A Fast Algorithm for Orthogonalizing Polynomials on an Arbitrarily Shaped Region

Dr. ir. W. Philips

Department of Electronics and Information Systems (ELIS)
Medical Image and Signal Processing group (MEDISIP)
Institute Biomedical Technology (IBITECH)

ELIS Technical Report WP 96-01
January 1996

UNIVERSITEIT GENT
Elektronica en Informatiesystemen

MEDISIP

IBITECH
Institute Biomedical Technology
GENT

A Fast Algorithm for Orthogonalizing Polynomials on an Arbitrarily Shaped Region

Wilfried Philips
Department of Electronics and Information Systems,
University of Gent,
St.-Pietersnieuwstraat 41,
B9000 Gent,
Belgium
Tel: (32) 9 264.33.85 Fax: (32) 9 264.35.94
E-mail: philips@elis.rug.ac.be

January 1996

Abstract

This report presents a faster version of the polynomial recursive orthogonalization (PRO) algorithm, which is used in segmented image coding to generate orthonormal base functions on a non-rectangular discrete image. It shows that up to 25% of the floating point operations can be eliminated by using previously computed results; also, some orthogonalizations can be avoided altogether, because they are guaranteed to produce zero functions. In practice, the PRO implementation in this report is 20 to 25% faster than the old algorithm and two to three times faster than Gram-Schmidt orthogonalization.

1 Introduction

Segmented image coding (SIC) is a second generation coding technique [1] that describes images in terms of the contours and the texture of non-rectangular regions corresponding to objects in the image. This report deals with fast texture coding.

In all previously proposed methods, the texture of each region is approximated by a bi-variate polynomial, i.e., by a linear combination of monomials x^iy^j. For many reasons it is preferable to express the approximating polynomial as a linear combination of base functions which are orthonormal on the region being coded [2]. Unfortunately, an orthonormal base necessarily depends on the shape of the region and must be constructed for every single region in the image. Therefore, fast procedures for generating such bases are vital.

Gilge [2] proposed to generate an orthonormal base by orthogonalizing the monomials x^iy^j in the *natural order* 1, x, y, x^2, xy, y^2, x^3, ... using the Gram-Schmidt procedure (GS) [3]. GS requires $\mathcal{O}(k^2/2)$ floating point operations per region pixel (flopp) to compute the k-th orthonormal base function. In [4], the author presented the polynomial recursive orthogonalization (PRO) algorithm, which is based on a recurrence [5]. PRO requires only $\mathcal{O}(k)$ flopp for computing the k-th base function.

Table 1: Details concerning the new PRO recurrence

	$s_{m,n}$	terms
$m \geq 2$	$(m-2,n)$	$2(m+n)-1$
$m=1$	$(n,0)$	$2(m+n)-1$
$m=0, n \geq 2$	$(0,n-2)$	$2(m+n)+1$
$m=0, n=1$	$(0,0)$	2

	$s'_{m,n}$	coeff.
$m \geq 1$	$(m-1,n)$	$m+n$
$m=0, n \geq 1$	$(0,n-1)$	$2(m+n)$

Section 3 of this report shows that the computational complexity of PRO can be reduced by up to 25%, by cleverly reusing the results of some calculations. Also, it shows that some orthogonalizations need not be performed, because their result is guaranteed to be a zero function. Finally, section 4 compares the complexities of the old PRO, the improved PRO and of GS, both theoretically and experimentally.

2 Definitions and notations

Let Ω be the set of pixel coordinates of the region for which a base must be generated. A *row* of Ω is a non-empty subset containing all pixels with the same y-coordinate. A *column* is defined similarly. With N and M we denote the number of rows and columns in Ω; Furthermore, let n_i and m_j be the number of pixels in the i-th column and in the j-th row respectively. We assume that the rows and columns are numbered such that $m_0 \geq m_1 \cdots \geq m_{N-1}$ and $n_0 \geq n_1 \cdots \geq n_{M-1}$.

The functions $f(x,y)$ and $g(x,y)$ are *orthogonal* on Ω if their *inner product* $f \cdot g \triangleq \sum_{(x,y)\in\Omega} f(x,y)g(x,y)$ vanishes. A function $f(x,y)$ is *normalized* if its squared *norm* $||f||^2 \triangleq f \cdot f$ equals unity. *Orthonormal* functions are normalized and mutually orthogonal.

The natural orthogonalization order proposed by Gilge [2] sorts the functions $x^m y^n$ by increasing total degree $m+n$, or by increasing y-degree n, if they have the same total degree. It corresponds to an ordering of the couples (m,n) of non-negative integers: $(m,n) \prec (m',n')$ iff. $m+n < m'+n'$ or $m+n = m'+n'$ and $n < n'$. With $(m,n)-1$ we denote the natural predecessor of (m,n), i.e., $(m+1,n-1)$ if $n \neq 0$ and $(0,m-1)$ if $n=0$. In the following, $p_{m,n}(x,y)$ denotes the polynomial that results from orthonormalizing $x^m y^n$.

3 The improved PRO algorithm

The PRO algorithm constructs the orthonormal polynomials $p_{m,n}(x,y)$ faster than GS and without computing the monomials $x^m y^n$. It is based on the following recurrence which expresses $p_{m,n}(x,y)$, or rather a multiple $p'_{m,n}(x,y)$ of it, in terms of its natural predecessors:

$$p'_{m,n}(x,y) = q_{m,n}(x,y) - \sum_{(i,j)=\mathrm{s}_{m,n}}^{(m,n)-1} \alpha_{i,j}^{m,n} p_{i,j}(x,y). \tag{1}$$

In this equation, $q_{m,n}(x,y)$ equals $xp_{m-1,n}(x,y)$ if $m \neq 0$ and $yp_{m,n-1}(x,y)$ otherwise. The value of $\mathrm{s}_{m,n}$ is defined in table 1, along with the number of terms in the sum. The coefficients in the recurrence are $\alpha_{i,j}^{m,n} = q_{m,n} \cdot p_{i,j}$. Finally, $p_{m,n} = p'_{m,n}/N_{m,n}$, where $N_{m,n} = \left\| p'_{m,n} \right\|$.

Note that $p_{m,n}(x,y)$ cannot be computed when $p'_{m,n} \equiv 0$ on Ω; this happens when $x^m y^n$ is linearly dependent on its predecessors (LDP) on Ω. In that case, $p_{m,n}(x,y)$ (as defined above) does not exist. Note that $p_{m,n}(x,y)$ appears in the right-hand side of the recurrences of some of its successors. However, those recurrences continue to hold when any term containing $p_{m,n}(x,y)$ is removed from their right-hand side [6]. Also, when $x^m y^n$ is LDP, any $x^i y^j$ with $i \geq m$ and $j \geq n$ is also LDP [6]. The corresponding $p_{i,j}(x,y)$ must also be removed from further recurrences.

To avoid unnecessary computations, it would be useful to know whether a certain $x^m y^n$ is LDP, *before computing* $p'_{m,n}$. The following theorem uses geometrical information about the region Ω to predict whether $x^m y^n$ is LDP:

Theorem: Let $S_1 = \{(i,j) : 0 \leq j < N, 0 \leq i < m_j\}$ Then any $x^m y^n$ with $(m,n) \succ (i,j), \forall (i,j) \in S_1$ is LDP; any $x^m y^n$ for which (m,n) and all its predecessors $(i,j) \prec (m,n)$ belong to S_1 is certainly not LDP. These conclusions are also true for the index set $S_2 = \{(i,j) : 0 \leq i < M, 0 \leq j < n_i\}$. Furthermore, if $S_1 = S_2$, the monomial $x^m y^n$ is LDP iff. $(m,n) \notin S_1 = S_2$.

This theorem is proved in [6]. One of its consequences is that $x^m y^n$ is LDP whenever $m \geq M$ or $n \geq N$.

We now show that some of the coefficients $\alpha_{i,j}^{m,n}$ need not be computed. This reduces the complexity of PRO below its theoretical value. First suppose that $m > 0$; in that case $\alpha_{i,j}^{m,n} = (xp_{m-1,n}) \cdot p_{i,j}$. From the definition of the inner product, it is easily seen that $\alpha_{i,j}^{m,n} = \alpha_{m-1,n}^{i+1,j}$. The coefficients $\alpha_{m-1,n}^{i+1,j}$ with $(i,j) \prec (m-1,n)$ have been computed when $p_{m-1,n}(x,y)$ was generated and can thus be reused when computing $p_{m,n}(x,y)$. Furthermore, it can be shown that for $m > 1$, $\alpha_{m-2,n}^{m,n} = N_{m-1,n}$. A detailed analysis shows that for $m \geq 1$, only $m+n$ of the $2(m+n)-1$ coefficients need to be computed. Unfortunately, in the case $m = 0$, only one coefficient can be reused, namely $\alpha_{m,n-2}^{m,n} = N_{m,n-1}$. The index $\mathrm{s}'_{m,n}$ of the first coefficient to compute is listed in table 1, together with the exact number of coefficients to compute. As in most of the recurrences $m \neq 0$, about half of all coefficient computations can be avoided.

We now present a new PRO algorithm which uses the information provided by the theorem and which reuses coefficients. We assume that a set of flags $f_{m,n}$ is computed before applying the actual PRO algorithm. For each (m,n), $f_{m,n}$ should be set to D when the theorem above states that $x^m y^n$ is LDP and to I when $x^m y^n$ is not LDP; when the theorem provides no information, $f_{m,n}$ should be set to U (unknown).

PRO algorithm:

set $p'_{0,0}(x,y) = 1$, compute N_{00} and $p_{0,0}(x,y)$
for $(m,n) = (1,0),(0,1),(2,0),\ldots$ in natural order
    if $m > 0$ and $f_{m-1,n} = \mathrm{D}$ or $n > 0$ and $f_{m,n-1} = \mathrm{D}$

```
    set f_{m,n} = D
  if f_{m,n} ≠ D
    compute q_{m,n}(x,y) and set q(x,y) = q_{m,n}(x,y)
    for (i,j) = s_{m,n}, ..., (m,n) - 1
      if f_{i,j} ≠ D
        if necessary, compute α_{i,j}^{m,n}
        set q(x,y) = q(x,y) - α_{i,j}^{m,n} p_{i,j}(x,y)
    compute N_{m,n} = ||q||
    if f_{m,n} ≠ I
      compute ||q_{m,n}||
      if N_{m,n} < ε||q_{m,n}|| set f_{m,n} = D
      else set f_{m,n} = I
    if f_{m,n} = I
      set p_{m,n}(x,y) = q(x,y)/N_{m,n}
```

This algorithm implements a mathematically equivalent, but numerically more stable version of eq. (1), based on a modification also employed in the "modified" GS algorithm [3, pp. 218]. It skips those (m,n) for which $x^m y^n$ is LDP; if it skips $x^m y^n$, then it also skips all $x^i y^j$ with $i \geq m$ and $j \geq n$ which are also LDP. When it is not known whether $x^m y^n$ is LDP, the algorithm computes $p'_{m,n}$ and then determines whether or not $p'_{m,n} \equiv 0$, by comparing $\sin\phi \triangleq \|p'_{m,n}\|/\|q_{m,n}\|$ to a small threshold ϵ; geometrically, ϕ is the angle between $q_{m,n}$ and the space spanned by the base functions $p_{i,j}$ with $(i,j) \prec (m,n)$. Hence $\phi = 0$ iff. $x^m y^n$ is LDP.

4 Computational complexity

Assuming that none of the $x^m y^n$ with $m+n \leq p$ are LDP, the new PRO requires (slightly fewer than) $2p^3 + 8p^2 + 12p - 3$ flopp and $p^2/2 + p$ square roots to generate all base functions with $m+n \leq p$. The old PRO (which computes all coefficients $\alpha_{i,j}^{m,n}$) requires $8p^3/3 + 8p^2 + 40p/3 - 5$ flopp (and the same number of roots) to perform this task. For large p, the new PRO requires 25% fewer computations and is therefore about 33% faster; for smaller values of p in the range 3 to 8, i.e., when computing 10 to 45 base functions, it is 13 to 26% faster than the old version.

In practice, some monomials will be LDP, especially for small regions and this further lowers the complexity. For instance, because all $x^m y^n$ with $m \geq M$ or with $n \geq N$ are LDP, the number of terms in the sum of eq. (1) can never exceed $2\min(M,N)$. Also, many (m,n) are skipped.

To estimate its complexity in realistic circumstances, the new PRO algorithm was included in a SIC program, which was used to compress two 256×256 images: a magnetic resonance (MR) image containing 403 regions and the "peppers" image, which was segmented into 574 regions. In each region, the number of base functions b was selected as $b = \min(0.2n_p, n_b)$, where n_p is the number of pixels in the region and n_b is an upper limit which was either 32 or 64. The mean number $\overline{b}$ of base functions per image region is indicated in table 2 and is much smaller than n_b in all cases. The proposed strategy for determining the number of base functions was also used in earlier SIC methods [2].

Table 2 shows that the new PRO algorithm is actually about 21 to 24% faster than the

Table 2: Computational complexities (Mflop)

				Old PRO		
Image	n_b	$\overline{b}$	PRO	actual	theor.	GS
MR	64	15	159.1	196.6	211.4	471.8
MR	32	13	58.9	71.0	78.6	127.9
Peppers	64	10	144.2	177.9	191.6	422.1
Peppers	32	9	56.8	68.3	75.8	122.5

old PRO algorithm which does not reuse coefficients. The table also shows that the actual complexity of the old algorithm is 7 to 10% lower than its theoretical value estimated using table 1. This demonstrates that the complexity is indeed lowered when LDP monomials are encountered. Finally, the new PRO algorithm is 2.2 times (for $n_b = 32$) to 3 times (for $n_b = 64$) faster than the GS algorithm.

Acknowledgment

This work was financially supported by the Belgian National Fund for Scientific Research (NFWO) through a mandate of "postdoctoral research fellow" and through the projects 39.0051.93 and 31.5831.95.

References

[1] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques", *IEEE Proceedings*, vol. 73, no. 4, pp. 549–574, Apr. 1985.

[2] M. Gilge, T. Engelhardt, and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform", *Signal Processing: Image Communication*, , no. 1, pp. 153–180, 1989.

[3] G.H. Golub and C.F. Van Loan, *Matrix Computations*, John Hopkins University Press, Tweede uitgave edition, 1991.

[4] W. Philips, "A fast algorithm for the generation of orthogonal base functions on an arbitrarily shaped region", in *Proceedings of ICASSP92*, Mar. 1992, vol. 3, pp. 421–424, San Fransisco.

[5] D. Jackson, "Formal properties of orthogonal polynomials in two variables", *Duke Mathematical Journal*, vol. 2, pp. 423–434, 1936.

[6] W. Philips, "Orthogonal base functions on a discrete two-dimensional region", Tech. Rep. DG 91-20, ELIS, RUG, Gent, Belgium, 1992.