L_2 optimal oracles and compression strategies for semiorthogonal wavelets

L_2 Optimal Oracles and Compression Strategies for Semiorthogonal Wavelets

Markus Gross

Computer Science Department
Swiss Federal Institute of Technology (ETH)
Zürich, Switzerland
Computer Science Department Internal Report No. 254
Version 1.0, December 20, 1996

ABSTRACT

The following paper discusses the problem of optimal coefficient rejection for approximations with semiorthogonal wavelets. A rejection strategy, a so-called oracle, computes the significance of an individual wavelet coefficient and is essential for lossy compression schemes. The oracles proposed in this paper are based on the L_2 norm of the underlying functional space. The report starts with an error analysis in semiorthogonal settings and presents a scheme to compute the fractional energy in each complement space of a given iteration m. However, as opposed to orthogonal settings the minimization of the overall approximation error while rejecting K out of N wavelets cannot be solved by sorting of the coefficients. Moreover, geometric interpretations of this global optimization task relate to common combinatorial problems of linear algebra and computational geometry. As a result, we propose a greedy construction scheme for an L_2 oracle, which computes the conditional significance of individual wavelet coefficients and which operates in quadratic time. The compression strategy is first derived for 1D functions and extended to 2D nonstandard tensor product functions. Some results on real world data sets compare the oracle with standard rejection schemes for orthogonal bases.

1 LIST OF SYMBOLS

For convenience, the following table presents all symbols used in the subsequent analysis.

Table 1: Notations

m	: $0 \ldots M$	iteration
i, j	: $1 \ldots N/2^m$	indices for wavelets and scaling functions
k, l	:	indices
p, q	:	indices
V_m	:	approximation space of iteration m
$\tilde{V}_m$	:	dual approximation space
W_m	:	orthogonal complement space
$\tilde{W}_m$	:	dual orthogonal complement space
ϕ_i^m	:	scaling function in space V_m
$\tilde{\phi}_i^m$	:	dual scaling function in space V_m
ψ_i^m	:	wavelet in W_m
$\tilde{\psi}_i^m$	:	wavelet in $\tilde{W}_m$
$\mathbf{\Phi}^m$	:	inner product matrix $\langle \phi_i^m, \phi_j^m \rangle$
$\mathbf{\Psi}^m$	:	inner product matrix $\langle \psi_i^m, \psi_j^m \rangle$
$f(x)$	:	finite energy function
$\| \ \|_{L_2}$	:	L_2-norm
c_i	:	scaling function coefficient
d_i	:	wavelet coefficient
$\mathbf{c}$	:	scaling function coeff. vector
$\mathbf{d}$	:	wavelet coeff. vector
$\mathbf{C}$	:	scaling function coeff. matrix
$\mathbf{D}$	:	wavelet coeff. matrix
Δf^m	:	difference function in space W_m
$\otimes$	:	tensor product operator
$\bullet$	:	dot product operator
$\tilde{c}_i$	:	dual scaling function coefficient
$\tilde{d}_i$	:	dual wavelet coefficient
$\tilde{\mathbf{c}}$	:	dual scaling function coeff. vector
$\tilde{\mathbf{d}}$	:	dual wavelet coeff. vector
$\tilde{\underline{d}}_{ij}$	:	dual elements of wavelet coefficient matrix

2 METRIC AND ENERGY IN SEMIORTHOGONAL SETTINGS

This section elaborates a computational scheme for the L_2 approximation error in semiorthogonal settings. It is derived that the norm can be computed from a basis transform of the wavelet coefficient vector. To this end, let‘s first recall finite dimensional orthogonal approximations and let‘s assume $\{\phi_i(x)\}_{i=1,\ldots,N}$ to be some bases, which expands a function $f(x)$ as

$$f(x) = \sum_{i=1}^{N} c_i \cdot \phi_i(x) \tag{1}$$

where c_i denote the coordinates in some functional space V spanned by $\{\phi_i\}$.

Truncating the sum in (1) by an upper bound $K < N$ affects the approximation and results in $f'(x)$. The L_2 error is determined by the linear combination of the rejected basis functions

$$\| f(x) - f'(x) \|_{L_2}^2 = \left\| \sum_{i=K+1}^{N} c_i \cdot \phi_i(x) \right\|_{L_2}^2 \tag{2}$$

and can be rewritten as

$$\left\langle \sum_{i=K+1}^{N} c_i \cdot \phi_i(x), \sum_{i=K+1}^{N} c_i \cdot \phi_i(x) \right\rangle = \sum_{i=K+1}^{N} |c_i|^2 \tag{3}$$

< , > stands for the inner product operator.
That is, the squared magnitude of the coefficient c_i corresponds precisely to the fraction of energy provided by the corresponding basis ϕ_i. Note, that the orthogonality $\langle \phi_i, \phi_j \rangle = \delta_{ij}$ cancels out all intermediate terms in (3) and simplifies the relation.
For convenience, we will refer to the terminus „*energy*“ as the squared norm of a function throughout the paper. Hence, given a function *f(x),* it’s energy is computed by

$$\| f(x) \|_{L_2}^2 = \langle f(x) | f(x) \rangle = \int_{-\infty}^{\infty} f(x) \cdot f(x) dx \tag{4}$$

Needless to say that all functions whose upper inner product is limited belong the Hilbert space of square integrable or finite energy functions which conforms to a bandlimitation in Fourier space. Along these lines the optimal L_2 approximation i.e. compression problem can be formulated in terms of two different, but similar variants:

Variant I — *Find a subset of K out of N coefficients, which minimizes the overall approximation error induced by rejection of the associated basis functions. These hold for optimal compression of a predefined ratio N : (N-K).*

Variant II — *Reject a maximum number of coefficients while keeping the approximation error below a predefined bound E_b.*

Obviously, globally optimal compression can be achieved by sorting coefficients according to their magnitude and by rejection of the K smallest ones [2]. This strategy provides an L_2 optimal oracle for orthogonal bases, which is commonplace in wavelet applications. It can be computed in $O(N\log(N))$ using fast sorting algorithms [7].
Unfortunately, oracle construction becomes more difficult for semiorthogonal wavelets, as the ones used in this paper. Following the fundamental relations of semiorthogonality [1], [5] or [6] we assume $\phi_i^m(x)$ to be a scaling function of approximation space V_m and $\psi_i^m(x)$ to be the corresponding wavelet spanning the orthogonal complement space W_m. Semiorthogonality states orthogonality between two different complement spaces $W_m \perp W_n, \quad m \neq n$ and in addition that $V_m \perp W_m$. From there, it follows immediately that

$$\langle \phi_i^m, \psi_j^m \rangle = 0 \qquad \langle \psi_i^m, \psi_j^n \rangle = 0, \forall i, j \tag{5}$$

Similar relations hold for the duals. However in an individual space W_m, we generally obtain

$$\langle \psi_i^m, \psi_j^m \rangle \neq 0 \tag{6}$$

Let's now assume our space V_0 to have some finite dimension N and let's further assume a dyadic scaling of the multiresolution analysis bounding the dimension of any W_m and V_m to $N/2^m$ *, $m = 1,\ldots,M$. Thus, we rewrite the expansion of $f(x)$ as

$$\begin{aligned} f(x) &= \sum_{i=1}^{N} c_i^0 \phi_i^0(x) = \sum_{i=1}^{N/2^M} c_i^M \phi_i^M(x) + \sum_{m=1}^{M} \sum_{i=1}^{N/2^m} d_i^m \psi_i^m(x) \\ &= f^M(x) + \sum_{m=1}^{M} \Delta f^m(x) \\ &= f^M(x) + \Delta f(x) \end{aligned} \tag{7}$$

with the coefficients c_i^m and d_i^m to be computed by inner products of the dual wavelets and scaling functions:

$c_i^m = \langle f, \tilde{\phi}_i^m \rangle \quad , \quad d_i^m = \langle f, \tilde{\psi}_i^m \rangle$.

The L_2 norm of the difference of the two approximations $f(x)$ and $f^M(x)$ yields

$$\| f(x) - f^M(x) \|_{L_2}^2 = \| \Delta f(x) \|_{L_2}^2 = \left\langle \sum_{m=1}^{M} \sum_{i=1}^{N/2^m} d_i^m \psi_i^m(x), \sum_{m=1}^{M} \sum_{i=1}^{N/2^m} d_i^m \psi_i^m(x) \right\rangle \tag{8}$$

*To simplify notion, we carry out all analysis for unbounded wavelets on the real axis i.e. for periodic end conditions. Hence, the dimension of the nested subspaces V_m and W_m are given by $N/2^m$. For spline wavelets bounded to the interval, an offset has to be added to account for the knot multiplicity at the end points. This offset is given by *order* - 1 [5], [6].

Recalling the properties of semiorthogonality, the inner products of wavelets between different resolutions $m \neq n$ are canceled out and we obtain

$$\|\Delta f(x)\|_{L_2}^2 = \sum_{m=1}^{M} \left\langle \sum_{i=1}^{N/2^m} d_i^m \psi_i^m(x), \sum_{i=1}^{N/2^m} d_i^m \psi_i^m(x) \right\rangle = \sum_{m=1}^{M} \|\Delta f^m(x)\|_{L_2}^2 \tag{9}$$

From (9) we observe that the squared error between different resolutions can be summed up, a result which follows immediately from the orthogonality of two different complement spaces $W_m \perp W_n$. Obviously, when searching for an oracle to compute the significance of a single wavelet ψ_i^m, it is sufficient to analyze its relationship to all $\psi_j^m \in W_m$. To do so, let's formalize the norm of the difference signal $\Delta f^m(x)$:

$$\begin{aligned} \|\Delta f^m(x)\|_{L_2}^2 &= \left\langle \sum_{i=1}^{N/2^m} d_i^m \psi_i^m(x), \sum_{i=1}^{N/2^m} d_i^m \psi_i^m(x) \right\rangle \\ &= \sum_{i=1}^{N/2^m} \sum_{j=1}^{N/2^m} d_i^m d_j^m \cdot \int \psi_i^m(x) \psi_j^m(x) dx \\ &= \sum_{i=1}^{N/2^m} \sum_{j=1}^{N/2^m} d_i^m d_j^m \cdot \langle \psi_i^m(x), \psi_j^m(x) \rangle \\ &= 2 \sum_{i=1}^{N/2^m} \sum_{j>i}^{N/2^m} d_i^m d_j^m \cdot \langle \psi_i^m(x), \psi_j^m(x) \rangle + \sum_{i=1}^{N/2^m} (d_i^m)^2 \end{aligned} \tag{10}$$

The last step follows from the symmetry of the relation. Note furthermore, that (10) can be rewritten in matrix form by introducing the inner product matrix Ψ^m

$$\Psi^m = \begin{bmatrix} \langle \psi_1^m, \psi_1^m \rangle & \cdots & \langle \psi_1^m, \psi_{N/2^m}^m \rangle \\ \vdots & & \vdots \\ \langle \psi_{N/2^m}^m, \psi_1^m \rangle & \cdots & \langle \psi_{N/2^m}^m, \psi_{N/2^m}^m \rangle \end{bmatrix} = \begin{bmatrix} 1 & \cdots & \langle \psi_1^m, \psi_{N/2^m}^m \rangle \\ \vdots & 1 & \vdots \\ \langle \psi_{N/2^m}^m, \psi_1^m \rangle & \cdots & 1 \end{bmatrix} \tag{11}$$

Let $\mathbf{d}^m$ be the vector of wavelet coefficients in space W_m, we compute the duals as

$$\tilde{\mathbf{d}}^m = \Psi^m \cdot \mathbf{d}^m \tag{12}$$

As a result (10) collapses into the following dot product of two vectors

$$\left\|\Delta f^m(x)\right\|_{L_2}^2 = \mathbf{d}^m \bullet \tilde{\mathbf{d}}^m = \sum_{i=1}^{N/2^m} d_i^m \cdot \tilde{d}_i^m = \mathbf{d}^m \bullet (\Psi^m \cdot \mathbf{d}^m) \tag{13}$$

The relation from above states that the energy of difference function $\Delta f^m(x)$ can be computed by a dot product of the coefficient vector with its copy projected into dual space.

Note that the sparsity of the matrix Ψ^m enables to compute the transform in linear time for most wavelets.

Similar relations hold for the scaling functions: The energy of the low-pass approximation $f^M(x)$ is computed by the following dot product

$$\left\|f^M(x)\right\|_{L_2}^2 = \mathbf{c}^M \bullet \tilde{\mathbf{c}}^M = \sum_{i=1}^{N/2^M} c_i^m \cdot \tilde{c}_i^m = \mathbf{c}^M \bullet (\Phi^M \cdot \mathbf{c}^M) \tag{14}$$

where Φ^M stands for the inner product matrix of the scaling functions:

$$\Phi^M = \begin{bmatrix} \langle \phi_1^M, \phi_1^M \rangle & \cdots & \langle \phi_1^M, \phi_{N/2^M}^M \rangle \\ \vdots & & \vdots \\ \langle \phi_{N/2^M}^M, \phi_1^M \rangle & \cdots & \langle \phi_{N/2^M}^M, \phi_{N/2^M}^M \rangle \end{bmatrix} = \begin{bmatrix} 1 & \cdots & \langle \phi_1^M, \phi_{N/2^M}^M \rangle \\ \vdots & 1 & \vdots \\ \langle \phi_{N/2^M}^M, \phi_1^M \rangle & \cdots & 1 \end{bmatrix} \tag{15}$$

Thus the energy of our function *f*(*x*) is computed by

$$\left\|f(x)\right\|_{L_2}^2 = \mathbf{c}^M \bullet \tilde{\mathbf{c}}^M + \sum_{m=1}^{M} \mathbf{d}^m \bullet \tilde{\mathbf{d}}^m \tag{16}$$

3 COMPUTATION OF APPROXIMATION ERRORS

This section addresses the conditional behavior of the approximation error upon rejection of individual wavelet coefficients. Due to the orthogonality of different complement spaces and following (8) and (9) it is sufficient to analyze the error norm of a single space W_m. In order to derive an incremental method we assume *K* out of $N/2^m$ coefficients to vanish. The approximation error is determined by the following relation

$$\begin{aligned} \left\|\Delta f^m(x) - \Delta f'^m(x)\right\|_{L_2}^2\Big|_{Rej(K)} &= \left\langle \sum_{i \in Rej(K)} d_i^m \psi_i^m(x), \sum_{i \in Rej(K)} d_i^m \psi_i^m(x) \right\rangle \\ &= \sum_{i \in Rej(K)} \sum_{j \in Rej(K)} d_i^m d_j^m \cdot \langle \psi_i^m(x), \psi_j^m(x) \rangle \end{aligned} \tag{17}$$

where $\Delta f'^m(x)$ represents the residual approximation and *Rej*(*K*) denotes the set of all coefficients being rejected from the initial transform. In a next step we compute how the upper error behaves when

rejecting an arbitrary $d_k^m \neq 0$ assuming K coefficients already being rejected in previous procedures. That is, we compute an expression for the error increment invoked by a single coefficient.

$$\left\|\Delta f^m(x) - \Delta f'^m(x)\right\|_{L_2}^2\Big|_{Rej(K,\,k)} = \left\|\Delta f^m(x) - \Delta f'^m(x)\right\|_{L_2}^2\Big|_{Rej(K)} + (d_k^m)^2 + 2 \sum_{i \in Rej(K)} d_k^m d_i^m \cdot \langle \psi_k^m(x), \psi_i^m(x) \rangle \tag{18}$$

Equation (18) expresses the dependence of the error on its increment from the rejection set. We will refer to it as the *conditional approximation error* in all subsequent discussions of the paper. The factor of 2 follows immediately from the symmetry of the inner product matrix. Apparently, the conditional increment is computed by adding one row and one column to the matrix type structure representing the double summation of (17), such as depicted in fig. 1. In summary, the error can be updated by adding the products of the coefficient d_k^m and the elements of the rejection set times the associated entry of the inner product matrix. Note that this error can be considered as a score, which reflects the *conditional importance* of an individual coefficient.

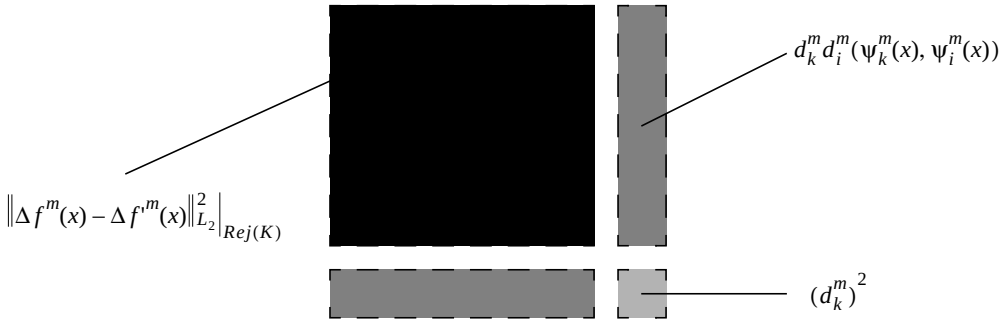

Figure 1: Illustration of the conditional incremental approximation error

Another issue deserves further attention: In signal processing applications, the quality of the residual approximation is often measured by the so-called signal-noise ratio, which in essence conforms to the ratio between the error energy and the energy of the residual approximation

$$S/N = \frac{\left\|f^M(x)\right\|_{L_2}^2 + \sum_{m=1}^{M} \left\|\Delta f'^m(x)\right\|_{L_2}^2}{\sum_{m=1}^{M} \left\|\Delta f^m(x) - \Delta f'^m(x)\right\|_{L_2}^2}\Bigg|_{Rej(K)} = \frac{\mathbf{c}^M \bullet \tilde{\mathbf{c}}^M + \sum_{m=1}^{M} \sum_{i \notin Rej(K)} \sum_{j \notin Rej(K)} d_i^m d_j^m \cdot \langle \psi_i^m(x), \psi_j^m(x) \rangle}{\sum_{m=1}^{M} \sum_{i \in Rej(K)} \sum_{j \in Rej(K)} d_i^m d_j^m \cdot \langle \psi_i^m(x), \psi_j^m(x) \rangle} \tag{19}$$

We assume that all scaling functions are preserved.

The derived relations represent an essential step towards the development of an oracle.They allow to predict how the approximation error changes when rejecting an individual wavelet coefficient under the precondition that *K* other coefficients had been rejected earlier. Based on this fundamental relationship, it is possible to develop an incremental rejection algorithm, which will be explained in detail in section 5.

4 A GEOMETRIC INTERPRETATION OF L_2 OPTIMAL ORACLES

At this point in time, we recall our initial problem of finding the set of *K* out of $N/2^m$ coefficients while preserving most of the energy of the residual approximation of $\Delta f^m(x)$. Obviously, unlike orthogonal settings, the magnitude $|d_i|$ does not compute the fractional error imposed by the associated wavelet. Even worse, brut force combinatorial analysis states, that there are $\binom{K}{N/2^m}$ possibilities to combine *K* coefficients to find the global optimum for each step *m*. Building a bridge to vector spaces in linear algebra the problem can be reformulated as follows:

Let $\{d_i \cdot \vec{\psi}_i\}_{i=1,N}$ be a set of *N* nonorthogonal vectors of unit length and scaled by some factor d_i in some finite-dimensional vector space. We represent a vector $\Delta f = \sum_{i=1}^{N} d_i \vec{\psi}_i$ as illustrated in fig. 2 for *N*=3. The problem is now to find a subset of *K* vectors, whose linear combination is closest to $\mathbf{0}^*$.

$$\left| \sum_{i=K+1}^{N} d_i \vec{\psi}_i \right| = min > 0 \tag{20}$$

Some special cases illustrate again the problem: If our *N* initial vectors are linear independent, the associated orthogonal space will have equal dimension *N*. Conversely, if all vectors are linear dependent, i.e. all $\{\vec{\psi}_i\}_{i=1\ldots N}$ point into the same direction, the space becomes one-dimensional. Let‘s now assume all coefficients d_i to be positive numbers, the second variant of the optimization problem has an interesting interpretation: *Find a maximum number out of a set of lines, which can be packed into another line of given length* E_b. In this case, the optimal solution can be found again in O(*N* log(*N*)) by sorting of the coefficients and relates closely to knapsack type problems [7].

5 A GREEDY COMPRESSION STRATEGY

As a result of the computational framework derived earlier, we propose a greedy oracle, which computes a minimum error rejection set of coefficients. In essence, the greedy oracle operates as follows: It first assigns an initial score to all wavelet coefficients of all iterations *m*. The score is defined by the overall conditional approximation error, which governs the oracle. In a second step, the oracle selects iteratively the coefficient with the minimum score, rejects it and recomputes the scores of all other coefficients. The iteration loop runs up to a predefined number of cycles *K* or up to a predefined error

*This correspondence can only be established, if the vectors are expandable by an orthogonal bases spanning the vector space. For nonorthogonal vectors, such as our wavelets, appropriate basis could be found by orthogonalization methods which are, however, beyond the scope of this report.

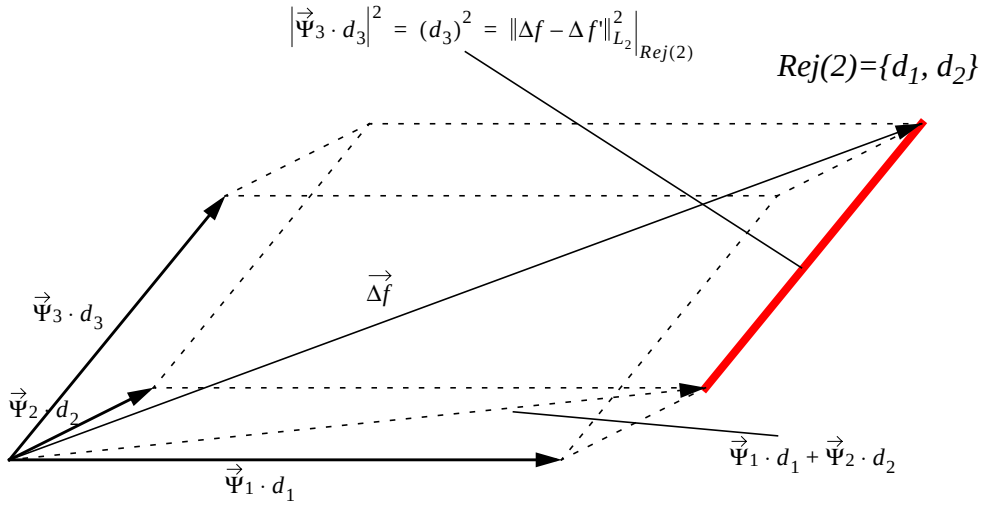

Figure 2: Geometric interpretation of optimal compression

bound E_b. Corresponding to equation (18) the score can be recomputed by an appropriate increment after each iteration, Thus we end up with a simple reject-and-update scheme for our oracle which can be implemented straightforwardly:

- **Initialize**: score[*i*,*m*] <- *d*[*i*,*m*]*d*[*i*,*m*]
- **For** *i* <- 1 **to** *K*:
- **For** *m*<- 1 **to** *M*:
- **Do**
- **Search:** coefficient | score[i_{rej}, m_{rej}]=min!=0;
- **Reject:** *d*[i_{rej},m_{rej}] <- 0;
- **Update:** score[*i*,*m*] <- score[*i*,*m*] + increment[*i*,*m*];
- **End**;

After each step, the score[*i*,*m*] of a coefficient d[*i*,*m*] represents the overall conditional approximation error when rejecting *d*[*i*,*m*]. The following C-style code fragments give the full oracle. In order to simplify notion, the C language convention was relaxed for some statements:

```
// minimizes ||f(x) - f'(x)||_{L_2}^2 |_{Rej(K)}   using a greedy algorithm; //
// corresponds to variant II of the compression problem //

E_b=const;                              // user defines global approximation error
                                           bound in [0..1] //
old_score=0;                            // stores current approximation error //

for (m=1,m<=M,m++)                   // channelwise energy computation for all wavelets d^m
   {                                       and scaling functions c^M //
   E_tot = E_tot + d^m • d~^m
   for (i=1, i<=N/2^m, i++)             // initial score assignment; score represents
         {                                 overall conditional approximation error //
         score[i,m]=d[i,m]*d[i,m];
         }
   }
E_tot = E_tot + c^M • c~^M
while (old_score/E_tot<E_b)
      {
      (i_rej, m_rej) <- Min(score);// find the index pair of the coefficient, which
                                      induces minimum conditional approximation
                                      error upon rejection (|score[i,m]|=min!=0)|//
      for (m=1,m<=M,m++)             // only wavelets are rejected //
         {                           // update score for all wavelets over all
         for (i=1, i<=N/2^m, i++)    // channels/subspaces //
            {
            if ((m==m_rej)&&(score[i,m]!=0))// if basis in same subspace* //
               score[i,m]+= 2*d[i,m]*d[i_rej,m_rej]* ψ[i, i_rej, m]
                           + score[i_rej,m_rej] - old_score;
            else if (score[i,m]!=0)
               score[i,m]+= score[i_rej,m_rej] - old_score; // update score with
            }                                                  error increment //
         }
      old_score=score[i_rej,m_rej];    // store current approximation error //
      score[i_rej,m_rej]=0;            // zero score indicates rejected wavelet //
   }
```

This fragment illustrates that the oracle can be implemented efficiently. In this implementation the oracle operates over all iterations and rejects the wavelet or scaling function with the minimum score. Note, that the score stores the overall conditional approximation error. At a first glance, it might be somewhat difficult to understand the incremental update scheme. However, when thoroughly analyzing (18) and taking a view from any coefficient $d_k^m \notin Rej(K)$ the sum $2 \sum_{i \in Rej(K)} d_k^m d_i^m \cdot \langle \psi_k^m(x), \psi_i^m(x) \rangle$ can be interpreted as follows: Each time a coefficient d_i^m with $i \neq k$ is added into the rejection set, the *conditional approximation error* for d_k^m changes by $2 \cdot d_k^m d_i^m \cdot \langle \psi_k^m(x), \psi_i^m(x) \rangle$ which corresponds precisely to our update term. For coefficients belonging to a different subspace orthogonality cancels out the inner product and the term vanishes.

*An extension of the oracle to scaling function rejection can be accomplished straightforwardly using little more control structures in the code.

Similar code fragments can be given for the variant I of our compression problem. In this case the oracle is controlled by a maximum number of rejected coefficients *K*.

```
// minimizes ||f(x) - f'(x)||_{L_2}^2 |_{Rej(K)}   using a greedy algorithm
   corresponds to variant I of the compression problem //

K=const;                              // user defines a maximum number of rejected
                                         coefficients/compression ratio K<N //
old_score=0;                          // stores current approximation error //

for (m=1,m<=M,m++)                 // search for vanishing coefficients in the initial
   {                                  setting //
   for (i=1, i<=N/2^m, i++)           // initial score assignment; score represents
         {                               overall conditional approximation error //
         score[i,m]=d[i,m]*d[i,m];
         if (score[i,m]==0)
            {
            K--;
            if (K==0)
               exit(1);
            }
         }
   }
while (K>0)
      {
      (i_rej, m_rej) <- Min(score);// find the index pair of the coefficient, which
                                      induces minimum conditional approximation
                                      error upon rejection (|score[i,m]|=min!=0)//
      for (m=1,m<=M,m++)           // only wavelets are rejected //
         {                         // update score for all wavelets over all
         for (i=1, i<=N/2^m, i++)  // channels/subspaces //
            {
            if ((m==m_rej)&&(score[i,m]!=0))// if basis in same subspace //
               score[i,m]+= 2*d[i,m]*d[i_rej,m_rej]* ψ[i, i_rej, m]
                            + score[i_rej,m_rej] - old_score;
            else if (score[i,m]!=0)
               score[i,m]+= score[i_rej,m_rej] - old_score; // update score with
            }                                                  error increment //
         }
      old_score=score[i_rej,m_rej];    // store current approximation error //
      score[i_rej,m_rej]=0;            // zero score indicates rejected wavelet //
      K--;
   }
```

In both cases the oracle operates in $O(N^2)$ assuming that **Min**(score) takes at most $O(N)$ operations to find the minimum, updating takes linear time as well and K or E_b depend linear on N.

6 2D NONSTANDARD TENSOR PRODUCT EXTENSIONS

This section extends the previous results to multiple dimensions. Therefore, we briefly review the fundamentals of nonstandard tensor product constructions of multidimensional wavelets. The separability of the construction allows us to come up with similar analyses, which are implemented by matrix operations. Interestingly enough, these schemes work for any dimensions and implementations con-

form to those of the one-dimensional setting from section 5.

In the non-standard tensor product constructions [2], the two-dimensional approximation spaces V_m^2 and orthogonal complements W_m^2 are defined as follows:

$$\begin{aligned} V_m^2 &= V_m \otimes V_m \\ W_m^{2,1} &= V_m \otimes W_m \\ W_m^{2,2} &= W_m \otimes V_m \\ W_m^{2,3} &= W_m \otimes W_m \end{aligned} \tag{21}$$

The two-dimensional wavelets and scaling functions are defined by

$$\phi_{ji}^m(x,y) = \phi_i^m(x) \cdot \phi_j^m(y) \tag{22}$$

with 3 types of wavelets

$$\begin{aligned} \psi_{ji}^{m,1}(x,y) &= \phi_i^m(x) \cdot \psi_j^m(y) \\ \psi_{ji}^{m,2}(x,y) &= \psi_i^m(x) \cdot \phi_j^m(y) \\ \psi_{ji}^{m,3}(x,y) &= \psi_i^m(x) \cdot \psi_j^m(y) \end{aligned} \tag{23}$$

If the one-dimensional relatives are semiorthogonal similar relationships follow for the 2D setting, since for instance

$$\begin{aligned} \langle \psi_{ji}^m(x,y) | \psi_{lk}^m(x,y) \rangle &= \int_{-\infty}^{\infty} \psi_i^m(x) \cdot \psi_k^m(x) dx \\ &\cdot \int_{-\infty}^{\infty} \psi_j^m(y) \cdot \psi_l^m(y) dy \\ &= (\langle \psi_i^m(x) | \psi_k^m(x) \rangle \cdot \langle \psi_j^m(y) | \psi_l^m(y) \rangle) \end{aligned} \tag{24}$$

From there we obtain immediately the orthogonality relations between the subspaces:

$$\begin{aligned} V_m^2 &\perp W_m^{2,t} \\ W_m^{2,1} &\perp W_m^{2,2} \\ W_m^{2,2} &\perp W_m^{2,3} \\ W_m^{2,1} &\perp W_m^{2,3} \\ W_m^{2,t} &\perp W_n^{2,s}, n \neq m | s,t = 1,2,3 \end{aligned} \tag{25}$$

Given the expansion of any two-dimensional finite energy function $f(x, y)$ as

$$\begin{aligned} f(x, y) &= \sum_{j=1}^{N} \sum_{i=1}^{N} c_{ji}^{0} \phi_{ji}^{0}(x, y) = \sum_{j=1}^{N/2^M} \sum_{i=1}^{N/2^M} c_{ji}^{M} \phi_{ji}^{M}(x, y) + \sum_{t=1}^{3} \sum_{m=1}^{M} \sum_{j=1}^{N/2^m} \sum_{i=1}^{N/2^m} d_{ji}^{m,t} \psi_{ji}^{m,t}(x, y) \\ &= f^M(x, y) + \sum_{t=1}^{3} \sum_{m=1}^{M} \Delta f^{m,t}(x, y) \\ &= f^M(x, y) + \Delta f(x, y) \end{aligned} \tag{26}$$

with the coefficients c_{ji}^{m} and $d_{ji}^{m,t}$ to be computed by inner products of the dual wavelets and scaling functions:

$$c_{ji}^{m} = \langle f, \tilde{\phi}_{ji}^{m} \rangle \quad , d_{ji}^{m,t} = \langle f, \tilde{\psi}_{ji}^{m,t} \rangle .$$

The L_2 norm of the difference of the two approximations $f(x, y)$ and $f^M(x, y)$ yields

$$\begin{aligned} &\| f(x, y) - f^M(x, y) \|_{L_2}^2 = \| \Delta f(x, y) \|_{L_2}^2 = \\ &\left\langle \sum_{t=1}^{3} \sum_{m=1}^{M} \sum_{j=1}^{N/2^m} \sum_{i=1}^{N/2^m} d_{ji}^{m,t} \psi_{ji}^{m,t}(x, y), \sum_{t=1}^{3} \sum_{m=1}^{M} \sum_{j=1}^{N/2^m} \sum_{i=1}^{N/2^m} d_{ji}^{m,t} \psi_{ji}^{m,t}(x, y) \right\rangle \end{aligned} \tag{27}$$

In analogy to the 1D version the properties of semiorthogonality of (25) cancel out all inner products of different resolutions $m \neq n$ and we obtain

$$\begin{aligned} &\| \Delta f(x, y) \|_{L_2}^2 = \\ &\sum_{t=1}^{3} \sum_{m=1}^{M} \left\langle \sum_{j=1}^{N/2^m} \sum_{i=1}^{N/2^m} d_{ji}^{m,t} \psi_{ji}^{m,t}(x, y), \sum_{j=1}^{N/2^m} \sum_{i=1}^{N/2^m} d_{ji}^{m,t} \psi_{ji}^{m,t}(x, y) \right\rangle = \sum_{t=1}^{3} \sum_{m=1}^{M} \| \Delta f^{m,t}(x, y) \|_{L_2}^2 \end{aligned} \tag{28}$$

Again, let's formalize the norm of the difference signal $\Delta f^{m,t}(x, y)$:

$$
\begin{aligned}
\left\|\Delta f^{m,t}(x,y)\right\|_{L_2}^2 &= \left\langle \sum_{j=1}^{N/2^m}\sum_{i=1}^{N/2^m} d_{ji}^{m,t}\psi_{ji}^{m,t}(x,y), \sum_{j=1}^{N/2^m}\sum_{i=1}^{N/2^m} d_{ji}^{m,t}\psi_{ji}^{m,t}(x,y)\right\rangle \\
&= \sum_{j=1}^{N/2^m}\sum_{i=1}^{N/2^m}\sum_{l=1}^{N/2^m}\sum_{k=1}^{N/2^m} d_{ji}^{m,t} d_{lk}^{m,t} \langle \psi_{ji}^{m,t}(x,y), \psi_{lk}^{m,t}(x,y)\rangle \\
&= \sum_{l=1}^{N/2^m}\sum_{j=1}^{N/2^m} \langle \psi_j^{m,t}(y), \psi_l^{m,t}(y)\rangle \cdot \sum_{k=1}^{N/2^m} d_{lk}^{m,t} \cdot \sum_{i=1}^{N/2^m} d_{ji}^{m,t}((\langle \psi_i^{m,t}(x), \psi_k^{m,t}(x)\rangle)) \\
&= \sum_{l=1}^{N/2^m}\sum_{j=1}^{N/2^m} \langle \psi_j^{m,t}(y), \psi_l^{m,t}(y)\rangle \cdot \sum_{k=1}^{N/2^m} d_{lk}^{m,t} \cdot \tilde{d}_{jk}^{m,t} \\
&= \sum_{l=1}^{N/2^m}\sum_{k=1}^{N/2^m} d_{lk}^{m,t} \cdot \sum_{j=1}^{N/2^m} \langle \psi_j^{m,t}(y), \psi_l^{m,t}(y)\rangle \cdot \tilde{d}_{jk}^{m,t} \\
&= \sum_{l=1}^{N/2^m}\sum_{k=1}^{N/2^m} d_{lk}^{m,t} \cdot \tilde{\tilde{d}}_{lk}^{m,t}
\end{aligned}
\tag{29}
$$

From (29) we observe that the computation of the approximation error can be separated into operations along the *x*- and *y*-axes respectively. This important property follows from the separability of the approach. To simplify notion we introduce the coefficient matrix $\mathbf{D}^{m,t}$

$$
\mathbf{D}^{m,t} = \begin{bmatrix} d_{11}^{m,t} & \cdots & d_{1,(N/2^m)}^{m,t} \\ \vdots & & \vdots \\ d_{(N/2^m),1}^{m,t} & \cdots & d_{(N/2^m),(N/2^m)}^{m,t} \end{bmatrix} \tag{30}
$$

Now, the equation collapses into a compact form provided by

$$
\left\|\Delta f^{m,t}(x,y)\right\|_{L_2}^2 = \mathbf{D}^{m,t} \bullet (((\mathbf{D}^{m,t}) \cdot \Psi^{m,t}(x))^T \cdot \Psi^{m,t}(y))^T = \mathbf{D}^{m,t} \bullet \tilde{\mathbf{D}}^{m,t} \tag{31}
$$

where $\tilde{\mathbf{D}}^{m,t}$ denotes the tensor product wise transform of the coefficient matrix. It is easy to see that some algebraic transform revealed that the relations for 1D scale straightforwardly to multiple dimensions.

Similar relations hold for the scaling functions: The energy of the low-pass approximation $f^M(x,y)$ is computed by the following matrix dot product

$$
\left\|f^M(x,y)\right\|_{L_2}^2 = \mathbf{C}^M \bullet (((\mathbf{C}^M) \cdot \Psi^{m,t}(x))^T \cdot \Psi^{m,t}(y))^T = \mathbf{C}^M \bullet \tilde{\mathbf{C}}^M \tag{32}
$$

where Φ^M is the inner product matrix of the scaling functions and $\mathbf{C}^M$ the coefficient matrix.
Thus the energy of our function *f*(*x*,*y*) is computed by

$$\| f(x, y) \|_{L_2}^2 = \mathbf{C}^M \bullet \tilde{\mathbf{C}}^M + \sum_{t=1}^{3} \sum_{m=1}^{M} \mathbf{D}^{m,t} \bullet \tilde{\mathbf{D}}^{m,t} \tag{33}$$

Computations of the approximation error can be accomplished by pursuing a similar strategy as in section 3. Due to the orthogonality constraints of (25) it is sufficient to consider an individual channel in *t*. A conditional error can be formalized by assuming K out of $N^2/2^{2m}$ *c*oefficients to vanish. Let $\Delta f'^{m,t}(x, y)$ denote the residual approximation

$$\begin{aligned} \left\| \Delta f^{m,t}(x, y) - \Delta f'^{m,t}(x, y) \right\|_{L_2}^2 \Big|_{Rej(K)} &= \left\langle \sum_{(j,i) \in Rej(K)} d_{ji}^{m,t} \psi_{ji}^{m,t}(x, y), \sum_{(j,i) \in Rej(K)} d_{ji}^{m,t} \psi_{ji}^{m,t}(x, y) \right\rangle \\ = \sum_{(j,i) \in Rej(K)} \sum_{(l,k) \in Rej(K)} & d_{ji}^{m,t} d_{lk}^{m,t} \cdot \langle \psi_j^{m,t}(y), \psi_l^{m,t}(y) \rangle \cdot \langle \psi_i^{m,t}(x), \psi_k^{m,t}(x) \rangle \end{aligned} \tag{34}$$

In order to determine an error increment, we reject an arbitrary $d_{qp}^{m,t}$ under the condition of *Rej*(*K*) and yield

$$\begin{aligned} \left\| \Delta f^{m,t}(x, y) - \Delta f'^{m,t}(x, y) \right\|_{L_2}^2 \Big|_{Rej(K,(q,p))} &= \left\| \Delta f^{m,t}(x, y) - \Delta f'^{m,t}(x, y) \right\|_{L_2}^2 \Big|_{Rej(K)} + (d_{qp}^{m,t})^2 \\ &+2 \cdot \sum_{(j,i) \in Rej(K)} d_{ji}^{m,t} d_{qp}^{m,t} \cdot \langle \psi_{ji}^{m,t}(x, y), \psi_{qp}^{m,t}(x, y) \rangle \end{aligned} \tag{35}$$

where

$$\langle \psi_{ji}^{m,t}(x, y), \psi_{qp}^{m,t}(x, y) \rangle = \langle \psi_j^{m,t}(y), \psi_q^{m,t}(y) \rangle \cdot \langle \psi_i^{m,t}(x), \psi_p^{m,t}(x) \rangle \tag{36}$$

Equation (35) governs the score of our 2D oracle which is carried out as an extension of the 1D version. The following fraction of C-style code provides all essential steps:

```
// minimizes ||Δf^m(x,y) − Δf'^m(x,y)||²_{L2}|_{Rej(K)}  using a greedy algorithm
   corresponds to version II of the compression problem //

E_b=const;/

old_score=0;

for (m=1,m<=M,m++)
   {
   for (t=1,t<=3,t++)
      {
      E_tot = E_tot + D^{m,t} • D̃^{m,t}
      for (i=1, i<=N/2^m, i++)
         for (j=1, j<=N/2^m, j++)
         {
         score[i,j,m,t]=d[i,j,m,t]*d[i,j,m,t];
         }
      }
   }
 E_tot = E_tot + C^M • C̃^M
while (old_score/E_tot<E_b)
      {
      (p,q, m_rej, t_rej) <- Min(score); // (|score[i,j,m,t]|=min!=0)|//
      for (m=1,m<=M,m++)
         {
         for (t=1,t<=3,t++)            // only wavelets are rejected //
            {
            for (i=1, i<=N/2^m, i++)
               {
               for (j=1, j<=N/2^m, j++)
                  {
                  if ((m==m_rej)&&(t==t_rej)&&(score[i,j,m,t]!=0))
score[i,j,m,t]+=2*d[i,j,m,t]*d[p,q,m_rej,t_rej]*⟨ψ_j^{m,t_(y)}, ψ_q^{m,t_(y)}⟩⟨ψ_i^{m,t_(x)}, ψ_p^{m,t_(x)}⟩
                     + score[p,q,m_rej,t_rej] - old_score;
                  else if (score[i,j,m,t]!=0)
                           score[i,j,m,t]+=score[p,q,m_rej,t_rej] - old_score;
                  }
               }
            }
         }
      old_score=score[p,q,m_rej,t_rej];
      score[p,q,m_rej,t_rej]=0;
      }
```

Note that the oracle operates over all tensor product kernels. Extensions to higher dimensions, such as volumes, can be constructed straightforwardly using the same approach.

7 RESULTS

The following computations compare a 1D version of the proposed oracle with a magnitude based rejection criterion as described in section 1. In fig. 3 wavelet approximations of a discontinuous function are depicted at different compression ratios. We observe that the performance of the new oracle shows up against magnitude based methods especially for high compression gains. All computations

are performed using cubic B-spline WT's, such as in [6]. The data set was sampled with 1024+3 coefficients.

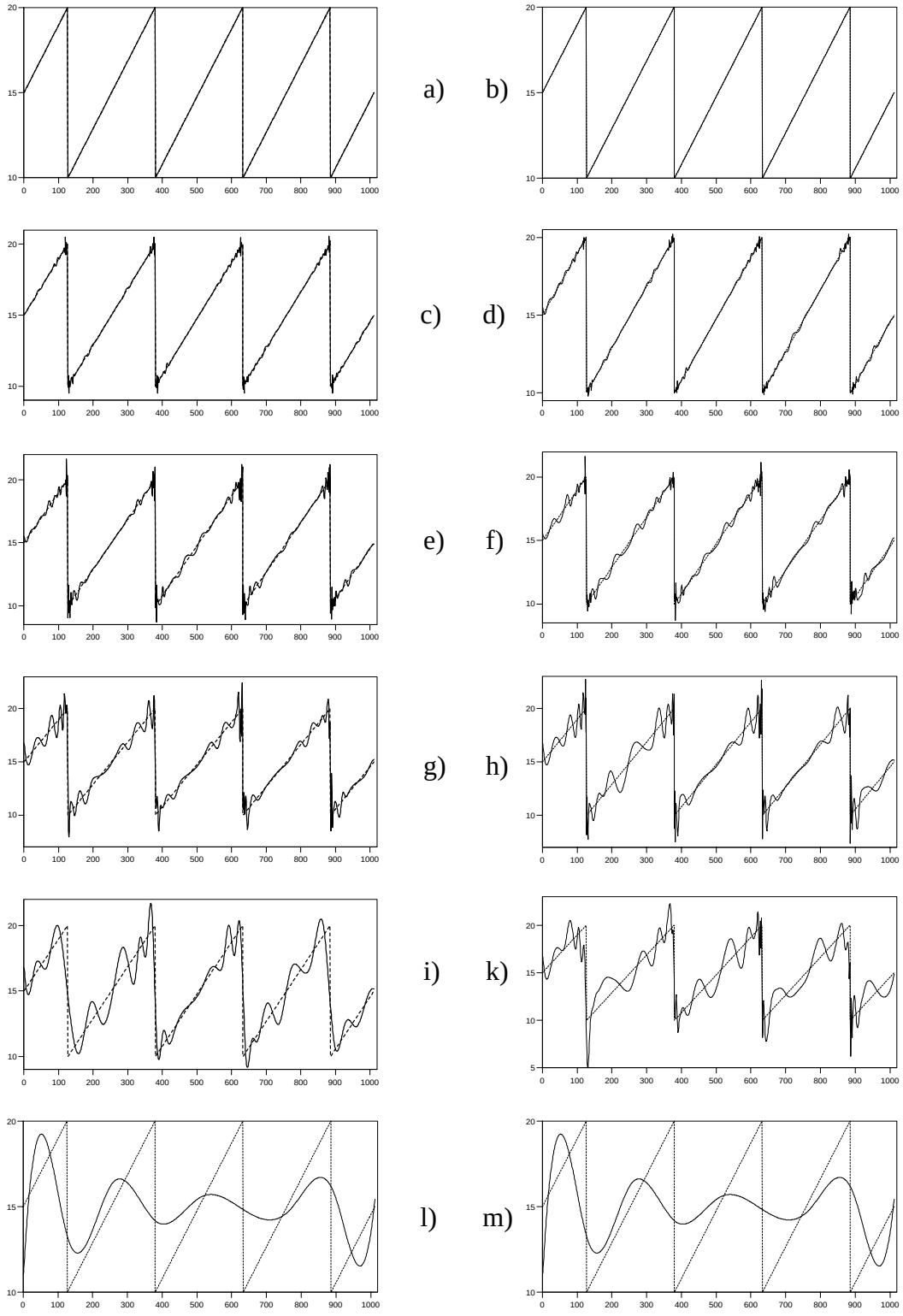

Figure 3: Comparison of oracle based (left hand series) and magnitude based coefficient rejection for the approximation of a dicontinuous function. Provided approximations are superimposed.

Table 2 summarizes the parameter settings and gives the associated L_2 errors. Note that the errors obtained by the oracle based rejection differ significantly for high compression gains.

Table 2: Parameter settings and L_2 approximation errors for fig. 3.

Figure 3	**% coefficients**	**L_2 error magn.**	**L_2 error oracle**
a, b	30	2.12145e-05	2.37568e-05
c, d	15	0.0112864	0.0127405
e, f	10	0.10158	0.0850593
g, h	5	0.687663	0.434002
i, k	3	2.18505	1.43214
l, m	scaling functions	5.86314	5.86314

Similar results are displayed in fig. 4 for a cross section of a digital terrain data set of size 256+3. Again the oracle performs mostly forhigh compression ratios. This tendency is confirmed by the L_2 errors reported in table 3. We observe that in ranges being important for most practical applications the magnitude based rejection performs equally and could hence be preferred with respect to computational complexity.

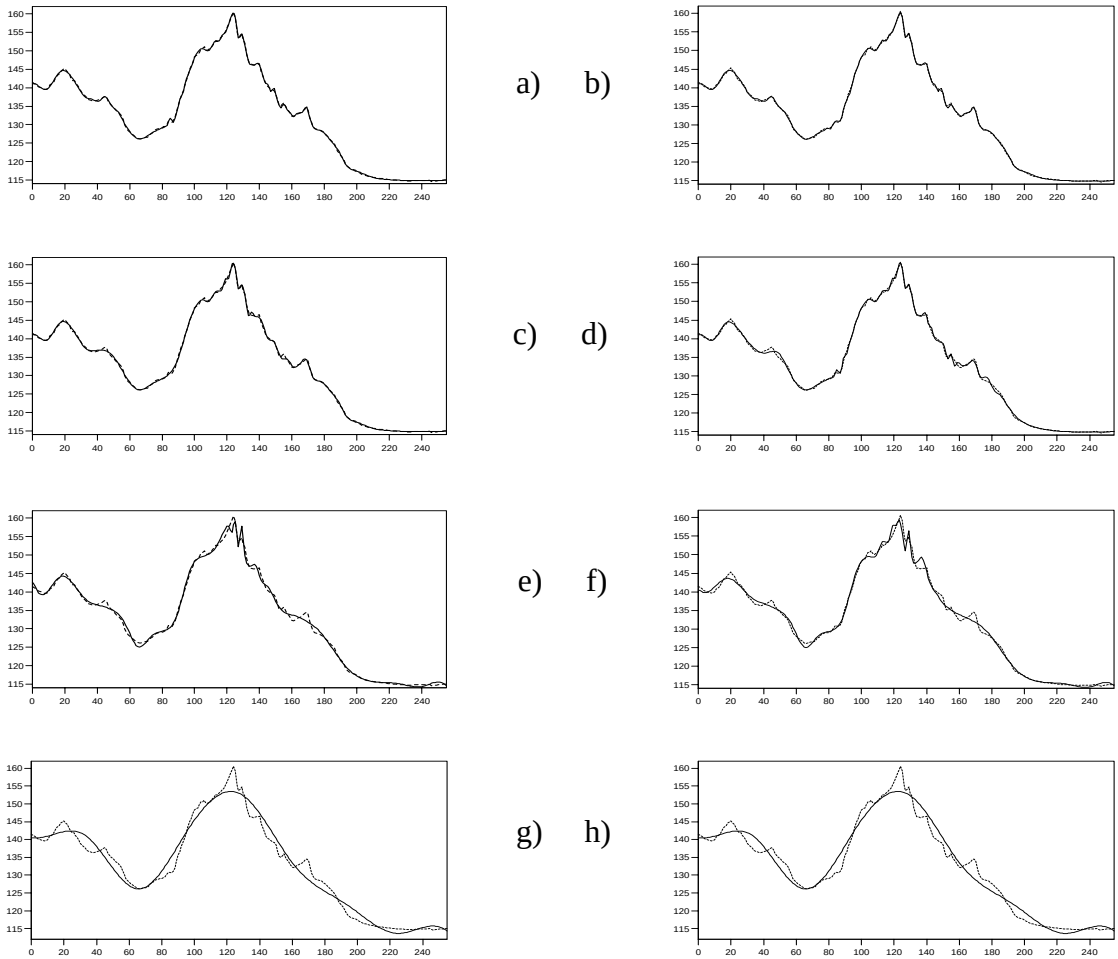

Figure 4: Greedy oracle contrasted against magnitude based coefficient rejection for different compression ratios.

Table 3: Parameter settings and L_2 approximation errors for fig. 4.

Figure 4	**% coefficients**	**L_2 error magn.**	**L_2 error oracle**
a, b	20	0.0616906	0.0671372
c, d	15	0.182863	0.117063
e, f	9	0.827001	0.65917
g, h	scaling functions	0.45991	0.45991

8 CONCLUSIONS

We presented an mathematical analysis of the L_2 approximation error in semiorthogonal wavelet spaces. It was shown that the computation is figured out by dot products of the wavelet coefficients with its counterparts transformed by the inner product matrix of all basis functions. As a result we came up with an incremental scheme to score the conditional significance of an individual basis function. This computational scheme enables to build a greedy oracle which rejects unimportant coefficients and which can be implemented straightforwardly. It is clear that the underlying framework can be used to control advanced wavelet based compression algorithms, which have to be addressed in future research activities. It has to be stated, however, that we did not prove the global optimality of our method. Moreover, the greedy algorithm does not necessarily feature global optima and is vulnerable to local minima. Furthermore, different norms, such as L_∞ have to be considered and require further investigation.

9 REFERENCES

[1] Cohen, A.; Daubechies, I.; Feauveau, J.: Bi-Orthogonal Bases of Compactly Supported Wavelets. Comm. Pure and Applied Math., Vol. 45, pp. 485-560, 1992

[2] Stollnitz, E.; DeRose, T.; Salesin, D.: Wavelets for Computer Graphics: A Primer, Part 1 and 2. IEEE Computer Graphics and Applications, Vol. 15, No. 3, pp. 76-84, 1995

[3] Gross, M.; Staadt, O.; Gatti, R.: Efficicnt Triangular Surface Approximations Using Wavelets and Quadtree Data Structures, IEEE Transactions on Visualization and Computer Graphics, Vol. 2, No. 2, pp. 130-143, 1996

[4] Fournier, A.: Wavelets and their Applications in Computer Graphics. Course Notes SIGGRAPH 95, 1995

[5] Chui, C.K.: An Introduction to Wavelets. Academic Press, Boston, 1992

[6] Quak, E.; Weyrich, N.: Decomposition and Reconstruction Algorithms for Spline Wavelets on a Bounded Interval. Appl. Comput. Harmon. Anal., Vol. 1, No.3, pp. 217-231, 1994.

[7] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.: Introduction to Algorithms. MIT Press, Cambridge, Massachusetts, 1994.