Fast Coding of Arbitrarily Shaped Image Segments using Weakly Separable Bases

Wilfried Philips

Abstract—

Segmented Image Coding (SIC) produces images of better subjective quality than techniques such as the Joint Photographic Experts Group (JPEG) standard at very low bit rates. It segments images into regions of similar texture and codes their contour and texture separately. Most SIC techniques represent the texture as a linear combination of orthonormal base functions, which differ from region to region. In earlier SIC-methods, the orthonormal bases were generated by the Gram–Schmidt (GS) procedure, whose excessive computational requirements are unacceptable in many applications. This paper introduces the new class of weakly-separable (WS) orthonormal bases. The WS base functions are the product of two components, which can be generated very quickly and using a modest amount of memory. The class of WS base functions includes polynomials, cosines and warped polynomials. The paper describes the spatial properties of these base functions. Experimental data shows that for typical coding parameters, WS SIC is 8 to 40 times faster than traditional SIC, at the expense of a 3% to 6% larger rms error. However, the subjective quality of the compressed images is only slightly smaller for the WS method. Considering its computational advantages, WS SIC is therefore a promising technique for fast low bit rate coding.

I. Introduction

Segmented image coding (SIC) [1] is a second-generation image compression technique that segments images into non-rectangular regions of similar texture. The region contours are coded, e.g., by chain codes, while the texture inside a region is approximated by a bivariate polynomial or some other smoothly varying function. At high compressions, SIC allows a better subjective image quality than block transform methods such as the Joint Photographic Experts Group (JPEG) standard [2], in which the well-known block-effect and the blurring of edges seriously degrade the image quality.

In SIC, edge blurring is avoided by coding the contours and the texture separately. Also, the block effect is eliminated even though the texture approximation is discontinuous at the region boundaries. This is because the discontinuities coincide with the region boundaries and are therefore masked by the image edges [3, p. 269]. The gentle degradation of the image quality with decreasing bit rate is an important advantage of SIC in high-compression coding and in the progressive transmission of images.

This paper deals with the representation of the gray values of the image regions, and not with segmentation techniques [4], [5] nor with methods for representing the shape of the image regions [6], [7]. The earliest SIC variants [1] approximated regions by bivariate polynomials of low degree (e.g., 1 or 2), which cannot represent large regions accurately. Therefore, large regions must be subdivided artificially, but this increases the bit-rate because more contours and more coefficients must be coded. Also, the contours separating artificial regions do not correspond to real image edges and therefore do not mask discontinuities.

To avoid these problems, recent SIC methods represent the gray values by a linear combination of many (e.g., 16 to 64) base functions. The most promising functions are polynomials [8], but cosines [9], [10] and other functions [11], [9] have also been used. In all cases, the coefficients in the linear combination are computed so as to minimize the mean-squared approximation error. This computation is greatly simplified when the base functions are normalized and mutually orthogonal on the region being coded [8]. Unfortunately, the computation of such orthonormal base functions is very time consuming, especially when many are needed. Also, orthonormal base functions depend on the shape of the region and must be computed for every single region. Typically, the computation of the orthonormal bases requires many more operations than segmenting the image and computing the texture coefficients. Therefore, fast algorithms for the construction of orthonormal SIC bases are essential.

The first high-degree SIC methods generated orthonormal bases by applying the Gram-Schmidt (GS) procedure to nonorthonormal *starting functions.* For instance, Gilge [8] orthogonalized the *naturally ordered* functions $1, x, y, x^2, xy, y^2, x^3, \ldots$ to obtain orthonormal polynomials $P^c_{m,n}(x,y)$, which we will call the *classical* base functions.

GS requires $\mathcal{O}(b^2p)$ flop to generate b orthonormal base functions for a region of p pixels. Also, even though each base function is used only once, namely for computing a single coefficient, the GS approach requires that all base functions be kept in memory until all the coefficients have been computed. This is because the computation of a given base function involves all its predecessors. Therefore, $\mathcal{O}(bp)$ intermediate storage locations are needed. The classical base can be generated faster by the *polynomial recursive orthogonalization* (PRO) algorithm [12] which requires only $\mathcal{O}(b^{3/2}p)$ flop and $\mathcal{O}(\sqrt{b}p)$ storage locations. Even so, the computational and memory requirements can be prohibitively large when b is large.

Section II of this paper introduces a new set $\mathcal{B}$ of orthonormal polynomial base function. The new base is *weakly separable* (WS) [14], which means that its base functions

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.

W. Philips is with the Department of Electronics and Information Systems, University of Gent, St.-Pietersnieuwstraat 41, B9000 Gent, Belgium. E-mail: philips@elis.rug.ac.be

$P_{m,n}(x,y)$ are of the form $P_{m,n}(x,y) = g_m(x,y)h_{m,n}(y)$. Section II also shows that the WS base functions $P_{m,n}(x,y)$ can be computed many times faster than the classical base functions $P^c_{m,n}(x,y)$ and using less memory. The precise computational and storage requirements depend on the shape of the region, on which base functions are computed and on in which order these are computed. Typically, only $\mathcal{O}\left(p\sqrt{b} + b\sqrt{p}\right)$ computations and from $\mathcal{O}\left(\sqrt{bp}\right)$ to $\mathcal{O}(p)$ memory locations are required.

Section III presents a more general class of orthonormal WS bases, consisting of *warped polynomials*, i.e., functions $\Phi_{m,n}(x,y)$ of the form $\Phi_{m,n}(x,y) = f_{m,n}\left(X(x,y), Y(x,y)\right)$, where $f_{m,n}(u,v)$ is a bivariate polynomial in u and v. It shows that such a base exists whenever the mapping $(x,y) \to (x'y')$, where $x' = X(x,y)$ and $y' = Y(x,y)$ is bijective on the image region. In that case, $X(x,y)$ and $Y(x,y)$ are called *warping functions*. The class of warped WS bases includes cosine and polynomial bases as special cases. The recently introduced shape-adaptive discrete cosine transform (DCT) [15] is also an example of a WS base contained in the above class. However, the shape-adaptive DCT only works well on very small regions, contained within a rectangle of 8 by 8 pixels [15]. The WS bases in this paper do not have this restriction.

In the important specific case where $X(x,y) = X(x)$ and $Y(x,y) = Y(y)$, section III shows that under nonrestrictive conditions, the functions $\Phi_{m,n}(x,y)$ are well approximated by products of two amplitude- and frequency- modulated cosines. The spatial properties (i.e., local amplitude and local spatial frequency) of the WS base functions are determined by the warping functions. This result allows a qualitative theoretical comparison of the coding properties of different WS bases.

Section IV experimentally evaluates the computational complexity and the coding properties of polynomial and cosine WS bases and compares them with those of the classical base. The results show that the WS bases can typically be computed 8 to 40 times faster than the classical base, when the latter is computed by (the fastest version of) PRO. The price to be paid is a typically 3% to 6% larger root mean-squared (rms) approximation error. However, the corresponding reduction in *subjective* image quality is quite small and doesn't outweigh the benefits of the much faster computation. The results also show that the polynomial WS base produces slightly better images than the cosine bases. However, in regions bounded by weak edges, it introduces larger discontinuities than one of the cosine bases. This suggests the use of the cosine base in regions with weak edges and of the polynomial base in regions with strong edges.

II. Weakly separable bases

In the first part of this section, we define the new bases and show that they are indeed WS. Also, we describe how to generate their components quickly. The second part of this section evaluates the precise computational requirements of WS SIC.

A. WS bases and their generation

Before defining the WS bases, we introduce some notations. Let Ω be the set of the (x,y)-coordinates of the pixels of the region to be coded. The region Ω can always be expressed as the union of n_r different rows $R_k \triangleq \{(x,y) \in \Omega : y = y_k\}$, $0 \le k < n_r$. Let p be the number of pixels in Ω, r_k the number of pixels in R_k and c_k the number of rows with at least $k+1$ pixels. Without loss of generality, we assume that the rows are indexed such that $r_i \le r_j$ if $i > j$. It is important to keep this assumption in mind, because many of the following equations depend on it.

With $\langle f|g\rangle \triangleq \sum_{(x,y)\in\Omega} f(x,y)g(x,y)$ we denote the *inner product* of $f(x,y)$ and $g(x,y)$ on Ω. The functions $f(x,y)$ and $g(x,y)$ are orthogonal if their inner product vanishes. A function $f(x,y)$ is *normalized* if its norm $||f|| \triangleq \sqrt{\langle f|f\rangle}$ equals unity. SIC approximates the image intensity $s(x,y)$ in Ω by a least mean-squares (LMS) approximation $\tilde{s}(x,y) \triangleq \sum_{(m,n)\in I} A_{m,n}P_{m,n}(x,y)$, where I is a suitably chosen index set, $P_{m,n}(x,y)$ are orthonormal base functions and $A_{m,n} = \langle s|P_{m,n}\rangle$.

The WS base functions $P_{m,n}(x,y)$ are defined as the functions that result from orthogonalizing the starting functions $x^m y^n$ with $0 \le n < n_r$ and $0 \le m < r_n$ in the *lexicographical order* $1, y, y^2, \ldots, x, xy, xy^2, \ldots$ It has been shown [16] that these starting functions are linearly independent on Ω, which implies that the orthogonalization produces exactly p orthonormal functions. This means that $\tilde{s}(x,y)$ always identically equals $s(x,y)$ when $\tilde{s}(x,y)$ includes *all* base functions $P_{m,n}(x,y)$. Error-free image coding can therefore always be achieved in SIC, just as in block coding. Of course, error-free coding is often not required, but the fact that it can be achieved is important, e.g., in progressive image transmission.

As can be seen from the above definition, the WS base functions are very asymmetric functions of x and y. This will become even more apparent later and is also the case for the dual base, obtained by orthogonalizing in the order $1, x, x^2, \ldots, y, xy, x^2y, \ldots$ However, it can be shown that for rectangular regions, both bases are equal. This means that the degree of symmetry of the base functions depends on the symmetry of the region.

In the following paragraphs, we will show that $\mathcal{B} \triangleq \{P_{m,n}(x,y) : 0 \le n < n_r : 0 \le m < r_n\}$ is WS. Also, we present a procedure for computing the (components of) the base functions much faster than is possible with orthogonalization techniques such as GS.

In [16] it was shown that the polynomials $P_{m,n}(x,y)$ satisfy the three-term recurrence

$$P'_{m,n}(x,y) = yP_{m,n-1}(x,y) - \sum_{l=1}^{2} \alpha^{(l)}_{m,n} P_{m,n-l}(x,y) \quad (1)$$

$$P_{m,n}(x,y) = P'_{m,n}(x,y)/N_{m,n}, \quad (2)$$

where $\alpha^{(l)}_{m,n} = \langle yP_{m,n-1}|P_{m,n-l}\rangle$ when $l \le n$ and 0 otherwise and where $N_{m,n} = ||P'_{m,n}||$. The recurrence is started with $P'_{0,0}(x,y) = 1$. By induction on n it is easily shown that $P_{m,n}(x,y) = P_{m,0}(x,y)Q_{m,n}(y)$, where $Q_{m,n}(y)$ is a polynomial of degree n in y. This means that $\mathcal{B}$ is weakly separable. The WS property allows considerable computational savings; indeed, for $n > 0$ it allows the computation of $P_{m,n}(x,y)$ for p values of (x,y) to be replaced by the computation of the corresponding $Q_{m,n}(y)$ for only n_r values of y. We now describe the computation of the base function components in detail.

First, we consider the functions $P_{m,0}(x,y)$; in [16] it was shown that these can be expressed in terms of the orthonormal polynomials $p_{m,k}(x)$, $m = 0, 1, \ldots$ associated with the *row inner product* $\langle f|g\rangle_{r,k} \triangleq \sum_{x\in R_k} f(x)g(x)$, i.e., the polynomials defined by the recurrence

$$p'_{m,k}(x) = xp_{m-1,k}(x) - \sum_{l=1}^{2} \beta^{(l)}_{m,n} p_{m-l,k}(x), \qquad (3)$$

$$p_{m,k}(x) = p'_{m,k}(x)/n_{m,k}, \qquad (4)$$

where $\beta^{(l)}_{m,k} = \langle xp_{m-1,k}|p_{m-l,k}\rangle_{r,k}$ when $l \leq m$ and 0 otherwise and where $n_{m,k} = \|p'_{m,k}\|_{r,k}$. The recurrence holds for $m > 0$ and is started with $p'_{0,k}(x) = 1$.

More precisely, $P_{m,0}(x, y_k)$ is proportional to $p_{m,k}(x)$ on R_k, i.e., $P_{m,0}(x,y_k) = s_{m,k}p_{m,k}(x)$ for all $x \in R_k$ when $0 \leq m < r_k$; on the other hand, $P_{m,0}(x,y_k) = 0$ for all $x \in R_k$ when $m \geq r_k$. The constants $s_{m,k}$ are given by $s_{m,k} = t_{m,k}/T_m$, where $t_{m,k} = \prod_{j=0}^{m} n_{j,k}$ and $T_m^2 = \sum_{k=0}^{c_m-1} t^2_{m,k}$. In practice, it is not advisable to compute the constants $s_{m,k}$ by these equations, because $t_{m,k}$ varies exponentially with m which might cause numerical problems. Instead, it is better to compute $s_{m,k}$, $m = 0, 1, \ldots$ recursively using the equations $z_{m,k} = n_{m,k}s_{m-1,k}$, $d_m^2 = \sum_{k=0}^{c_m-1} z^2_{m,k}$ and $s_{m,k} = z_{m,k}/d_m$.

Because the constants $s_{m,k}$ are positive it is always possible to find a set of non-negative functions $w_m(y)$ for which $w_m(y_k) = s^2_{m,k}$. We introduce these *weight functions* because they play an important role in the following. Note that only the values $w_m(y)$ for $y = y_k$ are needed in actual computations; nevertheless, it is often convenient to think of $w_m(y)$ as a function that is defined for all real y.

The previous discussion shows that $P_{m,n}(x,y_k) = p_{m,k}(x)q_{m,n}(y_k)$, where $q_{m,n}(y) = \sqrt{w_m(y)}Q_{m,n}(y)$. As we will explain later, from a computational point of view it is more efficient to compute $q_{m,n}(y)$ instead of $Q_{m,n}(y)$. By setting $y = y_k$ in eqs. (1) and (2) and dividing both sides of these equations by $p_{m,k}(x)$ it is seen that the functions $q_{m,n}(y)$ satisfy the recurrence

$$q'_{m,n}(y) = yq_{m,n-1}(y) - \sum_{l=1}^{2} \alpha^{(l)}_{m,n} q_{m,n-l}(y) \qquad (5)$$

$$q_{m,n}(x,y) = q'_{m,n}(y)/N_{m,n}, \qquad (6)$$

which holds for $n > 0$ and which is started with $q'_{m,0}(x,y) = \sqrt{w_m(y)}$. The recurrence coefficients $\alpha^{(l)}_{m,n}$, and $N_{m,n}$ are the same as those in eqs. (1-2), but they can also be expressed as $\alpha^{(l)}_{m,n} = \langle yq_{m,n-1}|q_{m,n-l}\rangle_{c,m}$ and $N_{m,n} = \|q'_{m,n}\|_{c,m}$, where $\langle f|g\rangle_{c,m} \triangleq \sum_{k=0}^{c_m-1} f(y_k)g(y_k)$ is the *column inner product* and $\|f\|_{c,m} \triangleq \sqrt{\langle f|g\rangle_{c,f}}$ the corresponding norm. Using these new expressions, the recurrence coefficients can be computed much faster, because the column inner product only requires $2c_m$ flop instead of $2p$ flop. Also, the expressions show that the polynomials $Q_{m,n}(x,y)$, $n \geq 0$ depend on $w_m(y)$ only which in turn depends only on $P_{m,0}(x,y)$.

Note that the polynomials $Q_{m,n}(y)$ also satisfy the recurrence (5-6). However, computing the recurrence coefficients in terms of $Q_{m,n}(y)$ requires $3c_m$ flop instead of $2c_m$ flop. This is the reason for expressing all computations in terms of $q_{m,n}(y)$ and not in terms of $Q_{m,n}(y)$.

From a computational point of view, it is important to note that $\alpha^{(2)}_{m,n} = N_{m,n-1}$ and $\beta^{(2)}_{m,n} = n_{m-1,n}$. This follows immediately from the orthonormality of the functions $P_{m,n}(x,y)$ and of the functions $p_{m,k}(x)$. It means that there is no need to compute $\alpha^{(2)}_{m,n}$ and $\beta^{(2)}_{m,n}$.

Finally, we consider the computation of the LMS texture coefficients $A_{m,n}$ and of the reconstructed texture $\tilde{s}(x,y)$. Because of the weak separability of the base $\mathcal{B}$, the coefficients $A_{m,n}$ can be computed by

$$A_{m,n} = \langle s|P_{m,n}\rangle = \sum_{k=0}^{c_m-1} q_{m,n}(y_k)B_{m,k}, \qquad (7)$$

where $B_{m,k} \triangleq \sum_{x\in R_k} p_{m,k}(x)s(x,y_k)$. On Ω, the texture $s(x,y)$ can be reconstructed from the coefficients $A_{m,n}$ using the inverse equations:

$$B_{m,k} = \sum_{n=0}^{c_m-1} A_{m,n}\, q_{m,n}(y_k) \qquad (8)$$

$$s(x,y_k) = \sum_{m=0}^{r_k} B_{m,k}\, p_{m,k}(x). \qquad (9)$$

When only some of the coefficients $A_{m,n}$ are computed, the LMS texture approximation $\tilde{s}(x,y)$ is also obtained by the above equations, in which the remaining coefficients are set to zero. In that case, many of the intermediate coefficients $B_{m,k}$ are zero and need not be computed.

B. Computational complexity

We now estimate the computational complexity and the memory requirements of WS SIC. First, we concentrate on the generation of the base functions, and next on the computation of the texture coefficients $A_{m,n}$ and of the reconstruction $\tilde{s}(x,y)$ from these coefficients.

Both the number of computations and the memory requirements depend in a complex way on the shape of Ω and on which texture coefficients are computed. In practice, the coefficients are computed in the *natural* order $(m,n) = (0,0),(1,0),(1,1),(2,0),(1,1),(0,2),\ldots$ or in the *rectangular* order $(m,n) = (0,0),(1,0),\ldots,(M-1,0),\ldots,(1,0),(1,1),\ldots(M-1,M-1)$ or in orders which are variants thereof. If we call the number of computed

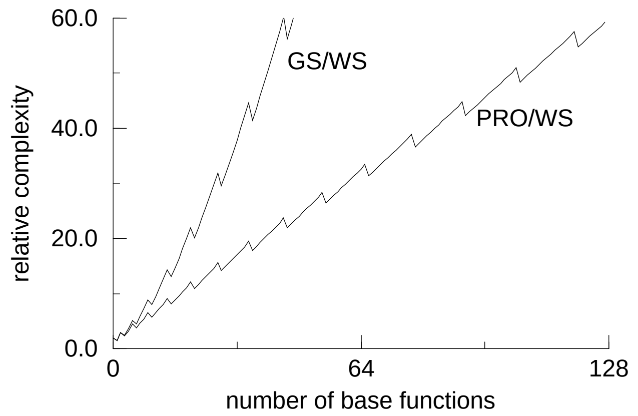

Fig. 1. Computational complexity of the GS and PRO algorithms, relative to the complexity of WS in function of the number of base functions for a square region. The base functions are computed in natural order.

base functions b, then we have approximately $0 \leq m+n < \sqrt{2b}$ in the first case and $0 \leq m, n < \sqrt{b}$ in the second.

In texture coding, the object is to compute b texture coefficients $A_{m,n}$. For each (m,n) this requires computing in turn $p_{m,k}(x)$, $s_{m,k}$ and $B_{m,k}$ for $0 \leq k < c_m$ and finally $q_{m,n}(y)$. The base function components are computed by the three-term recurrences of section II-A. For fixed m, n, these recurrences require about $10p$ flop to compute $p_{m,k}(x)$, $0 \leq k < c_m$ and about $10c_m$ flop to compute $q_{m,n}(y)$. Similarly, about $4c_m$ flop is needed to compute $s_{m,k}$ for $0 \leq k < c_m$.

As $c_m \leq n_r$, the total number of base computations needed to compute b texture coefficients is upper bounded by $10pM+14bn_r$, where $M-1$ is the maximum value of m. For nearly square regions, $n_r \approx \sqrt{p}$ and the complexity is about $10p\sqrt{2b}+14b\sqrt{p}$ for the natural coefficient order and $10p\sqrt{b}+14b\sqrt{p}$ for the rectangular order. For elongated regions, the complexity per region pixel is lowest when the region extends mainly in the x-direction. If this is not the case, it is advisable to use the dual base.

The number of base computations is far lower for the WS base functions than for the classical base functions. Originally, the latter were computed by the GS procedure, which requires $\mathcal{O}\left(b^2p\right)$ flop; the PRO algorithm [12] is faster but still requires $\mathcal{O}\left(b^{3/2}p\right)$ flop. Figure 1 demonstrates that for a square region and for the natural coefficient order, the WS base computation is 9 to 60 times faster than the fastest version of the PRO algorithm and 15 to 250 times faster than GS for values of b ranging from 16 to 128.

The computation of the texture coefficients $A_{m,n}$ using eq. (7) is also faster than in classical SIC where they are computed using the equation $A_{m,n} = \langle s|P^c\rangle$ which requires $2pb$ flop to compute b coefficients. Indeed, with $M-1$ the maximum value of m, the intermediate coefficients $B_{m,k}$ must be computed for $0 \leq m < M$ and $0 \leq k < c_m$. This requires at most $2Mp$ flop. Next, computing the b coefficients $A_{m,n}$ requires at most $2bn_r$ flop. Therefore, the total complexity is upper bounded by $2Mp + 2bn_r$, which is typically smaller than $2bp$. For a square region the complexity becomes $2\gamma p\sqrt{b} + 2b\sqrt{p}$, where $\gamma = \sqrt{2}$ for the natural order and 1 for the rectangular order.

Finally, the computational complexity of the texture reconstruction from the coefficients $A_{m,n}$ is comparable to that of the texture coding. Indeed, the computation of $\tilde{s}(x,y)$ requires the generation of the base function components, and of the intermediate coefficients $B_{m,k}$; the computational complexity of the first task was discussed above, and it is easily verified that computing $\tilde{s}(x,y)$ from a given number of coefficients $A_{m,n}$ requires about the same computation time as computing those coefficients from $s(x,y)$.

We now consider the storage requirements of WS SIC. These storage requirements can be kept very small if the computations are performed in the following order:

1. Compute $p_{m,k}(x)$, $s_{m,k}$ and $B_{m,k}$ in the order $(k,m) = (0,1), (0,2), \ldots, (1,0), (1,1), \ldots$
2. Compute $q_{m,n}(y)$ and $A_{m,n}$ in the order $(m,n) = (0,1), (0,2), \ldots, (1,0), (1,1), \ldots$

In the first step of this procedure, $p_{m,k}(x)$ is only needed for computing $p_{m+1,k}(x)$, $p_{m+2,k}(x)$ and $B_{m,k}$; therefore, only three of the functions $p_{m,k}(x)$ need to be kept in memory at any time. Similarly, in the second step, only three of the functions $q_{m,n}(y)$ must be stored. Consequently, intermediate storage is mainly needed for the intermediate coefficients $B_{m,k}$ and the numbers $s_{m,k}$. The latter have to be stored, because they are needed in the second step. This requires no more than Mn_r memory units. In the case of a square region, this becomes $\gamma b\sqrt{p}$ where $\gamma = \sqrt{2}$ for the natural order and $\gamma = 1$ for the rectangular order.

In progressive image transmission, only one texture coefficient is initially computed in each region. Next, the number of coefficients per region is increased by one and the additional coefficients are computed in all regions. This process is repeated until all possible coefficients have been computed or until the available transmission time has expired. As the coefficients must be transmitted at a more or less even rate and as it is desirable to spread the total computational load evenly over the total transmission time, it is better not to perform the computations in the above order. Instead, in progressive coding $p_{m,k}(x)$, $s_{m,k}$, $B_{m,k}$ and $q_{m,n}(y)$ should be generated at the moment they are actually needed to compute a coefficient $A_{m,n}$.

In that case, the precise storage requirements depend on the order in which the coefficients are computed. However, an upper bound on the memory required for each region is easily obtained. Indeed, there exist only p numbers $s_{m,k}$. Furthermore, only $3p$ memory units are needed to store $p_{m,k}(x)$, $0 \leq k < n_r$ for three successive values of m. Similarly, storing $q_{m,n}(y)$, $m = 0, 1, \ldots$ for three successive values of n requires no more than $3p$ memory units. Therefore, the required base storage is upper bounded by $7p$.[1] Finally, p memory units are sufficient to store all the intermediate coefficients $B_{m,k}$. The number of memory units needed to progressively transmit a complete image is therefore about eight times the number of pixels in the image.

[1] Actually, only $5p$ memory units are needed if $p_{m-2,k}(x)$ and $q_{m,n-2}(y)$ are overwritten by $p_{m,k}(x)$ and $q_{m,n}(y)$, respectively.

The storage requirements of WS SIC are typically lower than those of classical SIC, especially when the number of base functions per region is high. Indeed, classical SIC requires bp memory units per region, when using GS and about $p\sqrt{2b}$ units when using PRO. Therefore, classical SIC seems less suited for progressive transmission.

III. Warped weakly separable bases

In the first part of this section we define a more general class of WS orthonormal bases, consisting of so-called warped polynomials. We investigate their properties and present a fairly general analytical approximation of them. The results show the limitations of warped WS base functions. They apply in particular to the (non-warped) bases of the previous section. In the second part, we present a few examples of warped WS bases, which will be used in the experimental evaluation of section IV.

A. Definition and properties

Let the mapping $(x, y) \rightarrow (x'y')$ where $x' = X(x, y)$ and $y' = Y(x, y)$ be a bijection on Ω and let Ω' be the set of (x', y') coordinates of the pixels of Ω. Using the theory of the previous section it is possible to obtain a set of WS separable polynomial base functions $P_{m,n}(x', y')$ which are orthonormal on Ω'. The corresponding functions $\Phi_{m,n}(x, y) \triangleq P_{m,n}\Big(X(x, y), Y(x, y)\Big)$ are orthonormal on Ω. They are called *orthonormal warped polynomials* and $X(x, y)$ and $Y(x, y)$ are called *warping functions*.

By selecting different sets of warping functions, many different WS orthonormal bases can be obtained. The coding properties of the bases depend on the precise values of $X(x, y)$ and $Y(x, y)$. However, their computational properties depend only on the *structure* of the transformed set Ω', i.e., the number of rows in Ω' and the number of pixels in these rows.

In this paper we restrict the warping function $Y(x, y) = Y(y)$ to be a function of y only. In that case, Ω and Ω' have exactly the same structure, which implies that the corresponding warped WS bases for a region can be computed just as quickly and using the same amount of memory as the WS base for that region. Furthermore, we assume that $Y(y)$ is a strictly increasing function of y and that $X(x, y)$ is a strictly increasing function of x for all fixed y. These last two conditions guarantee that the mapping $(x, y) \rightarrow (x', y')$ is bijective. Also, they facilitate the study of the spatial properties of the warped WS base functions.

Now let $\phi_{m,k}(x) \triangleq p_{m,k}(X(x, y_k))$ and $\varphi_{m,n}(y) \triangleq q_{m,n}(Y(y))$ be the components of the base functions $\Phi_{m,n}(x, y)$. The function $\phi_{m,n}(x)$ is a uni-variate warped polynomial, while $\varphi_{m,n}(y)$ is the product of the uni-variate warped polynomial $Q_{m,n}(Y(y))$ and the function $w_m(Y(y))$. In the following paragraphs, it is convenient to introduce the function $\phi'_m(x, y)$ which equals $\phi_{m,k}(x)$ when $y = y_k$ and whose values are irrelevant when $y \neq y_k$.

The properties of uni-variate warped polynomials with strictly increasing warping functions are known in great detail. Indeed, such functions can be well approximated by amplitude- and frequency modulated cosines [17]. In [18], this result was used to derive approximations of the component functions $\phi'_m(x, y)$ and $\varphi_{m,n}(y)$ for regions Ω of the form $\Omega = \{(x, y) : x_-(y) \leq x < x_+(y), y_- \leq y < y_+\}$, where x and y are integers. The approximations are most easily expressed in terms of the *relative pixel coordinates* $x_r(x, y)$ and $y_r(y)$ defined by

$$x_r(x, y) \triangleq -1 + 2\frac{X(x, y) - X(x_-(y), y)}{X(x_+(y), y) - X(x_-(y), y)}, \quad (10)$$

$$y_r(y) \triangleq -1 + 2\frac{Y(y) - Y(y_-)}{Y(y_+) - Y(y_-)}. \quad (11)$$

In terms of these coordinates,

$$\phi'_m(x, y) \approx A(x, y) \cos\left(m \arccos x_r(x, y) + \Gamma(x, y)\right) \quad (12)$$

$$\varphi_{m,n}(y) \approx A'(y) \cos\left(n \arccos y_r(y) + \Gamma'(y)\right) \quad (13)$$

in which $A(x, y) \triangleq \sqrt{\frac{2}{\pi}\left|\frac{\partial}{\partial x} \arccos x_r(x, y)\right|}$ and $A'(y) \triangleq \sqrt{\frac{2}{\pi}\left|\frac{d}{dy} \arccos y_r(y)\right|}$. The approximations hold for $(x, y) \in \Omega$ only. They predict the spatial variation of the base functions reasonably well[2] when $5 < m \ll \min_k r_k$ and $5 < n \ll c_m$. The functions $\Gamma(x, y)$ and $\Gamma'(y)$ are generally slowly varying; for large enough m and n they have little influence on the frequencies of the cosines in the approximations.

According to eqs. (12) and (13), the base function components $\phi'_m(x, y)$ and $\varphi_{m,n}(y)$ are amplitude- and frequency modulated cosines. For sufficiently large m and n, the spatial variation of these components is mainly determined by the spatial variation of $x_r(x, y)$ and $y_r(y)$. In turn, the spatial variation of $x_r(x, y)$ depends strongly on the functions $X_-(y) \triangleq X(x_-(y), y)$ and $X_+(y) \triangleq Y(x_+(y), y)$, which describe the geometry of the transformed set Ω'.

The spatial behavior of the base function components is most easily described in terms of the location of their zero crossings. Indeed, the local orientation and local density of the zero crossings of a component determine its local spatial frequency. For high enough m and n, the spatial variation of $\Gamma(x, y)$ and $\Gamma'(y)$ can be ignored and the zero crossings of $\phi'_m(x, y)$ and $\varphi_{m,n}(y)$ are approximately lines of constant $x_r(x, y)$ and constant $y_r(y)$, respectively.

First assume that (x, y) is not too close to the boundary of Ω such that we can approximate $\arccos(x_r)$ by $\pi/2 - x_r$ in eq. (12). Eq. (10) then implies that the horizontal distance between neighboring zero crossings of $p'_m(x, y)$ is proportional to the row width $X_+(y) - X_-(y)$ of Ω' and inversely proportional to the local derivative $\partial x_r/\partial x$. This means that the local horizontal spatial frequency of this component is proportional $\partial x_r/\partial x$ and inversely proportional to the row width of Ω'. On the other hand, the vertical spatial frequency of this component depends on the *variation* of the row width $X_+(y) - X_-(y)$ and of the row position $X_-(y)$ with y and on $\partial x_r/\partial y$.

[2] See [17] for more details.

The previous discussion is not entirely correct near the edges of Ω; there the zero crossings of $\phi'_m(x,y)$ are pushed closer together owing to the presence of the arccos function in eq. (12). This increases the local horizontal spatial frequency near the region edges.

The second component $\varphi_{m,n}(y)$ is a function of y only; hence its horizontal spatial frequency is always zero. Its vertical local spatial frequency is inversely proportional to $\partial y_r/\partial y$ and proportional to the height $Y(y_+) - Y(y_-)$ of Ω'. Again, the presence of the arccos function causes the spatial frequency to increase for y near y_- or near y_+.

In SIC, regions are supposed to be areas of similar texture, which implies that the local spatial frequency spectrum of this texture is theoretically close to position-independent. The base functions $\Phi_{m,n}(x,y)$ match this behavior best when the spatial frequencies of the components $\phi'_m(x,y)$ and $\varphi_{m,n}(y)$ are also position-independent. Clearly, this is not always the case for the warped WS base functions. This is (at least theoretically) a disadvantage. In this context, note that a discontinuous change in the row width or row position induces a discontinuous change in $\phi'_m(x,y)$, which is undesirable.

In practice, the texture spectrum within a region is never completely position-independent. In particular, the texture properties generally change near the region edges. In this context we must distinguish between *strong edges* which correspond to rapid and large changes in texture properties and *weak edges* which correspond to more gradual changes. In the first case, the compression of the zero crossings of $\phi'_m(x,y)$ near the boundary of Ω owing to the presence of the arccos function in eq. (12) is beneficial. This is because base functions with a higher spatial frequency near the edges will approximate the texture better.[3] In the case of weak edges, the situation is not so clear.

The local properties of the base functions can be influenced by appropriately selecting $X(x,y)$ and $Y(x,y)$. These functions can be used to change the location of the zero crossings. However, they cannot modify the *total number* of zero crossings which always equals m for $\phi'_m(x,y)$ and n for $\varphi_{m,n}(y)$. This means that it is generally impossible to enforce a (more or less) constant spatial frequency on $\phi'_m(x,y)$. On the other hand, it is possible to modify segmentation methods so that they produce mainly regions with slowly varying row width. This can be achieved by artificially splitting regions, possibly at the expense of a lower coding gain. On the other hand, regions with a rapidly varying row width may have a slowly varying column width. In that case, the dual base could be used. In this paper, we will not address these questions any further.

B. Examples

We now present three examples of warped WS bases. The first one is the polynomial (non-warped) base POL, for which $X(x,y) = X(x) = x$ and $Y(y) = y$. The

[3] In [17] it is explained that optimal one-dimensional coding of a signal is achieved when the local frequency of the base functions matches the local signal bandwidth. Here, the signal is the texture on a single row.

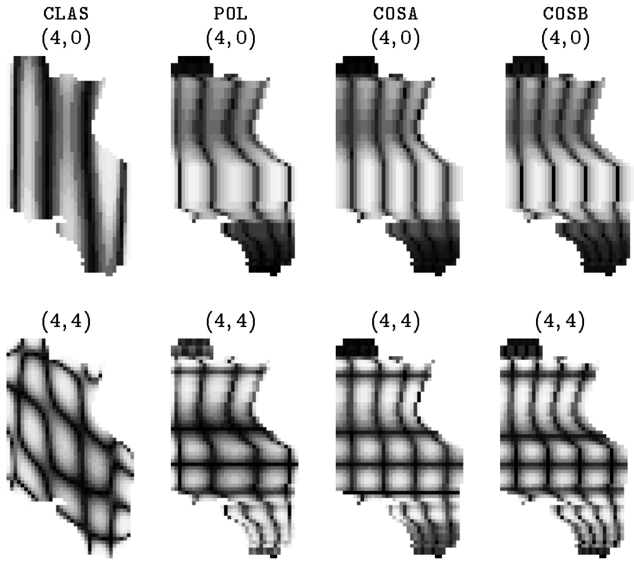

Fig. 2. Examples of classical base functions $P^c_{m,n}(x,y)$ and warped WS base functions $\Phi_{m,n}(x,y)$ for the same region and for two values of (m,n). The figure displays absolute values (dark corresponds to small values).

other bases are cosine bases, which we will call COSA, and COSB. The COSA base is derived from $X(x,y) = X(x) = \cos((x - x_-)\pi/w)$ and $Y(y) = \cos((y - y_-)\pi/h)$, where w and h are the width and height of the smallest rectangle which encloses Ω and (x_-, y_-) its top-left corner. For the COSB base, $X(x, y_k) = w_k \cos((x - x_k)\pi/w_k)$, where x_k is the x-coordinate of the left-most pixel in R_k and w_k the width of R_k. The warping function $Y(y)$ is the same as for COSA. It can be shown that the COSA and COSB base functions approximately equal those of the two-dimensional DCT when Ω is a rectangle. In fact, this approximation turns into an equality when x_-, y_-, and x_k are replaced by $x_- - 1/2$, $y_- - 1/2$ and $x_k - 1/2$, respectively. Therefore, the COSA and COSB bases can be considered generalizations of the JPEG base for non-rectangular regions.

For the COSB base, $\arccos(x_r(x, y_k)) = (x - x_k)\pi/w_k$. Therefore, the COSB base functions do not have an increased spatial frequency near the region edges, in contrast to those of the POL base. This means that the COSB base is less suitable for coding strong edges. On the other hand, the amplitude $A(x,y)A'(y)$ of the COSB base functions is constant, while that of the POL base increases near the region edges. This suggests that the COSB base is less susceptible to introduce visible discontinuities at weak region edges.

The COSA base has properties somewhere in between those of the POL and COSB bases: when $x_- \ll x_k$ and $x_k + w_k \ll x_- + w$, then $X(x, y_k) \approx (x - x_-)\pi/w$. From eq. (10) it follows that $x_r(x, y_k)$ then equals the relative x-coordinate of the POL base. On the other hand, when $x_k \approx x_-$ and $w_k \approx w$, $x_r(x, y_k)$ approximately equals the relative x-coordinate of the COSA base.

Figure 2 shows some of the POL, COSA and COSB base functions for the same image region. For comparison, the

TABLE I

COMPARISON OF THE POL, POL' AND CLAS BASE ON THE "CAMERAMAN" IMAGE. THE CLAS BASE IS COMPUTED BY PRO OR BY GS. THE QUANTIZATION PARAMETER ϵ EQUALS 15.

			Computations (Mflop)				Rms-error		
α	n_b	CR	POL	POL'	PRO	GS	POL	POL'	CLAS
10	16	31	2.7	1.7	20	33	13.4	13.5	13.0
10	32	29	3.8	2.8	55	120	13.1	13.2	12.6
10	64	28	5.3	4.0	149	440	12.9	13.0	12.4
20	16	27	2.8	1.8	21	34	12.7	12.8	12.2
20	32	25	3.9	3.0	58	126	12.3	12.4	11.6
20	64	23	5.6	4.3	158	470	11.9	12.0	11.2
50	16	23	2.8	1.9	21	35	12.1	12.2	11.5
50	32	20	4.1	3.1	61	132	11.3	11.4	10.6
50	64	17	5.9	4.5	168	502	10.6	10.8	9.9

corresponding classical base functions are also shown. The base functions in Fig. 2 are of relatively low degree and the above approximations do not apply to them. Nevertheless, the WS base functions in Fig. 2 qualitatively behave as predicted by the above theory: there are two distinct sets of zero crossings, consisting of lines of approximately constant x_r and constant y_r, respectively. Also, for $(m, n) = (4, 4)$, the local amplitude of the WS base functions is higher in regions of higher spatial frequency, i.e., there where the zero crossings are close together. This agrees with the fact that $A^2(x, y)$ and $A'^2(y)$ are proportional to $\left|\frac{\partial}{\partial x} \arccos x_r(x, y)\right|$ and $\left|\frac{d}{dy} \arccos y_r(y)\right|$, respectively. For $(m, n) = (4, 0)$ this is not true because the approximation (13) does not hold for $n = 0$.

The classical base functions (CLAS) in Fig. 2 display a different spatial behavior. No analytical approximation of these base functions is known. However, the zero crossings of the traditional base functions are more evenly distributed over the image region; also, their local amplitude and frequency are largely position-independent. As mentioned before, this means that they should have better coding properties.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Section II shows that the computational and memory requirements of WS SIC are far lower than those of classical SIC, where non-separable base functions are used. However, the actual computational advantage of WS SIC is difficult to predict, because computational requirements depend strongly on the shapes of the image regions. Therefore, this section measures the actual computational requirements of WS SIC for compressing a real-world image for a range of typical coding parameters.

According to the theory of section III, it is to be expected that WS SIC will not achieve the same image quality as classical SIC at the same bit rate. This is because the local frequency of the WS base functions varies with position. This section evaluates how large the loss in image quality actually is. Also, it compares different types of warped WS bases in this respect.

All results in this section are obtained on the "cameraman" image, which is segmented using an improved version of the edgmentation algorithm [19] which produces exactly 300 regions in this case. In the different texture compression methods investigated in this paper, the number of base functions b in a region is selected as $b = \min(\alpha p/100, n_b)$, where p is the number of pixels in the region, α a number in the range $0 < \alpha \leq 100$ and n_b an upper limit on the number of base functions. This strategy for determining the number of base functions was also used in classical SIC [8], [20]. It assigns more coefficients to larger regions, but limits the computational requirements to a maximum level. This limitation is essential in classical SIC, but is not really needed in WS SIC, which is much faster.

The texture coefficients $A_{m,n}$ are uniformly quantized and entropy coded; the same quantizer is used in all evaluated methods. The quantization step Δ of the uniform quantizer is adapted to the number of region pixels p and to the number of base functions b according to the equation $\Delta = \sqrt{12\epsilon p/b}$, in which ϵ is a parameter. This equation tries to enforce the same rms quantization error in each region. Indeed, assuming that the quantization errors on the individual coefficients are independent, uniformly distributed random variables, this rms quantization error equals $b\Delta^2/(12p)$, which is region independent for the proposed value of Δ.

Table I lists the number of operations needed to compress the "cameraman" image using the classical base and two types of polynomial WS bases for different values of α and n_b. The classical base is computed using two methods: PRO and GS. The two WS bases differ in the set of base functions they compute: the POL method employs the first b naturally ordered base functions, while the POL' method uses the first b rectangularly ordered functions. Table I also lists the compression ratios and rms approximation errors which are obtained for the different values of α and n_b (and for $\epsilon = 15$).

Table I shows that the POL base can be computed much faster than the classical base CLAS; the speedup factor is pretty much independent of α. For $n_b = 16$ to $n_b = 64$, it ranges from 8 to 28 if the classical base is computed by PRO and from 12 to 85 if it is computed by GS. The POL' base can be computed 1.3 to 1.5 times faster than POL, but its computational advantage decreases with increasing n_b. Note that Table I shows that the different methods differ very little in compression ratio. Therefore, the table gives an idea of the number of operations required by the different texture coding methods to achieve a given compression ratio.

The memory requirements are not shown in Table I. However, for $n_b = 16$ to $n_b = 64$, they range from 274Kbyte to 300Kbyte for POL and from 263Kbyte to 284Kbyte for POL'; for the classical base, they range from 887Kbyte to 1.65Mbyte when using PRO and from 1.26Mbyte to 5.04Mbyte when using GS. These figures are the amount of memory required to process the largest region; in progressive coding the memory requirements would be many times larger. The figures show that the required memory is 3 to 6 lower for WS SIC than for classical SIC using PRO

Fig. 3. Compressed images using the POL base and the classical base. In both cases, $\alpha = 20$, $n_b = 32$, $\epsilon = 15$ and CR$\approx$ 25.

and 4 to 17 times lower than for classical SIC using GS.

The fact that the computational and memory requirements of both classical and WS SIC depend little on α for $n_b \leq 64$ indicates that most resources are spent on coding large regions: indeed, increasing α increases b in small regions only, because in large regions $b = n_b$.

Because the computational requirements increase rapidly with n_b, this parameter needs to be kept small in classical SIC, especially in conjunction with GS orthogonalization. Unfortunately, this limits the accuracy with which large regions can be represented. For this reason, large regions had to artificially split in earlier classical SIC methods [8]. This leads to a larger overhead for describing the segmented image and can adversely affect the image quality. No such problems arise in WS-based SIC, where the complexity only moderately increases with n_b and where no restriction on b is needed.

The above results clearly demonstrate the computational superiority of the WS bases. On the other hand, the rms-errors listed in Table I show that for the same number of base functions the rms-error is consistently lowest in classical SIC. This agrees with the theoretical predictions of section III. More precisely, the rms-error of the POL base is 3% to 6% higher than that of the classical base. The rms-error of the POL' base is always slightly higher than that of the POL base. Of course, these rms-errors can only be fairly compared at the same compression ratio (CR). However, as mentioned before, the differences in compression ratio between the different methods are insignificant. Table I lists the values of CR in function of α and n_b, assuming that coding the region contours requires about 1.6 bits per contour pixel. This assumption is somewhat pessimistic though, as some contour coding techniques require only 1.2 bits per contour pixel [7].

Figure 3 shows the classical and POL images for a compression ratio of about 25, which corresponds to about 0.32 bits per pixel. (The original, segmented image is shown in Fig. 4.) The subjective quality of these images is comparable, even though the rms-error of the POL base is significantly higher. This might be explained as follows: due to the low number of base functions per regions, the approximated texture is very smooth in all regions and is "uninformative" to the human eye, in comparison to the edge information. Therefore, small differences in the texture are not easily noticeable. Of course, the smoothness of the compressed images is largely the result of the small value of n_b. From a subjective image quality point of view many more base functions should be computed in, e.g., the "grass" areas. In that case, the subjective quality of the POL image might be noticeably lower. Nevertheless, the results show that WS SIC is a viable alternative for classical SIC when b is low. On the other hand, the complexity of classical SIC prohibits using high values of b, which makes WS SIC a clear winner in all cases.

Finally, we experimentally compare the POL base with the COSA and COSB bases introduced in section III-B. In this comparison we put $n_b = +\infty$, i.e., we do not limit the maximum number of base functions per region. Table II shows the rms-errors and the compression ratios for different values of α and ϵ. The values of ϵ are larger than that used in Table I to compensate for the larger number of base functions; without this compensation, the CR

Fig. 4. Original segmented image and compressed images using three different bases. In each case, $\alpha = 20$ and $\epsilon = 100$ and CR$\approx$ 21.

would be rather low. The computational complexity and memory requirements are the same for all three bases and range from 12Mflop and 410Kbyte for $\alpha = 5$ to 27Mflop and 578Kbyte for $\alpha = 20$.

Table II shows that the differences in CR and in rms-error between the different bases are low and that the POL base consistently produces the lowest rms-error. Figure 4 displays the compressed images for $\alpha = 20$ and $\epsilon = 100$. It shows that POL and COSB produce a comparable image quality. The COSB image is slightly better in the "Sky" area, where the POL base introduces larger discontinuities than the COSB base. This agrees with the discussion in section III-B. The COSA-compressed image is of slightly lower quality than the other images. This is mainly due to the ringing-like artifacts between the legs of the tripod.

The methods evaluated above could be improved in sev-

TABLE II
COMPARISON OF THE POL, COSA AND COSB BASES ON THE "CAMERAMAN" IMAGE.

		Rms-error			CR		
α	ϵ	POL	COSA	COSB	POL	COSA	COSB
5	50	13.9	14.0	14.0	31.4	32.1	32.4
5	100	14.3	14.4	14.4	35.4	34.7	35.3
5	200	14.8	14.9	14.9	38.6	37.3	37.8
10	50	12.7	12.8	12.9	25.6	25.4	24.8
10	100	13.3	13.4	13.4	29.0	28.8	28.7
10	200	13.9	14.0	14.1	33.5	33.1	33.8
20	50	11.3	11.4	11.4	17.7	17.8	17.9
20	100	11.9	12.1	12.1	20.6	20.8	20.7
20	200	12.8	12.9	12.9	26.4	25.9	26.1

eral ways. For instance, an analysis of the region's shape could be used to switch between the POL base and its dual (see section III-A). Also, different types of bases could be used in different regions. For instance, each region could be compressed using both the POL and the COSB base and the best approximation would be retained. Of course, this would double the complexity of the coder (but not of the decoder). Possibly, the selection of the optimal base could be guided by the edge strength. However, this remains to be investigated.

V. CONCLUSION

This paper introduces weakly separable polynomial orthonormal bases for SIC and describes their structural and computational properties. The WS base functions are products of two component functions, which can be computed separately by three-term recurrences. All SIC computations can be expressed in terms of the component functions and the component functions can be computed very quickly. As a result, WS SIC is much faster and requires less memory than traditional SIC.

The paper introduces the class of warped WS bases of which polynomial WS bases are a special case. It presents an analytical approximation of these functions and uses this approximation to predict their properties.

The paper experimentally compares the coding and computational properties of WS and classical bases. The results show that the former can be computed 8 to 40 times faster than the latter. On the other hand, WS SIC introduces larger rms coding errors, but this is not reflected into the subjective image quality which is pretty much the same. A comparison of polynomial and cosine WS bases finally shows that the image obtained using the polynomial base has the best over-all quality. However, the differences are small and one of the cosine bases introduces smaller errors in regions bordered by weak edges.

VI. ACKNOWLEDGMENTS

The author would like to thank the Belgian National Fund for Scientific Research (NFWO) for financially supporting this work through a mandate of "postdoctoral research fellow" and through the NFWO projects 9.0051.93 and 9.4594.93. He is also grateful to the anonymous reviewers.

REFERENCES

[1] M. Kunt, M. Bénard, and R. Leonardi, "Recent results in high-compression image coding," *IEEE Trans. Circ. Syst.*, vol. 34, pp. 1306–1336, Nov. 1987.
[2] G. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, pp. 30–44, Apr. 1991.
[3] R. Clarke, *Transform Coding of Images.* Londen: Academic Press, 1985.
[4] F. Marqués, J. Cunillera, and A. Gasull, "Hierarchical segmentation using compound Gauss-Markov random fields," in *Proceedings of ICASSP-92*, vol. 3, pp. 53–56, Mar. 1992. San Fransisco.
[5] M. Bister, J. D. Becker, C. Vanhove, G. Demonceau, and J. Cornelis, "4-d segmentation of cardiac SPECT images," in *Proceedings of MEDICON-92*, vol. 2, pp. 939–942, July 1992. Capri,Italy.
[6] D. Moran, T. Vlachos, O. Morris, and A. Constantinides, "Region image coding techniques," in *IEE Colloquim on Low Bit Rate Image Coding*, May 1990. London.
[7] M. Eden and M. Kocher, "On the performance of a contour coding algorithm in the context of image coding–part i: Contour segment coding," *Signal Processing*, vol. 8, pp. 381–386, July 1985.
[8] M. Gilge, T. Engelhardt, and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform," *Signal Processing: Image Communication*, vol. 1, pp. 153–180, Oct. 1989.
[9] M. Gilge, *Regionorientierte Transformationcodierung in der Bildkommunikation.* PhD thesis, Rheinisch-Westfälisch Technischen Hochschule, Aachen, 1989. In German.
[10] A. Kaup and T. Aach, "A new approach towards description of arbitrarily shaped image segments," in *Workshop Notes of the IEEE International Workshop on Intelligent Signal Processing and Communication Systems*, pp. 543–553, Mar. 1992.
[11] U. Franke, R. Mester, and T. Aach, "Constrained iterative restoration techniques: A powerful tool in region oriented texture coding," in *Signal Processing IV: Theories and Applications*, (Amsterdam), pp. 1145–1148, North Holland, 1988. Proceedings of EUSIPCO-88, Grenoble, Frankrijk.
[12] W. Philips, "A fast algorithm for the generation of orthogonal base functions on an arbitrarily shaped region," in *Proceedings of ICASSP92*, vol. 3, pp. 421–424, Mar. 1992. San Fransisco.
[13] W. Philips, "A fast algorithm for orthogonalizing polynomials on an arbitrarily shaped region," *IEEE Signal Processing Letters.* Submitted.
[14] W. Philips, "On the properties of base functions that are orthogonal on a non-rectangular region," in *IEEE Benelux and PRORISC Workshop on Circuits, Systems and Signal Processing*, pp. 105–110, Apr. 1992. HouthalenBelgium.
[15] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 59–62, Feb. 1995.
[16] W. Philips, "Orthogonal base functions on a discrete two-dimensional region," Tech. Rep. DG 91-20, ELIS, RUG, Gent, 1992.
[17] W. Philips, "Coding properties of time-warped polynomial transforms," *Signal Processing*, vol. 37, pp. 229–242, May 1994.
[18] W. Philips, "Weakly separable bases for fast segmented image coding," in *Hybrid Image and Signal Processing IV*, vol. 2238, pp. 153–163, Apr. 1994. SPIE Conference, Orlando.
[19] R. Deklerck, J. Cornelis, and M. Bister, "Segmentation of medical images," *Image and Vision Computing*, vol. 11, pp. 486–503, Oct. 1993.
[20] W. Philips and C. Christopoulos, "Fast segmented image coding using weakly separable bases," in *Proceedings of ICASSP94*, vol. 5, pp. 345–348, Mar. 1994. Adelaide.
[21] M. Biggar, O. Morris, and A. Constantinides, "Segmented-image coding: Performance comparison with the discrete cosine transform," *IEE Proceedings*, vol. 135, Pt. F, pp. 121–132, Apr. 1988.
[22] M. Cermelli, F. Lavagetto, and M. Pampolini, "A fast algorithm for region-oriented texture coding," in *Proceedings of ICASSP94*, pp. 289–292, 1994.
[23] G. Golub and C. Van Loan, *Matrix Computations.* John Hopkins University Press, second ed., 1991.

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

[25] M. Gilge, "Region-oriented transform coding (ROTC) of images," in *Proc. ICASSP-90*, pp. 2245–2248, Apr. 1990. Albuquerque, New Mexico.

[26] C. Gu and M. Kunt, "Contour simplification by a new nonlinear filter for region-based coding," in *Visual Communications and Image Processing, VICIP'94*, Sept. 1994. Chicago.

[27] R. Schafer, R. Mersereau, and M. Richards, "Constrained iterative restoration algorithms," *IEEE Proceedings*, vol. 69, pp. 432–450, Apr. 1981.

[28] V. Tom, T. Quatieri, M. Hayes, and J. McClellan, "Convergence of iterative nonexpansive signal reconstruction algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, pp. 1052–1058, Oct. 1981.

[29] D. Jackson, "Problems of closest approximation on a two-dimensional region," *American Journal of Mathematics*, vol. 60, pp. 436–446, 1938.

[30] H. Chang and S. Chen, "Progressive image transmission by dynamic binary thresholding," *Optical Engineering*, pp. 586–595, Feb. 1994.

[31] A. Kaup and T. Aach, "Region-based image coding using functional approximation," in *Picture Coding Symposium Proceedings*, Mar. 1993.

[32] A. Kaup and T. Aach, "Segment-oriented coding of textured images based on successive approximation," in *International Symposium on Speech, Image Processing and Neural Networks Proceedings*, pp. 197–200, Apr. 1994.

[33] O. Kwon and R. Chellappa, "Segmentation-based image compression," *Optical Engineering*, vol. 32, pp. 1581–1587, July 1993.

[34] D. Graham, "Image transmission by two-dimensional contour coding," *IEEE Proceedings*, pp. 336–346, Mar. 1967.

[35] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *IEEE Proceedings*, vol. 73, pp. 549–574, Apr. 1985.

[36] G. Szegö, *Orthogonal Polynomials*, vol. 23 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Fourth ed., 1975.