Signal Processing 5 (1983) 229–240
North-Holland Publishing Company

RECURSION IN SHORT-TIME SIGNAL ANALYSIS

Michael UNSER
Signal Processing Laboratory, EPFL, 16 ch. de Bellerive, CH-1007 Lausanne, Switzerland

Received 14 July 1982
Revised 27 September 1982

Abstract. The problem of finding a recursive structure for the evaluation of features through a 'running' window is investigated. A general closed form expression is found for features satisfying a direct or indirect recursion condition. It is shown that most of the commonly used features (mean value, energy, autocorrelation function , DFT, Z-transform, entropy, etc.) satisfy these analytic expressions. The recursive, step by step, feature evaluation method is compared with the conventional method where features are evaluated for positions of the observation window with a 50% overlap. These two methods are equivalent in computation time for features satisfying the direct recursion condition. However, there might be some loss of information when using the last approach. The use of indirect recursion is advantageous for small window sizes. The results are then generalised to bidimensional signal processing.

Zusammenfassung. Es wird untersucht, wie man Parameter eines gleitenden Fensters rekursiv schätzen kann. Es wird ein allgemeiner Ausdruck für den Fall angegeben, das die Schätzfunktion einer direkten oder indirekten Rekursionsbedingung genügt. Diesen Rekursionsbedingungen genügen z.B. Mittelwert, Energie, die Autokorrelationsfunktion, die diskrete Fouriertransformation, die z-Transformation, die Entropie u.a.m. Die rekursiv geschätzten Parameter werden mit den üblichen Schätzwerten verglichen, wenn die Fenster je zu 50% überlappen. Genügt ein Parameter der direkten Rekursionsbedingung, sind beide Berechnungsmethoden bezüglich Rechenzeit äquivalent. Allerdings kann bei der letztgenannten Methode unter Umständen ein Teil der Information verloren gehen. Die indirekte Rekursion hat besonders bei kleinen Fenstern Vorteile. Die Ergebnisse werden auf den zweidimensionalen Fall verallgemeinert.

Résumé. Le problème de la définition d'une structure récursive pour l'évaluation de paramètres au travers d'une fenêtre glissante, est étudié. Une formulation analytique générale est proposée pour des mesures satisfaisant des conditions de recursion directe et indirecte. Il est à remarquer que la plupart des grandeurs couramment utilisées (moyenne, energie, fonction d'autocorrélation, TFD, transformée en Z, entropie, etc.) satisfont ces conditions. La méthode d'estimation récursive (pas à pas) est comparée avec l'approche conventionnelle où les grandeurs sont évaluées pour des positions de fenêtre avec un recouvrement de moitié. Ces deux approches sont équivalentes en temps calcul pour des mesures satisfaisant la condition de récursion directe. Cependant, il est montré qu'une perte d'information peut résulter de l'emploi de l'approche conventionnelle. L'utilisation de la récursion indirecte est particulièrement avantageuse pour des fenêtres de taille réduite. Les résultats sont ensuite généralisés pour le traitement de signaux à deux indices.

Keywords. Recursion, sliding window, feature evaluation, short-time signal analysis.

1. Introduction

In various signal processing or pattern recognition problems, the assumption that some properties of a signal change relatively slowly in time (or in space), can be made. For example in speech processing [16], it is well known that the short time spectral components have very little variation when compared with the original signals activity. Thus, they provide a very useful signal representation. In the image processing domain, a picture can be seen as an arrangement of different homogeneous textured fields. This observation makes us aware that there exist some local picture properties (textural features) that vary very slowly in the space domain, within a region of given texture. This assumption leads to a variety of 'short time' or 'short space' processing methods in which small segments from a signal are isolated and processed as if they were short signal parts with fixed properties. These short segments (sometimes called 'analysis frames') correspond to the observation of the signal through a fixed size window. For a given position of the observation window, a set of features (usually space- or time-invariant) can be extracted. Features are understood as distinguishing primitives or attributes of a signal field. Such an approach will produce a 'time-dependent' (or space dependent) sequence of a local property vector which can serve as a representation of the original signal. This type of representation can be very useful for signal segmentation, classification and understanding. For example, information patterns may be automatically extracted by grouping local property vectors into clusters.

This paper deals with the computational aspect of feature evaluation over a running window. Recursive algorithms, where the underlying idea is the 'updating' of some numeric quantities, have been proposed in various fields (e.g. time series analysis [3, 4], automatic and control [14], signal processing, pattern recognition and others) for the evaluation of some specific features or parameters. The recursive implementation of the moving average filter (evaluation of the local mean) is well known in signal processing [13]. The recursive structure of the running DFT has been investigated in [1, 2, 6, 7, 10]. The goal of this paper is to propose a generalisation and an extension of these results. A general form for features satisfying a direct or indirect recursion condition is proposed. The problem of sampling the features at a lower rate than the original signal rate is investigated. It is shown that the recursive step by step evaluation of feature satisfying the direct recursion condition is equivalent, in computation time, to the conventional method where features are computed at the minimum sampling rate (windows with a 50% overlap). Nevertheless, it is very important to be aware of the fact that some information can be lost when using this last approach. It is also shown that the indirect recursive computation of features derived from local probability functions is much faster than the conventional approach for small window sizes. This result could be of interest, for example, in some texture analysis problem when using features extracted from local co-occurrence matrices.

2. Local property vector

A segment of a signal is isolated and global features are estimated on it, as if this segment was a part of an ideal signal with fixed properties satisfying the Stationarity and Ergodicity conditions. For every possible segment of fixed size, a local property vector is defined.

Let x_k be a discrete signal. We observe x_k through a sliding window of length N and whose position is fixed in respect to the index k. For further developments we will choose the window starting at k (anti-causal convention) as shown in Fig. 1. It is very easy to adapt the subsequent

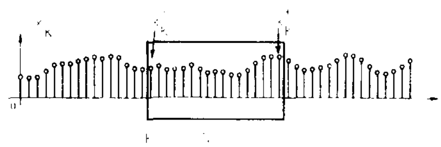

Fig. 1. Analysis of a signal through a running window of size N.

results for other positions relative to k (window centered on k, window finishing at k, etc.). For every position of the window, a set of q features is computed. A feature is a function of all possible samples which are available in the window:

$$f_i(k) = F_i(x_k, \ldots, x_{k+N-1}), \quad i = 1, \ldots, q. \quad (1)$$

The local property vector is given by

$$P(k) = \begin{pmatrix} f_1(k) \\ \vdots \\ f_q(k) \end{pmatrix}. \quad (2)$$

Let us define the two following values which correspond respectively to the first and last samples of the signal segment seen through the window at step k.

$$x_k^+ = x_{k+N-1},$$
$$x_k^- = x_k. \quad (3)$$

When moving the running window of one step, one removes the value x_{k-1}^- and adds x_k^+ to the observed samples. It is therefore reasonable to think that every component of the local property vector could satisfy the recursive equation:

$$f_i(k) = U(f_i(k-1), x_{k-1}^-, x_k^+) \quad (4)$$

where $U(\cdot)$ is an updating function depending only on the previous value of the feature $f_i(k-1)$ and on the values x_{k-1}^- and x_k^+. In the more general case, $U(\cdot)$ can be a function of x_{k-1}^- and x_k^+ and some of their relative neighbours or depend on some auxiliary recursive variables. This last case is referred to as indirect recursion. In the next sections, specific forms of (1) will be derived starting from recursion condition (4) with some particular updating functions.

3. Direct recursion

3.1. First order direct recursion

Theorem 1. *A feature $f(k)$, being a function of the samples $x_k, \ldots, x_{k+N-1}$, satisfies the first order recursion condition*

$$f(k) = w f(k-1) + w_+ F(x_k^+) + w_- F(x_{k-1}^-) \quad (5)$$

where w, w_-, w_+ are complex values and $F(\cdot)$ an arbitrary function, if and only if

$$f(k) = c \sum_{l=0}^{N-1} w^{-l} F(x_{k+l}) \quad \text{and} \quad \begin{cases} w_+ = c\, w^{-N+1} \\ w_- = -cw \end{cases} \quad (6)$$

where c is a constant.

Proof. Starting with $f(k)$, we apply N times the recursion equation (5), giving

$$f(k) = w^N f(k-N) + \sum_{i=0}^{N-1} w^i w_+ F(x_{k+N-i})$$
$$+ \sum_{i=0}^{N-1} w^i w_- F(x_{k-1-i}).$$

This result can be written in the following form by an appropriate change of variable

$$f(k) - w^N f(k-N) = w_+ w^{N-1} \sum_{l=0}^{N-1} w^{-l} F(x_{k+l})$$
$$+ w_- w^{N-1} \sum_{l=0}^{N-1} w^{-l} F(x_{k+l-N}).$$

$f(k)$ is a function of $x_k, \ldots, x_{k+N-1}$ and $f(k-N)$ a function of $x_{k-N}, \ldots, x_{k-1}$. Therefore, we can map

$$\begin{cases} f(k) = w_+ w^{N-1} \sum_{l=0}^{N-1} w^{-l} F(x_{k+l}), \\ f(k-N) = -w^{-N} w_- w^{N-1} \sum_{l=0}^{N-1} w^{-l} F(x_{k+l-N}). \end{cases}$$

Comparing those two expressions, finally we obtain

$$w_+ w^{N-1} = -w_- w^{-1} = c = cst.$$

The derivation into the other direction has been omitted.

The general form of eq. (6) allows us to construct a large number of features satisfying the direct recursion condition. Table 1 gives a list of some of the most commonly used features having this property. Most of recursive equations shown in Table 1 have been used in some particular applications. For example, the well known moving average filter (local mean) can be implemented using the recursive equation given for the mean. The recursive structure of the Discrete Fourier Transformation (DFT) has been investigated by many authors [1, 2, 6, 7, 10]. The recursive evaluation of features is well adapted for real time applications. When all the values are required it is much more economical than the conventional approach.

3.2. Second order direct recursion

The generalisation of the above results for second and higher order features is straightforward.

Table 1

Examples of features satisfying the first order direct recursion condition

Features	General equation	Recursive equation
general	$f(k)=\sum_{l=0}^{N-1} w^{-l}F(x_{k+l})$	$f(k)=w[f(k-1)-F(x_{k-1})]+w^{-N+1}F(x_{k+N-1})$
mean	$m(k)=\frac{1}{N}\sum_{l=0}^{N-1} x_{k+l}$	$m(k)=m(k-1)+\frac{1}{N}(x_{k+N-1}-x_{k-1})$
qth order moment	$m_q(k)=\frac{1}{N}\sum_{l=0}^{N-1} x_{k+l}^q$	$m_q(k)=m_q(k-1)+\frac{1}{N}(x_{k+N-1}^q-x_{k-1}^q)$
DFT coefficient n	$X(k;n)=\sum_{l=0}^{N-1} x_{k+l}\,e^{-i2\pi nl/N}$	$X(k;n)=e^{i2\pi n/N}[X(k-1;n)-x_{k-1}]+e^{-i(N-1)2\pi n/N}x_{k+N-1}$
z transform (value $z=z_0$)	$X(k;z_0)=\sum_{l=0}^{N-1} x_{k+l}z_0^{-l}$	$X(k;z_0)=z_0[X(k-1;z_0)-x_{k-1}]+z_0^{-(N-1)}x_{k+N-1}$

Let us consider the following notation:

$$\begin{cases} x_{1k}=x_k,\\ x_{2k}=x_{k+d}. \end{cases} \tag{7}$$

Theorem 2. *A feature $f(k)$, being a function of the samples $x_k, \ldots, x_{k+N-1}$, satisfies the second order recursion condition*

$$f(k)=w\,f(k-1)+w_+F(x_{1k}^+, x_{2k}^+) + w_-F(x_{1k-1}^-, x_{2k-1}^-) \tag{8}$$

where w, w_+, w_- are complex values and $F(\cdot)$ an arbitrary function, if and only if

$$f(k)=c\sum_{l=0}^{N-1} w^{-l}F(x_{1k+l}, x_{2k+l})$$

$$\text{and}\quad \begin{cases} w_+=c\,w^{-N+1},\\ w_-=-c\,w. \end{cases} \tag{9}$$

The proof of this theorem is nearly the same than for Theorem 1. Examples of commonly used features satisfying this condition are shown in Table 2. The features marked by a '[a]' are mainly used in bidimensional texture analysis [9]. They are usually computed from a co-occurrence matrice [8].

Table 2

Examples of features satisfying the 2nd order direct recursion condition

Feature	General equation	Recursive equation
general	$f(k)=\sum_{l=0}^{N-1}\omega^{-l}F(x_{k+l}, x_{k+l+d})$	$f(k)=\omega[f(k-1)-F(x_{k-1}, x_{k-1+d})]+\omega^{-N+1}F(x_{k+N-1}, x_{k+N-1+d})$
correlation	$\hat{R}_x(k;d)=\frac{1}{N}\sum_{l=0}^{N-1} x_{k+l}x_{k+l+d}$	$\hat{R}_x(k;d)=\hat{R}_x(k-1;d)+\frac{1}{N}[x_{k+N-1}x_{k+N-1+d}-x_{k-1}x_{k-1+d}]$
contrast[a]	$\bar{d}(k;d)=\frac{1}{N}\sum_{l=0}^{N-1}\lvert x_{k+l}-x_{k+l+d}\rvert$	$\bar{d}(k;d)=\bar{d}(k-1;d)+\frac{1}{N}[\lvert x_{k+N-1}-x_{k+N-1+d}\rvert-\lvert x_{k-1}-x_{k-1+d}\rvert]$
diff. average power[a]	$\bar{d}^2(k;d)=\frac{1}{N}\sum_{l=0}^{N-1}(x_{k+l}-x_{k+l+d})^2$	$\bar{d}^2(k;d)=\bar{d}^2(k-1;d)+\frac{1}{N}\{(x_{k+N-1}-x_{k+N-1+d})^2-(x_{k-1}-x_{k-1+d})^2\}$
local homogeneity[a]	$h(k;d)=\frac{1}{N}\sum_{l=0}^{N-1}\frac{1}{1+(x_{k+l}-x_{k+l+d})^2}$	$h(k;d)=h(k-1,d)+\frac{1}{N}\left\{\frac{1}{1+(x_{k+N-1}-x_{k+N-1+d})^2}-\frac{1}{1+(x_{k-1}-x_{k-1+d})^2}\right\}$

[a] Feature mainly used in bidimensional texture analysis.

3.3. Spectral considerations

Let us consider the problem of sampling the features satisfying the direct recursion condition at a lower rate than the original signal. This is equivalent to calculating features at some particular positions of the observation window.

Let us define y as being the result of a non-linear transformation $F(\cdot)$ applied to the signal x:

$$y_k = F(x_k). \tag{10}$$

The first order direct recursion condition can be rewritten as the following difference equation when setting the constant equal to one:

$$f(k) = w f(k-1) + w^{-N+1} y_{k+N-1} - w y_{k-1}. \tag{11}$$

Thus, y and f are both time- (or space-) dependent sequences which are respectively the input and the output of a linear system. The Z-transform of the impulse response of this system is given as

$$H(z) = F(z)/Y(z) = \frac{w^{-N} z^{N} - 1}{w^{-1} z^{-1} - 1}. \tag{12}$$

Using properties of the Z-transform, a power of the complex variable z can be interpreted in terms of elementary delays. Therefore, this equation suggests the implementation of the feature evaluating filter reported in Fig. 2. This structure

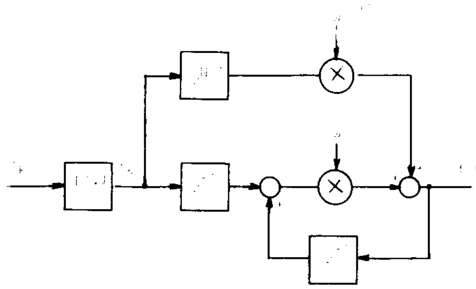

Fig. 2. Implementation of the recursive feature evaluation filter.

is well suited for real time computation. The frequency response is obtained by setting $z = e^{j2\pi f}$

$$H(f) = \frac{w^{-N} e^{j2\pi f N} - 1}{w^{-1} e^{j2\pi f} - 1}. \tag{13}$$

This expression will be discussed for particular values of w:

Case 1: $w = 1$. In this case (13) can be rewritten as

$$H(f) = e^{i\pi(N-1)f} \frac{\sin(N\pi f)}{\sin(\pi f)}. \tag{14}$$

The absolute value of the frequency response of this filter is shown in Fig. 3. This filter is a

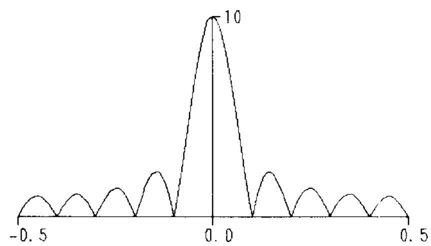

Fig. 3. Frequency response of the direct feature evaluation filter for $N = 10$.

lowpass filter. Though, it will be possible to sample $f(k)$ at a lower rate applying Shannon's Sampling Theorem [12]. If one assumes that most part of the energy of $f(k)$ is contained in the first lobe of the sinc function, it is possible to sample $f(k)$ at a rate of $\frac{1}{2}N$. This is equivalent to say that one will compute $f(k)$ only for window positions with a 50% overlap. The above assumption is not always true and one has to be very careful in order to avoid aliasing. The loss of information can be quite important. Let us consider the problem of the estimation of the local mean for a white noise. The aliasing average power is respectively 10% and 23% for the $\frac{1}{2}N$ and N sampling rates. Thus, the error will be very important when reconstructing the missing samples.

Case 2: $w = e^{j2\pi n/N}$. For this special choice for w, the output of the linear filter will be a coefficient of the running DFT. The Z-transform and frequency response of this filter are respectively

$$H(z) = \frac{z^{N} - 1}{e^{-j2\pi n/N} z - 1}, \tag{15}$$

$$H(f) = e^{i\pi(N-1)(f-n/N)} \frac{\sin \pi N(f - n/N)}{\sin \pi(f - n/N)}. \tag{16}$$

Such a filter can be decomposed as a comb filter, whose transfer function is $(z^N - 1)$, followed by a complex oscillator. The module of frequency response is given in Fig. 4. The complex oscillators

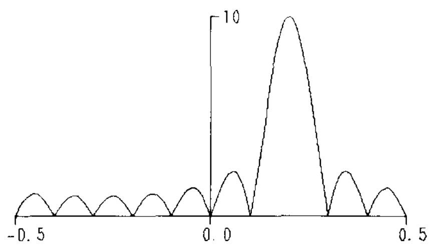

Fig. 4. Frequency response of the running DFT filter ($n = 2$) for $N = 10$.

pole is cancelled by a comb filters zero. The resulting filter is a bandpass filter. Such a structure is commonly referred to as 'frequency sampling filter' (FSF). The parallel connection of M FSF, whose associated oscillators frequencies are evenly distributed all around the unit circle, allows us, by appropriated weighting, to construct an arbitrary M points FIR filter [5]. Thus, a FIR filter can be realised by the weighted summation of the outputs of a bank of bandpass recursive filters (running DFT coefficients). This is illustrated by Fig. 5. As for the previous case, it is possible to sample $f(k)$ at the rate of $\frac{1}{2}N$. Nevertheless, the reconstruction will be more difficult (interpolation + modulation).

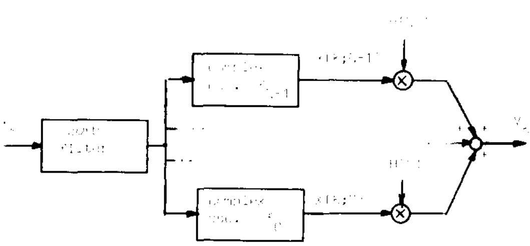

Fig. 5. Recursive implementation of an arbitrary N points FIR filter using running DFT filters.

A feature satisfying the direct recursion condition, can be seen as the output of a system composed of a non-linear transformation $F(\cdot)$ followed by a filter (usually lowpass averaging filter), (see Fig. 6). It is possible to approximate the averaging filter by a first order lowpass filter with a slight ability to 'forget its past'. The output of such a system is

$$f(k) = w f(k-1) + w_+ y_{k+N-1} \quad \text{with } |w| < 1. \tag{17}$$

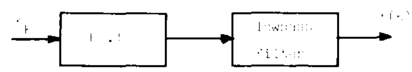

Fig. 6. Block diagram of a general feature evaluation filter.

3.4. Comparison between the conventional and recursive algorithms

It is interesting to compare the conventional and recursive approaches on the basis of their computation time. Let us define the following average times:

t_f: time for the evaluation of the function $F(\cdot)$,
t_a: time for an addition,
t_m: time for a multiplication.

The calculation of one feature for a fixed position of a window of length N, requires an average time of

$$T_w = N(t_f + t_a + \alpha t_m)$$

$$\text{with } \alpha = \begin{cases} 1 & \text{for } w \neq 1, \\ 0 & \text{for } w = 1, \end{cases} \tag{18}$$

using the conventional approach (eq. (6)), and

$$T_w = 2t_f + 2t_a + \alpha 2t_m \tag{19}$$

using the recursive evaluation (eq. (5)). These results for different window sizes are reported in Fig. 7. In the recursive approach, the computation time is not dependent on the size of the window. When the evaluation of a feature is needed at every step, the recursive approach is much quicker than the conventional one.

Nevertheless, in the above section, it has been shown that most of the information is preserved when evaluating feature for position of the observation window with 50% overlap. Therefore, we shall compare the computation time of the conventional method with a 50% overlap and the

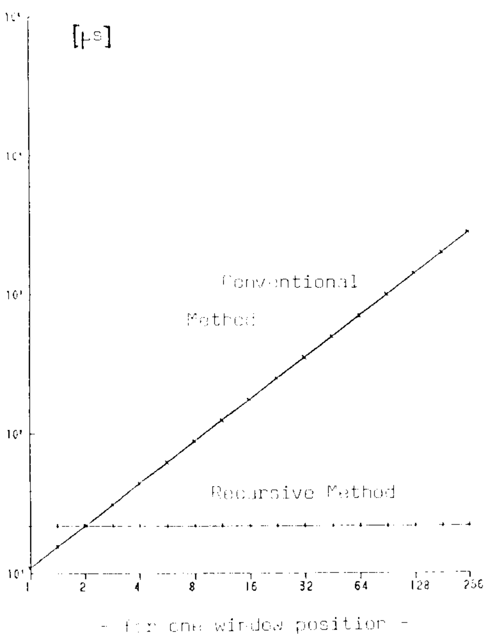

Fig. 7. Computation time as a function of the window size for both methods used for the evaluation of features satisfying the direct recursion condition ($t_a = 2$ µs, $t_m = 3$ µs, $t_f = 6$ µs).

recursive step by step evaluation for a signal with M samples. Thus, the analysis time of M samples, with a window of size N, is

$$T_t = 2M(t_f + t_a + \alpha t_m) \qquad (20)$$

when using the conventional algorithm, and

$$T_t = 2M(t_f + t_a + \alpha t_m) \qquad (21)$$

for the recursive approach.

It is interesting to note that the two approaches are exactly equivalent if one considers the computational aspect. But it is important to keep in mind that for the conventional method, with one half overlapping, there is some loss of information, while in the recursive approach all the information is preserved.

4. Indirect recursion

When a feature can be evaluated from an auxiliary recursive variable, it is said to have an indirect recursion property. There is a great variety of indirect recursive variables. For example the variance can be computed from the average power and from the mean, which both have the direct recursion property. In this section, we will restrict ourselves to choosing a particular auxiliar variable: the estimate of a probability density function for the observed signal segment.

4.1. Updating the probability density function

An estimate of the PDF for a signal segment is the normalised count of the signal's levels. Let us define $p_k(i)$, being the ith entry of the estimated PDF at recursion step k where i represents a particular discrete level of the signal. This quantity is given by

$$p_k(i) = \#\{l = k, \ldots, k + N - 1 \mid x_l = i;\ i \in G\}/N \qquad (22)$$

where N is the size of the window, G is the set of all the possible discrete levels of the signal and $\#$ denotes the number of elements of a specified set.

As pointed out previously, the only changing samples when moving the window of one step, are x_{k-1}^- and x_k^+. Therefore, the PDF satisfies the recursion relation

$$\begin{cases} p_k(x_k^+) = p_{k-1}(x_k^+) + 1/N, \\ p_k(x_{k-1}^-) = p_{k-1}(x_{k-1}^-) - 1/N, \\ p_k(i) = p_{k-1}(i) \quad \text{for } i \neq x_k^+,\ x_{k-1}^-. \end{cases} \qquad (23)$$

This property has been used in [11] for designing a fast median filtering algorithm. Updating the PDF at every step requires only two additions.

4.2. Indirect recursion via PDF

In pattern recognition, many features (see Table 3) are computed from PDFs. It will be shown that most of them can be estimated recursively.

Theorem 3. *A feature $f(k)$, being a function of the* PDF *estimates at step k $p_k(1), \ldots, p_k(i), \ldots,$*

$p_k(i_{max})$, satisfies the indirect recursion condition

$$f(k)=f(k-1)+F(p_k(x_k^+);x_k^+)$$
$$+F(p_k(x_{k-1}^-);x_{k-1}^-)$$
$$-F(p_{k-1}(x_k^+);x_k^+)$$
$$-F(p_{k-1}(x_{k-1}^-);x_{k-1}^-) \quad (24)$$

where $F(\cdot)$ is an arbitrary function, if and only if

$$f(k)=\sum_{i\in G} F(p_k(i);i). \quad (25)$$

Proof. As shown by eq. (23), p_k and p_{k-1} are unchanged for all entries different from x_k^+ and x_{k-1}^-. So (24) is equivalent to

$$f(k)=f(k-1)$$
$$+\sum_{i\in G}(F(p_k(i);i)-F(p_{k-1}(i);i)), \quad (24a)$$

finally after regrouping the terms

$$f(k)=\sum_{i\in G} F(p_k(i);i).$$

It is easy, using eq. (25), to make the derivation into the other direction. This result can be generalised to higher order PDFs. Eq. (25) is a very general form and allows to construct a large set of features. Examples of such first and second order features are shown in Table 3 with their associated indirect recursive equations. It is interesting to see that features like the entropy can be computed recursively. In some problems it can be interesting to compare a local PDF with a reference PDF. This is usually done with the help of a distance measure [15]. It is worthwhile to note that the Euclidian, Battacharrya, Divergence, Kullback information, etc. distance measures to a reference PDF satisfy the indirect recursion condition.

4.3. Comparison between the conventional and recursive algorithms

In order to compare these two approaches, let us define

Table 3

Examples of features satisfying the first order indirect recursion condition (via PDF)

Type	Feature	Closed form
	General	$\sum_i F\{p_k(i);i\}$
local histogram characteristics	mean	$\sum_i p_k(i)$
	qth order moment	$\sum_i i^q p_k(i)$
	entropy	$-\sum_i i\log[p_k(i)]$
	inertia	$\sum_i p_k^2(i)$
Similarity measurements with reference PDF $q(x)$	Euclidean distance	$\sum_i \{p_k(i)-q(i)\}^2$
	Battacharrya distance	$\sum_i \sqrt{p_k(i)\cdot q(i)}$
	Divergence	$\sum_i \{p_k(i)-q(i)\} \times\log\left\{\frac{p_k(i)}{q(i)}\right\}$

g: number of discrete levels of the signal,
q: order of the estimated PDF,
N: size of the observation window.

When using the conventional approach (eq. (25)) and using the previous definition for t_a, t_f and t_m, the computation of one feature for a fixed position of the window requires an average time of

$$T_w=N t_a+g^q(t_a+t_f) \quad (26)$$

where $N t_a$ is the time necessary to compute the histogram. For the recursive approach, one has:

$$T_w=6t_a+4t_f. \quad (27)$$

These values are reported in Fig. 8(a) as functions of the window size. As in the direct recursion case, the advantage is to the recursive approach where the computation time is not dependent on the window size.

When not all the values are required, it is more realistic to compare the SCM (sampled conventional method) with a 50% overlap and the RM

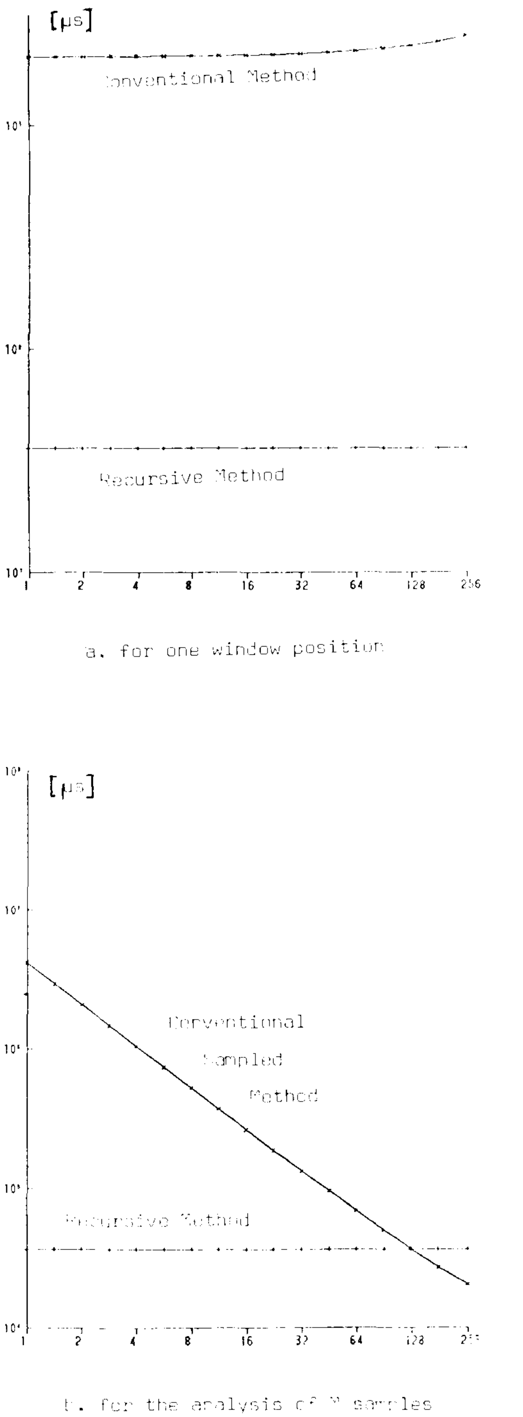

Fig. 8. Computation time as a function of the window size for both methods used for the evaluation of features satisfying the indirect recursion condition ($t_a = 2$ μs, $t_m = 3$ μs, $t_f = 6$ μs, $g = 256$, $q = 1$, $M = 1024$).

(recursive method). The analysis time of M samples, with a window of size N, is

$$T_t = 2Mt_a + \frac{2M}{N} g^q (t_a + t_f) \tag{28}$$

when using the SCM, and

$$T_t = 2Mt_a + 4M(t_a + t_f) \tag{29}$$

for the recursive step by step evaluation. In this last expression, the time necessary for initialisation has been neglected. As it turns out when looking at Fig. 8(b), the two methods are not equivalent. For small windows ($N < 2g^q$) the RM is more economical and it has the advantage to preserve all the information. Nevertheless, the time required for the evaluation or the updating of the histogram is the same for the two methods.

5. Bidimensional recursion

The generalisation of the previous results to bidimensional signal processing appears as straightforward.

5.1. Bidimensional generalisation

Let us consider a square window of size $N \times N$. When moving the running window of one step one removes N values and adds N other values to the observed sample set, when moving the window of one horizontal step. The following equations can be found for features satisfying the direct recursion condition.

(1) Horizontal:

$$f(k,l) = w_1 f(k-1,l) + w_{1+} \sum_{u=0}^{N-1} F(x_{k+N-1,l+u}) + w_{1-} \sum_{u=0}^{N-1} F(x_{k-1,l+u}). \tag{30}$$

(2) Vertical:

$$f(k,l)=w_2 f(k,l-1)+w_{2+}\sum_{u=0}^{N-1} F(x_{k+u,l+N-1})$$

$$+w_{2-}\sum_{u=0}^{N-1} F(x_{k+u,l-1}), \qquad (31)$$

$$f(k,l)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} w_1^{-u} w_2^{-v} F(x_{k+u,l+v}). \qquad (32)$$

The general closed form for features satisfying the indirect recursion condition is

$$f(k,l)=\sum_{i\in G} F(p_{k,l}(i);i) \qquad (33)$$

It is straightforward to rewrite the bidimensional equivalents of Eqs. (23) and (24) for a move of one vertical or horizontal step. The consequence of such a generalisation is that the computation time by the recursive method becomes now proportional to N and is not any more independent of the window size. It is very important to choose a scanning of the picture such as the features can be evaluated continuously (without any jump). Yet, it will be shown that for features satisfying the direct recursion condition, it is possible to reduce the computation time and to make it independent of the window size.

5.2. Separability property

Let $f(k,l)$ be a feature satisfying the two-dimensional direct recursion condition. The general expression for $f(k,l)$ is given by eq. (32) which can be rewritten as

$$f(k,l)=\sum_{v=0}^{N-1} w_2^{-v} g(k,l+v) \qquad (34)$$

with $g(k,l)$ being an auxiliary measurement

$$g(k,l)=\sum_{u=0}^{N-1} w_1^{-u} F(x_{k+u,l}). \qquad (35)$$

$g(k,l)$ can be evaluated by a line by line scanning. When considering one line, $g(k,l)$ has one-dimensional nature and furthermore, it has the direct recursion property. The same is true for $f(k,l)$ when computing it from the array $g(k,l)$ with a row by row scanning. $f(k,l)$ is said to have the separability property because it can be computed from two successive transformations along the lines and rows.

5.3. Comparison between the conventional and recursive approaches

A feature satisfying the direct recursion property is separable. Thus, computation time is independent of the window's size, when using the RM. All previous results are exactly transposable to this case. It is worthwhile to point out that some features used in texture analysis (correlation, contrast, homogeneity) and which are usually computed from the co-occurrence matrices [8, 9] can be evaluated with this approach. The advantage is double, first it does not require the explicit evaluation of the matrices (no memory storage) and second the computation is much quicker.

For features satisfying the indirect recursion condition (via a PDF), the situation is slightly different than the previous case. The computation time for the RM, has become proportional to N (side of the window). For a fixed position of the observation window (size: $N\times N$), the following computation times are required

$$T_w = N^2 t_a + g^q(t_a+t_f) \qquad (36)$$

for the conventional evaluation, and

$$T_w = 6Nt_a + 4Nt_f \qquad (37)$$

for the recursive evaluation (see Fig. 9(a)).

The comparison of the SCM with a 50% overlap and the RM when applied to the analysis of a $M\times M$ picture through an $N\times N$ window, gives the following times

$$T_t = 4M^2 t_a + \frac{4M^2}{N^2} g^q(t_a+t_f) \qquad (38)$$

when using the SCM, and

$$T_t = M^2(6Nt_a + 4Nt_f) \qquad (39)$$

for the recursive step by step evaluation.

These results are shown in Fig. 9(b) for the evaluation of a second order feature (for example: the entropy or the energy of a co-occurrence

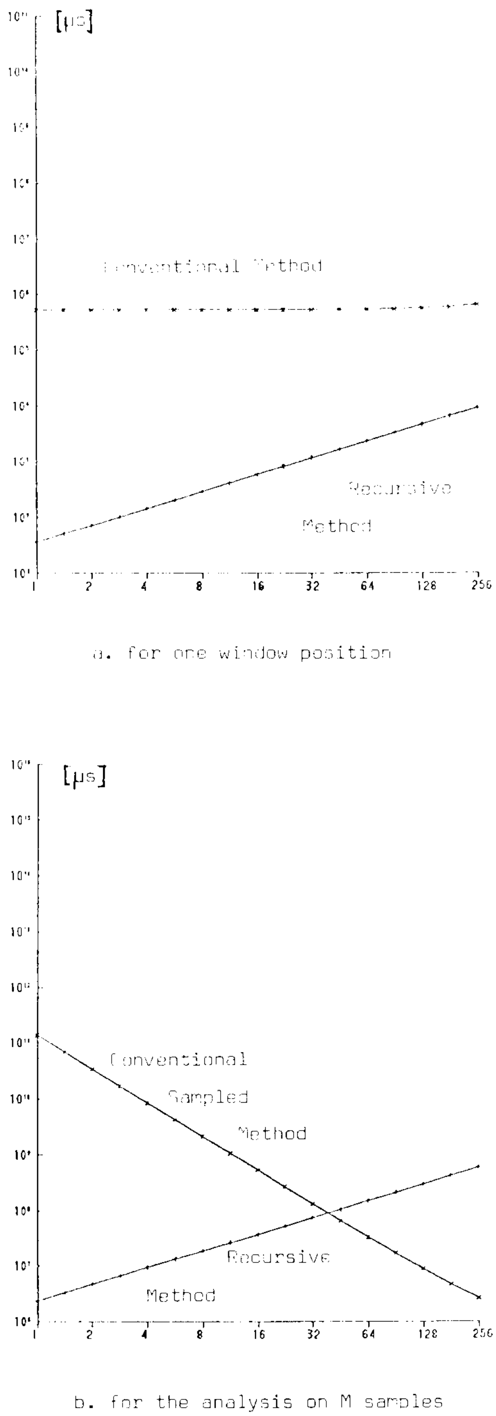

Fig. 9. Computation time as a function of the window size for both methods used for the evaluation of bidimensional features satisfying the indirect recursion condition ($t_a = 2$ μs, $t_m = 3$ μs, $t_f = 6$ μs, $g = 256$, $q = 2$, $M = 256^2$).

matrice). Once again the RM is more economic for small window sizes ($N < g^{q/3}$).

6. Conclusion

This study has shown that recursion can be successfully applied to the evaluation of features through a running window. The recursive approach seems to be computationally very interesting in short-time or short-space signal analysis. It is well adapted for hardware implementation in real time applications. The recursive step by step evaluation method has been compared with the conventional approach where features are evaluated for window positions with an overlapping of one half. In this last approach features are evaluated at a minimum rate (Shannon's Sampling Theorem) and there might be some loss of information due to aliasing. Furthermore, it turns out that the indirect recursive approach is computationally more economic for small window sizes. The two methods are equivalent for features satisfying the direct recursion condition. An other advantage for a system using a recursive implementation is that the output rate (local feature) is equal to the input rate. The size of the observation window can be easily changed without affecting the computation time. A feature can be seen as the result of a local transformation applied to the original signal. It is hoped that these results will find some applications in discrete signal analysis. For example, in texture analysis it might be interesting to perform a tone-to-texture transform. This means transforming the original grey level picture in a set of feature-pictures with interpretable texture information. These features-pictures can then be successfully used for segmentation or classification.

References

[1] R.R. Bitmead, "On recursive discrete Fourier transformation", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-30, No. 2, April 1982, pp. 319–322.

[2] G. Bongiovanni, P. Corsini and G. Frosini, "Procedures for computing the discrete Fourier transform on staggered blocks", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-24, No. 2, April 1976, pp. 132–137.

[3] B.W. Dickinson, "Two recursive estimates of autoregressive models based on maximum likelihood", *J. Stat. Comp. Simul.*, Vol. 7, 1978, pp. 85–92.

[4] B. Friedlander, "A recursive maximum likelihood algorithm for ARMA spectral estimation", *IEEE Trans. Information Theory*, Vol. IT-28, No. 4, July 1982, pp. 639–646.

[5] B. Gold and C.M. Rader, *Digital Processing of Signals*, McGraw-Hill, New York, 1969.

[6] B.G. Goldberg, "A continuous recursive DFT analyser – the discrete coherent memory filter", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-28, No. 6, December 1980, pp. 760–762.

[7] J.H. Halberstein, "Recursive complex Fourier analysis for real time applications", *Proc. IEEE* (*Lett.*), Vol. 54, June 1966, p. 903.

[8] R. Haralick, K. Shanmugan and I. Dinstein, "Texture features for image classification", *IEEE Trans. Systems Man Cybernet.*, Vol. SMC-3, Nov. 1973, pp. 610–621.

[9] R. Haralick, "Staistical and structural approaches to textures", *Proc. IEEE*, Vol. 67, No. 5, May 1979, pp. 786–804.

[10] G.H. Hostetter, "Recursive discrete Fourier transformation", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-28, No. 2, April 1980, pp. 184–190.

[11] T.S. Huang, G.J. Yang and G.Y. Tang, "A fast two-dimensional median filtering algorithm", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-27, No. 1, Feb. 1979, pp. 13–18.

[12] Abdul J. Jerri, "The Shannon sampling theorem – its various extensions and applications: a tutorial review", *Proc. IEEE*, Vol. 65, No. 11, Nov. 1977, pp. 1565–1596.

[13] M. Kunt, H. Rey and A. Ligtenberg, "Preprocessing of Electrocardiograms by Digital Techniques", *Signal Processing*, Vol. 4, 1982, pp. 215–222.

[14] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, Cambridge MIT, 1982.

[15] G. Lowitz, "Extraction de caricatures et texture depuis un histogram local", *8ème Colloque dur le Traitement due Signal et ses Applications*, Nice, 1–5 June 1981, pp. 793–800.

[16] L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ, 1978.