show the effectiveness the suggested method on real images.

Fig. 2 represents sharpening of blurred images using two-dimensional least-square SVD algorithm. The original image is shown on Fig. 2a. The image was blurred using a 5-by-5 low-pass convolution filter. Some uncorrelated noise has been added at the end of the blurring procedure (Fig. 2b). Image has been restored using a 7-by-7 adaptive algorithm; the result is shown on Fig. 2c. The results of the inverse filtering of the same image are presented on Fig. 2d. From the result we can deduct that the proposed algorithm is far less sensible than classical inverse filtering procedure. The image blurred without presence of noise and the noiseless image sharpened by using 2D LS SVD algorithm are shown on Figs. 2e and 2f, respectively.

The same procedure applied in noise removal is shown on Fig. 3. From the images it is clear, that the 2D LS SVD algorithm does not converge to expected low-pass solution. A big amount of the uncorrelated noise has been successfully removed from the image preserving sharp edges of the image. The softening of the image contours is a common problem when low-pass filters are used for noise removal (Fig. 2d).

7. COMMENTS

The paper shows that the Wiener filtering principle can successfully be implemented in linear algebraic image restoration. Methods well known from the linear algebra theory may be applied instead of classical methods based on Fourier transformation.

Applying SVD to an entire signal matrix $\mathbf{X}$ is a powerful, yet computing and storage demanding task by itself. This is not the only existing solution of the normal equation. The idea opened for further work is to apply an updating matrix reduction method that may lead to the same result with a more optimal computing power consumption.

REFERENCES

- [1] S. Haykin: *Modern Filters*, McMillan, New York, 1989.
- [2] S. Haykin: *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, N.J., 1991.
- [3] J. Tasic et. al.: *Eigenanalysis in Adaptive FIR Filtering*, Internal Report, University of Westminster, January 1993.



Fig. 2. Sharpening of blurred image using 2D LS SVD algorithm. (a) original image, (b) image, blurred with 5-by-5 low pass convolution filter, additive noise, (c) b, restored with 7-by-7 LS SVD filter, (d) b, inverse filtered, (e) b, noiseless, (f) e, restored



Fig. 3. Removing of noise using 2D LS SVD algorithm. (a) noisy image, (b) original image, (c) a, 2D LS SVD filtered, (d) a, low-pass filtered

- [4] R. C. Gonzales, R. C. Woods: *Digital Image Processing*, Addison-Wesley Publishing Company, 1992.
- [5] H. C. Andres, B. R. Hunt: *Digital Image Restoration*, Prentice Hall, Englewood Cliffs, New Jersey, 1977.

"ADJUSTABLE POLYGONS": A NOVEL ACTIVE CONTOUR MODEL FOR OBJECTS TRACKING ON COMPLEX BACKGROUND

P. DELAGNES, J. BENOIS, and D. BARBA

SEI/LABORATOIRE D'ANALYSE ET DE TRAITEMENT D'IMAGES - IRESTE
LA CHANTRERIE CP3003, 44087 NANTES CEDEX 03
FRANCE

1. INTRODUCTION

Active contour models (known as "snakes" [1]) can easily be applied to deformable objects tracking in the 2D spatial domain. But they yield good results only in relatively simple contexts, for instance when the tracked object moves on a uniform or low-contrasted background [2], or when its shape is rather smooth (like cells [3]). In this paper, we consider the most general case of the spatio-temporal segmentation of deformable objects moving on a complex background.

In the next section, the tracking is performed with a classical active contour model. Thus, the different problems that may arise will be listed. The general "Adjustable Polygons" model is then introduced. Its application to object tracking is finally discussed. For a better object detection on a complex background, the external energy field of active contour models is extended. So, for a given current frame, this term stresses the object that was segmented by a corresponding adjustable polygon in the previous frame.

2. OBJECT TRACKING WITH USUAL ACTIVE CONTOURS

Basically, an active contour is a curve $\{v(s) = (x(s), y(s))\}$ that can move over the image by minimizing its energy E^* defined by:

$$E^* = \int_{curve} (E_{int}(v(s)) + E_{ext}(v(s))) ds.$$

The internal energy, which can be written:

$$E_{int}(v) = \alpha \cdot \left| \frac{dv}{ds} \right|^2 + \beta \cdot \left| \frac{d^2v}{ds^2} \right|^2,$$

serves as a smoothness constraint (the first-order term controls the curve elasticity, the second-order term its flexibility); the external energy, $E_{ext}(v) = -|\nabla I(v)|^2$, pulls the active contour towards nearby edges of image I .

The active contour model we used here was the "B-snakes" model [4], where the curve is composed of B-spline functions which interpolate a set of control points $\{v_i\}$. Thus, this model makes it possible to develop a corner in the curve using a triple control point (whereas a basic snake cannot overcome the smoothness constraint).

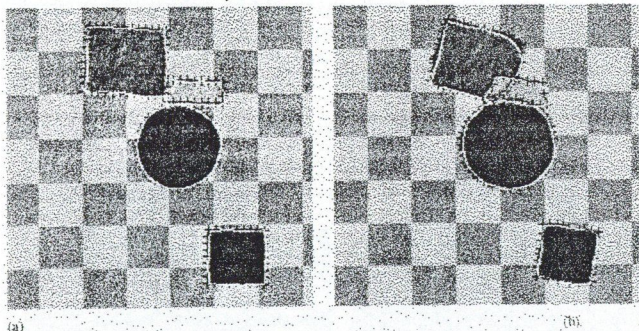


Fig. 1. B-snakes tracking. (a) Frame I_1 ; (b) Frame I_6 (white: converged B-snake; black: initial B-snake; black crosses: initial control points in the frame)

However, B-snakes shown unsatisfying tracking results when used in a rather complex image sequence (Fig. 1), mainly for the following reasons:

- The value of coefficient β (α can be set to 0) is very difficult to find (particularly on the objects corners) because a strong smoothness leads to a loss of the strong curvature points, whereas a weak smoothness provides a bad description of edges (noisy oscillations). Furthermore, in many cases, there is no satisfactory value to be found. This is the main restriction for the use of snakes in most applications.
- The two problems of the tracking by active contours appear and remain unsolved:
 - Attraction of the snake by a "wrong" edge, which belongs to the background or to another object (appears on the right side of the large rectangle in Fig. 1a). This is a normal behaviour of the snake, as the interfering edge may yield a higher contrast than the object border.
 - Snake divergence when the contrast between the background and the object lowers (see the small rectangle in Fig. 1b). This happens when an object moves onto a part of the background having a luminance close to that of the object. In this case, E_{ext} weakens because of the low contrast, so E_{int} becomes predominant and makes the snake leave the object edges.

Some tried to solve these two problems by adding a temporal smoothness constraint [2], [5], but it reinforces the difficulties described in point a).

3. THE "ADJUSTABLE POLYGONS" MODEL

In order to overcome the weaknesses mentioned above, a novel active contour model, called "Adjustable Polygons", is introduced.

This model represents a polygonal contour, which is a set of several "active" segments $\{p_1, \dots, p_n\}$. It allows not to consider any smoothness constraint, because of the segments own rigidity. The active contour energy in this model is the mean external energy E_{ext} computed along every segment. The convergence of an adjustable polygon towards the object edges is achieved in 3 steps:

- First, the energy is minimized for each segment *independently* (Fig. 2). For every segment, a rectangular window is first constructed, named "convergence window". Then, the initial segment $p_i = [v_i v_{i+1}]$ is transformed into an articulated two-segment active curve $\{[v_i S_i], [S_i v_{i+1}]\}$ which moves onto the edges detected in this convergence window. Points v_i and v_{i+1} can move on the window itself (so that the active curve does not shrink), and the new point S_i can move anywhere inside the window (see Fig. 2a). This new point is created to take into account possible deformations of the object border.
- Then, in order to restore the continuity of the polygonal border, all the initial vertices, such as point v_i in Fig. 3, must be moved to cope with both resulting segments on their left and on their right. So the end vertices of adjacent segments are reaccorded geometrically (see Fig. 3).
- Finally, useless vertices are eliminated (those which are aligned with their neighbours or are too close to them).

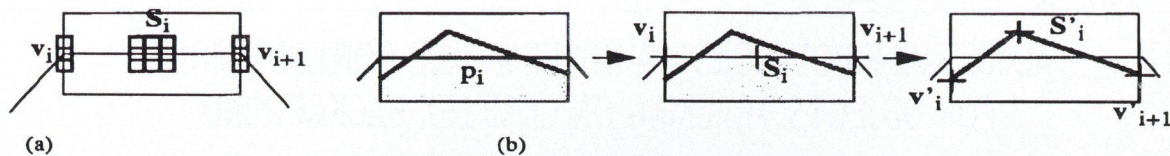


Fig. 2. Convergence of an "active segment" in the Adjustable Polygons model. (a) Convergence window; (b) An active segment converging to a real contour

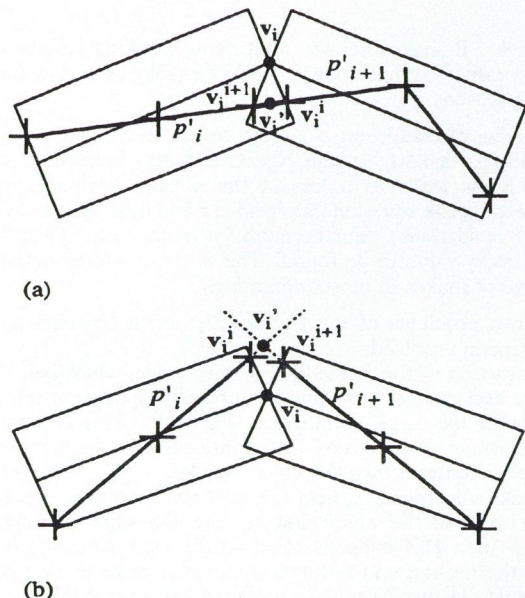


Fig. 3. Geometrical restoration of the polygonal border continuity. (a) Case of colinear resulting segments v'_i is placed in the middle of $[v_i^i v_{i+1}^i]$; (b) Case of non-colinear resulting segments $v'_i = p'_i \cap p'_{i+1}$

This model provides a better segmentation; the adjustable polygons are positioned accurately on the edges and corners of the objects (compared to B-snakes). But they can still be attracted by an interfering background edge. In order to overcome this attraction by a neighbouring contour with stronger energy (that is stronger contrast), a new energy E_{ext} was introduced.

4. MODIFICATION OF THE EXTERNAL ENERGY

In order to push a segment onto the contour of a given object, we use the local texture characteristics m and σ of this object, computed in a confined area along the segment.

The statistical parameters m and σ (i.e. the mean and standard deviation of the gray-levels) of the tracked object in the segment neighbourhood, are determined in the current frame (time t) of the sequence, where the object is already segmented. In the next frame (time $t + 1$), the probability of every pixel to be an object's one (assuming that the texture locally follows a Gaussian law $N(m, \sigma)$) is computed. Finally we consider the gradient $\nabla N_{(m, \sigma)}(x, y)$ of this probability field. The real object contour is stressed by this process.

For each pixel in the convergence window of a segment, the expression of the new external energy field is then given by:

$$E_{ext}(x, y) = [|\nabla N_{(m, \sigma)}(x, y)| \cdot |\nabla I(x, y)|^2].$$

Fig. 4 provides an illustration of this on the entire image, for the upper rectangle. For every pixel, the probability of belonging to the rectangle is shown on Fig. 4a. The gradient of this probability field is then computed (Fig. 4b).

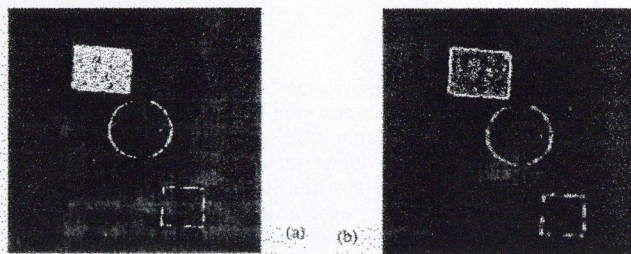


Fig. 4. (a) Probability field and (b) gradient $\nabla N_{(m, \sigma)}$ for the upper rectangle

5. RESULTS AND DISCUSSION

The result of the final "Adjustable Polygons" model, with the new expression of E_{ext} , are presented on Fig. 5 (the occlusion areas were not treated in this case). Both problems of the tracking by active contours are solved:

- Interfering background edges are attenuated by the texture-based external energy field. To make a comparison with Fig. 1, the right side of the upper rectangle is accurately segmented, in spite of a more contrasted chessboard edge being very close.
- An active segment divergence in edgeless area can be avoided easily: if its energy decrease is too slight during the minimization process, it is kept on its initial position.

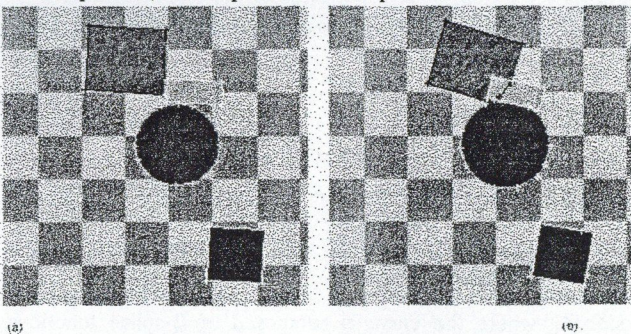


Fig. 5. Adjustable Polygons tracking (crosses: polygonal contour vertices). (a) Frame I_1 ; (b) Frame I_6

Let us notice that these results could be achieved in a simple way because we considered several local minimizations instead of a global one as it is the case for any other active contour model. Also, the minimization process described in section 3 can be easily parallelized.

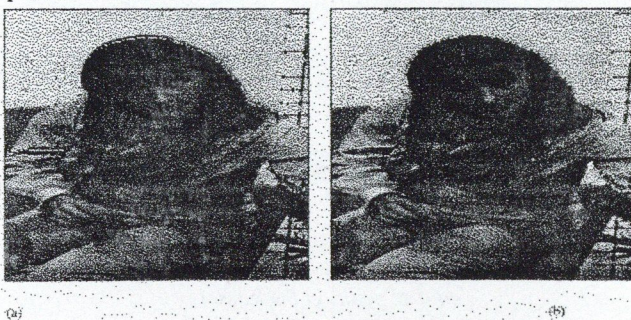


Fig. 6. Adjustable Polygons segmentation from regions predicted shape (black points: vertices). (a) Prediction; (b) Adjustment

The model was recently included in an object-oriented image sequence encoder [6], where it yielded a satisfactory behaviour.

Fig. 6 shows the segmentation of three moving regions by adjustable polygons, starting from the regions predicted position

REFERENCES

- [1] M. Kass, A. Witkin, D. Terzopolous: "Snakes: active contour models", *Int. Journal of Computer Vision*, Vol. 1, No. 4, pp. 321-331, 1988.
- [2] K. Fujimura, N. Yokoy, K. Yamamoto: "Motion tracking of deformable objects by active contour models using multiscale dynamic programming", *Journal of Visual Comm. & Image Repr.*, Vol. 4, No. 4, pp. 382-391, Dec. 1993.
- [3] F. Leymarie, M. D. Levine: "Tracking deformable objects in the plane using an active contour model", *IEEE Tr. on PAMI*, Vol. 15, No. 6, pp. 617-633, June 1993.

in the image plane. It can be seen that quite different types of region shapes can be handled by the model.

ACKNOWLEDGEMENTS

This work was supported by the CCETT, France.

- [4] S. Menet, P. St-Marc, G. Medioni: "Active contours models: overview, implementation and applications", in *System, Man & Cybernetics*, pp. 194-199, 1990.
- [5] M. Takayama, M. Imai, S. Tsuji: "Determining motion of non-rigid objects by active tubes", *11th ICPR*, pp. 647-650, The Hague, Sept. 1992.
- [6] J. Benois, L. Wu, D. Barba: "Joint contour based and motion based sequences segmentation for TV image coding at very low bit rate", submitted to *VCIP 94*.

CAD SYSTEM FOR PATTERN RECOGNITION AND DSP WITH USE OF FAST TRANSLATION INVARIANT TRANSFORMS

J. TURÁN, L. KÖVESI, and M. KÖVESI

DEPARTMENT OF RADIOELECTRONICS
TECHNICAL UNIVERSITY OF KOŠICE
SLOVAKIA

1. INTRODUCTION

The class of fast translation invariant transforms (CT — Certain Transforms) [1] is well known in the field of pattern recognition which may be used for position invariant pattern classification problems. The class CT originated from rapid transform (RT) developed in [2]. The RT results from a minor modification of the Walsh—Hadamard transform (WHT). The signal flow graph for the RT is identical to that of the WHT, except that the absolute value of the output of each stage of the iteration is taken before feeding it to the next stage. This is not an orthogonal transform, as no inverse exists. RT has some interesting properties such as invariance to cyclic shift, reflection of the data sequence, and slight rotation of a two-dimensional pattern. Various properties of RT have been developed in [2], [3], [4]. RT was used in recognition of alphanumeric characters [2], [5], [6], robotics [7] and scene analysis [4], [8]. With the help of additional data, however, the original signal can be recovered from the RT sequence [9], [10], [11], so it is possible to compute inverse rapid transform (IRT). Then IRT may be used for one- or two-dimensional signal coding and nonlinear filtering [11]. In some applications (character recognition), it is required that the transform be noninvariant under reflection, however, RT is invariant under reflection. The modified rapid transform (MRT) was presented to break undesired invariances of the RT [12], which leads to a loss of information about the original pattern. This is achieved by combining the RT with preprocessing steps using a neighbour operator. If the neighbour operator is asymmetric, the undesired invariance under reflection can be overcome with only one preprocessing step. Thus the MRT can distinguish many more patterns from one another than the original RT can.

The programme code CT-CAD was developed as a CAD system for projecting pattern recognition systems with use of RT, MRT and other transforms members of the class CT for invariant feature selection and use of IRT for one- or two-dimensional signal coding and nonlinear filtering.

In the following, short outline of the class CT and properties of MRT and IRT are first reviewed; then the applications of CT, MRT and IRT in pattern recognition and DSP are presented. Finally, the developed programme package CT-CAD will be outlined.

2. THE CLASS OF FAST TRANSLATION INVARIANT TRANSFORMS

Regarding only finite discretized patterns we use the following vector notation for one-dimensional patterns:

$$\mathbf{x} = \{x_s\}, s = 0, 1, \dots, N-1, \quad (1)$$

where $N = 2^n$ and n is a positive integer. For the uniform division of a vector in m subvectors we choose the following notation:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{1/m} \\ \mathbf{x}_{2/m} \\ \vdots \\ \mathbf{x}_{m/m} \end{bmatrix}, m \text{ is divisor of } N, \quad (2)$$

where $\mathbf{x}_{i/m}$ denotes the i -th of subvectors containing N/m consecutive elements of $\mathbf{x}$. A transformed vector will be written as

$$\bar{\mathbf{x}} = \tau(\mathbf{x}), \quad (3)$$

where $\tau \in \text{CT}$. The one-dimensional class of transform CT^N may now be defined recursively [13] as (Fig. 1)

$$\bar{\mathbf{x}} = \begin{bmatrix} \tilde{f}_1(\mathbf{x}_{1/2}, \mathbf{x}_{2/2}) \\ \tilde{f}_2(\mathbf{x}_{1/2}, \mathbf{x}_{2/2}) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{1/2}^{(1)} \\ \mathbf{x}_{2/2}^{(2)} \end{bmatrix} = \dots, \quad (4)$$

where f_1, f_2 are arbitrary commutative operators. Similar recursive definition may be used also in two-dimensional case $\text{CT}^{N \times N}$. The more useful members of CT are the transforms RT, NT, MT, QT and BT defined with the choosing of pairs of commutative operators (Table 1). The class CT may be generalized to GCT [13]. With the use of a set of symmetric operators we may extend the transform patterns of arbitrary lengths, where N is no a power of 2. In particular, when $N = 2^n$, GCT results in CT.

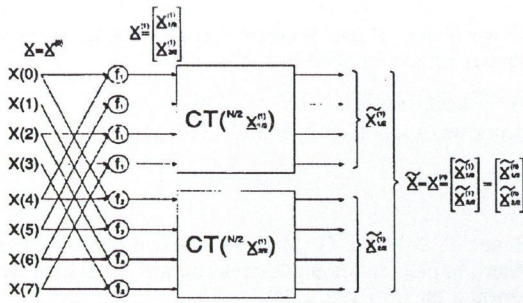


Fig. 1. The recursive definition of the class CT

Table 1. Class of fast translation invariant transforms

Transform Operator	N_{RT}	N_{QT}	N_{NT}	N_{MT}	N_{BT}
$f_1(a, b)$	$a + b$	$a + b$	$\max\{a, b\}$	$a \vee b$	$a \vee_m b$
$f_2(a, b)$	$ a - b $	$(a - b)^2$	$\min\{a, b\}$	$a \wedge b$	$a \wedge_m b$
Patterns	real-valued	real-valued	real-valued	binary-valued	integer-valued

The most interesting properties of CT is the translation invariance. The class CT maps the complete set of translations of a pattern x into one point of the transform domain

$$\tau(\pi(x)) = \tau(x) = \bar{x}, \quad \forall \tau \in CT, \quad (5)$$

where $\pi(x)$ denotes the complete set of elements generated from (x) applying all operators of the set of translation operations. A class CT is not only invariant to translations, but also to a more general class of permutations, which want to reduce the complete set of invariants of a pattern x under the class of transform CT

$$\pi_{CT}(x) = \{x_x | \tau(x_x) = \tau(x), \quad \forall \tau \in CT\}. \quad (6)$$

The subset of these permutations are the cyclic as well as the dyadic permutations. The reflection or mirror pattern is an exceptional element of the class of dyadic permutation, then CT is invariant of reflection of input patterns.

3. MODIFIED AND INVERSE RT

The modified rapid transform (MRT) [12] was presented to break undesired invariances (invariance to reflection) of the RT which leads to a loss of information about the original pattern. This is achieved by combining the RT with preprocessing steps using an asymmetric neighbour operator α . This operator is used to break undesirable invariances but keep the shift invariance of MRT.

Signal graph of MRT (Fig. 2) results from signal graph of RT with adding in general k preprocessing steps $x' = \alpha x$. This maps the element $x(i)$ of input vector x to element $x'(i)$ of vector x' by working on elements $x(i)$, $x(i+1)$ and $x(i+2)$

$$x'(i) = f_0(x(i), x(i+1), x(i+2)). \quad (7)$$

Operator f_0 may be realized in the following simple manner [12]

$$x'(i) = f_0(x(i), x(i+1), x(i+2)) = x(i) + |x(i+1) - x(i+2)|. \quad (8)$$

For two-dimensional patterns one can use either one-dimensional RTs in sequence, for the horizontal (x) and vertical (y) directions, or one two-dimensional RT. Then we can choose the following symmetries of the neighbour operators:

$$f_0^x : x'(i, j) = x(i, j) + |x(i+1, j) - x(i+2, j)| \quad (9a)$$

$$f_0^y : x'(i, j) = x(i, j) + |x(i, j+1) - x(i, j+2)| \quad (9b)$$

$$f_0^{x+y} : x'(i, j) = x(i, j) + |x(i+1, j) - x(i+2, j)| + |x(i, j+1) - x(i, j+2)| \quad (9c)$$

$$f_0^{xy} : x'(i, j) = x(i, j) + |x(i+1, j) - x(i, j+1)|, \quad (9d)$$

where $x'(i, j)$ and $x(i, j)$, $i, j = 0, \dots, N-1$ are pixels of the desired preprocessed and input two-dimensional patterns. The concept of modification can be used for all transforms from the class CT based on two commutative operators. So the class of modified certain transforms (MCT) may be created. The efficiency of the modification may be different for each transform.

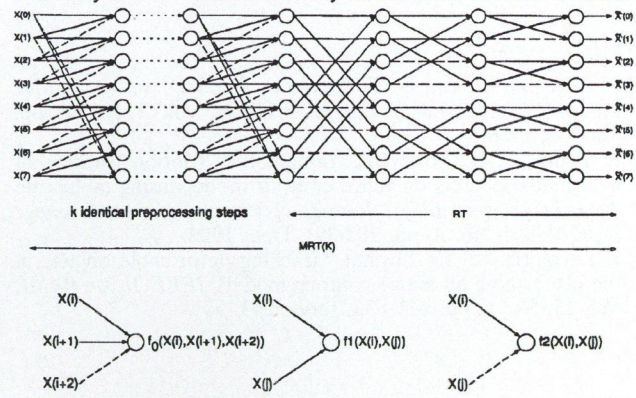


Fig. 2. Signal flow graph of the modified rapid transform

Even if RT is nonlinear and thus noninvertible transform, with computation of additional data in transform process of RT it is possible to compute inverse RT (IRT) [9], [10], [11]. Thus additional data are known as a matrix of states K . For one-dimensional RT the matrix of states may be computed as follows:

$$k(i, r-1) = 0, \quad \text{if } x^{(r-1)}(i) - x^{(r-1)}(i + N/2) < 0 \\ k(i, r-1) = 0, \quad \text{if } x^{(r-1)}(i) - x^{(r-1)}(i + N/2) \geq 0, \quad (10)$$

where $x^{(r-1)}(i)$ and $x^{(r-1)}(i + N/2)$ are values of input vector x after $r-1$ steps of RT transform process. The dimension of binary matrix K is $n \times N/2$. From transform coefficients of the RT $\bar{x} = RT\{x\}$ and its matrix of states K we can obtain the original input data x , with use of IRT defined as follows:

$$x^{(r+1)}(i) = x^{(r)}(2i) + (-1)^{(1+k(i, n-r+1))} x^{(r)}(2i+1) \quad (11) \\ x^{(r+1)}(i + N/2) = x^{(r)}(2i) + (-1)^{(1+k(i, n-r+1))} x^{(r)}(2i+1),$$

where $i = 0, 1, \dots, N/2-1$ and $r = 0, 1, \dots, n-1$. After n inverse steps (11) and dividing the resulted data by N we obtain the original input vector x .

4. CT, MCT AND IRT APPLICATIONS

Let first consider the applications of CT and MCT in pattern recognition. The basic block scheme of the recognition system is on Fig. 3. It contains the following sub-systems;

- Original digital picture preprocessing system CSPO-II [14] was used to accept the physical input picture and then transduces it into a measurable matrix of $N \times N$ pixels.
- The CT processor according to its function may be also called a feature extractor. A two-dimensional CT or MCT of all prototypes is taken in this stage. Than feature selection is carried out in the CT or MCT "spectral" domain, on various basis (maximal value of spectral coefficients, variance zonal sampling and inter class standard deviation).
- The selected CT or MCT features of patterns are in the teaching process feeded into the memory. Thus the memory unit learn apriori knowledge of each class before the system can be used to make any decision. In the recognition process the selected CT or MCT features are feeded into the classifier, which discriminates each pattern and assigns a category (class) to it by some decision rule. We used simple classifiers based on Euclidean or maximum absolute distance.

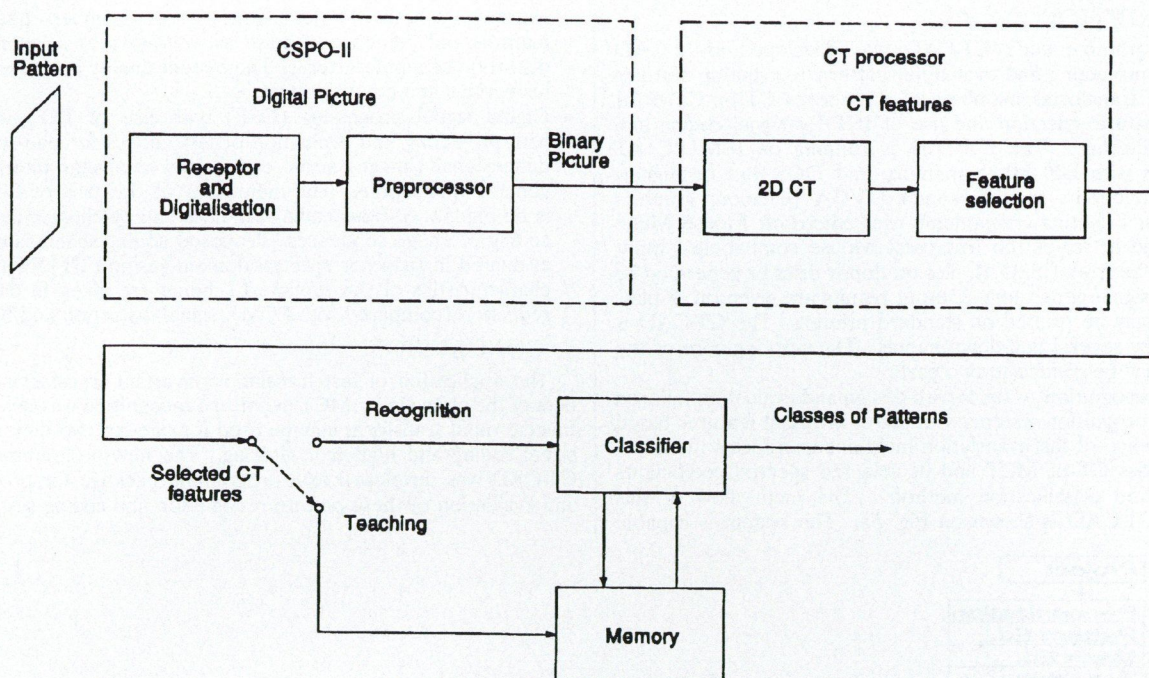


Fig. 3. Basic block of the recognition system

RT and IRT may be used for coding and data compression. Application of RT to signal x we obtain transform coefficients $\tilde{x}$, which are together with the matrix of states K transmitted to the decoder. With reduction of the number of used transform coefficients in transmitter we may reduce the needed data flux. This concept may be used for coding one- or two-dimensional signals and various coding schemes may be considered [15]. For the simplicity we consider the block scheme of IRT coder with reduction of transform coefficients and number of transmitted fields showed on Fig. 4. Input digital image is divided into blocks of pixels,

i.e. vectors which are input of RT processor. The processor compute transform coefficients of these vectors and appropriate matrices of states of each block. Data flux needed to transmission of these vectors and matrices may be reduced with reduction of used (transmitted) transform coefficients. Further reduction of data flux may be obtained with reduction of transmitted pictures of the picture sequence. It means that we transmit only reduced number of transform coefficients of some choosed pictures and matrices of states of all pictures from the sequence.

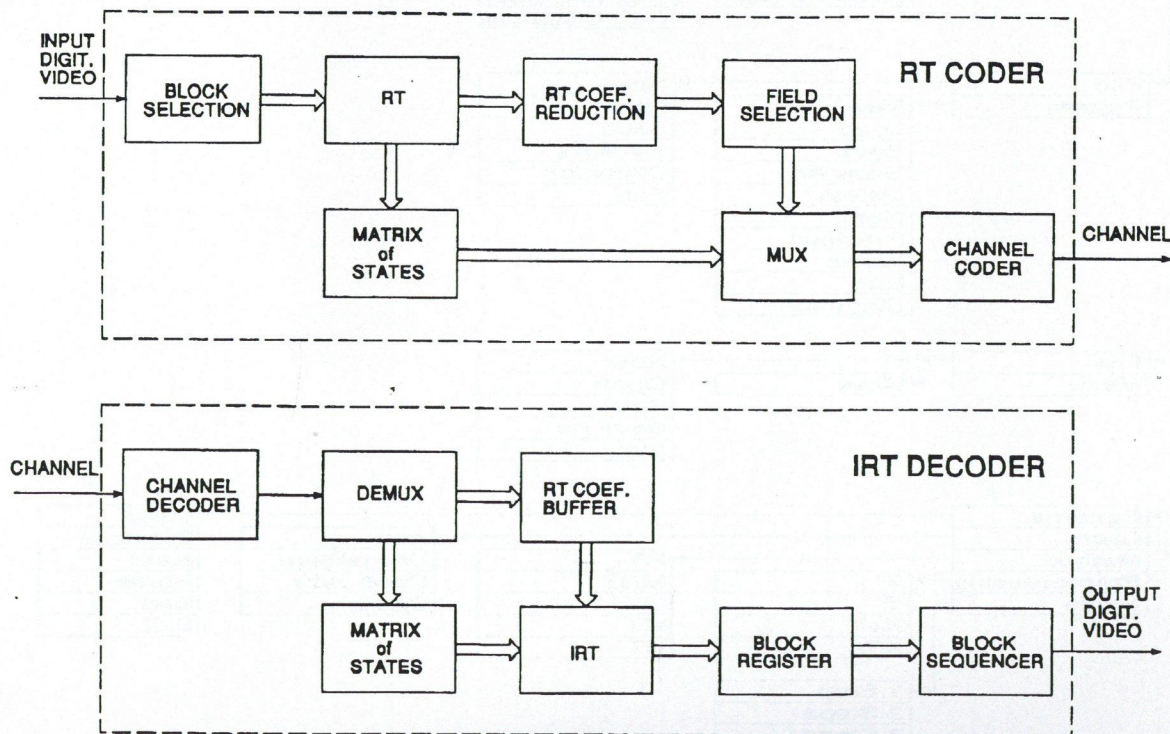


Fig. 4. Block scheme of IRT codec with reduction of transform coefficients and number of transmitted fields

The quality of the coding/decoding process of using IRT may be evaluated in signal (image) or spectral domain. The processed

signals may be compared with using simple Euclidean distance or signal to noise ratio (SNR).

5. PROGRAMME PACKAGE CT-CAD
DESCRIPTION

The programme code CT-CAD was developed as a CAD system for projecting and evaluation pattern recognition systems with use of transforms members of the classes CT or CTM for invariant feature selection and use of IRT for signal coding and nonlinear filtering. The CT-CAD is running on IBM PC AT compatibles with 640 kByte memory and DOS 4.xx or higher. System components require standard VGA 16 colour graphics display. For selecting components uses Microsoft Mouse Menu software and a two-button Microsoft Mouse compatible. Input signals may be from CSPO-II, files on floppy discs or generated by embedded signal generators. Output results are given on display, as files or may be printed on standard printers. The CT-CAD is controlled by several pull-down menus. The working core of the CT-CAD may be divided to two parts:

- Pattern recognition — deals with design and evaluation tailored pattern recognition systems with use of invariant features based on the choice of fast translation invariant transforms members of the class CT or MCT and its selected spectral coefficients and desired classification method. The menu flow of this part of CT-CAD is shown on Fig. 5a. The system is capable

to evaluate recognition efficiency of the proposed feature extraction scheme, its dependence on the number of selected features and influence of noise as well as various (geometric) distortion of input patterns. The system quality is expressed by histograms and confusion matrix.

- Digital signal processing (DSP) with use of IRT — deals with projecting and evaluation of fast codecs for one- or two-dimensional (image) signal coding and nonlinear filtering for general applications. The menu flow of this part of CT-CAD is on Fig. 5b. Various coding schemes can be chosen, including coding of image sequences. Proposed coding schemes may be evaluated in signal or spectral domain (using FFT). Numerical characteristics of the proposed scheme are given in data flux reduction (compared with PCM), signal distortion and SNR.

6. CONCLUSION

The application of fast translation invariant transforms members of the class CT or MCT in pattern recognition was described. Inverse rapid transform may be used for one- or two-dimensional signal coding and nonlinear filtering. The new programme code CT-CAD was developed as a programme package for projecting and evaluation of these pattern recognition and coding systems.

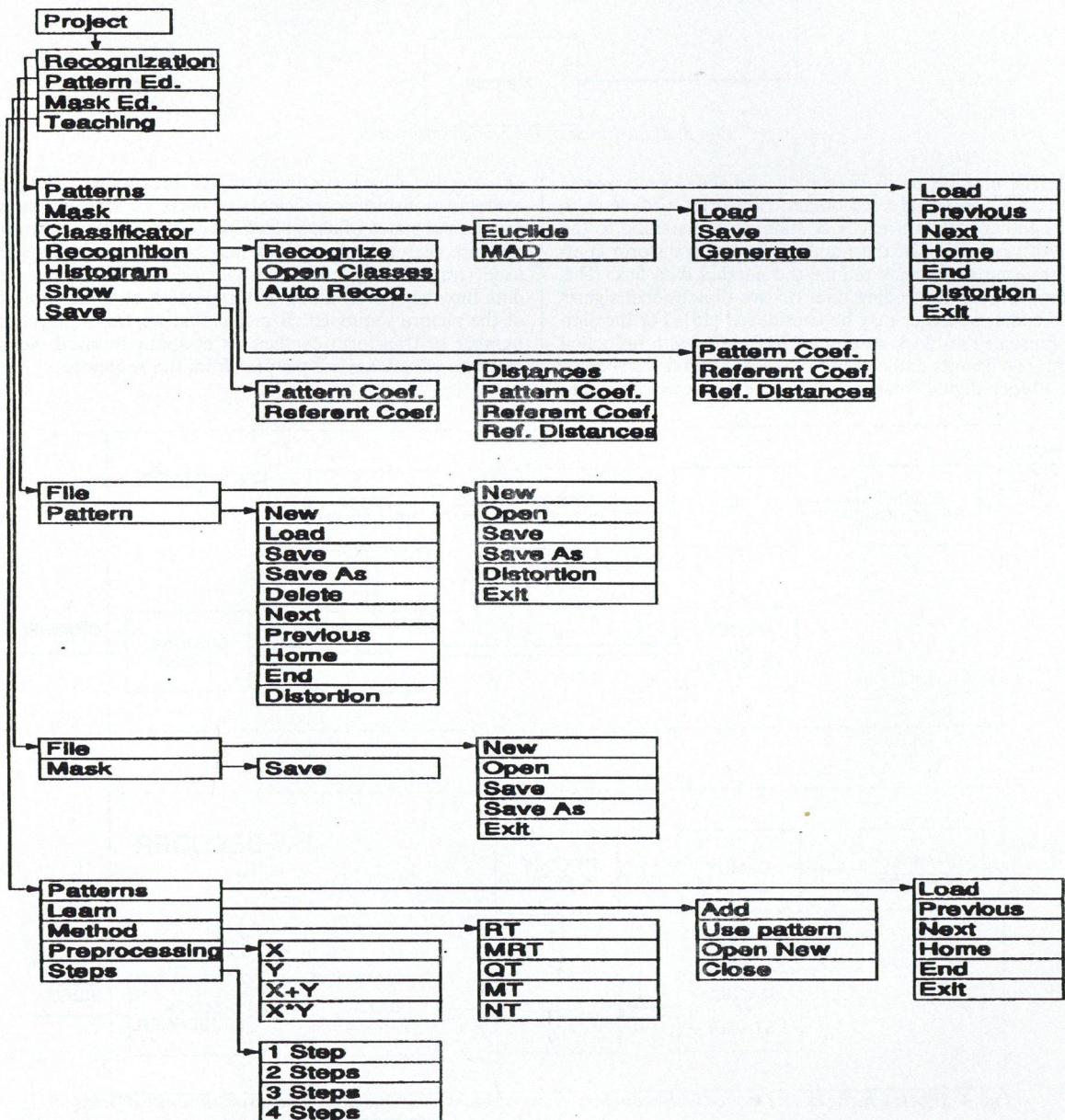


Fig. 5a. Menu flow of programme package CT-CAD: pattern recognition

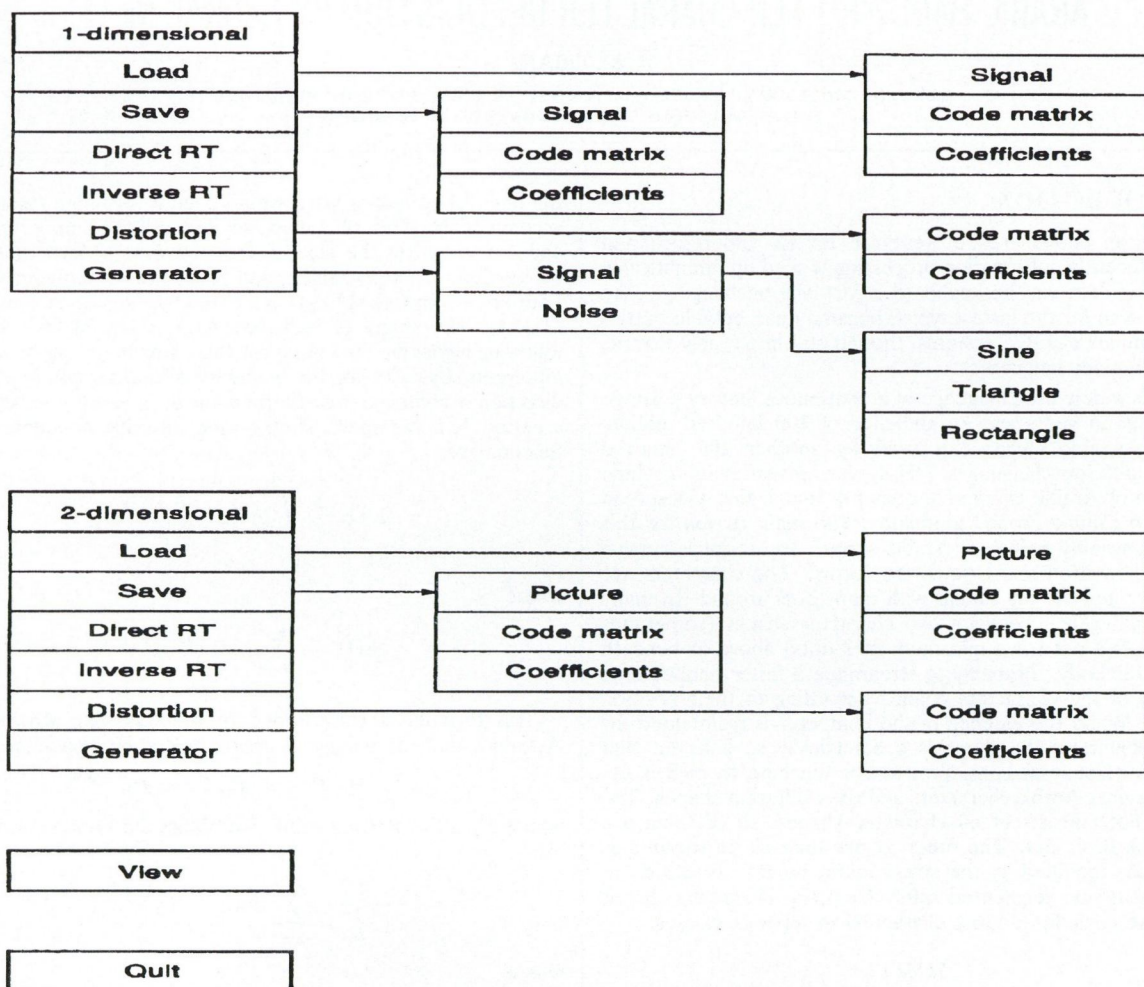


Fig. 5b. Menu flow of programme package CT-CAD: digital signal processing

REFERENCES

- [1] M. D. Wagh, S. V. Kanetkar: "A Class of Translation Invariant Transforms", *IEEE Trans. on Acoustic, Speech and Signal Proc.*, Vol. ASSP-25, No. 3, 1977, pp. 203-205.
- [2] H. Reitboeck, T. P. Brody: "A Transformation with Invariance Under Cyclic Permutation for Application in Pattern Recognition", *Inf. and Control*, Vol. 15, 1969, pp. 130-154.
- [3] M. D. Wagh, S. V. Kanetkar: "A multiplexing theorem and generalisation of R-transform", *Int. J. Comp. Math. Sec. A*, Vol. 5, 1975, pp. 163-171.
- [4] H. Burkhardt: "Transformationen zur lageinvarianter Merkmalgewinnung". *Fortschnittsbericht*. (Reihe 10, Nr. 7), VDI-Verlages, Düsseldorf, 1979.
- [5] P. O. Wang, R. C. Schiau: "Machine recognition of printed chinese characters via transformation algorithms", *Pattern Recognition*, Vol. 5, 1973, pp. 303-321.
- [6] M. A. Nasarimhan, V. Devarajan, K. R. Rao: "Simulation of alphanumeric machine print recognition", *IEEE Trans.*, Vol. SMC-10, 1980, pp. 270-275.
- [7] J. Chmurný, J. Turán: "Two-dimensional Fast Translation Invariant Transforms and Their Use in Robotics", *Electronic Horizon*, Vol. 15, No. 5, 1984, pp. 211-220 (in Slovak).
- [8] H. Schütte, S. Frydrychowicz, J. Schröder: "Scene matching with translation invariant transforms". *5-ICPR*, Miami, USA, 1980, pp. 195-198.
- [9] V. Vlasenco, K. R. Rao: "Unified matrix treatment of discrete transforms", *IEEE Trans.*, Vol. C-28, 1979, pp. 934-938.
- [10] J. Turán, J. Chmurný: "Two-dimensional Inverse R Transform", *Computers and Art. Intelligence*, Vol. 2, No. 5, 1983, pp. 473-477.
- [11] M. Fang: "Class of Invertible Shift Invariant Transforms", *Signal Processing*, Vol. 23, No. 4, 1991, pp. 35-44.
- [12] M. Fang, G. Häusler: "Modified Rapid Transform", *Applied Optics*, Vol. 28, No. 6, 1989, pp. 1257-1262.
- [13] H. Burkhardt, X. Müller: "On Invariant Sets of a Certain Class of Fast Translation — Invariant Transforms", *IEEE Trans.*, Vol. ASSP-28, 1980, pp. 517-523.
- [14] J. Turán, J. Mihal'ik, M. Chudáček: "Digital picture preprocessing system", *Journal of Electrical Eng.*, Vol. 40, No. 9, 1989, pp. 693-698 (in Slovak).
- [15] J. Turán, J. Gamec, L. Kövesi: "Projecting of Rapid Transform Codec", *MICRO-CAD SYSTEMS 93*, Košice, Nov. 9-10, 1993, pp. 201-212.
- [16] J. Turán: "Use of Rapid and Modified Rapid Transform in Optical Character Recognition", *Int. Conf. on DSP'93*, Kosice, Sept. 7-8, 1993, pp. 194-197.
- [17] J. Turán, J. Gamec: "Application of 1D-Inverse Rapid Transform in Picture Coding", *Intelligent Terminals and Source and Channel Coding*, COST 229, Budapest, Sept. 7-9, 1993, pp. 293-294.
- [18] J. Gamec, J. Turán: "Motion Estimation with Use of 1D-Inverse Rapid Transform", *Intelligent Terminals and Source and Channel Coding*, COST 229, Budapest, Sept. 7-9, 1993, pp. 213-218.

ARABIC HANDWRITTEN CHARACTER RECOGNITION BY NEURAL NETS

A. M. OBAID

DEPT. OF MICROWAVE TELECOM., TECHNICAL UNIVERSITY OF BUDAPEST
H-1111, GOLDMAN TÉR 3., BUDAPEST, HUNGARY

1. INTRODUCTION

OCR is an active area of research for its importance and urgency. Recently, information processing is sped up dramatically, yet inputting data via the keyboard is virtually unchanging. The keyboard, even for the fastest typist, became unacceptable bottleneck. Commercial OCR systems, though still intolerably erratic, appeared in many languages.

Arabic is widely used language of a continuous literary heritage of 1.5 thousand years and an audience of 200 hundred million. Moreover, Arabic alphabet is used by another 200 hundred million in different languages. However, research in machine recognition of Arabic characters does not match this widespread use, though gaining ground gradually. The main reason for this lag is the unwieldy nature of Arabic script. The script is cursive in both handwritten and typewritten forms. The characters are very slender and oftenly similar with many dots around. In many cases the difference between two character shapes (sometimes more) is only a dot (or a maximum of 3 dots) above or beneath the character body. Probably to streamline a faster handwriting, the shapes of the characters change according to their position in a word. While the essence of the character is maintained for human recognition, the shape is geometrically so different that it warrants a separate consideration for machine to pick it up. Table 1 presents Arabic characters and their different shapes. The Extended Set consists of 60 character shapes, 29 of them are those of the Basic Set. The other 31 are their shape variants as well as those modified by the stress mark, hamza. Because the hamza can not be segmented safely from the characters that it modifies, we considered those characters in separate classes.

Table 1.

Basic set	ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي
	ط ظ ع غ ف ق ك ل م ن ه و ي
Extended set	أ إ ب ب ت ث ث ج ح ح خ خ د ذ ر ز س س ش ش ص ص ض ض ط ط ظ ظ ع ع غ غ ف ف ق ق ك ك ل ل م م ن ن ه ه و و ي ي
	ذ ر ز س س ش ش ص ص ض ض ط ط ظ ظ ع ع غ غ ف ف ق ق ك ك ل ل م م ن ن ه ه و و ي ي
	ع ع غ غ ف ف ق ق ك ك ل ل م م ن ن ه ه و و ي ي
	ن ن ه ه و و ي ي

OCR systems consist broadly of two parts; a feature extractor and a classifier. The first part is the most difficult to realise, since it very much depends on the nature of the feature that might be so fuzzy to define or difficult to extract with a reasonable cost. Usually to extract features, preprocessing and segmentation stages are indispensable. Every stage is a field of research on its own right.

For the classification part, different techniques have been tried. Recently, neural nets have showed promising results. They have been tried for classifying alphanumeric characters in different European and Asian languages. Our objective in this paper is to use the traditional MLP with its backpropagation learning algorithm to classify a handwritten Arabic characters. Since, as we mentioned above, Arabic script is cursive, we assume that the characters are already segmented and we present them to our network in this form. Segmentation stage is beyond the scope of this paper.

2. FEATURE EXTRACTION

The segmented handwritten characters are scanned into binary image of 300 dpi resolution. As it can be seen from Table 1, using a simple pixel representation as feature input to the classifier is not viable since the difference might be as small as a dot. Besides,

the fine discriminating parts would need a very high resolution to accommodate. This, of course, will increase the number of inputs, and consequently the size of the network. More time would be needed for both training and recognition. Instead, we used Fourier Descriptors (FD's) augmented by Topological Descriptors (TD's). The shape of each character is traced by a contour-following algorithm that produces the outer boundary in a form of a polygon. By following the boundary of the polygon in clockwise direction a complex periodic function $u(l) = x(l) + jy(l)$, with a period L , is obtained. The Fourier Descriptors defined on this function are:

$$a_n = \frac{1}{L} \int_0^{2\pi} u(l) e^{-jn(\frac{2\pi}{L})l} dl,$$

where

$$u(l) = \sum_{n=-\infty}^{\infty} a_n e^{jn(\frac{2\pi}{L})l}.$$

This formula is transformed to the following algorithm [2]. Assuming that the polygon is approximated by m points

$$P_0, P_1, \dots, P_{m-1} = P_0,$$

where P_0 is the starting point. We obtain the Fourier Descriptors as:

$$a_n = \frac{1}{L(\frac{n2\pi}{L})^2} \sum_{k=1}^m (b_{k-1} - b_k) e^{-jn(\frac{2\pi}{L})l_k},$$

where

$$l_k = \sum_{i=1}^k |p_i - p_{i-1}|$$

$$b_k = \frac{p_{k+1} - p_k}{|p_{k+1} - p_k|}.$$

As it can be seen from Table 1, some characters have almost identical shapes. This means that their FD's are similar. To differentiate between these similar shapes, a set of TD's is considered. It is observed that the number and position of dots is a good measure that discriminate between characters uniquely. An algorithm that works in association with that of FD's identifies the dot or cluster of dots and their relative position to the body of the characters. Thus, the input is represented by a feature vector of 43 elements; 40 of them is FD's and 3 TD's. The FD's were given as real numbers, while TD's in binary coded numbers.

Table 2.

No	Feature	Pattern
1	No dots	0 0 0
2	One dot above	0 0 1
3	One dot beneath	0 1 0
4	Two dots above	0 1 1
5	Two dots beneath	1 0 0
6	Three dots above	1 0 1
7	Hamza (cluster) above	1 1 0
8	Hamza (cluster) beneath	1 1 1

Three bits were sufficient to accommodate all possible variations in the topological feature space as shown in Table 2.

3. EXPERIMENTATION

A data base of 1800 segmented handwritten characters is obtained from 10 different writers, where every one wrote the Extended Set three times. The shapes were scanned with a resolution of 300 dpi. A shape matrix of 60×60 pixels were used by the feature extracting algorithms to obtain the FD's and the TD's. The size of the segmented character — though handwritten — is approximately equivalent to 15 points of printed type. The large size of the shape matrix places a heavy burden on the feature extraction stage. However, this price can not be avoided if we were to obtain a faithful feature representation. The features are presented to the network in a vector of 43 elements. Thus, the network has 43 inputs. The number of hidden units is 45 and the output is 60. The database is divided evenly (900 characters) between the training set and the testing set. Training took a total of 90 thousand passes with a hundred pass for each character shape. Since rejection is practically more important than misclassification, we minimized the latter by placing a restrictive criterion on output levels. The highest output is considered a reject if it does not exceed a given threshold t_1 or that the difference between the two highest levels is smaller than a threshold t_2 . Otherwise the character is classified to the class associated with the highest activation level. The learning rate η and the momentum rate α were set to 0.9 and 0.65 respectively. The recognition rate without rejection was 90.09 % and misclassification rate was 9.01 %. While with rejection recognition was 84.3 %, misclassification was 1.7 % and rejection was 14 %.

4. ANALYSIS AND CONCLUSION

Since there is no objective measure of good handwriting, it is

REFERENCES

- [1] A. Amin, G. Masini: "Machine Recognition of Cursive Arabic Words", *SPIE*, Vol. 359, Application of Digital Image Processing IV, 1982.
- [2] A. Krzyzak, W. Dai, C. Suen: "Classification of large Set of Handwritten Characters Using Modified Back propagation Model", *IEEE Int. Conf. on Neural Networks*, Vol. III, pp. 225, Seattle, 1991.
- [3] B. Muhlbaier, A. Koop: "SCRIPT-A Prototype for the Recognition of Continuous, Cursive, Handwritten Input by Means of a Neural Network Simulator", *IEEE Int. Conf. on Neural Networks*, pp. 1672, San Francisco, Cal. USA, April 1993.
- [4] C. Suen, C. Nadal, R. Legault, T. Mai, L. Lam: "Computer Recognition of Unconstrained Handwritten Numerals", *Proceeding of the IEEE*, Vol. 80, No. 7, pp. 1162, July 1992.
- [5] G. Granlund: "Fourier Preprocessing for Hand Print Character Recognition", *IEEE Transactions on Computers*, Feb. 1972.
- [6] H. Al-Muallim, S. Yamaguchi: "A method of recognition of Arabic cursive Handwriting", *IEEE Trans. on PAMI*, Vol. 9, No. 5, pp. 715-722, September 1987.
- [7] H. Al-Yousefi, S. Udpa: "Recognition of Arabic Characters", *IEEE transaction on PAMI*, Vol. 14, No. 8, August 1992, pp. 853-857.
- [8] J. Mantas: "An overview of Character Recognition Methodologies", *Pattern Recognition*, Vol. 19, No. 6, pp. 425, 1986.
- [9] J. Simon: "Off Line Cursive Word Recognition", *Proceeding of the IEEE*, Vol. 80, No. 7, pp. 1150, July 1992.
- [10] J. Tou, R.C. Gonzalez: "Recognition of Handwritten Characters by Topological Feature Extraction and Multilevel Categorization", *IEEE Transaction on Computers*, pp. 776, July 1972.
- [11] J. Wang, J. Jean: "Multiresolution Neural Networks for Omnifont Character Recognition", *IEEE Int. Conf. on Neural Networks*, pp. 1588, San Francisco, Cal. USA, April, 1993.
- [12] K. Badie, M. Shimura: "Machine Recognition of Arabic

very difficult to draw a line between what is acceptable and what is not. The FD's of a badly handwritten characters would vary widely. While we refrained from placing any restriction on our writers, we advised them to be as good as if they were writing an official letter. Style, mood, background, ... etc. are factors that influence the way a person writes. Even is human recognition of an average handwriting, an estimated rate of 4 % misclassification is unavoidable. Moreover, humans usually use context as the main discriminator. To put this into perspective, an Arab reading a Farsi handwritten text (same alphabet) or an English reading Italian would make much more misclassification. This means that having a 100 % recognition rate of handwritten characters without contextual information is near to impossible.

From the results it was observed that most of the characters were classified correctly. However, even with our combined feature extractor, there were still similar shapes! The misclassified shapes were

(ذ, ذ), (ل, ل), (ك, ك), (ف, ف), (ث, ث), (ع, ع)

It is possible to enhance the topological feature extractor to consider these relatively rare cases, but the price will be high in terms of time. Contextual information, that can be obtained before the segmentation stage, is more helpful. For instance, in the first three pairs the first character can not be connected from the left side, while the second must be. In the last pair, the first character must be at the beginning of a word (or subword) and the second at the end.

Considering previous works on Arabic handwritten OCR, the attained recognition rate is very satisfactory.

- Handprinted Scripts", *The Transaction of the IECE of Japan*, Vol. E 65, No. 2, pp. 107, Feb. 1982.
- [13] M. El-Wakil, A. Shoukry: "On-Line recognition of Handwritten Isolated Arabic Characters", *Pattern Recognition*, Vol. 22, No. 2, pp. 97, 1989.
- [14] M. Shridhar, A. Badreldin: "High Accuracy Character Recognition Algorithm Using Fourier And Topological Descriptors", *Pattern Recognition*, Vol. 17, No. 5 pp. 515, 1984.
- [15] O. Alhusain: "High Accuracy Recognition Algorithm of Arabic Characters", *Periodica Polytechnica Seer. El. Eng.*, Vol. 35, No. 1, pp. 3, 1991.
- [16] R. Bozinovic, S. Srihari: "Off Line Cursive Script Word Recognition", *IEEE Trans. on PAMI*, Vol. 11, No. 1, pp. 68, January 1989.
- [17] R. Lippmann: "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, pp. 4, April 1987.
- [18] S. Al-Emami, M. Usher: "On-Line Recognition of Handwritten Arabic Characters", *IEEE Trans. on PAMI*, Vol. 12, No. 1, pp. 704, July 1990.
- [19] S. Khobragade, A. Ray: "Connectionist Network for Feature Extraction and Classification of English Alphabetic Characters", *IEEE Int. Conf. on Neural Networks*, pp. 1606, San Francisco, Cal. USA, April 1993.
- [20] S. Knerr, L. Personnaz, G. Dreyfus: "Handwritten Digit Recognition by Neural Networks with Single-Layer Training", *IEEE Transaction on Neural Networks*, Vol. 3, No. 6, pp. 962, Nov. 1992.
- [21] S. Mori, C. Suen, K. Yamamoto: "Historical Review of OCR Research and Development", *Proc. of IEEE*, Vol. 80, No. 7, pp. 1029-1157, July 1992.
- [22] S. Mori, K. Yamamoto, M. Yasuda: "Research on Machine Recognition of Handprinted Characters", *IEEE Trans. on PAMI*, Vol. 6, No. 4, pp. 386, July 1984.
- [23] T. El-Sheik, R. Guindi: "Automatic Recognition of Isolated Arabic Characters", *Signal Processing*, Vol. 14, No. 2, pp. 177-184, March 1988.

3rd International Conference on INTELLIGENCE in NETWORKS

BORDEAUX — FRANCE — Palais des Congrès

October 11, 12, 13, 1994

The ICIN 94 conference will build on the success of the previous two conferences held in BORDEAUX, France. It will bring together world-wide specialists in communications, computing and services to consider current developments and future directions. Topics will include intelligent network applications, intelligent network architecture, mobile networks, current implementation and experience, service creation and management. The views of users, vendors and operators will be represented.

**Secrétariat du COLLOQUE INTERNATIONAL
SUR L'INTELLIGENCE DANS LES RESEAUX (ICIN)
ADERA
B.P. 196
33608 PESSAC CEDEX
FRANCE**

Information for authors

JOURNAL ON COMMUNICATIONS is published monthly, alternately in English and Hungarian. In each issue a significant topic is covered by selected comprehensive papers.

Other contributions may be included in the following sections:

- INDIVIDUAL PAPERS for contributions outside the focus of the issue,
- PRODUCTS-SERVICES for papers on manufactured devices, equipments and software products,
- BUSINESS-RESEARCH-EDUCATION for contributions dealing with economic relations, research and development trends and engineering education,
- NEWS-EVENTS for reports on events related to electronics and communications,
- VIEWS-OPINIONS for comments expressed by readers of the journal.

Manuscripts should be submitted in two copies to the Editor in chief (see inside front cover). Papers should have a length of up to 30 double-spaced typewritten pages (counting each figure as one page). Each paper must include a 100–200 word abstract at the head of the manuscript. Papers should be accompanied by brief biographies and clear, glossy photographs of the authors.

Contributions for the PRODUCTS-SERVICES and BUSINESS-RESEARCH-EDUCATION sections should be limited to 16 double-spaced typewritten pages.

Original illustrations should be submitted along the manuscript. All line drawings should be prepared on a white background in black ink. Lettering on drawings should be large enough to be readily legible when the drawing is reduced to one- or two-column width. On figures capital lettering should be used. Photographs should be used sparingly. All photographs must be glossy prints. Figure captions should be typed on a separate sheet.



Do you know?

- ... that you can use the telephone number of Country Direct in Hungary as well...?
- ... that after dialing your Country Direct number, the local operator answers and you can ask her in your native tongue to put through...?
- ... that after landing you can immediately call your home as you do not have to bother changing money...?
- ... that there is a reverse charge for calls through Country Direct...?

The  **HUNGARIAN TELECOMMUNICATIONS COMPANY LIMITED** welcomes you to Hungary with its new service so you can phone your home quickly and easily at any time:

Hello! Greetings from Hungary!

Please, remember your Country Direct number!

USA Direct (AT&T)	00 36 0111*	Ireland Direct	00 800 03531
	00 800 01111	Japan Direct (KDD)	00 800 08111
(in Hungarian)	00 800 01112		00 36 8111*
USA Direct (MCI)	00 800 01411	Canada Direct	00 800 01211
USA Direct (USSprint)	00 800 01877		00 36 1111*
Australia Direct	00 36 6111*	Korea Direct (KT)	00 800 08211
	00 800 06111	Korea Direct (DACOM)	00 800 08212
Austria Direct	00 800 04311	Germany Direct	00 800 04911
UK Direct (BT)	00 800 04411	Norway Direct	00 800 04711
UK Direct (Mercury)	00 800 04412	Italy Direct	00 800 03911
Denmark Direct	00 800 04511		00 36 3911
Finland Direct	00 800 03581	Portugal Direct	00 800 03511
	00 36 3581*	Sverige Direct	00 800 04611
France Direct	00 800 03311	Singapore Direct	00 800 06511
Netherlands Direct	00 800 03111	New Zealand Direct	00 800 06411
	00 36 3111*	Turkey Direct	00 800 03531
Hongkong Direct	00 800 08521	* The old telephone numbers can be called until 31 Dec.1994.	
Israel Direct	00 800 09721		

The use of Direct service in Hungary is free of charge.

A coin must be inserted into a public phone, but the telephone will eject the coin when the receiver is replaced.

Only the red and blue coin phones may be used for Direct service.

At present 5, 10 or 20 Forint coins can be used.

Phone card phones also accept Direct service. The phone card must be used, but this will not affect the number of units.

TEXTURE ANALYSIS

- Image processing system for detection of weaving defects T. Thomas and M. Cattoen 41
Analyzing texture anisotropy via gray-level difference features D. Chetverikov 43
Texture analysis based on wavelet decomposition Cs. Kiss, G. Németh, P. Csillag, and L. Böröczky 47

MULTI-DIMENSIONAL PROCESSING

- Mapping strategies for signal and image processing algorithm parallelization I. Erényi and I. Vassányi 50
Robust morphological and textural description of micro- and
macroproperties on machine finished surfaces I. Loványi and Á. Nagy 52
A data-flow approach to develop image processing applications B. Várkonyi 55
Anatomic segmentation, feature extraction and pseudocoloring
for biomedical image measurements A. Vlaicu, S. Lungu, E. Meciu, and S. Iacob 59
Surface interpolation technique and 3D segmentation
for reconstructing 3D objects from serial sections A. Vlaicu, S. Lungu, and C. Rusu 64
Color image archivization for medical purposes M. Domański and M. Bartkowiak 66

POSTERS I.

- Quantizing accuracy for high quality color image processing O. Gan, K. Kotani, and M. Miyahara 69
Accelerated image coding scheme based on iterated
function systems K. Tanaka, M. Iwakiri, and K. Matsui 71
One application of homomorphic filtering H. Hudecová, J. Polec, R. Vargic, and J. Pavlovičová 73
New ordering of sequences and bases functions of discrete Fourier
and discrete hartley transforms for transform coders J. Polec, J. Pavlovičová, and R. Vargic 75
Kohonen and Grossberg learning in neural networks for image compression M. Oravec 77

POSTERS II.

- Two-dimensional least square SVD algorithm U. Burnik, J. Tasič, and G. Cain 80
"Adjustable polygons": a novel active contour model
for objects tracking on complex background P. Delagnes, J. Benois, and D. Barba 83
CAD system for pattern recognition and DSP with use of fast
translation invariant transforms J. Turán, L. Kövesi, and M. Kövesi 85
Arabic handwritten character recognition by neural nets A. M. Obaid 90
-

1994 DEC 01