Fast FIR filtering algorithms for image processing

Ryszard STASIŃSKI[*]

The Norwegian Institute of Technology, Division of Telecommunications
O.S. Bragstads plass 2B, N-7034 Trondheim, Norway

RÉSUMÉ

Dans cette article on présente plusieurs algorithmes efficaces pour le filtrage de signaux à 2 dimensions. Les algorithmes sont dérivés de ceux utilisés pour le calcul de convolutions lineaires courtes, présentés en [1] et [2]. Les resultats de cette etude illustrent bien les limites de l'application de la technique du *nesting* dans le cas de filtres séparable. Les algoritmes sont relativement simple et ont efficaces realisations techniques. En outre, on explique comment optimiser le choix de l'algorithme en fonction des paramètres de l'ordinateur.

ABSTRACT

In the paper a set of efficient filtering two-dimensional algorithms for short filters is derived. The algorithms are based on linear convolution ones for short data vectors derived in [1], and [2]. In the case of separable filters limits of the *nesting* technique use are illustrated. The algorithms are relatively simple and have efficient program realizations. Moreover, it is explained in the paper how to make the optimal choice of the algorithm depending on the computational machine available.

1. INTRODUCTION

The construction of efficient filtering algorithms is one of the most important tasks in the domain of digital signal processing. The time efficiency of algorithms, being always important, becomes their critical feature when image processing, and in general, processing of multidimensional data is to be done. The fast FIR filtering is usually acomplished by using a fast convolution algorithm for a data block combined with the overlap-and-save, or overlap-and-add method [3], [4]. There is a great variety of convolution algorithms that can be applied here: FFT method, circular convolution, and linear convolution based ones. When a FIR filter to be realized is rather short, the last method is the best. This is the image processing case [5].

The derived in the paper two-dimensional filtering algorithms are overlap-and-add ones based on linear convolution ones from [1], and [2], section 2. The main feature of the latter algorithms consists in the fact that they fill the gap between very efficient methods for long data vectors, and algorithms for the shortest data vectors from [3], [4]. In this way a set of algorithms for short-length multidimensional data vectors is obtained, section 3, Table I. The nonseparable filters can be implemented using two-dimensional algorithms, only, but in the case of separable ones row-column technique can be applied, and it is shown in section 3 which algorithms are better in the latter case. Additionally, some comments on appropriate proportions between filter and data block sizes in fast filtering methods are given.

2. METHOD

Recently a simple method of generating efficient linear convolution algorithms for short data, based on the Chinese Remainder Theorem [3], [4], has been introduced for choice of polynomials modulo which computations are done [1]:

$$Z^R, Z^T-1, \text{ and } Z^S-\infty. \qquad (1)$$

Computations modulo Z^T-1 are in fact T-point circular convolutions, reduction modulo Z^R is equivalent to rejection of all but first R polynomial coefficients resulting in "truncated convolutions" [1], and $Z^S-\infty$

[*] The author is on leave from Department of Electronics and Communications, Technical University of Poznań, Poland.

reduction is a shorthand notation for the operation of rejecting all but S last polynomial coefficients [4], also resulting in truncated convolutions. So, reductions and reconstructions modulo polynomials (1) are in fact very simple, and for computations modulo $Z^T - 1$ we have at disposal a wide range of ready to use algorithms [3], [4]. Only non-trivial 3-point truncated convolution is computed using 5 multiplications and 5 additions [2].

In this paper the two-dimensional data are processed either by using algorithms obtained by nesting [6], [3] the one-dimensional ones, or by using the row-column technique, of course, the latter one for separable filters, only. Similarly as in the case of Winograd's Fourier Transform Algorithms, inside the nesting algorithm's structure $T \times T$-point circular convolution algorithms can be found [3], [1]. The best way of computing them is through the application of polynomial transforms. The technique has been used for generating efficient masking algorithms for image processing [7], too.

3. RESULTS

The FIR filters used in image processing are typically rather small, which is not only due to computational costs, but also due to filter approximation problems [5]. In the paper it is assumed that the filters are separable, or nonseparable, but not symmetric ones of size 2×2, 3×3, 4×4, and 5×5. The relevant results are given in Table I. The results have been obtained from those for linear convolution algorithms by adding to their addition counts operations linked with the overlap-and-add technique [3], [4]. Apart from multiplication (mults), and addition (adds) counts, operation numbers per input data samples are given:

$$\mu = mult(M)/M, \;\; \alpha = add(M)/M,$$

M is the size of input data block in the overlap-and-add method ('Input size').

Parameter ρ is defined as the ratio of the total multiplication time and the total addition time, characteristic of an computational equipment on which a given algorithm is implemented. From the knowledge of the ρ value for the equipment at disposal the Table I provides the information on the choice of the best algorithm. The best algorithm corresponds to the one in Table I whose ρ value is nearest to but smaller than that of the equipment. For example, if the ρ value for the equipment is 2 and the nonseparable filter size is 3×3, then the best algorithm is that for 6×6-point data blocks, see Table I ($\rho > 0.50$). The operations should be counted together with all accompanying non-arithmetical operations (e.g. data transfers, index modifications, etc.), hence, the evaluation of the "true" ρ value is never absolutely exact. It can be seen from Table I that even for the smallest L values there exist fast algorithms, i.e. algorithms more efficient than the direct formula.

When considering an algorithm for an L-tap FIR filter an optimum algorithm for size $N = K + M - 1$ exists. In the two-dimensional case, when a $L_0 \times L_1$-tap filter is non-separable, the L_i values are [8]:

$$L_i = \frac{N_i}{\ln N_T + const}, \;\; N_T = N_0 \cdot N_1, \;\; i = 0, 1, \qquad (2)$$

while for a separable filter '$\ln N_T$' should be replaced by '$\ln N_i$'. In the paper, without any loss of generality, it is assumed that $N_0 = N_1 = N$, $L_0 = L_1 = L$, and $M_0 = M_1 = M$. As can be seen from the formula (2), the optimal longitude of data segment in a fast filtering algorithm should be greater than that of the filter. Notice that in fact even for such small set of filter examples effects of (2) can be observed, at least for nonseparable filters, Table I.

The results for separable filters provide an excellent illustration of choice between the nesting, and row-column technique [9] ('r-c' in Table I means 'row-column'). Namely, we can see here a direct manifestation of the *rule of number 2* [10]. The rule states that the nesting is preferable if:

$$\rho(\mu_0 + \mu_1 - \mu_0\mu_1) > \alpha_1(\mu_0 - 1), \qquad (3)$$

(operation in dimension '1' is nested inside that for dimension '0'). The left-hand side of equation (3) describes the gain in the number of multiplications when replacing row-column method by the nesting one, while the right-hand side the loss in the number of additions. If $\mu_0 = \mu_1 = 2$ the gain is null, hence, the name of the rule. Notice that indeed, this special case, i.e. the case when multiplication count per data sample for a row-column algorithm is close to 4, discriminate between nesting and row-column algorithms for big ρ values. For small ρ the 'price' of extra additions per saved multiplication becomes important, however, for smallest the μ values in the case of 2×2-tap filters the nesting technique is always preferable.

Table I

Fast two-dimensional linear convolution algorithms

Filter	Input size	T[1]	mults	adds	μ	α	ρ range[2]
Non-separable filters							
2 × 2	direct form.	-	4	3	4	3	$\rho \geq 0$
	2 × 2	1	9	22	2.25	5.5	$\rho > 1.43$
	3 × 3	1	16	57	1.78	6.33	$\rho > 1.77$
3 × 3	direct form.	-	9	8	9	8	$\rho \geq 0$
	4 × 4	4	46	214	2.88	13.38	$\rho > 6.43$
	5 × 5	4	78	295	3.12	11.8	$\rho > 6.13$
	6 × 6	4	118	390	3.28	10.83	$\rho > 0.50$
4 × 4	direct form.	-	16	15	16	15	$\rho \geq 0$
	5 × 5	4	118	388	4.72	15.52	$\rho > 0.05$
	6 × 6	4	166	561	4.61	15.58	$\rho > 0.58$
	7 × 7	6	184	866	3.76	17.67	$\rho > 2.44$
		6[3]	154	1221	3.14	24.92	$\rho > 11.83$
5 × 5	direct form.	-	25	24	25	24	no range
	8 × 8	6	312	1355	4.88	21.17	$\rho \geq 0$
	10 × 10	6[3]	426	2380	4.26	23.8	$\rho > 4.27$
Separable filters							
2 × 2	direct r-c	-	2+2	1+1	4	2	$\rho \geq 0$
	2 r-c	1	3+3	4+4	3	4	no range
	2 × 2	1	9	21	2.25	5.25	$\rho > 1.86$
	3 r-c	1	4+4	8+8	2.67	5.33	no range
	3 × 3	1	16	56	1.78	6.22	$\rho > 2.06$
3 × 3	direct r-c	-	3+3	2+2	6	4	$\rho \geq 0$
	4 r-c	4	8+8	14+14	4	7	$\rho > 1.5$
	4 × 4	4	46	210	2.88	13.13	$\rho > 6.06$
	5 × 5	4	78	291	3.12	11.64	$\rho > 6.05$
	6 r-c	4	11+11	25+25	3.67	8.33	$\rho > 4$
4 × 4	direct r-c	-	4+4	3+3	8	6	$\rho \geq 0$
	4 r-c	1 × 1[4]	9+9	18+18	4.5	9	$\rho > 0.86$
	6 r-c	6	12+12	41+41	4	12.33	$\rho > 6.67$
	7 × 7	6[3]	154	1212	3.14	24.73	$\rho > 14.47$
5 × 5	direct r-c	-	5+5	4+4	10	8	$\rho \geq 0$
	5 r-c	4[5]	14+14	29+29	5.6	11.6	$\rho > 0.82$
		4	13+13	31+31	5.2	12.4	$\rho > 2$
	8 r-c	6	18+18	56+56	4.5	14	$\rho > 2.29$
	10 r-c	6[3]	21+21	85+85	4.2	17	$\rho > 10$
	10 × 10	6[3]	426	2364	4.26	23.64	no range

[1] i.e. the sizes of circular convolutions used (1).

[2] In fact, the best algorithms for $\rho > 1$, only. Namely, for $\rho \leq 1$ some "obvious" algorithm's derivation rules are irrelevant [2], hence, the ρ values are provided for completness of the table, only.

[3] Apart from the $T = 6$-point circular convolution algorithm the polynomial product modulo $Z^2 + 1$ algorithm is used, [2].

[4] Optimized algorithm obtained by nesting of 2-point ones [3], [4].

[5] 3-point truncated convolution computed using definition formula.

4. CONCLUSION

Various small filters have been considered in this paper. For each of these filters a number of fast algorithms for filtering of two-dimensional data is presented. The choice of the optimum algorithm depends on the value of ρ — the ratio of multiplication time and addition time — of the computing machine. It is explained how to make the optimal choice once the value of ρ is known. It is also shown when the nesting technique is recommended when a separable filter is implemented. The possibility and indeed the construction of fast, efficient algorithms for very small filter sizes are clearly demonstrated.

References

[1] R. Stasiński, "Extending sizes of effective convolution algorithms", Electron. Lett., vol. 26, No. 19, pp. 1602–1604, 1990.

[2] R. Stasiński, "Metody algebry wielomianowej w cyfrowym przetwarzaniu sygnalow" (in Polish), ed. Tech. University of Poznań, 1991.

[3] H.J. Nussbaumer, "Fast Fourier transform and convolution algorithms", Springer-Verlag, 2nd edition 1982.

[4] R.E. Blahut, "Fast algorithms for digital signal processing", Addison-Wesley, Reading, Mass., 1985.

[5] A.K. Jain, "Fundamentals of Digital Image Processing", Prentice-Hall, 1989.

[6] R.C. Agarwal, C.S. Burrus, "Fast one-dimensional digital convolutions by multidimensional techniques", IEEE Trans. Acoust., Speech, Signal Proces., vol. ASSP-22, pp. 1-10, 1974.

[7] R. Stasiński, A.K. Nandi "Fast masking algorithms for image analysis", Electron. Lett., vol. 28, No. 18, pp. 1680-1681, 1992.

[8] X. Li, G. Qian, "Block size considerations for multidimensional convolution and correlation", IEEE Trans. Signal Proces., vol. 40, pp. 1271–1273, 1992.

[9] R. Stasiński, "Selective nesting of circular convolution algorithms", Proc. ICASSP'92, vol. V, pp. (V-) 1–4, 1992.

[10] R. Stasiński, "Easy generation of small-N discrete Fourier transform algorithms", IEE Proc., Pt. G, vol. 133, pp. 133–139, 1986.