

Best Basis Search in Lapped Dictionaries

Yan Huang, Ilya Pollak, *Senior Member, IEEE*, Charles A. Bouman, *Fellow, IEEE*, and Minh N. Do, *Member, IEEE*

Abstract—This paper proposes, analyzes, and illustrates several best basis search algorithms for dictionaries consisting of lapped orthogonal bases. It improves upon the best local cosine basis selection based on a dyadic tree [10], [11] by considering larger dictionaries of bases. It is shown that this can result in sparser representations and approximate shift invariance. An algorithm that is strictly shift invariant is also provided. The experiments in this paper suggest that the new dictionaries can be advantageous for time-frequency analysis, compression, and noise removal. Accelerated versions of the basic algorithm are provided that explore various tradeoffs between computational efficiency and adaptability. It is shown that the proposed algorithms are in fact applicable to any finite dictionary comprised of lapped orthogonal bases. One such novel dictionary is proposed that constructs the best local cosine representation in the frequency domain, and it is shown that the new dictionary is better suited for representing certain types of signals.

Index Terms—Best basis, lapped transforms, time-frequency analysis.

I. INTRODUCTION

THE contributions of our paper are in the area of best basis search algorithms where the aim is to adaptively select, from a dictionary of orthonormal bases, the basis that minimizes a cost for a given signal [3], [10], [11]. Such methods have been demonstrated to be effective for compression [27], [28], [46], [52], estimation [13]–[15], [25], [26], [30], [38], [43], [53], and time-frequency (or space-frequency) analysis [12], [18], [19], [50], [51], [54].

The original work on best basis search [10], [11] exploited the fact that a dictionary consisting of local cosine bases [9], [32], [33], [47] on dyadic intervals can be represented as a single dyadic tree. This made it possible to find the best basis, for an additive cost function, via an efficient tree pruning algorithm. On the other hand, it has been noticed that, for an additive cost function, the optimal segmentation of a one-dimensional (1-D) signal can be efficiently found using dynamic programming. This has been exploited in many contexts such as piecewise polynomial approximation [2], [41], [44], best basis

Manuscript received June 10, 2004; revised March 29, 2005. This work was supported in part by a National Science Foundation (NSF) CAREER award CCR-0093105, an NSF Grant IIS-0329156, a Purdue Research Foundation grant, and an NSF CAREER award CCR-0237633. All experiments were generated with the help of Wavelab 802 [16]. Preliminary results were reported in [20] and [21]. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Anuj Srivastava.

Y. Huang, I. Pollak, and C. A. Bouman are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA.

M. N. Do is with the Department of Electrical and Computer Engineering and Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

Digital Object Identifier 10.1109/TSP.2005.861771

search in time-varying wavelet packet [54] and Modified Discrete Cosine Transform (MDCT) [40] dictionaries, estimation of abrupt changes in a linear predictive model [45], and optimal selection of cosine-modulated filter banks [39]. In this paper, we exploit a similar idea to remove the restriction of [10], [11] that the supports of local cosine basis functions be dyadic and use a dynamic programming algorithm to find the best basis in a much larger collection of local cosine bases. Through examples, we illustrate the advantages of our approach in three application areas: time-frequency analysis, compression, and noise removal. Specifically, these examples show that our algorithms result in the following:

- sparser and more accurate time-frequency descriptions;
- lower entropy, even when the side information is taken into account;
- improved noise removal performance, as measured by the signal-to-noise ratio (SNR).

In addition, we extend our basic algorithm in several ways. We argue that our algorithm is approximately shift invariant and, moreover, show that it can be made strictly shift invariant by using a procedure similar to the one developed in [12]. We furthermore propose two accelerated versions of the algorithm which explore various tradeoffs between computational efficiency and adaptability, and which are based on the idea of two-stage processing of the data: first, small pieces of a signal are processed using dynamic programming within each piece, and then the results are combined using another dynamic programming sweep.

The use of our algorithms is not restricted to local cosine dictionaries. For example, lapped bases in the frequency domain were used in [24], [30], [37]. We propose a novel construction that represents the discrete cosine transform (DCT) of a signal in a local cosine dictionary and therefore corresponds to representing the signal in a dictionary whose elements are the inverse DCTs of the local cosine functions. We give examples where noise removal using this new dictionary yields a higher SNR than the best local cosine representation. While we develop and illustrate our algorithms using two dictionaries—the local cosines in the time domain and in the DCT domain—we show in Section IV that our algorithms are applicable to any finite dictionary comprised of lapped orthogonal bases.

II. LOCAL COSINE DECOMPOSITIONS

A. Best Basis Search Problem

The general best basis search problem is formulated, for example, in [10], [11], and [30]. We consider a dictionary \mathcal{D} that is a set of orthonormal bases for \mathbb{R}^N , $\mathcal{D} = \{B^\lambda\}_{\lambda \in \Lambda}$, where each basis B^λ consists of N vectors, $B^\lambda = \{g_m^\lambda\}_{1 \leq m \leq N}$. The cost

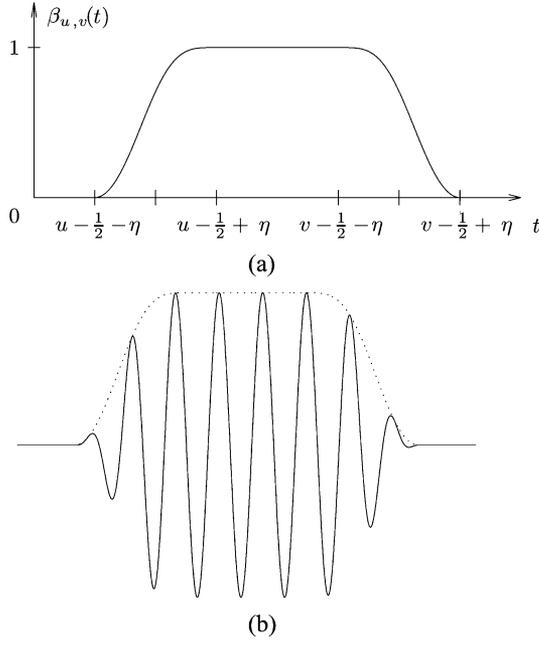


Fig. 1. Window function $\beta_{u,v}$ and an element of a local cosine dictionary. (a) Window function $\beta_{u,v}$. (b) Local cosine basis function.

of representing a signal f in B^λ is typically defined as follows [10], [11], [30]:

$$C(f, B^\lambda) = \sum_{m=1}^N \Phi \left(\frac{|\langle f, g_m^\lambda \rangle|^2}{\|f\|^2} \right) \quad \text{or} \quad (1)$$

$$C(f, B^\lambda) = \sum_{m=1}^N \Phi \left(|\langle f, g_m^\lambda \rangle|^2 \right),$$

where Φ is application dependent. Any basis that achieves the minimum of the cost $C(f, B^\lambda)$ over all the bases in the dictionary is called the *best basis*.

B. A Local Cosine Dictionary

We identify each vector in \mathbb{R}^N with a signal $f(n)$ defined for $n = 0, 1, \dots, N - 1$. A local cosine basis [9], [30], [32], [33], [47] for \mathbb{R}^N is defined using cosine functions multiplied by overlapping smooth windows. For each discrete interval $[u, v - 1] \subset [1, N - 2]$, we define a window function $\beta_{u,v}$ [see Fig. 1(a)], which gradually ramps up from zero to one around $u - 1/2$ and goes down from one to zero around $v - 1/2$:

$$\beta_{u,v}(t) = \begin{cases} r \left(\frac{t - (u - 1/2)}{\eta} \right), & \text{if } u - \frac{1}{2} - \eta \leq t < u - \frac{1}{2} + \eta \\ 1 & \text{if } u - \frac{1}{2} + \eta \leq t < v - \frac{1}{2} - \eta \\ r \left(\frac{(v - 1/2) - t}{\eta} \right) & \text{if } v - \frac{1}{2} - \eta \leq t \leq v - \frac{1}{2} + \eta \\ 0 & \text{otherwise} \end{cases}$$

where the parameter $\eta \in \mathbb{R}$ controls how fast the window tapers off, and r is a profile function which monotonically increases from $r(t) = 0$ for $t < -1$ to $r(t) = 1$ for $t > 1$ and satisfies

$r^2(t) + r^2(-t) = 1 \forall t \in \mathbb{R}$. Following [30], we define the discrete local cosine family $\mathcal{B}_{u,v}$ as follows:

$$\mathcal{B}_{u,v} = \left\{ \beta_{u,v}(n) \sqrt{\frac{2}{v-u}} \times \cos \left[\pi \left(\kappa + \frac{1}{2} \right) \frac{n - (u - \frac{1}{2})}{v-u} \right] \right\}_{0 \leq \kappa < v-u}$$

where $n \in \mathbb{Z}$ is a discrete time parameter and $\kappa \in \mathbb{Z}$ is a discrete frequency parameter. One signal from such a family is depicted in Fig. 1(b). It can be shown [30] that this set of signals is orthonormal if $v - u \geq 2\eta$.

For a signal f of length N , we search for the best basis in the local cosine dictionary

$$\mathcal{D} = \bigcup_{\lambda \in \Lambda} B^\lambda \quad (2)$$

which consists of the following local cosine bases:

$$B^\lambda = \bigcup_{k=0}^{K_\lambda - 1} \mathcal{B}_{n_k, n_{k+1}} \quad (3)$$

where λ is a set of partition points $\{n_k\}_{0 \leq k \leq K_\lambda}$ of the domain of f . If the partition points are such that only adjacent windows overlap (i.e., if $n_{k+1} - n_k \geq 2\eta$ for all k) then B^λ is an orthonormal basis for \mathbb{R}^N [30]. In order to achieve this, we impose that the finest cell size be some fixed integer $M \geq 2\eta$, i.e., we require the partition points to be integer multiples of M

$$n_0 = 0 < n_1 < \dots < n_{K_\lambda - 1} < n_{K_\lambda} = N \quad (4)$$

$$n_k \text{ is divisible by } M \text{ where } M \geq 2\eta \text{ is a fixed integer.} \quad (5)$$

We will refer to the resulting \mathcal{D} as a MOD-M dictionary. We note that a MOD-M dictionary is larger than the local cosine tree dictionary of [10], [11]. In fact, if we choose M such that $N/M = 2^J$, where J is the maximum depth of the local cosine tree of [10], [11], it can be easily shown that the local cosine tree dictionary of [10], [11] will be a subset of the MOD-M dictionary.

C. Best Basis Algorithm

We now describe an efficient best basis search algorithm for our MOD-M dictionary. It is a dynamic programming algorithm whose variants have been widely used in literature since [2] to find the best segmentation of a 1-D signal. Our exposition closely follows [54] where it was used to find the best block wavelet packet basis.

Let $0 \leq u < v \leq N$, and let the best basis associated with the window $\beta_{u,v}$ be denoted by $\mathcal{O}_{u,v}$. For $v - u > M$, we have (6), shown at the bottom of the next page, where

$$d^* = \arg \min_{d: u < d < v, d \text{ is a multiple of } M} C(f, \mathcal{B}_{u,d}) + C(f, \mathcal{O}_{d,v}).$$

```

 $\mathcal{O}_{0,N} \leftarrow \text{best.basis}(f) \{$ 
  for  $u = N - M, N - 2M, \dots, 2M, M, 0 \{$ 
     $\mathcal{O}_{u,N} \leftarrow \mathcal{B}_{u,N};$  //Initialize  $\mathcal{O}_{u,N}$ 
     $C_{u,N}^* \leftarrow C(f, \mathcal{B}_{u,N});$  //Initialize  $C_{u,N}^*$ 
    for  $d = u + M, u + 2M, \dots, N - M \{$ 
      if  $C(f, \mathcal{B}_{u,d}) + C_{d,N}^* < C_{u,N}^* \{$ 
         $\mathcal{O}_{u,N} \leftarrow \mathcal{B}_{u,d} \cup \mathcal{O}_{d,N};$ 
         $C_{u,N}^* \leftarrow C(f, \mathcal{B}_{u,d}) + C_{d,N}^*;$ 
      }
    }
  }
  save  $\mathcal{O}_{u,N}$  and  $C_{u,N}^*$  in an internal data structure;
}
return  $\mathcal{O}_{0,N};$ 
}
    
```

Fig. 2. Pseudocode specification of a fast dynamic programming algorithm for the best local cosine basis search. The cost of the best basis $\mathcal{O}_{u,N}$ is denoted by $C_{u,N}^*$.

(Note that, since the cost function is additive, the cost of $\mathcal{B}_{u,d} \cup \mathcal{O}_{d,v}$ is $C(f, \mathcal{B}_{u,d}) + C(f, \mathcal{O}_{d,v})$.) The initial condition is that for $v - u = M$,

$$\mathcal{O}_{u,v} = \mathcal{B}_{u,v}.$$

Then the best basis $\mathcal{O}_{0,N}$ for signal f can be calculated via dynamic programming, by repeatedly applying (6). The C pseudocode for this algorithm is shown in Fig. 2. In Fig. 2, we use $C_{u,v}^*$ to denote the cost of the best basis $\mathcal{O}_{u,v}$ and assume that the costs $C(f, \mathcal{B}_{u,v})$ have been precomputed. The algorithm calculates $\mathcal{O}_{u,N}$ and $C_{u,N}^*$ for $u = N - M, N - 2M, \dots, 2M, M, 0$. The calculation of each $\mathcal{O}_{u,N}$ involves a loop over $u + M, u + 2M, \dots, N - M$, with $O(1)$ computations within each iteration of the loop. Therefore, the dynamic programming has time complexity $O(L^2)$, where $L = N/M$. The major computational burden is associated with computing the costs $C(f, \mathcal{B}_{u,d})$. The calculation of $C(f, \mathcal{B}_{u,v})$ via the definition (1) involves $O(a)$ additions where $a = v - u$, as well as the computation of the inner product of f with each basis function in $\mathcal{B}_{u,v}$, which requires $O(a \log a)$ operations using a fast local cosine transform algorithm [30], [33]. In the process of calculating $\mathcal{O}_{0,N}$, we need the values for $C(f, \mathcal{B}_{u,v})$ with $u = pM, v = qM$, where $p = 0, 1, \dots, L - 1$ and $q = p + 1, p + 2, \dots, L$. It is easy to show that this results in the overall time complexity¹ of $O(L^2 N \log N)$.

D. Shift Invariance: A Qualitative Discussion

We call a best basis search algorithm n_0 -shift-invariant if circularly shifting any signal by an arbitrary integer multiple of n_0 leads to shifting its best basis by the same multiple of n_0 . When $n_0 = 1$ —i.e., when the algorithm is invariant to any integer shift—we simply call it shift invariant.

¹The time complexities of all our algorithms are summarized in a table in the Appendix.

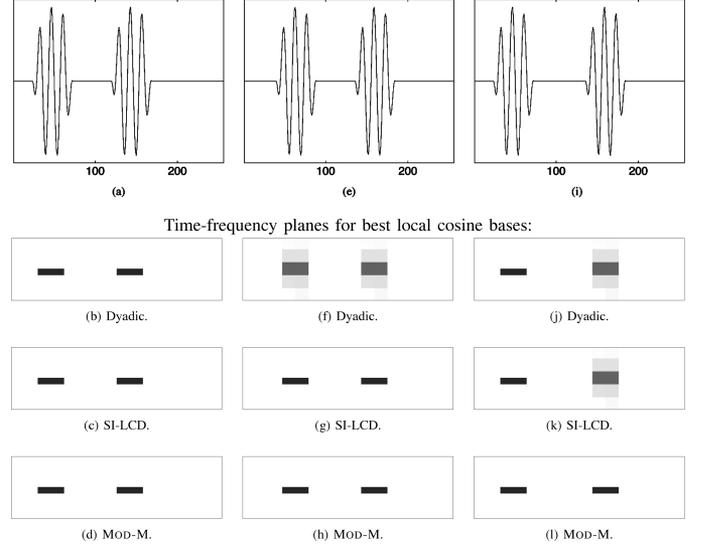


Fig. 3. Original signals and time-frequency representations of the best local cosine basis with smallest cell size $M = 16$: (a) signal consisting of two local cosine basis functions; (b) time-frequency tiling for the best local cosine basis of [10], [11]; (c) time-frequency tiling for the shift-invariant local cosine decomposition [12]; (d) time-frequency tiling for the best MOD-M local cosine basis; (e)–(h) similar experiment for the signal in (a) shifted by 16 samples; and (i)–(l) similar experiment for a signal where the two local cosine bumps are shifted by different amounts. The darkness of each rectangle in (b)–(d), (f)–(h), (j)–(l) is proportional to the magnitude of the corresponding local cosine coefficient.

The MOD-M method described in Section II-C is, strictly speaking, not M -shift-invariant, since we always require the leftmost basis function to start at the leftmost point of the signal. It is, however, M -shift-invariant, modulo these boundary effects: i.e., it is invariant to shifts by integer multiples of M for signals whose support is well within the interval $[0, N - 1]$. The dyadic best local cosine basis algorithm of [10], [11] is fundamentally not shift invariant since it uses a dyadic tree. Its variant introduced in [12] is formally shift invariant; however, Fig. 3 shows that the MOD-M method offers certain advantages.

For each best basis experiment in Fig. 3, the smallest cell size M is chosen to be 16. (For the methods in [10]–[12], this means that the maximal tree depth is set to $J = \log_2 N - \log_2 M = 4$.) We follow [10] and [11] and use the entropy cost function

$$C(f, B^\lambda) = - \sum_{m=1}^N \frac{|\langle f, g_m^\lambda \rangle|^2}{\|f\|^2} \ln \frac{|\langle f, g_m^\lambda \rangle|^2}{\|f\|^2}.$$

The 256-point signal depicted in Fig. 3(a) consists of two local cosine basis functions with $\eta = 8$, one with $u = 32$ and $v = 64$ (i.e., supported on $[24, 71]$), and another one with $u = 128$ and $v = 160$ (i.e., supported on $[120, 167]$). Since both these functions are in the dyadic dictionary, all three methods produce the same best basis, as illustrated by the time-frequency

$$\mathcal{O}_{u,v} = \begin{cases} \mathcal{B}_{u,d^*} \cup \mathcal{O}_{d^*,v}, & \text{if } C(f, \mathcal{B}_{u,d^*}) + C(f, \mathcal{O}_{d^*,v}) < C(f, \mathcal{B}_{u,v}) \\ \mathcal{B}_{u,v}, & \text{otherwise.} \end{cases} \quad (6)$$

tilings² of Fig. 3(b)–(d), which respectively correspond to the dyadic search of [10], [11], the shift-invariant local cosine decomposition (SI-LCD) proposed in [12] which essentially considers N shifted versions of the dyadic dictionary, and our MOD-M method. Shifting each of the two width-32 local cosine basis functions³ to the right by 16 samples—as shown in Fig. 3(e)—takes them both out of the dyadic dictionary and forces the dyadic algorithm to represent each of them by a number of width-16 local cosine basis functions, six of which are clearly visible in Fig. 3(f) (the others have small coefficients and are not visible in the figure). Both the SI-LCD and MOD-M algorithms, however, are invariant to this shift [Figs. 3(g) and (h)]. The third column of Fig. 3 illustrates shifting the two local cosine functions by different amounts: the first one stays in place, and the second one is shifted by 16. The MOD-M algorithm is still invariant to this change, as evidenced by Fig. 3(l). SI-LCD, however, produces a different basis.

E. Strictly Shift-Invariant Algorithm

The qualitative discussion of Section II-D shows that the MOD-M algorithm possesses the desired shift-invariant properties, even though it is not, strictly speaking, shift invariant. We now show, in addition, that we can make it strictly invariant to any integer shift, using a method similar to [12].

For a discrete signal f of length N , we extend both the signal and the basis functions periodically with period N so that all shifts of all signals will effectively be circular shifts. We expand the dictionary of Section II-B by adding in the shifts of the basis signals. We define \mathcal{D}_0 to be the same as the dictionary of (2), (3), and let \mathcal{D}_s be \mathcal{D}_0 shifted by s to the left, as follows:

$$\mathcal{D}_s = \bigcup_{\lambda \in \Lambda} B_s^\lambda \quad \text{where } B_s^\lambda = \bigcup_{k=0}^{K_\lambda-1} \mathcal{B}_{n_k+s, n_{k+1}+s}.$$

The new dictionary \mathcal{D}_{SI} is defined to be the union of the N shifted subdictionaries

$$\mathcal{D}_{\text{SI}} = \bigcup_{s=0}^{N-1} \mathcal{D}_s.$$

Now the best basis search involves finding the subdictionary \mathcal{D}_{s^*} that contains the best basis and searching for the best basis in \mathcal{D}_{s^*} . Using an argument similar to the one in Section II-C, it can be shown that the optimal solution is achieved in $O(N^3 \log N)$ time. In addition, we present in the Appendix a suboptimal solution based on the method in [12], to result in the time complexity similar to that of Section II-C.

F. Examples With the Entropy Cost

To further illustrate our methods, we use two examples that compare our proposed MOD-M local cosine decomposition with the dyadic best local cosine basis selection [10], [11]. We again use the entropy cost function. We set $\eta = 8$.

²To depict a coefficient corresponding to a local cosine basis function with frequency \varkappa and window $\beta_{u,v}$, we use a rectangle that extends horizontally from u to v and vertically from $\varkappa/(v-u)$ to $(\varkappa+1)/(v-u)$. The intensity of the rectangle is proportional to the magnitude of the coefficient.

³We use the term *width* to mean the number of samples in the interval $[u, v-1]$.

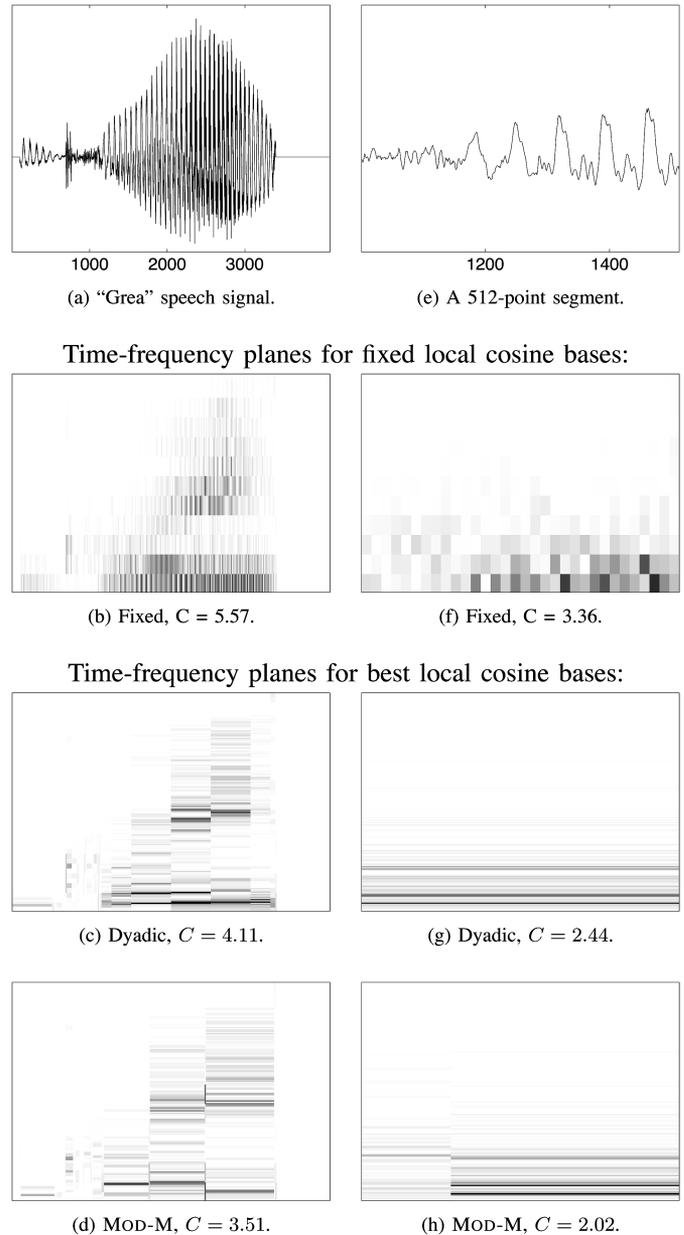
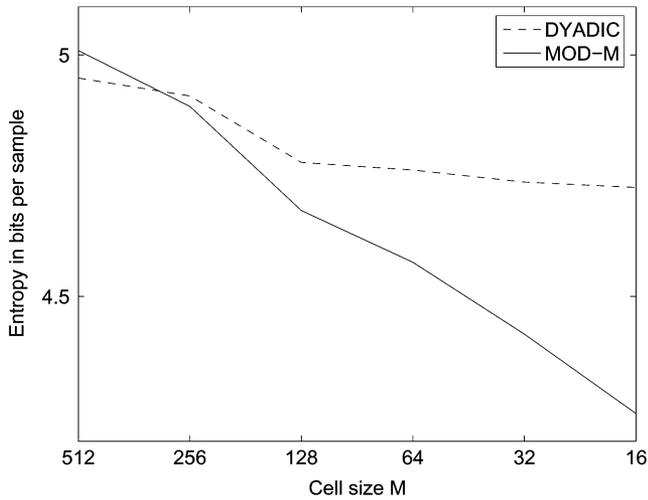
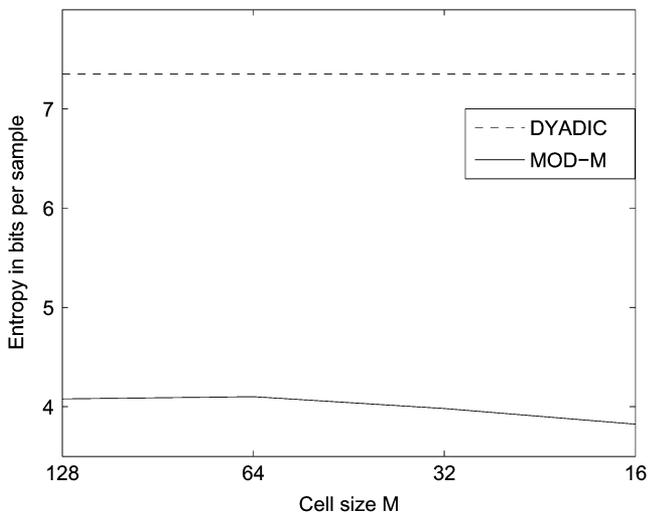


Fig. 4. Two signals and the time-frequency pictures of their best bases: (a) segment “grea” of the speech signal “greasy”; (b), (c), (d) time-frequency tilings for a fixed local cosine basis, the best local cosine basis of [10], [11], and the best MOD-M local cosine basis, respectively; (e–h) similar experiment for a shorter segment of the speech signal. (a) “Grea” speech signal. (b) Fixed, $C = 5.57$. (c) Dyadic, $C = 4.11$. (d) MOD-M, $C = 3.51$. (e) 512-point segment. (f) Fixed, $C = 3.36$. (g) Dyadic, $C = 2.44$. (h) MOD-M, $C = 2.02$.

Figs. 4(a), 4(c), and 4(d) show a speech signal of length $N = 4096$ and the time-frequency pictures for the best bases selected by the two methods. The minimal cell size for these experiments was set at $M = 16$ for both methods. The resulting costs are 4.11 for the dyadic dictionary and 3.51 for ours. In addition, note the sparser time-frequency representation in Fig. 4(d) resulting from our method. For comparison, we provide the result for a fixed local cosine decomposition with $M = 16$ —i.e., the local cosine decomposition where each segment has width 16. The cost of this basis is 5.57. The difference of the costs $C_{\text{dyadic}} - C_{\text{MOD-M}}$, while smaller than $C_{\text{fixed}} - C_{\text{dyadic}}$, is



(a)



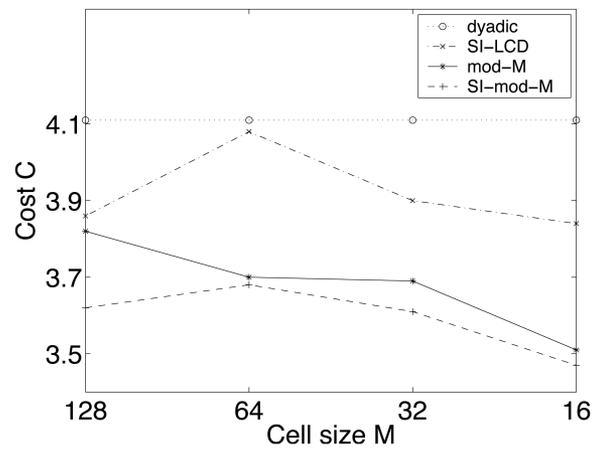
(b)

Fig. 5. Total entropy as a function of the minimal cell size M for the dyadic and MOD-M algorithms. (a) Estimated number of bits per sample for “grea.” (b) Estimated number of bits per sample for the 512-point segment.

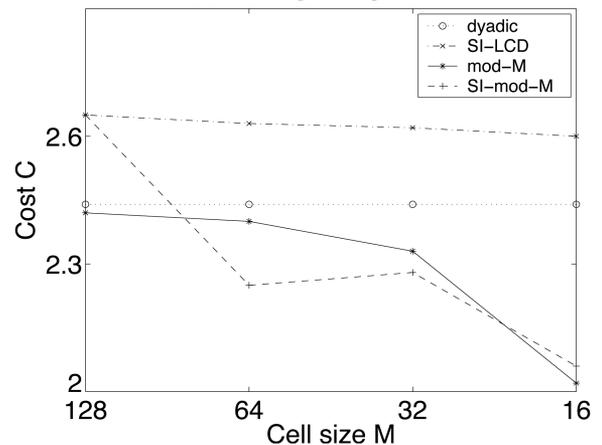
substantial, showing that the improvement achieved by our algorithm is significant.

In Fig. 4(e), we zoom into the samples 1001 through 1512 of the signal in Fig. 4(a). For this 512-point segment, we compute the best basis with the two methods, again setting $M = 16$, and again comparing to the fixed basis. This results in the following costs: 3.36 for the fixed basis, 2.44 for the dyadic dictionary, and 2.02 for ours. Again, the representation resulting from our method corresponds to a sparser time-frequency tiling. Moreover, the transition between two phonemes (in the neighborhood of the sample 1150) is missed by the best dyadic basis but is accurately captured by the best MOD-M basis.

To demonstrate the potential of our method for compression applications, we estimate the total entropy, in bits per sample, for both the dyadic and MOD-M best bases, for different values of the minimal cell size M . The number of bits required to encode the coefficients is estimated by constructing a histogram of the coefficients and computing the entropy for the histogram. The number of overhead bits in the MOD-M case—i.e., the bits required to encode the partition points n_k —is estimated



(a) “Grea” speech signal.



(b) A 512-point segment.

Fig. 6. Performance of four algorithms for extracting the best local cosine basis: dyadic [10], [11] (dotted), SI-LCD [12] (dashed-dotted line), the proposed MOD-M algorithm (solid line), and the proposed shift-invariant version of the MOD-M algorithm (dashed line). The optimal cost is depicted as a function of the minimal allowed cell size: (a) 4096-point “grea” speech signal and (b) 512-point segment of the signal.

by constructing a histogram of the distances between adjacent partition points and computing the entropy for this histogram. The number of overhead bits in the dyadic case is assumed to be equal to the number of nodes in the dyadic tree, since at each node of the tree one binary decision needs to be encoded: whether or not the node has children. The total estimates of the numbers of bits necessary to encode both the coefficients and the side information are shown, for several examples, in Fig. 5. Fig. 5(a) shows the results for the 4096-point “grea” signal, for six values of the minimal cell size M : 512, 256, 128, 64, 32, and 16, corresponding to the tree depths of the dyadic decomposition 3, 4, 5, 6, 7, and 8. In each of these experiments, the contribution of the overhead, for both dictionaries, does not exceed 0.5% of the overall entropy. The fact that the overhead is negligible is not surprising since both best bases have relatively few partition points. For very large cell sizes, the dyadic and MOD-M best bases are similar and result in similar entropies; as the cell size gets smaller, however, the MOD-M best basis can achieve a significantly lower entropy. A similar experiment for the 512-point segment of this signal is shown in Fig. 5(b).

Fig. 6 summarizes an experiment where we compared the dyadic tree algorithm and the MOD-M algorithm with the suboptimal shift-invariant versions of the two algorithms:

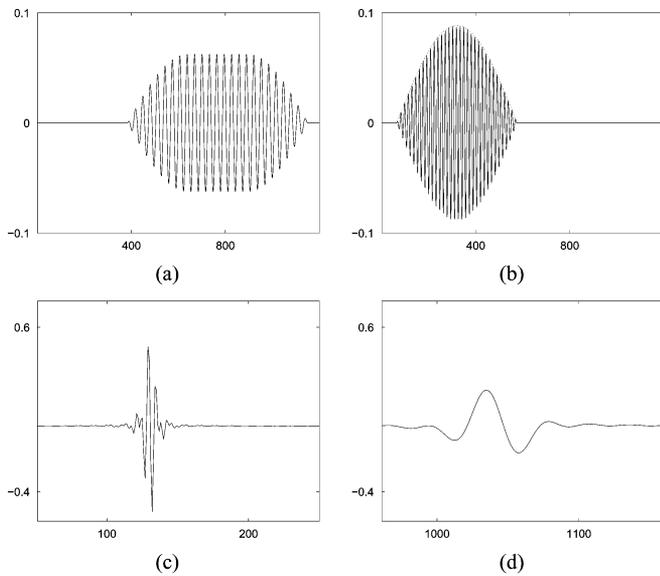


Fig. 7. Top row: Two local cosine functions. Bottom row: Two functions from the FDLC dictionary obtained by taking the inverse DCT-IV of the functions in the top row.

the SI-LCD [12] and our suboptimal shift-invariant MOD-M algorithm described in the Appendix. The resulting costs⁴ for the four algorithms are plotted as a function of M . Note that in both cases, the whole curve for the MOD-M algorithm is below each of the outcomes for the algorithms in [11], [12]. This is to be expected since we perform the search over a much larger dictionary. The price to pay is the time complexity of the algorithm, which, as indicated above, is higher than the time complexity for the dyadic best basis search algorithm. Note, however, that for small signal lengths the running time of the two algorithms is similar. For example, in our experiment with the 512-point signal, the running times⁵ for $M = 128, 64, 32, 16$ are 0.01, 0.02, 0.02, and 0.03 s, respectively, for the dyadic algorithm and 0.01, 0.01, 0.05, and 0.19, respectively, for the MOD-M algorithm. This suggests that the most practical way of using this algorithm is on blocks whose length is a small multiple of M . Sections III-B and III-C investigate this idea.

G. Frequency-Domain Local Cosines

Frequency-domain lapped bases have been suggested in, e.g., [24], [30], and [37]. For example, it was shown in [24] that decomposing a signal in a Meyer wavelet basis [37] is equivalent to decomposing its spectrum in a lapped trigonometric basis. We propose a new dictionary of lapped bases in the frequency domain, which we call the MOD-M FDLC dictionary. This dictionary is obtained by taking the inverse DCT of each basis vector of the MOD-M local cosine dictionary defined in (2)–(5). (A dyadic FDLC dictionary can similarly be obtained by taking the inverse DCT of each basis vector of the dyadic local cosine dictionary.) Two FDLC basis vectors are depicted in the bottom row of Fig. 7; their DCTs are members of the local cosine dictionary and are shown in the top row of Fig. 7.

⁴Note that, even though the shift-invariant dictionaries are supersets of their nonshift-invariant counterparts, the latter can sometimes result in smaller costs since the shift-invariant search algorithms used here are suboptimal.

⁵The code was written in Matlab and run using Matlab 6.5 under Windows XP on a machine with a Pentium-M 1.4 GHz processor.

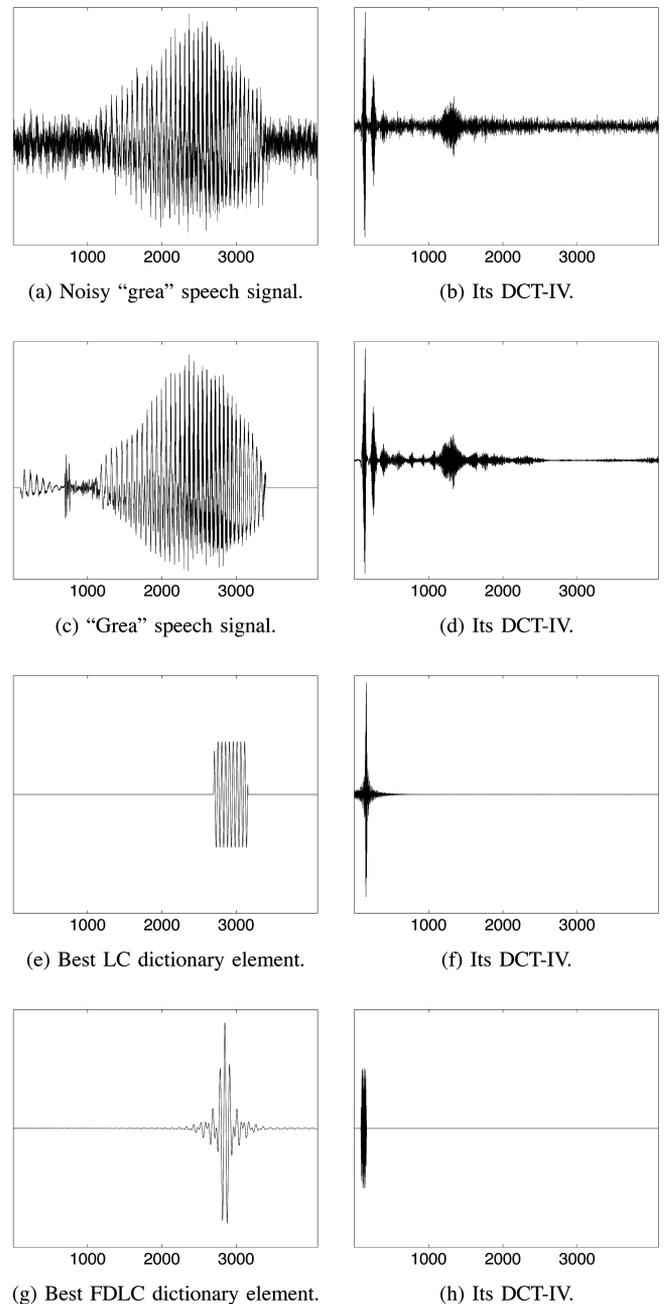


Fig. 8. (a) A noisy speech signal; (b) its DCT; (c) noise-free speech signal; (d) its DCT; (e) the basis vector from the best MOD-M local cosine basis whose inner product with the signal in (a) is the largest; (f) its DCT; (g) the basis vector from the best MOD-M FDLC basis whose inner product with the signal in (a) is the largest; (h) its DCT.

To find the best basis of a signal f in this dictionary, we calculate the DCT \hat{f} of f , and then find the best local cosine basis for \hat{f} using the MOD-M method described above. Since DCT is an orthogonal transform, $|\langle f, g_m^\lambda \rangle|^2 = |\langle \hat{f}, \hat{g}_m^\lambda \rangle|^2$ and $\|f\|^2 = \|\hat{f}\|^2$, and therefore the costs (1) computed in the DCT domain are identical to the costs in the time domain.

The FDLC dictionary offers alternative ways of tiling the time-frequency plane and is better suited than the local cosine dictionary to the analysis of some types of nonstationary signals, for example, those whose energy is mostly concentrated in several frequency bands. This is illustrated in Fig. 8, where

TABLE I
COMPARISON OF DYADIC AND MOD-M METHODS FOR THE NOISY “GREA” SIGNAL WITH SNR = 8.22 dB.

	SNR (db)	RMS error	cost
Dyadic Local Cosines	11.23	107.03	3.2e+007
MOD-M Local Cosines	11.62	102.38	2.1e+007
Dyadic Frequency-Domain Local Cosines	12.51	92.36	2.5e+007
MOD-M Frequency-Domain Local Cosines	12.62	91.29	1.8e+007

the noisy “grea” speech signal, y , and its DCT are shown in (a) and (b), respectively. The noise-free signal, x , and its DCT, are shown in Figs. 8(c) and (d), respectively. Fig. 8(e) shows the basis vector g_{LC} from the best MOD-M local cosine basis for y whose inner product with y is the largest, and Fig. 8(f) shows the DCT of this basis vector. Fig. 8(g) shows the basis vector g_{FDLC} from the best MOD-M FDLC basis whose inner product with y is the largest, and Fig. 8(h) shows the DCT of this basis vector. It is evident from Figs. 8(f) and (h) that g_{FDLC} is more sharply concentrated around the strongest resonant frequency of x than g_{LC} whose spectrum is more spread out. As we show in Section II.H and Table I, the noise removal performance in the problem of recovering x from this observation y is better for the best FDLC basis than for the best local cosine basis.

H. Noise Removal Examples

Following [13], we adopt the following procedure for estimating a signal x from its noisy measurement y : we find the best basis for y from a dictionary \mathcal{D} , decompose y in the best basis, threshold the coefficients, and reconstruct an estimate of x from the remaining coefficients. We use hard thresholding, i.e., we keep every coefficient whose absolute value is above a threshold T , and replace all other coefficients with zeros. As suggested in [26], [30], we use the following cost function for a basis $B^\lambda = \{g_m^\lambda\}_{1 \leq m \leq N}$:

$$C(y, B^\lambda) = \sum_{m=1}^N \Phi(|\langle y, g_m^\lambda \rangle|^2),$$

with

$$\Phi(u) = \begin{cases} u - \sigma^2 & \text{if } u \leq T^2, \\ \sigma^2 & \text{if } u > T^2. \end{cases}$$

In our experiments, we follow [16] and fix $T = 3.8\sigma$. In Table I and Figs. 9 and 10, we present the noise removal results for the dyadic and MOD-M versions of both the time-domain and frequency-domain local cosine (FDLC) dictionaries. To perform noise removal with a FDLC dictionary, a signal is transformed using DCT-IV. Both the best basis extraction and thresholding are then done in the DCT domain. Finally, the resulting DCT coefficients are transformed back using the inverse DCT-IV.

For both time-domain and FDLC dictionaries, the MOD-M dictionary leads to a lower cost, higher SNR, and lower RMS error than the dyadic dictionary. The FDLC dictionaries quite dramatically outperform their respective time-domain counterparts, achieving higher SNRs, lower root-mean-square (rms) errors, lower costs, and significantly sparser time-frequency tilings shown in the last row of Fig. 9. In addition, the last row of Fig. 9 shows that the FDLC dictionaries resolve the resonant

frequencies [i.e., frequencies corresponding to the peaks in the DCT plots of Fig. 9(b) and (d)] much better than the local cosine dictionaries.

Fig. 10 illustrates Monte Carlo experiments performed on three speech signals of length $N = 8192$. For each signal, 30 noisy versions were generated by adding 30 realizations of white Gaussian noise to the signal: ten realizations resulting in SNR = 10 dB, ten resulting in SNR = 15 dB, and ten resulting in SNR = 20 dB. The ten SNRs achieved by each of the four best bases (MOD-M FDLC, dyadic FDLC, MOD-M local cosines, and dyadic local cosines) are plotted in Fig. 10(a) for one of the signals and the input SNR level 10 dB. Note that the FDLC dictionaries outperform the time-domain local cosine dictionaries and that the MOD-M dictionaries typically outperform their dyadic counterparts. We have compared these four algorithms to the results of thresholding the coefficients of a fixed Meyer wavelet basis [37], for several different threshold values. The dotted line in Fig. 10(a) shows the best among these, which corresponds to the threshold value $T/2 = 1.9\sigma$, and achieves significantly lower SNRs than the four best basis algorithms.

The full results of the Monte Carlo simulations are shown in Figs. 10(b)–(d). Each curve is the plot of the average SNRs achieved by the corresponding algorithm, and the error bars are set at \pm two standard deviations. For example, the leftmost points of the curves in Fig. 10(b) are the average values of the corresponding curves in Fig. 10(a). Again, the FDLC dictionaries perform better than their time-domain counterparts and than the Meyer wavelet basis; the MOD-M dictionaries perform better than their dyadic counterparts.

III. FURTHER EXTENSIONS OF THE BASIC ALGORITHM

A. Extension 1, MIN-M: Allowing Arbitrary Positions for Windows

The MOD-M algorithm restricted the length of the local cosine windows to be integer multiples of the finest interval size M . We now allow arbitrary window length with a lower bound M . This results in a larger dictionary than the dictionary of Section II-B. The dictionary and the local cosine bases in it are defined in the same way as in (2) and (3), while the constraints (4) and (5) on the partition points are changed to the following:

$$n_0 = 0 < n_1 < \dots < n_{K_\lambda - 1} < n_{K_\lambda} = N \quad (7)$$

$$n_{k+1} - n_k \geq M \text{ for } k = 0, 1, \dots, K_\lambda - 1$$

where $M \geq 2\eta$ is a fixed integer. (8)

This new dictionary will be referred to as a MIN-M dictionary. The recursion formula for the best basis search is the same as (6); however, the search for d^* is now done over a different set.

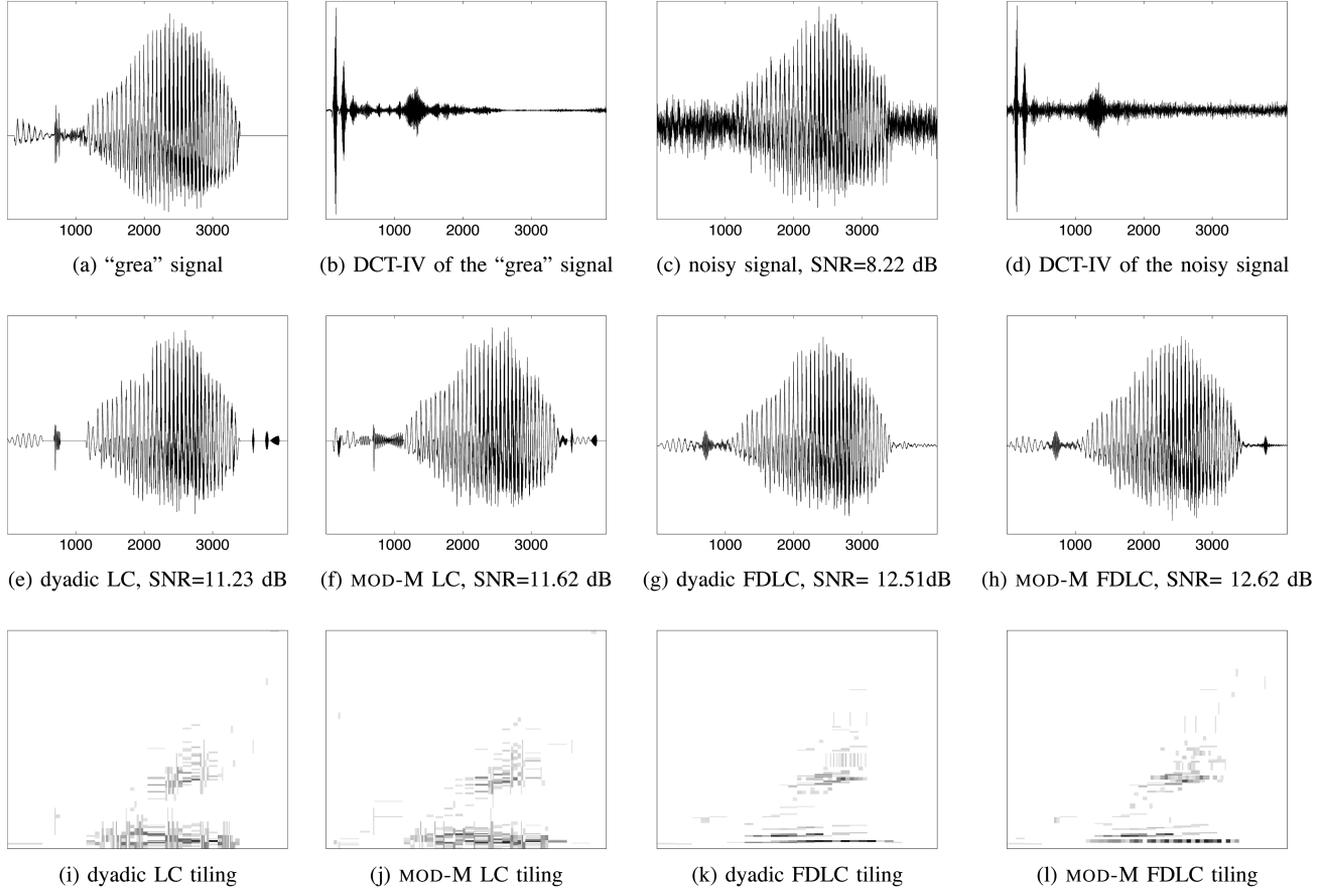


Fig. 9. Best basis thresholding with dyadic and MOD-M local cosine dictionaries in time domain and in frequency domain. The second row shows various estimates of the signal (a) based on its noisy observation (c), and the third row shows the corresponding tilings of the time-frequency plane. From left to right: (e), (i) dyadic local cosine dictionary; (f), (j) MOD-M local cosine dictionary; (g), (k) dyadic FDLC dictionary; (h), (l) MOD-M FDLC dictionary. Tiling, (k) dyadic FDLC tiling, and (l) MOD-M FDLC tiling.

Specifically, for $v - u \geq 2M$, see the equation shown at the bottom of the page, where

$$d^* = \arg \min_{d: u+M \leq d \leq v-M} C(f, \mathcal{B}_{u,d}) + C(f, \mathcal{O}_{d,v}).$$

The initial condition is that for $v - u < 2M$

$$\mathcal{O}_{u,v} = \mathcal{B}_{u,v}.$$

This method reduces the cost more significantly, but the price to pay is more computation. The time complexity of this method is $O(N^3 \log N)$, which is prohibitive for large signals. However, both this method and the MOD-M algorithm of Section II-C can be used on small blocks of a signal, and the results can be combined via a postprocessing step. By varying the size of each block, we can achieve various tradeoffs between the cost

and the time complexity. The next two subsections describe two procedures for blockwise application of our algorithms.

B. Extension 2: BLOCKS Algorithm

We divide a signal f of length N into blocks of equal size M_2 . Let $L_2 = N/M_2$ be the total number of blocks. For each block $f_{lM_2, (l+1)M_2}$, $l = 0, 1, \dots, L_2 - 1$, the best basis can be calculated using either the MOD-M or the MIN-M algorithm. Concatenating the partition points of the best bases calculated for all blocks, we get a partition for the signal

$$t_0 = 0 < t_1 < \dots < t_I = N$$

where I is the total number of segments. Since we imposed partition points at the block boundaries, we need a postprocessing step to remove the artifacts and further optimize the cost by considering the whole signal. The postprocessing will select a set of partition points among the ones we obtained for the blocks. Let

$$\mathcal{O}_{u,v} = \begin{cases} \mathcal{B}_{u,d^*} \cup \mathcal{O}_{d^*,v}, & \text{if } C(f, \mathcal{B}_{u,d^*}) + C(f, \mathcal{O}_{d^*,v}) < C(f, \mathcal{B}_{u,v}) \\ \mathcal{B}_{u,v}, & \text{otherwise.} \end{cases}$$

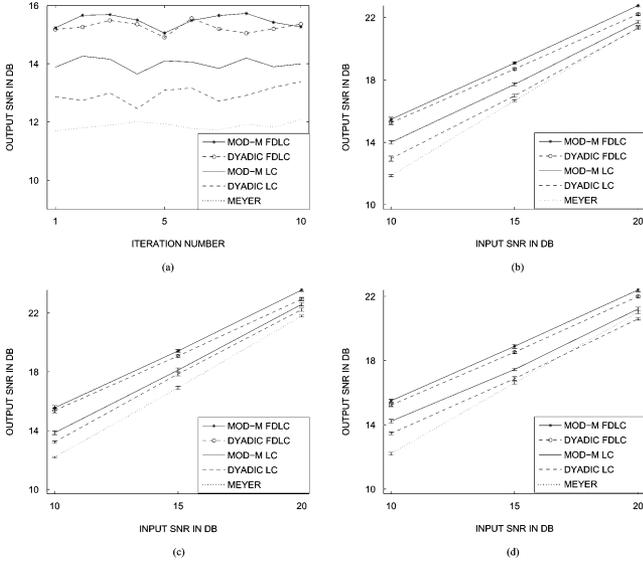


Fig. 10. Monte Carlo simulations for MOD-M LC and FDLC dictionaries, dyadic LC and FDLC dictionaries, and shift-invariant thresholding in Meyer wavelet basis. (a) SNRs for ten experiments for the noisy “zero” speech signal. (b) Average SNRs for three different input SNRs for the noisy “zero” signal. (c) Average SNRs for three different input SNRs for the noisy “one” signal. (d) Average SNRs for three different input SNRs for the noisy “two” signal.

\mathcal{O}_{t_u, t_v} be the best basis for f_{t_u, t_v} after postprocessing. Then the postprocessing can be done using the following recursive formula: when $v - u > 1$, see (9), shown at the bottom of the page, where

$$d^* = \arg \min_{d: u < d < v} C(f, \mathcal{B}_{t_u, t_d}) + C(f, \mathcal{O}_{t_d, t_v}). \quad (10)$$

When $v - u = 1$

$$\mathcal{O}_{t_u, t_v} = \mathcal{B}_{t_u, t_v}. \quad (11)$$

We now calculate the time complexity of the BLOCKS algorithm where the MOD-M algorithm of Section II-C is applied to each block. As mentioned in Section II-C, the calculation of the best basis via the MOD-M algorithm is $O(\bar{L}^2 M_2 \log M_2)$ for each block of size M_2 , where $\bar{L} = M_2/M$. For L_2 blocks, the total time complexity before postprocessing is therefore $O(L_2 \bar{L}^2 M_2 \log M_2) = O(\bar{L}^2 N \log M_2)$. The time complexity of the postprocessing step is calculated similarly to that of the MOD-M algorithm of Section II-C and is $O(I^2 N \log N)$, where I is the number of segments before postprocessing. Since in the worst case, I can be equal to the number of cells L , the worst-case time complexity of the postprocessing step alone is $O(L^2 N \log N)$, which is the same as the complexity of the MOD-M algorithm applied to the whole signal. In practice, however, if appropriate values M_2 are used, it is typical for I to be significantly smaller than L , leading to considerable computational savings. The overall time complexity of the two stages of the BLOCKS algorithm is $O(N \bar{L}^2 \log M_2 + I^2 N \log N) =$

TABLE II
RUNNING TIMES AND COSTS FOR DYADIC AND MOD-M ALGORITHMS

	time	cost
dyadic	0.22s	4.1
MOD-M, Section II-C	53s	3.5

$O(N \bar{L}^2 \log M_2 + L^2 N \log N)$. It can be similarly shown that if the BLOCKS algorithm is used in conjunction with the MIN-M algorithm of Section III-A, the overall time complexity will be $O(N M_2^2 \log M_2 + L^2 N \log N)$.

We illustrate the BLOCKS algorithm on the signal “Grea” whose length is $N = 4096$. We use the MOD-M algorithm within each block. We fix the cell size $M = 16$ and $\eta = 8$, and vary the block size M_2 . The results are shown in Table III(a). The results of dyadic and MOD-M methods with the same M and η are shown in Table II for comparison. By using different values of M_2 , we are able to obtain various tradeoffs between optimizing the cost and minimizing the running time. When M_2 is very large, there is essentially no difference between using BLOCKS and applying MOD-M to the whole signal. In fact, if $M_2 = N$, then the two algorithms produce identical results except the BLOCKS algorithm makes some unnecessary computations. This is illustrated by comparing the last line of Table III(a) and the last line of Table II. When M_2 is very small, the first stage of the BLOCKS algorithm tends to produce many partition points, and the bulk of the computation is done during the postprocessing stage. The algorithm is the fastest for the medium values of M_2 .

The fact that partition points are imposed at the block boundaries contributes to the poor performance of the first stage of the BLOCKS algorithm for small block sizes M_2 . This problem can be ameliorated by using overlapping blocks, as described in the next subsection and illustrated in Table III(b).

C. Extension 3: Overlapping-Blocks Algorithm

We process an N -point signal f using L_2 overlapping blocks that do not necessarily have the same length. We set $M_2 = N/L_2$. We denote the index of the leftmost and the rightmost points of the i th block ($i = 0, \dots, L_2 - 1$) by l_i and $r_i - 1$, respectively (i.e., the block itself is denoted by f_{l_i, r_i}). We fix $r_i = (i + 1)M_2$ and $l_0 = 0$. The point l_{i+1} and the basis $\mathcal{O}_{l_i, l_{i+1}}$ for $f_{l_i, l_{i+1}}$ are recursively found by applying either the MOD-M algorithm or the MIN-M algorithm to f_{l_i, r_i} subject to the constraint that the first partition point of the best basis is to the right of $r_{i-1} - 1$. Denoting the optimal partition points of f_{l_i, r_i} by n_0, n_1, \dots, n_{K_i} , we therefore have

$$n_0 = l_i < r_{i-1} \leq n_1 < \dots < n_{K_i-1} < n_{K_i} = r_i.$$

If $i = L_2 - 1$, we set $l_{i+1} = N$; otherwise, we set $l_{i+1} = n_{K_i-1}$. We let

$$\mathcal{O}_{l_i, l_{i+1}} = \mathcal{B}_{l_i, n_1} \cup \mathcal{B}_{n_1, n_2} \cup \dots \cup \mathcal{B}_{n_{K_i-2}, l_{i+1}}.$$

$$\mathcal{O}_{t_u, t_v} = \begin{cases} \mathcal{B}_{t_u, t_{d^*}} \cup \mathcal{O}_{t_{d^*}, t_v} & \text{if } C(f, \mathcal{B}_{t_u, t_{d^*}}) + C(f, \mathcal{O}_{t_{d^*}, t_v}) < C(f, \mathcal{B}_{t_u, t_v}) \\ \mathcal{B}_{t_u, t_v} & \text{otherwise,} \end{cases} \quad (9)$$

TABLE III
RUNNING TIMES AND COSTS FOR THE (a) BLOCKS ALGORITHM AND (b) OVERLAPPING-BLOCKS ALGORITHM, EACH USED WITH THE MOD-M ALGORITHM

	BLOCKS			
	before	postprocessing	after	postprocessing
M_2	time	cost	time	cost
32	0.16s	5.43	24s	3.54
64	0.19s	5.11	9.4s	3.63
128	0.29s	4.69	3.9s	3.70
256	0.58s	4.18	2.2s	3.75
512	1.3s	3.81	2.3s	3.66
1024	4.2s	3.75	5.0s	3.65
2048	16s	3.59	17s	3.59
4096	54s	3.51	54s	3.51

(A)

	OVERLAPPING-BLOCKS			
	before	postprocessing	after	postprocessing
M_2	time	cost	time	cost
32	0.23s	4.51	3.3s	3.68
64	0.26s	4.32	2.9s	3.68
128	0.37s	4.14	1.5s	3.79
256	0.67s	3.83	1.7s	3.80
512	1.5s	3.59	2.2s	3.59
1024	4.4s	3.61	5.0s	3.61
2048	16s	3.56	17s	3.56
4096	54s	3.51	55s	3.51

(B)

Once this is done for all $i = 0, 1, \dots, L_2 - 1$, we take the overall basis for f to be

$$\mathcal{O}_{l_0, l_1} \cup \mathcal{O}_{l_1, l_2} \cup \dots \cup \mathcal{O}_{l_{L_2-1}, l_{L_2}}.$$

We can again use the postprocessing procedure described in Section III-B, (9)–(11); however, it may not be needed since the Overlapping-Blocks algorithm does not typically result in blocking artifacts.

We illustrate the Overlapping-Blocks algorithm on the signal “grea” whose length is $N = 4096$. We use the MOD-M algorithm within each block. We fix the cell size $M = 16$ and $\eta = 8$ and vary the block size M_2 . The results are shown in Table III(b). The results of dyadic and MOD-M methods with the same M and η are shown in Table II for comparison.

While the worst-case time complexity of the Overlapping-Blocks algorithm can be shown to be the same as that of the MOD-M algorithm, we have observed that, in practice, the Overlapping-Blocks algorithm can be significantly faster for appropriate values of M_2 . The intuition described above for the Blocks algorithm holds here, as well, as illustrated in Table III(b): when M_2 is very large, the first stage takes a long time and when M_2 is very small, the postprocessing takes a long time; the algorithm is the fastest for the medium values of M_2 .

Note that overlapping-blocks is usually faster than blocks, without much difference in the achieved cost. The reason is that it is able to eliminate more partition points during stage 1, and therefore its postprocessing stage typically takes less time. Also note that, for medium and large values of M_2 , the Overlapping-Blocks algorithm does not, in fact, need the postprocessing stage, since postprocessing does not reduce the cost. Dispensing with the postprocessing stage further reduces the running time. In addition, this makes it possible to process the data in a sequential manner: once $\mathcal{O}_{l_i, l_{i+1}}$ is determined, the data for $[l_i, l_{i+1} - 1]$ can be discarded.

IV. BEST BASIS SEARCH IN LAPPED DICTIONARIES

General lapped orthogonal bases [6], [7], [30], [33] are not required to use cosine functions; they may use a more general family of orthogonal functions which satisfies certain symmetry conditions. Moreover, nonsymmetric windows can be used.

Specifically, a window for a discrete interval $[u, v - 1]$ may be defined as follows:

$$\beta_{u,v}^{\eta, \eta'}(t) = \begin{cases} r\left(\frac{t-(u-1/2)}{\eta}\right), & \text{if } u - \frac{1}{2} - \eta \leq t < u - \frac{1}{2} + \eta \\ 1, & \text{if } u - \frac{1}{2} + \eta \leq t < v - \frac{1}{2} - \eta' \\ r\left(\frac{(v-1/2)-t}{\eta'}\right), & \text{if } v - \frac{1}{2} - \eta' \leq t \leq v - \frac{1}{2} + \eta' \\ 0, & \text{otherwise.} \end{cases}$$

where r is a profile function just as in Section II-B, but η is not necessarily equal to η' .

As in Section II-B, we suppose that λ is a set of partition points, but we now assume that each partition point n_k comes with its own profile parameter η_k : $\lambda = \{(n_k, \eta_k)\}_{0 \leq k \leq K_\lambda}$. Then, provided that the functions $e_{\mathcal{X}, n_k, n_{k+1}}$ satisfy the appropriate symmetry and orthogonality properties, and that the partition points are such that only adjacent windows overlap (i.e., $n_{k+1} - n_k \geq \eta_k + \eta_{k+1}$ for all k), it can be shown that

$$\mathcal{B}_{n_k, n_{k+1}}^{\eta_k, \eta_{k+1}} \triangleq \{\beta_{n_k, n_{k+1}}^{\eta_k, \eta_{k+1}}(n) e_{\mathcal{X}, n_k, n_{k+1}}\}_{0 \leq \mathcal{X} < n_{k+1} - n_k}$$

is an orthonormal family, and

$$\mathcal{B}^\lambda \triangleq \bigcup_{k=0}^{K_\lambda-1} \mathcal{B}_{n_k, n_{k+1}}^{\eta_k, \eta_{k+1}}$$

is an orthonormal basis for \mathbb{R}^N [30]. A finite dictionary of such bases may be specified by allowing the same set of valid partitions as in (4) and (5) and restricting all valid profile parameters η_k to a finite set A . By adding the search over the set A , the MOD-M algorithm of Fig. 2 is modified to search for the best basis in this dictionary. The resulting modified algorithm is shown in Fig. 11. In this figure, $\mathcal{O}_{\eta, u, N}$ denotes the best basis associated the window $\beta_{u, N}^{\eta, 0}$, and $C_{\eta, u, N}^*$ denotes the corresponding cost. This modified algorithm is very generic and can be used to perform a best basis search for any MOD-M dictionary consisting of a finite number of lapped orthogonal bases. The extensions of the basic algorithm discussed above also apply to the generic algorithm. The complexity of the generic algorithm will depend on the size of the set A and, more generally, on the complexity of calculating the costs $C(f, \mathcal{B}_{u, d}^{\eta, \eta'})$.

```

 $\mathcal{O}_{0,N} \leftarrow \text{best\_basis}(f) \{$ 
  for  $u = N - M, N - 2M, \dots, 2M, M, 0 \{$ 
    if  $u == 0 \{$ 
       $A \leftarrow \{0\}$  //A Special definition for the leftmost endpoint
    }
    for  $\eta \in A \{$ 
       $\mathcal{O}_{\eta,u,N} \leftarrow \mathcal{B}_{u,N}^{\eta,0};$  //Initialize  $\mathcal{O}_{\eta,u,N}$ 
       $C_{\eta,u,N}^* \leftarrow C(f, \mathcal{B}_{u,N}^{\eta,0});$  //Initialize  $C_{\eta,u,N}^*$ 
      for  $d = u + M, u + 2M, \dots, N - M \{$ 
        for  $\eta' \in A \{$ 
          if  $C(f, \mathcal{B}_{u,d}^{\eta,\eta'}) + C_{\eta',d,N}^* < C_{\eta,u,N}^* \{$ 
             $\mathcal{O}_{\eta,u,N} \leftarrow \mathcal{B}_{u,d}^{\eta,\eta'} \cup \mathcal{O}_{\eta',d,N};$ 
             $C_{\eta,u,N}^* \leftarrow C(f, \mathcal{B}_{u,d}^{\eta,\eta'}) + C_{\eta',d,N}^*;$ 
          }
        }
      }
      save  $\mathcal{O}_{\eta,u,N}$  and  $C_{\eta,u,N}^*$  in an internal data structure;
    }
  }
}
 $\mathcal{O}_{0,N} \leftarrow \mathcal{O}_{0,0,N};$ 
return  $\mathcal{O}_{0,N};$ 
}

```

Fig. 11. Pseudocode specification of a fast dynamic programming algorithm for the best basis search in a lapped dictionary.

V. REMARKS ON PARAMETER SELECTION

Our algorithms use a number of parameters. The issue of parameter selection is an open research issue and is beyond the scope of our paper. Fortunately, these parameters are also used in existing best basis methods as well as other adaptive approximation algorithms, and therefore there exists substantial literature on parameter selection. In our examples, we have used parameter settings taken from the literature on dyadic best bases, demonstrating that our algorithms can produce significant improvements even without any optimization of the parameters. We now briefly sample the existing literature on parameter selection.

A. Shape of the Profile Function r and Window Taper Parameters η and η'

These parameters can be optimized with respect to various criteria, such as approximation of polynomials [5], [35] and good frequency localization [34]. A comprehensive evaluation of the choice of these parameters in the context of compression is provided in [36].

B. Minimum Cell Size M and Block Size M_2

Our examples demonstrate the improved performance of our algorithms, compared to the dyadic best basis search, for a variety of values of M and M_2 . Note that M is akin to the tree depth for dyadic local cosines, or, more generally, for any multiscale transforms such as wavelet or wavelet packet transforms. Selecting M_2 is similar to selecting the block size for a block transform. The selection of such parameters is addressed in any application which uses multiscale or block transforms, such as, for example, the compression of audio signals [23], images [42], [48], [49], or video [23].

C. Cost Function $C(f, B^\lambda)$

The original best basis papers [10], [11] proposed using an entropy cost. Since then, a number of different criteria have been proposed for different applications. For example, some

cost functions stem from optimal estimation frameworks [13], [25], [26], [26], [30], [38], [43], or optimal bit allocation in compression applications [18], [19], [46], [54]. Our methods can be used with any of these cost functions.

D. Threshold T

Based on an estimation-theoretic analysis of the best basis thresholding, the threshold value of $T = \sigma\sqrt{2\ln P}$ was proposed in [13], where P is the number of distinct signals in the dictionary. Many alternative thresholding strategies have been proposed since. Overviews and comparative evaluations of a large number of these methods are found in [1], [4].

VI. CONCLUSION

A large body of literature exists on methods for adaptively selecting vectors [17], [22], [31], bases [8], or subspaces [29] from dictionaries, based on various criteria such as the L^2 norm of the approximation error [31], the L^1 norm of the coefficients [8], and optimal recognition or classification performance [29].

In this paper, we have focused on best basis search algorithms to adaptively compute local cosine decompositions. Simple examples indicate that our algorithms can be applied in time-frequency analysis, compression, and noise removal. In applications where the speed is important, it is possible to use accelerated versions of our algorithms by first applying them to small blocks and then combining the results via a postprocessing step.

We have moreover introduced a new dictionary of FDLCS and showed that it can result in improved representations. We provided a generic version of our algorithms which can be used to find the best basis in any finite dictionary of lapped orthogonal bases.

APPENDIX

A SUBOPTIMAL STRICTLY SHIFT-INVARIANT ALGORITHM

Recall that M is the size of the finest cell we are considering, and $L = N/M$ is the total number of such cells. We let $s = lM + m$ where l is the cell where s appears, $0 \leq l < L$, and m is the position of s within the cell, $0 \leq m < M$. Instead of finding s^* , i.e., optimizing over m and l jointly, we first optimize over m and then optimize over l . We optimize over m using the method described in [12]. For each shift $m = 0, 1, \dots, M - 1$, define the following basis B_m :

$$B_m = \bigcup_{l=0}^{L-1} \mathcal{B}_{lM+m, (l+1)M+m}.$$

We calculate the cost C_m of approximating f with the basis B_m :

$$C_m = \sum_{l=0}^{L-1} C(f, \mathcal{B}_{lM+m, (l+1)M+m}).$$

Then m^* is found by minimizing the cost C_m over m :

$$m^* = \arg \min_{0 \leq m \leq M-1} C_m.$$

Using this value of m^* , we now choose l . Let $\mathcal{O}_{lM+m^*, N+lM+m^*}$ be the best basis for the signal

TABLE IV
TIME COMPLEXITY OF THE ALGORITHMS

Algorithm	Time complexity	Time complexity for fixed M, M_2	Time complexity for fixed L, L_2
Dyadic [10], [11]	$O(\log L \cdot N \log N)$	$O(N(\log N)^2)$	$O(N \log N)$
SI-LCD [12]	$O((M + \log L) \cdot N \log N)$	$O(N(\log N)^2)$	$O(N^2 \log N)$
MOD-M (Section II-C)	$O(L^2 \cdot N \log N)$	$O(N^3 \log N)$	$O(N \log N)$
SI-mod-M (Section II-E)	$O((M + L^2) \cdot N \log N)$	$O(N^3 \log N)$	$O(N^2 \log N)$
MIN-M (Section III-A)	$O(N^2 \cdot N \log N)$	$O(N^3 \log N)$	$O(N^3 \log N)$
BLOCKS with blockwise MOD-M (Section III-B)	$O(N\bar{L}^2 \log M_2 + L^2 \cdot N \log N)$	$O(N^3 \log N)$	$O(N \log N)$
BLOCKS with blockwise MIN-M (Section III-B)	$O(NM_2^2 \log M_2 + L^2 \cdot N \log N)$	$O(N^3 \log N)$	$O(N^3 \log N)$
OVERLAPPING-BLOCKS with blockwise MOD-M (Section III-C)	$O(L^2 \cdot N \log N)$	$O(N^3 \log N)$	$O(N \log N)$
OVERLAPPING-BLOCKS with blockwise MIN-M (Section III-C)	$O(N^2 \cdot N \log N)$	$O(N^3 \log N)$	$O(N^3 \log N)$

$f_{lM+m^*, N+lM+m^*}$ in the subdictionary \mathcal{D}_{lM+m^*} , for $l = 0, 1, \dots, L-1$. The best basis for each l is calculated in the same way as in Section II-C. Then we choose the best l , as follows:

$$l^* = \arg \min_{0 \leq l \leq L-1} C(f, \mathcal{O}_{lM+m^*, N+lM+m^*}).$$

The corresponding (suboptimal) shift-invariant best basis \mathcal{O} for the signal f is $\mathcal{O}_{l^*M+m^*, N+l^*M+m^*}$. Finding m^* requires calculating C_m for $m = 0, 1, \dots, M-1$. Each C_m is the sum of L costs, and calculating each cost requires $O(M \log M)$ operations. So the calculation of m^* requires $O(M \cdot L \cdot M \log M) = O(NM \log M)$ operations. To determine l^* , we do basis searches for L signals of length N using the method of Section II-C. We know that the recursion formula (6) takes $O(L^2)$ operations for one search, so L searches require $O(L^3)$ operations. The major part of the computational burden, however, is still the calculation of the costs in (6). Fortunately, many of the costs used in the L searches are repeated. We only need to compute the values for $C(f, \mathcal{B}_{u+m^*, v+m^*})$ with $u = pM, v = qM$, where $p = 0, 1, \dots, L-1$ and $q = p+1, p+2, \dots, p+L$. It can be shown that this computation has time complexity $O(L^2 N \log N)$, the same as in Section II-C. The two steps taken together, therefore, result in the overall complexity of $O((L^2 + M)N \log N)$, which is similar to the complexity of the basic MOD-M algorithm. For comparison, the algorithm of [12] is $O((\log L + M)N \log N)$.

As remarked above, the “best” basis found by this method, is suboptimal for the enlarged dictionary. However, in this way, we achieved strict shift-invariance essentially without increasing the time complexity of the algorithm.

We now show how to find the optimal basis which is strictly M -shift-invariant. We define the following dictionary:

$$\mathcal{D}_{\text{SIM}} = \bigcup_{s \text{ is a multiple of } M} \mathcal{D}_s$$

where the subdictionaries \mathcal{D}_s are defined as previously. We can then adapt the algorithm we just described to find the best basis in the dictionary \mathcal{D}_{SIM} . Specifically, we omit the step of finding m^* and set $m^* = 0$. The resulting best basis will be optimal for the dictionary \mathcal{D}_{SIM} , and it will be M -shift-invariant. The computational complexity for this procedure is $O(L^2 N \log N)$, the same as that of the MOD-M algorithm.

TIME COMPLEXITY OF VARIOUS ALGORITHMS

In Table IV, we summarize the time complexity of the various algorithms introduced in this paper, as well as the SI-LCD algorithm of [12] and the dyadic tree search of [10], [11]. Here, N is the size of the signal, M is the finest cell size, $L = N/M$, L_2 is the number of blocks, $M_2 = N/L_2$, and $\bar{L} = M_2/M$.

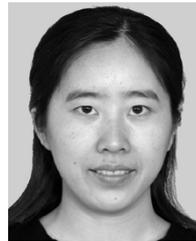
ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for very helpful comments.

REFERENCES

- [1] A. Antoniadis, J. Bigot, and T. Sapatinas, “Wavelet estimators in non-parametric regression: A comparative simulation study,” *J. Stat. Softw.*, vol. 6, no. 6, pp. 1–83, 2001.
- [2] R. Bellman, “On the approximation of curves by line segments using dynamic programming,” *Commun. ACM*, vol. 4, no. 6, p. 284, 1961.
- [3] N. N. Bennett, “Fast algorithm for best anisotropic walsh bases and relatives,” *Appl. Comput. Harmon. Anal.*, vol. 8, pp. 86–103, 2000.
- [4] P. Besbeas, I. De Feis, and T. Sapatinas, “A comparative simulation study of wavelet shrinkage estimators for Poisson counts,” *Int. Stat. Rev.*, vol. 72, pp. 209–237, 2004.
- [5] K. Bittner, “Error estimates and reproduction of polynomials for biorthogonal local trigonometric bases,” *Appl. Comput. Harmon. Anal.*, vol. 6, pp. 75–102, 1996.
- [6] P. M. Cassereau, “A new class of optimal unitary transforms for image processing,” S.M. thesis, Department of Electrical Engineering and Computer Science, MITT, Cambridge, MA, May 1985.
- [7] P. M. Cassereau, D. H. Staelin, and G. De Jager, “Encoding of images based on a lapped orthogonal transform,” *IEEE Trans. Commun.*, vol. 37, no. 2, pp. 189–193, Feb. 1989.
- [8] S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1999.
- [9] R. R. Coifman and Y. Meyer, “Remarques sur l’analyse de Fourier a fenetre,” *Comptes Rendus Acad. Sci.*, pp. 259–261, 1991.
- [10] R. R. Coifman, Y. Meyer, and M. V. Wickerhauser, “Wavelet analysis and signal processing,” in *Wavelets and Their Applications*, M. B. Ruskai et al., Ed. Boston, MA: Jones and Bartlett, 1992, pp. 153–178.
- [11] R. R. Coifman and M. V. Wickerhauser, “Entropy based algorithms for best basis selection,” *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 713–718, Mar. 1992.
- [12] I. Cohen, S. Raz, and D. Malah, “Orthonormal shift-invariant adaptive local trigonometric decomposition,” *Signal Process.*, vol. 57, no. 1, pp. 43–64, Feb. 1997.
- [13] D. L. Donoho and I. M. Johnstone, “Ideal denoising in an orthonormal basis chosen from a library of bases,” *Comptes Rendus Acad. Sci.*, ser. I, vol. 319, pp. 1317–1322, 1994.
- [14] D. L. Donoho, “CART and best-ortho-basis: A connection,” *Ann. Stat.*, vol. 25, pp. 1870–1911, 1997.
- [15] —, “Wedgelets: Nearly minimax estimation of edges,” *Ann. Stat.*, vol. 27, pp. 859–897, 1999.

- [16] D. Donoho, M. R. Duncan, X. Huo, O. Levi, J. Buckheit, M. Clerc, J. Kalifa, S. G. Mallat, and T. Yu. Wavelab 802 (software package). [Online]. Available: www-stat.stanford.edu/~wavelab
- [17] R. Gribonval, P. Depalle, X. Rodet, E. Bacry, and S. Mallat, "Sound signals decomposition using a high resolution matching pursuit," in *Proc. Computer Music Conf. (ICMC)*, Aug. 1996, pp. 293–296.
- [18] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli, "Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3341–3359, Dec. 1993.
- [19] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint space-frequency segmentation using balanced wavelet packet tree for least-cost image representation," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1213–1230, Sep. 1997.
- [20] Y. Huang, I. Pollak, C. A. Bouman, and M. N. Do, "New algorithms for best local cosine basis search," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Montreal, QC, Canada, May 2004, pp. 773–776.
- [21] —, "Time-Frequency analysis with best local cosine bases," in *Proc. SPIE*, vol. 5299, Computational Imaging II, C. A. Bouman and E. L. Miller, Eds., San Jose, CA, Jan. 2004, pp. 187–192.
- [22] S. Jaggi, W. C. Karl, S. Mallat, and A. S. Willsky, "High resolution pursuit for feature extraction," *J. Appl. Comput. Harmon. Anal.*, vol. 5, pp. 428–449, 1998.
- [23] *Coding of Moving Pictures and Audio*, Mar. 2002. ISO/IEC JTC1/SC29/WG11 N4668.
- [24] E. D. Kolaczyk, "Wavelet methods for the inversion of certain homogeneous linear operators in the presence of noisy data," Ph.D. dissertation, Dept. of Statistics, Stanford Univ., Stanford, CA, Oct. 1994.
- [25] H. Krim and J.-C. Pesquet, "On the statistics of best bases criteria," in *Wavelets and Statistics*, A. Antoniadis, Ed: Springer-Verlag, 1995, pp. 193–207. Lecture Notes in Statistics.
- [26] H. Krim, D. Tucker, S. Mallat, and D. Donoho, "On denoising and best signal representation," *IEEE Trans. Inform. Theory*, vol. 45, no. 7, pp. 2225–2238, Nov. 1999.
- [27] M. Lindberg, "Two-dimensional adaptive Haar–Walsh tilings," Licentiate thesis in applied mathematics, Åbo Akademi Univ., Åbo, Finland, Oct. 1999.
- [28] M. Lindberg and L. F. Villemoes, "Image compression with adaptive Haar–Walsh tilings," in *Proc. SPIE*, vol. 4119, Wavelet Applications in Signal Image Processing VIII, 2000.
- [29] X. Liu, A. Srivastava, and K. Gallivan, "Optimal linear representations of images for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 662–666, May 2004.
- [30] S. G. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. New York: Academic, 1999.
- [31] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [32] H. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 6, pp. 969–978, Jun. 1990.
- [33] —, *Signal Processing with Lapped Transforms*. Norwood, MA: Artech House, 1992.
- [34] —, "Biorthogonal and nonuniform lapped transforms for transform coding," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 1043–1053, Apr. 1998.
- [35] G. Matviyenko, "Optimized local trigonometric bases," *Appl. Comput. Harmon. Anal.*, vol. 3, pp. 301–323, 1996.
- [36] F. Meyer, "Image compression with adaptive local cosines: A comparative study," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 616–629, Jun. 2002.
- [37] Y. Meyer, "Principe d'incertitude, bases hilbertiennes et algèbres d'opérateurs," in *Séminaire Bourbaki*, vol. 662, Paris, France, 1986.
- [38] P. Moulin, "Signal estimation using adapted tree-structured bases and the MDL principle," in *Proc. IEEE-SP Int. Symp. Time-Frequency and Time-Scale Analysis (TFTS)*, Paris, France, Jun. 1996, pp. 141–143.
- [39] O. A. Niamut and R. Heusdens, "Flexible frequency decompositions for cosine-modulated filter banks," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, vol. 5, Hong Kong, Apr. 2003, pp. 449–452.
- [40] —, "RD optimal time segmentations for the time-varying MDCT," in *Proc. Eur. Signal Processing Conf. (Eusipco)*, Vienna, Austria, Sep. 6–10, 2004, pp. 1649–1652.
- [41] G. Papakonstantinou, "Optimal polygonal approximation of digital curves," *Signal Process.*, vol. 8, pp. 131–135, 1985.
- [42] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.
- [43] J.-C. Pesquet, H. Krim, D. Leporini, and E. Hamman, "Bayesian approach to best basis selection," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Atlanta, GA, May 1996, pp. 2634–2638.
- [44] J.-C. Perez and E. Vidal, "Optimum polygonal approximation of digitized curves," *Pattern Recogn. Lett.*, vol. 15, pp. 743–750, Aug. 1994.
- [45] P. Prandoni and M. Vetterli, "R/D optimal linear prediction," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 646–655, Nov. 2000.
- [46] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Process.*, vol. 2, no. 2, pp. 160–175, Apr. 1993.
- [47] J. H. Rothweiler, "Polyphase quadrature filters—A new subband coding technique," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Boston, MA, Mar. 1983, pp. 1280–1283.
- [48] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, 1996.
- [49] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.
- [50] C. M. Thiele and L. F. Villemoes, "A fast algorithm for adapted time-frequency tilings," *Appl. Comput. Harmon. Anal.*, vol. 3, pp. 91–99, 1996.
- [51] L. F. Villemoes. (1999, Jun.) Adapted bases of time-frequency local cosines. [Online]. Available: www.math.kth.se/old-home-pages/larsv/publ.html
- [52] M. B. Wakin, J. K. Romberg, H. Choi, and R. G. Baraniuk, "Rate-distortion optimized image compression using wedgelets," in *Proc. Int. Conf. Image Processing (ICIP)*, vol. III, Rochester, New York, Sep. 2002, pp. 237–240.
- [53] R. M. Willett and R. D. Nowak, "Platelets: A multiscale approach for recovering edges and surfaces in photon-limited medical imaging," *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 332–350, Mar. 2003.
- [54] Z. Xiong, K. Ramchandran, C. Herley, and M. T. Orchard, "Flexible tree-structured signal expansions using time-varying wavelet packets," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 333–345, Feb. 1997.



Yan Huang received the B.Eng. degree in electronic engineering from Tsinghua University, Beijing, China, in 2000 and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 2004.

She is currently a Postdoctoral Researcher at the School of Electrical and Computer Engineering, Purdue University. Her research interests are in image and signal processing.



Ilya Pollak (S'99–M'99–SM'05) received the B.S. and M.Eng. degrees in 1995 and Ph.D. degree in 1999, all from the Massachusetts Institute of Technology, Cambridge, all in electrical engineering.

From 1999 to 2000, he was a Postdoctoral Researcher with the Division of Applied Mathematics, Brown University, Providence, RI. Since 2000, he has been Assistant Professor of Electrical and Computer Engineering at Purdue University, West Lafayette, IN. He has held short-term visiting positions at the Institut National de Recherche en

Informatique et en Automatique, Sophia Antipolis, France, and at Tampere University of Technology, Finland. His research interests are in image and signal processing, specifically, hierarchical statistical models, fast estimation algorithms, nonlinear scale spaces, and adaptive representations.

Dr. Pollak received a CAREER award from the National Science Foundation in 2001. He is the Chair of the Signal Processing Chapter of the IEEE Central Indiana Section and a member of the IEEE Technical Committee on Signal Processing Theory and Methods.



Charles A. Bouman (S'86–M'89–SM'97–F'01) received the B.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1981, the M.S. degree from the University of California at Berkeley in 1982, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1989.

From 1982 to 1985, he was a full staff member at MIT Lincoln Laboratory. In 1989, he joined the faculty of Purdue University, West Lafayette, IN, where he is a Professor with a primary appointment in the School of Electrical and Computer Engineering and a

secondary appointment in the School of Biomedical Engineering. His research focuses on the use of statistical image models, multiscale techniques, and fast algorithms in applications including medical and electronic imaging.

Prof. Bouman is a Fellow of the American Institute for Medical and Biological Engineering (AIMBE), a Fellow of the Society for Imaging Science and Technology (IS&T), and a member of The International Society for Optical Engineers (SPIE) professional societies. He is also a recipient of IS&T's Raymond C. Bowman Award for outstanding contributions to digital imaging education and research and a University Faculty Scholar of Purdue University. He is currently the general Co-Chair of the SPIE/IS&T Symposium on Electronic Imaging, secretary of the IEEE Biomedical Image and Signal Processing Technical Committee, and a member of the Steering Committee for the IEEE TRANSACTIONS ON MEDICAL IMAGING. He has been an associate editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has also been the Awards Chair for the International Conference on Image Processing (ICIP) 1998 organizing committee, Co-Chair of the SPIE/IS&T conferences on Visual Communications and Image Processing 2000 (VCIP), and a member of the IEEE Image and Multidimensional Signal Processing Technical Committee, a Vice President of Publications, and a member of the Board of Directors for the IS&T Society. He is the founder and co-chair of the SPIE/IS&T conference on Computational Imaging.



Minh N. Do (M'05) was born in Thanh Hoa, Vietnam, in 1974. He received the B.Eng. degree in computer engineering from the University of Canberra, Australia, in 1997 and the Dr.Sci. degree in communication systems from the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland, in 2001.

Since 2002, he has been an Assistant Professor with the Department of Electrical and Computer Engineering and a Research Assistant Professor with the Coordinated Science Laboratory and the

Beckman Institute, University of Illinois at Urbana-Champaign. His research interests include wavelets, image and multidimensional signal processing, multiscale geometric analysis, and visual information representation.

Dr. Do received a Silver Medal from the 32nd International Mathematical Olympiad in 1991, a University Medal from the University of Canberra in 1997, the best doctoral thesis award from the Swiss Federal Institute of Technology Lausanne in 2001, and a CAREER award from the National Science Foundation in 2003.