SIAM REVIEW
Vol. 43, No. 1, pp. 129–159

Atomic Decomposition by Basis Pursuit*

Scott Shaobing Chen[†]
David L. Donoho[‡]
Michael A. Saunders[§]

Abstract. The time-frequency and time-scale communities have recently developed a large number of overcomplete waveform dictionaries—stationary wavelets, wavelet packets, cosine packets, chirplets, and warplets, to name a few. Decomposition into overcomplete systems is not unique, and several methods for decomposition have been proposed, including the method of frames (MOF), matching pursuit (MP), and, for special dictionaries, the best orthogonal basis (BOB).

Basis pursuit (BP) is a principle for decomposing a signal into an "optimal" superposition of dictionary elements, where *optimal* means having the smallest l^1 norm of coefficients among all such decompositions. We give examples exhibiting several advantages over MOF, MP, and BOB, including better sparsity and superresolution. BP has interesting relations to ideas in areas as diverse as ill-posed problems, abstract harmonic analysis, total variation denoising, and multiscale edge denoising.

BP in highly overcomplete dictionaries leads to large-scale optimization problems. With signals of length 8192 and a wavelet packet dictionary, one gets an equivalent linear program of size 8192 by 212,992. Such problems can be attacked successfully only because of recent advances in linear and quadratic programming by interior-point methods. We obtain reasonable success with a primal-dual logarithmic barrier method and conjugate-gradient solver.

Key words. overcomplete signal representation, denoising, time-frequency analysis, time-scale analysis, ℓ^1 norm optimization, matching pursuit, wavelets, wavelet packets, cosine packets, interior-point methods for linear programming, total variation denoising, multiscale edges, MATLAB code

AMS subject classifications. 94A12, 65K05, 65D15, 41A45

PII. S003614450037906X

1. Introduction.

Over the last several years, there has been an explosion of interest in alternatives to traditional signal representations. Instead of just representing signals as superpositions of sinusoids (the traditional Fourier representation) we now have available alternate dictionaries—collections of parameterized waveforms—of which the wavelets dictionary is only the best known. Wavelets, steerable wavelets, segmented wavelets, Gabor dictionaries, multiscale Gabor dictionaries, wavelet pack-

[*] Published electronically February 2, 2001. This paper originally appeared in *SIAM Journal on Scientific Computing*, Volume 20, Number 1, 1998, pages 33–61. This research was partially supported by NSF grants DMS-92-09130, DMI-92-04208, and ECS-9707111, by the NASA Astrophysical Data Program, by ONR grant N00014-90-J1242, and by other sponsors.

http://www.siam.org/journals/sirev/43-1/37906.html

[†] Renaissance Technologies, 600 Route 25A, East Setauket, NY 11733 (schen@rentec.com).

[‡] Department of Statistics, Stanford University, Stanford, CA 94305 (donoho@stat.stanford.edu).

[§] Department of Management Science and Engineering, Stanford University, Stanford, CA 94305 (saunders@stanford.edu).

ets, cosine packets, chirplets, warplets, and a wide range of other dictionaries are now available. Each such dictionary $\mathcal{D}$ is a collection of waveforms $(\phi_\gamma)_{\gamma\in\Gamma}$, with γ a parameter, and we envision a decomposition of a signal $\mathbf{s}$ as

$$\mathbf{s} = \sum_{\gamma\in\Gamma} \alpha_\gamma \phi_\gamma, \tag{1.1}$$

or an approximate decomposition

$$\mathbf{s} = \sum_{i=1}^{m} \alpha_{\gamma_i} \phi_{\gamma_i} + R^{(m)}, \tag{1.2}$$

where $R^{(m)}$ is a residual. Depending on the dictionary, such a representation decomposes the signal into pure tones (Fourier dictionary), bumps (wavelet dictionary), chirps (chirplet dictionary), etc.

Most of the new dictionaries are *overcomplete*, either because they start out that way or because we merge complete dictionaries, obtaining a new megadictionary consisting of several types of waveforms (e.g., Fourier and wavelets dictionaries). The decomposition (1.1) is then nonunique, because some elements in the dictionary have representations in terms of other elements.

1.1. Goals of Adaptive Representation. Nonuniqueness gives us the possibility of adaptation, i.e., of choosing from among many representations one that is most suited to our purposes. We are motivated by the aim of achieving simultaneously the following *goals*.

- *Sparsity.* We should obtain the sparsest possible representation of the object—the one with the fewest significant coefficients.
- *Superresolution.* We should obtain a resolution of sparse objects that is much higher resolution than that possible with traditional nonadaptive approaches.

An important *constraint*, which is perhaps in conflict with both the goals, follows.

- *Speed.* It should be possible to obtain a representation in order $O(n)$ or $O(n\log(n))$ time.

1.2. Finding a Representation. Several methods have been proposed for obtaining signal representations in overcomplete dictionaries. These range from general approaches, like the method of frames (MOF) [9] and the method of matching pursuit (MP) [29], to clever schemes derived for specialized dictionaries, like the method of best orthogonal basis (BOB) [7]. These methods are described briefly in section 2.3.

In our view, these methods have both advantages and shortcomings. The principal emphasis of the proposers of these methods is on achieving sufficient computational speed. While the resulting methods are practical to apply to real data, we show below by computational examples that the methods, either quite generally or in important special cases, lack qualities of sparsity preservation and of stable superresolution.

1.3. Basis Pursuit. Basis pursuit (BP) finds signal representations in overcomplete dictionaries by convex optimization: it obtains the decomposition that minimizes the ℓ^1 norm of the coefficients occurring in the representation. Because of the nondifferentiability of the ℓ^1 norm, this optimization principle leads to decompositions that can have very different properties from the MOF—in particular, they can be much sparser. Because it is based on global optimization, it can stably superresolve in ways that MP cannot.

BP can be used with noisy data by solving an optimization problem trading off a quadratic misfit measure with an ℓ^1 norm of coefficients. Examples show that it can stably suppress noise while preserving structure that is well expressed in the dictionary under consideration.

BP is closely connected with linear programming. Recent advances in large-scale linear programming—associated with interior-point methods—can be applied to BP and can make it possible, with certain dictionaries, to nearly solve the BP optimization problem in nearly linear time. We have implemented primal-dual log barrier interior-point methods as part of a MATLAB [31] computing environment called Atomizer, which accepts a wide range of dictionaries. Instructions for Internet access to Atomizer are given in section 7.3. Experiments with standard time-frequency dictionaries indicate some of the potential benefits of BP. Experiments with some nonstandard dictionaries, like the stationary wavelet dictionary and the heaviside dictionary, indicate important connections between BP and methods like Mallat and Zhong's [29] multiscale edge representation and Rudin, Osher, and Fatemi's [35] total variation-based denoising methods.

1.4. Contents. In section 2 we establish vocabulary and notation for the rest of the article, describing a number of dictionaries and existing methods for overcomplete representation. In section 3 we discuss the principle of BP and its relations to existing methods and to ideas in other fields. In section 4 we discuss methodological issues associated with BP, in particular some of the interesting nonstandard ways it can be deployed. In section 5 we describe BP denoising, a method for dealing with problem (1.2). In section 6 we discuss recent advances in large-scale linear programming (LP) and resulting algorithms for BP.

For reasons of space we refer the reader to [4] for a discussion of related work in statistics and analysis.

2. Overcomplete Representations. Let $\mathbf{s} = (s_t : 0 \leq t < n)$ be a discrete-time signal of length n; this may also be viewed as a vector in $\mathbf{R}^n$. We are interested in the reconstruction of this signal using superpositions of elementary waveforms. Traditional methods of analysis and reconstruction involve the use of orthogonal bases, such as the Fourier basis, various discrete cosine transform bases, and orthogonal wavelet bases. Such situations can be viewed as follows: given a list of n waveforms, one wishes to represent $\mathbf{s}$ as a linear combination of these waveforms. The waveforms in the list, viewed as vectors in $\mathbf{R}^n$, are linearly independent, and so the representation is unique.

2.1. Dictionaries and Atoms. A considerable focus of activity in the recent signal processing literature has been the development of signal representations outside the basis setting. We use terminology introduced by Mallat and Zhang [29]. A dictionary is a collection of parameterized waveforms $\mathcal{D} = (\phi_\gamma : \gamma \in \Gamma)$. The waveforms ϕ_γ are discrete-time signals of length n called *atoms*. Depending on the dictionary, the parameter γ can have the interpretation of indexing frequency, in which case the dictionary is a frequency or Fourier dictionary, of indexing time-scale jointly, in which case the dictionary is a time-scale dictionary, or of indexing time-frequency jointly, in which case the dictionary is a time-frequency dictionary. Usually dictionaries are complete or overcomplete, in which case they contain exactly n atoms or more than n atoms, but one could also have continuum dictionaries containing an infinity of atoms and undercomplete dictionaries for special purposes, containing fewer than n atoms. Dozens of interesting dictionaries have been proposed over the last few years; we focus in this paper on a half dozen or so; much of what we do applies in other cases as well.

2.1.1. Trivial Dictionaries. We begin with some overly simple examples. The *Dirac* dictionary is simply the collection of waveforms that are zero except in one point: $\gamma \in \{0, 1, \ldots, n-1\}$ and $\phi_\gamma(t) = 1_{\{t=\gamma\}}$. This is of course also an orthogonal basis of $\mathbf{R}^n$—the standard basis. The *heaviside* dictionary is the collection of waveforms that jump at one particular point: $\gamma \in \{0, 1, \ldots, n-1\}$; $\phi_\gamma(t) = 1_{\{t \geq \gamma\}}$. Atoms in this dictionary are not orthogonal, but every signal has a representation

$$\mathbf{s} = s_0 \phi_0 + \sum_{\gamma=1}^{n-1} (s_\gamma - s_{\gamma-1}) \phi_\gamma. \tag{2.1}$$

2.1.2. Frequency Dictionaries. A Fourier dictionary is a collection of sinusoidal waveforms ϕ_γ indexed by $\gamma = (\omega, \nu)$, where $\omega \in [0, 2\pi)$ is an angular frequency variable and $\nu \in \{0, 1\}$ indicates phase type: sine or cosine. In detail,

$$\phi_{(\omega,0)} = \cos(\omega t), \qquad \phi_{(\omega,1)} = \sin(\omega t).$$

For the *standard* Fourier dictionary, we let γ run through the set of all cosines with Fourier frequencies $\omega_k = 2\pi k/n$, $k = 0, \ldots, n/2$, and all sines with Fourier frequencies ω_k, $k = 1, \ldots, n/2 - 1$. This dictionary consists of n waveforms; it is in fact a basis, and a very simple one: the atoms are all mutually orthogonal. An *overcomplete* Fourier dictionary is obtained by sampling the frequencies more finely. Let ℓ be a whole number > 1 and let Γ_ℓ be the collection of all cosines with $\omega_k = 2\pi k/(\ell n)$, $k = 0, \ldots, \ell n/2$, and all sines with frequencies ω_k, $k = 1, \ldots, \ell n/2 - 1$. This is an ℓ-fold overcomplete system. We also use complete and overcomplete dictionaries based on discrete cosine transforms and sine transforms.

2.1.3. Time-Scale Dictionaries. There are several types of wavelet dictionaries; to fix ideas, we consider the Haar dictionary with "father wavelet" $\varphi = 1_{[0,1]}$ and "mother wavelet" $\psi = 1_{(1/2,1]} - 1_{[0,1/2]}$. The dictionary is a collection of translations and dilations of the basic mother wavelet, together with translations of a father wavelet. It is indexed by $\gamma = (a, b, \nu)$, where $a \in (0, \infty)$ is a scale variable, $b \in [0, n]$ indicates location, and $\nu \in \{0, 1\}$ indicates gender. In detail,

$$\phi_{(a,b,1)} = \psi(a(t-b)) \cdot \sqrt{a}, \qquad \phi_{(a,b,0)} = \varphi(a(t-b)) \cdot \sqrt{a}.$$

For the *standard* Haar dictionary, we let γ run through the discrete collection of mother wavelets with dyadic scales $a_j = 2^j/n$, $j = j_0, \ldots, \log_2(n) - 1$, and locations that are integer multiples of the scale $b_{j,k} = k \cdot a_j$, $k = 0, \ldots, 2^j - 1$, and the collection of father wavelets at the coarse scale j_0. This dictionary consists of n waveforms; it is an orthonormal basis. An *overcomplete* wavelet dictionary is obtained by sampling the locations more finely: one location per sample point. This gives the so-called stationary Haar dictionary, consisting of $O(n \log_2(n))$ waveforms. It is called *stationary* since the whole dictionary is invariant under circulant shift.

A variety of other wavelet bases are possible. The most important variations are smooth wavelet bases, using splines or using wavelets defined recursively from two-scale filtering relations [10]. Although the rules of construction are more complicated (boundary conditions [33], orthogonality versus biorthogonality [10], etc.), these have the same indexing structure as the standard Haar dictionary. In this paper, we use *symmlet*-8 smooth wavelets, i.e., Daubechies nearly symmetric wavelets with eight vanishing moments; see [10] for examples.

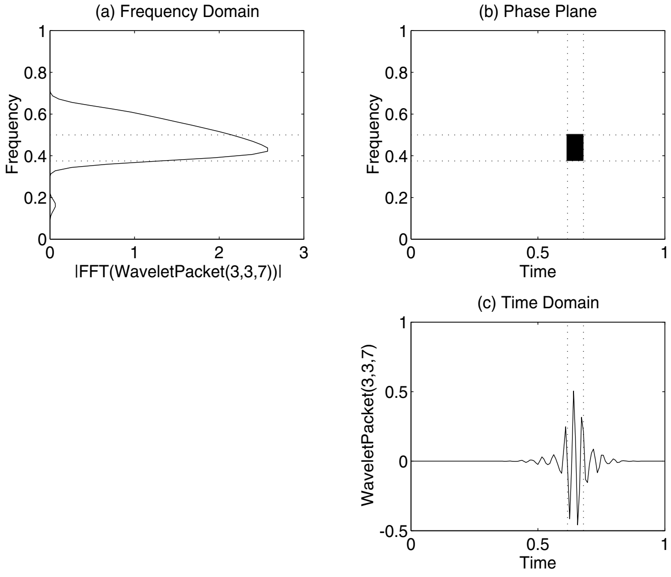

Fig. 2.1 *Time-frequency phase plot of a wavelet packet atom.*

2.1.4. Time-Frequency Dictionaries. Much recent activity in the wavelet communities has focused on the study of time-frequency phenomena. The standard example, the Gabor dictionary, is due to Gabor [19]; in our notation, we take $\gamma = (\omega, \tau, \theta, \delta t)$, where $\omega \in [0, \pi)$ is a frequency, τ is a location, θ is a phase, and δt is the duration, and we consider atoms $\phi_\gamma(t) = \exp\{-(t-\tau)^2/(\delta t)^2\} \cdot \cos(\omega(t-\tau)+\theta)$. Such atoms indeed consist of frequencies near ω and essentially vanish far away from τ. For fixed δt, discrete dictionaries can be built from time-frequency lattices, $\omega_k = k\Delta\omega$ and $\tau_\ell = \ell\Delta\tau$, and $\theta \in \{0, \pi/2\}$; with $\Delta\tau$ and $\Delta\omega$ chosen sufficiently fine these are complete. For further discussions see, e.g., [9].

Recently, Coifman and Meyer [6] developed the wavelet packet and cosine packet dictionaries especially to meet the computational demands of discrete-time signal processing. For one-dimensional discrete-time signals of length n, these dictionaries each contain about $n \log_2(n)$ waveforms. A wavelet packet dictionary includes, as special cases, a standard orthogonal wavelets dictionary, the Dirac dictionary, and a collection of oscillating waveforms spanning a range of frequencies and durations. A cosine packet dictionary contains, as special cases, the standard orthogonal Fourier dictionary and a variety of Gabor-like elements: sinusoids of various frequencies weighted by windows of various widths and locations.

In this paper, we often use wavelet packet and cosine packet dictionaries as examples of overcomplete systems, and we give a number of examples decomposing signals into these time-frequency dictionaries. A simple block diagram helps us visualize the atoms appearing in the decomposition. This diagram, adapted from Coifman and Wickerhauser [7], associates with each cosine packet or wavelet packet a rectangle in the time-frequency phase plane. The association is illustrated in Figure 2.1 for a certain wavelet packet. When a signal is a superposition of several such waveforms, we indicate which waveforms appear in the superposition by shading the corresponding rectangles in the time-frequency plane.

2.1.5. Further Dictionaries. We can always merge dictionaries to create mega-dictionaries; examples used below include mergers of wavelets with heavisides.

2.2. Linear Algebra. Suppose we have a discrete dictionary of p waveforms and we collect all these waveforms as columns of an n-by-p matrix Φ, say. The decomposition problem (1.1) can be written

$$\Phi\alpha = \mathbf{s}, \tag{2.2}$$

where $\alpha = (\alpha_\gamma)$ is the vector of coefficients in (1.1). When the dictionary furnishes a basis, then Φ is an n-by-n nonsingular matrix and we have the unique representation $\alpha = \Phi^{-1}\mathbf{s}$. When the atoms are, in addition, mutually orthonormal, then $\Phi^{-1} = \Phi^T$ and the decomposition formula is very simple.

2.2.1. Analysis versus Synthesis. Given a dictionary of waveforms, one can distinguish *analysis* from *synthesis*. *Synthesis* is the operation of building up a signal by superposing atoms; it involves a matrix that is n-by-p: $\mathbf{s} = \Phi\alpha$. *Analysis* involves the operation of associating with each signal a vector of coefficients attached to atoms; it involves a matrix that is p-by-n: $\tilde{\alpha} = \Phi^T\mathbf{s}$. Synthesis and analysis are very different linear operations, and we must take care to distinguish them. One should avoid assuming that the analysis operator $\tilde{\alpha} = \Phi^T\mathbf{s}$ gives us coefficients that can be used as is to synthesize $\mathbf{s}$. In the overcomplete case we are interested in, $p \gg n$ and Φ is not invertible. There are then many solutions to (2.2), and a given approach selects a particular solution. One does *not* uniquely and automatically solve the synthesis problem by applying a simple, linear analysis operator.

We now illustrate the difference between synthesis ($\mathbf{s} = \Phi\alpha$) and analysis ($\tilde{\alpha} = \Phi^T\mathbf{s}$). Figure 2.2a shows the signal `Carbon`. Figure 2.2b shows the time-frequency structure of a sparse synthesis of `Carbon`, a vector α yielding $\mathbf{s} = \Phi\alpha$, using a wavelet packet dictionary. To visualize the decomposition, we present a phase-plane display with shaded rectangles, as described above. Figure 2.2c gives an analysis of `Carbon`, with the coefficients $\tilde{\alpha} = \Phi^T\mathbf{s}$, again displayed in a phase plane. Once again, between analysis and synthesis there is a large difference in sparsity. In Figure 2.2d we compare the sorted coefficients of the overcomplete representation (synthesis) with the analysis coefficients.

2.2.2. Computational Complexity of Φ and Φ^T. Different dictionaries can impose drastically different computational burdens. In this paper we report computational experiments on a variety of signals and dictionaries. We study primarily one-dimensional signals of length n, where n is several thousand. Signals of this length occur naturally in the study of short segments of speech (a quarter-second to a half-second) and in the output of various scientific instruments (e.g., FT-NMR spectrometers). In our experiments, we study dictionaries overcomplete by substantial factors, say, 10. Hence the typical matrix Φ we are interested in is of size “thousands” by “tens-of-thousands.”

The nominal cost of storing and applying an *arbitrary* n-by-p matrix to a p-vector is a constant times np. Hence with an *arbitrary* dictionary of the sizes we are interested in, simply to *verify* whether (1.1) holds for given vectors α and s would require tens of millions of multiplications and tens of millions of words of memory. In contrast, most signal processing algorithms for signals of length 1000 require only thousands of memory words and a few thousand multiplications.

Fortunately, certain dictionaries have *fast implicit algorithms*. By this we mean that $\Phi\alpha$ and $\Phi^T\mathbf{s}$ can be computed, for arbitrary vectors α and $\mathbf{s}$, (a) without ever

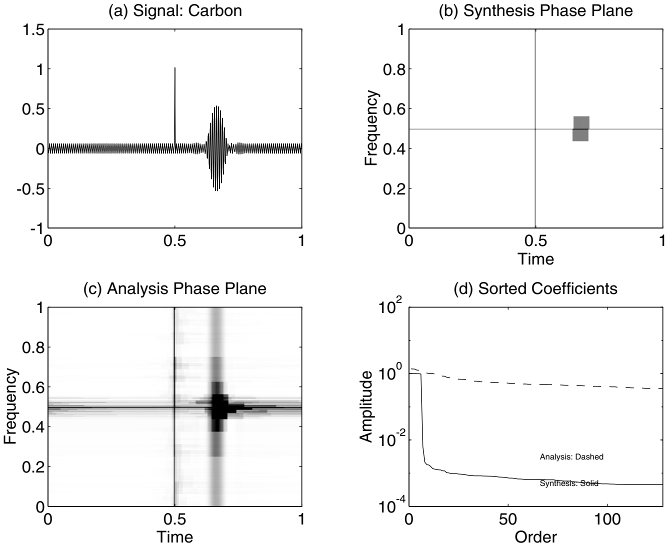

Fig. 2.2 *Analysis versus synthesis of the signal* `Carbon`*.*

storing the matrices Φ and Φ^T, and (b) using special properties of the matrices to accelerate computations.

The most well-known example is the standard Fourier dictionary for which we have the fast Fourier transform algorithm. A typical implementation requires $2 \cdot n$ storage locations and $4{\cdot}n{\cdot}J$ multiplications if n is dyadic: $n = 2^J$. Hence for very long signals we can apply Φ and Φ^T with much less storage and time than the matrices would nominally require. Simple adaptation of this idea leads to an algorithm for overcomplete Fourier dictionaries.

Wavelets give a more recent example of a dictionary with a fast implicit algorithm; if the Haar or S8-symmlet is used, both Φ and Φ^T may be applied in $O(n)$ time. For the stationary wavelet dictionary, $O(n \log(n))$ time is required. Cosine packets and wavelet packets also have fast implicit algorithms. Here both Φ and Φ^T can be applied in order $O(n \log(n))$ time and order $O(n \log(n))$ space—much better than the nominal $np = n^2 \log_2(n)$ one would expect from naive use of the matrix definition.

For the viewpoint of this paper, it only makes sense to consider dictionaries with fast implicit algorithms. Among dictionaries we have not discussed, such algorithms may or may not exist.

2.3. Existing Decomposition Methods.

There are several currently popular approaches to obtaining solutions to (2.2).

2.3.1. Frames.

The MOF [9] picks out, among all solutions of (2.2), one whose coefficients have minimum l^2 norm:

$$\min \|\alpha\|_2 \quad \text{subject to} \quad \Phi\alpha = \mathbf{s}. \tag{2.3}$$

The solution of this problem is unique; label it $\alpha^\dagger$. Geometrically, the collection of all solutions to (2.2) is an affine subspace in $\mathbf{R}^p$; MOF selects the element of this subspace closest to the origin. It is sometimes called a minimum-length solution. There is a

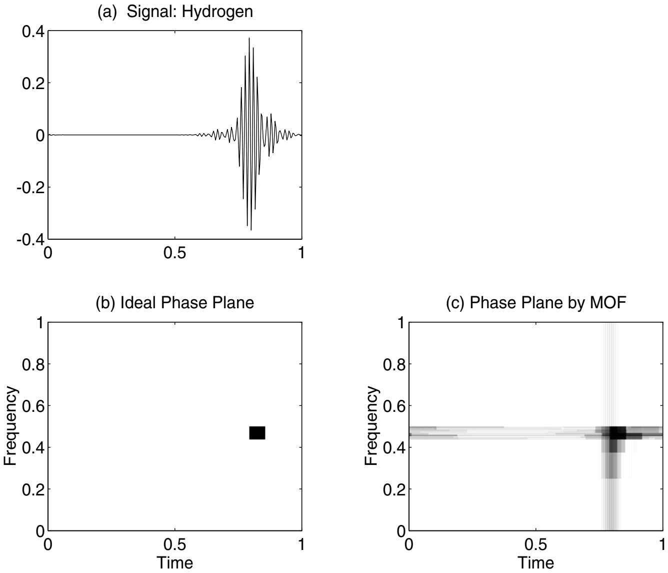

Fig. 2.3 *MOF representation is not sparse.*

matrix $\Phi^\dagger$, the generalized inverse of Φ, that calculates the minimum-length solution to a system of linear equations:

$$\alpha^\dagger = \Phi^\dagger \mathbf{s} = \Phi^T(\Phi\Phi^T)^{-1}\mathbf{s}. \tag{2.4}$$

For so-called tight frame dictionaries MOF is available in closed form. A nice example is the standard wavelet packet dictionary. One can compute that for all vectors $\mathbf{v}$, $\|\Phi^T\mathbf{v}\|^2 = L_n \cdot \|\mathbf{v}\|^2$, $L_n = \log_2(n)$. In short $\Phi^\dagger = L_n^{-1}\Phi^T$. Notice that Φ^T is simply the analysis operator.

There are two key problems with the MOF. First, MOF is not *sparsity preserving.* If the underlying object has a very sparse representation in terms of the dictionary, then the coefficients found by MOF are likely to be very much less sparse. Each atom in the dictionary that has nonzero inner product with the signal is, at least potentially and also usually, a member of the solution.

Figure 2.3a shows the signal `Hydrogen` made of a single atom in a wavelet packet dictionary. The result of a frame decomposition in that dictionary is depicted in a phase-plane portrait; see Figure 2.3c. While the underlying signal can be synthesized from a single atom, the frame decomposition involves many atoms, and the phase-plane portrait exaggerates greatly the intrinsic complexity of the object.

Second, MOF is intrinsically *resolution limited.* No object can be reconstructed with features sharper than those allowed by the underlying operator $\Phi^\dagger\Phi$. Suppose the underlying object is sharply localized: $\alpha = 1_{\{\gamma=\gamma_0\}}$. The reconstruction will not be α, but instead $\Phi^\dagger\Phi\alpha$, which, in the overcomplete case, will be spatially spread out. Figure 2.4 presents a signal `TwinSine` consisting of the superposition of two sinusoids that are separated by less than the so-called Rayleigh distance $2\pi/n$. We analyze these in a fourfold overcomplete discrete cosine dictionary. In this case, reconstruction by MOF (Figure 2.4b) is simply convolution with the Dirichlet kernel. The result is the synthesis from coefficients with a broad oscillatory appearance, consisting not of two

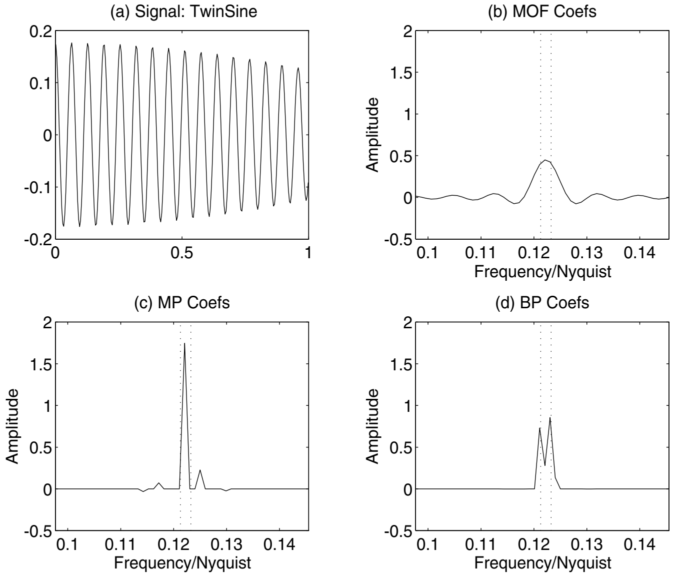

Fig. 2.4 *Analyzing* `TwinSine` *with a fourfold overcomplete discrete cosine dictionary.*

but of many frequencies and giving no visual clue that the object may be synthesized from two frequencies alone.

2.3.2. Matching Pursuit. Mallat and Zhang [29] discussed a general method for approximate decomposition (1.2) that addresses the sparsity issue directly. Starting from an initial approximation $s^{(0)} = 0$ and residual $R^{(0)} = \mathbf{s}$, it builds up a sequence of sparse approximations stepwise. At stage k, it identifies the dictionary atom that best correlates with the residual and then adds to the current approximation a scalar multiple of that atom, so that $\mathbf{s}^{(k)} = \mathbf{s}^{(k-1)} + \alpha_k \phi_{\gamma_k}$, where $\alpha_k = \langle R^{(k-1)}, \phi_{\gamma_k} \rangle$ and $R^{(k)} = \mathbf{s} - \mathbf{s}^{(k)}$. After m steps, one has a representation of the form (1.2), with residual $R = R^{(m)}$. Similar algorithms were proposed by Qian and Chen [39] for Gabor dictionaries and by Villemoes [48] for Walsh dictionaries. A similar algorithm was proposed for Gabor dictionaries by Qian and Chen [39]. For an earlier instance of a related algorithm, see [5].

An intrinsic feature of the algorithm is that when stopped after a few steps, it yields an approximation using only a few atoms. When the dictionary is orthogonal, the method works perfectly. If the object is made up of only $m \ll n$ atoms and the algorithm is run for m steps, it recovers the underlying sparse structure exactly.

When the dictionary is not orthogonal, the situation is less clear. Because the algorithm is myopic, one expects that, in certain cases, it might choose wrongly in the first few iterations and end up spending most of its time correcting for any mistakes made in the first few terms. In fact this does seem to happen.

To see this, we consider an attempt at superresolution. Figure 2.4a portrays again the signal `TwinSine` consisting of sinusoids at two closely spaced frequencies. When MP is applied in this case (Figure 2.4c), using the fourfold overcomplete discrete cosine dictionary, the initial frequency selected is in between the two frequencies making up the signal. Because of this mistake, MP is forced to make a series of alternating corrections that suggest a highly complex and organized structure. MP

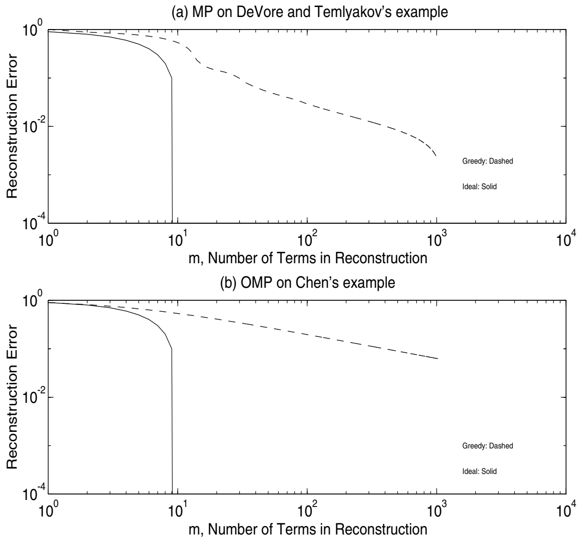

Fig. 2.5 *Counterexamples for MP.*

misses entirely the doublet structure. One can certainly say in this case that MP has failed to superresolve.

Second, one can give examples of dictionaries and signals where MP is arbitrarily suboptimal in terms of sparsity. While these are somewhat artificial, they have a character not so different from the superresolution example.

DeVore and Temlyakov's Example. Vladimir Temlyakov, in a talk at the IEEE Conference on Information Theory and Statistics in October 1994, described an example in which the straightforward greedy algorithm is not sparsity preserving. In our adaptation of this example, based on Temlyakov's joint work with DeVore [12], one constructs a dictionary having $n+1$ atoms. The first n are the Dirac basis; the final atom involves a linear combination of the first n with decaying weights. The signal $\mathbf{s}$ has an exact decomposition in terms of A atoms, but the greedy algorithm goes on forever, with an error of size $O(1/\sqrt{m})$ after m steps. We illustrate this decay in Figure 2.5a. For this example we set $A = 10$ and choose the signal $\mathbf{s}_t = 10^{-1/2} \cdot 1_{\{1 \le t \le 10\}}$. The dictionary consists of Dirac elements $\phi_\gamma = \delta_\gamma$ for $1 \le \gamma \le n$ and

$$\phi_{n+1}(t) = \begin{cases} c, & 1 \le t \le 10, \\ c/(t-10), & 10 < t \le n, \end{cases}$$

with c chosen to normalize ϕ_{n+1} to unit norm.

Shaobing Chen's Example. The DeVore–Temlyakov example applies to the original MP algorithm as announced by Mallat and Zhang in 1992. A later refinement of the algorithm (see Pati, Rezaiifar, and Krishnaprasad [38] and Davis, Mallat, and Zhang [11]) involves an extra step of orthogonalization. One takes all m terms that have entered at stage m and solves the least-squares problem

$$\min_{(\alpha_i)} \left\| \mathbf{s} - \sum_{i=1}^{m} \alpha_i \phi_{\gamma_i} \right\|_2$$

for coefficients $(\alpha_i^{(m)})$. Then one forms the residual $\bar{R}^{[m]} = \mathbf{s} - \sum_{i=1}^{m} \alpha_i^{(m)} \phi_{\gamma_i}$, which will be orthogonal to all terms currently in the model. This method was called *orthogonal matching pursuit* (OMP) by Pati, Rezaiifar, and Krishnaprasad [38]. The DeVore–Temlyakov example does not apply to OMP, but in 1993 Shaobing Chen found a similar example that does. In this example, a special signal and dictionary are constructed, with the following flavor. The dictionary is composed of atoms ϕ_γ with $\gamma \in \{1, \ldots, n\}$. The first A atoms come from the Dirac dictionary with $\gamma \in \{1, \ldots, A\}$, $\phi_\gamma = \delta_\gamma$. The signal is a simple equiweighted linear combination of the first A atoms: $\mathbf{s} = A^{-1} \sum_{i=1}^{A} \phi_i$. Dictionary atoms with $\gamma > A$ are a linear combination of the corresponding Dirac δ_γ and $\mathbf{s}$. OMP chooses all atoms *except* the first A before ever choosing one of the first A. As a result, instead of the ideal behavior one might hope for, terminating after just A steps, one gets n steps before convergence, and the rate is relatively slow. We illustrate the behavior of the reconstruction error in Figure 2.5b. We chose $A = 10$ and $n = 1024$. The dictionary was $\phi_i = \delta_i$ for $1 \leq i \leq 10$ and $\phi_i = \sqrt{a}\mathbf{s} + \sqrt{1-a}e_i$ for $11 \leq i \leq n$, where $a = 2/10$. With these parameters, $\|\bar{R}^{[m]}\|_2 = (1-a)/\sqrt{1+(m-1)a}$, whereas one might have hoped for the ideal behavior $\bar{R}^{[m]} = 0, m \geq 11$.

2.3.3. Best Orthogonal Basis. For certain dictionaries, it is possible to develop specific decomposition schemes custom tailored to the dictionary.

Wavelet packet and cosine packet dictionaries are examples; they have very special properties. Certain special subcollections of the elements in these dictionaries amount to orthogonal bases; in this way one gets a wide range of orthonormal bases (in fact $\gg 2^n$ such orthogonal bases for signals of length n).

Coifman and Wickerhauser [7] have proposed a method of adaptively picking from among these many bases a single orthogonal basis that is the "best basis." If $(s[\mathcal{B}]_I)_I$ denotes the vector of coefficients of s in orthogonal basis $\mathcal{B}$, and if we define the "entropy" $\mathcal{E}(s[\mathcal{B}]) = \sum_I e(s[\mathcal{B}]_I)$, where $e(s)$ is a scalar function of a scalar argument, they give a fast algorithm for solving

$$\min \{\mathcal{E}(s[\mathcal{B}]) : \mathcal{B} \text{ ortho basis} \subset \mathcal{D}\}.$$

The algorithm in some cases delivers near-optimal sparsity representations. In particular, when the object in question has a sparse representation in an orthogonal basis taken from the library, one expects that BOB will work well. However, when the signal is composed of a moderate number of highly nonorthogonal components, the method may not deliver sparse representations—the demand that BOB find an orthogonal basis prevents it from finding a highly sparse representation. An example comes from the signal `WernerSorrows`, which is a superposition of several chirps, sinusoids, and Diracs; see Figure 2.6a. When analyzed with a cosine packet dictionary and the original Coifman–Wickerhauser entropy, BOB finds nothing: it chooses a global sinusoid basis as best. The lack of time-varying structure in that basis means that all chirp and transient structure in the signal is missed entirely; see Figure 2.6b.

3. BP. We now discuss our approach to the problem of overcomplete representations. We assume that the dictionary is overcomplete, so that there are in general many representations $\mathbf{s} = \sum_\gamma \alpha_\gamma \phi_\gamma$.

The principle of BP is to find a representation of the signal whose coefficients have minimal ℓ^1 norm. Formally, one solves the problem

$$\min \|\alpha\|_1 \quad \text{subject to} \quad \Phi\alpha = \mathbf{s}. \tag{3.1}$$

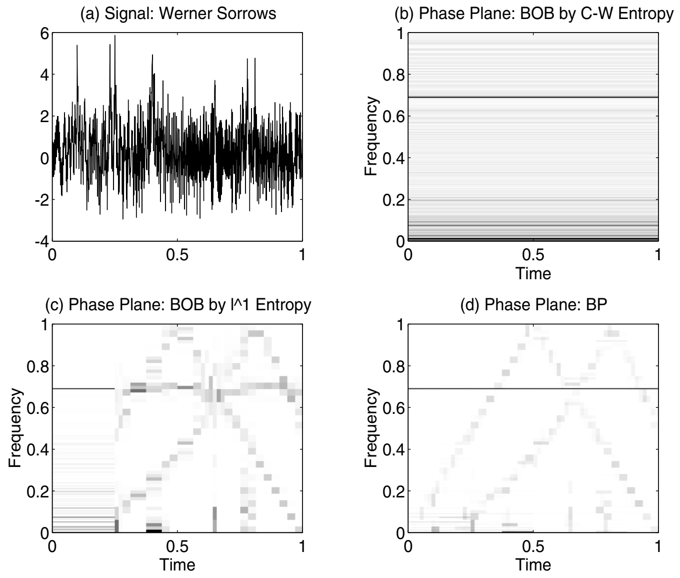

Fig. 2.6 *Analyzing the signal* `WernerSorrows` *with a cosine packet dictionary.*

From one point of view, (3.1) is very similar to the MOF (2.3): we are simply replacing the ℓ^2 norm in (2.3) with the ℓ^1 norm. However, this apparently slight change has major consequences. The MOF leads to a quadratic optimization problem with linear equality constraints and so involves essentially just the solution of a system of linear equations. In contrast, BP requires the solution of a convex, nonquadratic optimization problem, which involves considerably more effort and sophistication.

3.1. LP.

To explain the last comment and BP, we develop a connection with LP.

The linear program in so-called standard form [8, 21] is a constrained optimization problem defined in terms of a variable $\mathbf{x} \in R^m$ by

$$\min \ \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq 0, \tag{3.2}$$

where $\mathbf{c}^T\mathbf{x}$ is the objective function, $A\mathbf{x} = \mathbf{b}$ is a collection of equality constraints, and $\mathbf{x} \geq 0$ is a set of bounds. The main question is which variables should be zero.

The BP problem (3.1) can be equivalently reformulated as a linear program in the standard form (3.2) by making the following translations:

$$m \Leftrightarrow 2p, \quad A \Leftrightarrow (\Phi, -\Phi), \quad \mathbf{b} \Leftrightarrow \mathbf{s}, \quad \mathbf{c} \Leftrightarrow (1; 1), \quad \mathbf{x} \Leftrightarrow (\mathbf{u}; \mathbf{v}), \quad \alpha \Leftrightarrow \mathbf{u} - \mathbf{v}.$$

Hence the solution of (3.1) can be obtained by solving an equivalent linear program. (The equivalence of minimum ℓ^1 optimizations with LP has been known since the 1950s; see [2].) The connection between BP and LP is useful in several ways.

3.1.1. Solutions as Bases.

In the LP problem (3.2), suppose A is an n-by-m matrix with $m > n$, and suppose an optimal solution exists. It is well known that a solution exists in which at most n of the entries in the optimal $\mathbf{x}$ are nonzero. Moreover, in the generic case, the solution is so-called nondegenerate, and there are exactly n nonzeros. The nonzero coefficients are associated with n columns of A, and these columns make up a basis of R^n. Once the basis is identified, the solution is uniquely dictated by the basis. Thus finding a solution to the LP is identical to finding the optimal basis. In this sense, LP is truly a process of BP.

Translating the LP results into BP terminology, we have the decomposition

$$\mathbf{s} = \sum_{i=1}^{n} \alpha^{\star}_{\gamma_i} \phi_{\gamma_i} \, .$$

The waveforms (ϕ_{γ_i}) are linearly independent but not necessarily orthogonal. The collection γ_i is not, in general, known in advance but instead depends on the problem data (in this case $\mathbf{s}$). The selection of waveforms is therefore signal adaptive.

3.1.2. Algorithms. BP is an optimization principle, not an algorithm. Over the last 40 years, a tremendous amount of work has been done on the solution of linear programs. Until the 1980s, most work focused on variants of Dantzig's simplex algorithm, which many readers have no doubt studied. In the last ten years, some spectacular breakthroughs have been made by the use of so-called interior-point methods, which use an entirely different principle.

From our point of view, we are free to consider any algorithm from the LP literature as a candidate for solving the BP optimization problem; both the simplex and interior-point algorithms offer interesting insights into BP. When it is useful to consider BP in the context of a particular algorithm, we will indicate this by the label: either BP-simplex or BP-interior.

BP-Simplex. In standard implementations of the simplex method for LP, one first finds an initial basis B consisting of n linearly independent columns of A for which the corresponding solution $B^{-1}\mathbf{b}$ is feasible (nonnegative). Then one iteratively improves the current basis by swapping, at each step, one term in the basis for one term not in the basis, using the swap that best improves the objective function. There always exists a swap that improves or maintains the objective value, except at the optimal solution. Moreover, LP researchers have shown how one can select terms to swap in such a way as to guarantee convergence to an optimal solution (anticycling rules) [21]. Hence the simplex algorithm is explicitly a process of BP: iterative improvement of a basis until no improvement is possible, at which point the solution is achieved.

Translating this LP algorithm into BP terminology, one starts from any linearly independent collection of n atoms from the dictionary. One calls this the current decomposition. Then one iteratively improves the current decomposition by swapping atoms in the current decomposition for new atoms, with the goal of improving the objective function. By application of anticycling rules, there is a way to select swaps that guarantees convergence to an optimal solution (assuming exact arithmetic).

BP-Interior. The collection of feasible points $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq 0\}$ is a convex polyhedron in R^m (a "simplex"). The simplex method, viewed geometrically, works by walking around the boundary of this simplex, jumping from one vertex (extreme point) of the polyhedron to an adjacent vertex at which the objective is better. Interior-point methods instead start from a point $x^{(0)}$ well inside the interior of the simplex ($x^{(0)} \gg 0$) and go "through the interior" of the simplex. Since the solution of a linear program is always at an extreme point of the simplex, as the interior-point method converges, the current iterate $x^{(k)}$ approaches the boundary. One may abandon the basic interior-point iteration and invoke a "crossover" procedure that uses simplex iterations to find the optimizing extreme point.

Translating this LP algorithm into BP terminology, one starts from a solution to the overcomplete representation problem $\Phi\alpha^{(0)} = \mathbf{s}$ with $\alpha^{(0)} > 0$. One iteratively

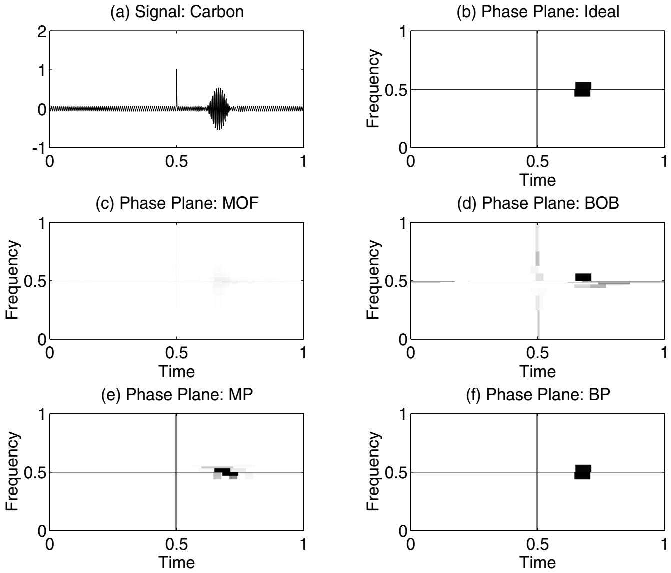

Fig. 3.1 *Analyzing the signal* `Carbon` *with a wavelet packet dictionary.*

modifies the coefficients, maintaining feasibility $\Phi\alpha^{(k)} = s$ and applying a transformation that effectively sparsifies the vector $\alpha^{(k)}$. At some iteration, the vector has $\leq n$ significantly nonzero entries, and it "becomes clear" that those correspond to the atoms appearing in the final solution. One forces all the other coefficients to zero and "jumps" to the decomposition in terms of the $\leq n$ selected atoms. (More general interior-point algorithms start with $a^{(0)} > 0$ but don't require the feasibility $\Phi\alpha^{(k)} = s$ throughout; they achieve feasibility eventually.)

3.2. Examples. We now give computational examples of BP in action.

3.2.1. Carbon. The synthetic signal `Carbon` is a composite of six atoms: a Dirac, a sinusoid, and four mutually orthogonal wavelet packet atoms, adjacent in the time-frequency plane. The wavelet packet dictionary of depth $D = \log_2(n)$ is employed, based on filters for symmlets with eight vanishing moments. (Information about problem sizes for all examples is given in Table 6.1.)

Figure 3.1 displays the results in phase-plane form; for comparison, we include the phase planes obtained using MOF, MP, and BOB. First, note that MOF uses all basis functions that are not orthogonal to the six atoms, i.e., all the atoms at times and frequencies that overlap with some atom appearing in the signal. The corresponding phase plane is very diffuse or smeared out. Second, MP is able to do a relatively good job on the sinusoid and the Dirac, but it makes mistakes in handling the four close atoms. Third, BOB cannot handle the nonorthogonality between the Dirac and the cosine; it gives a distortion (a coarsening) of the underlying phase plane picture. Finally, BP finds the "exact" decomposition in the sense that the four atoms in the quad, the Dirac, and the sinusoid are all correctly identified.

3.2.2. TwinSine. Recall that the signal `TwinSine` in Figure 2.4a consists of two cosines with frequencies closer together than the Rayleigh distance. In Figure 2.4d, we analyze these in the fourfold overcomplete discrete cosine dictionary. Recall that

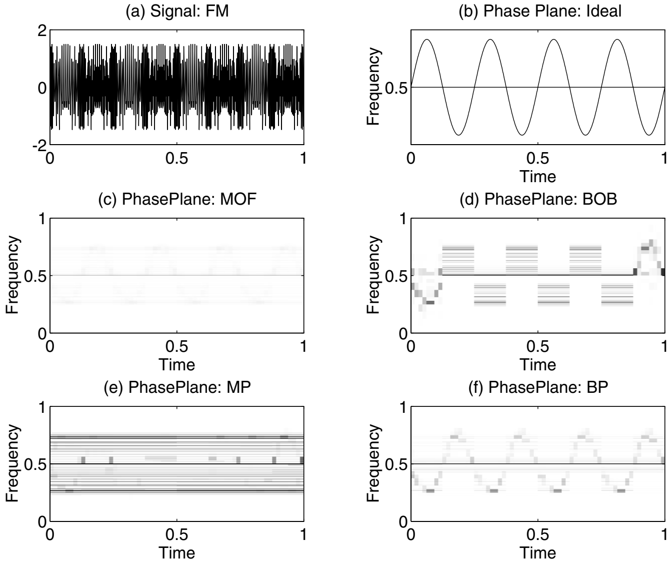

Fig. 3.2 *Analyzing the signal* `FM-Cosine` *with a cosine packet dictionary.*

in this example MP began by choosing at the first step a frequency in between the two ideal ones and then never corrected the error. In contrast, BP resolves the two frequencies correctly.

3.2.3. FM Signal. Figure 3.2a displays the artificial signal `FM-Cosine` consisting of a frequency-modulated sinusoid superposed with a pure sinusoid: $s = \cos(\xi_0 t) + \cos((\xi_0 t + \alpha \cos(\xi_1 t))t)$. Figure 3.2b shows the ideal phase plane.

In Figure 3.2c–3.2f we analyze it using the cosine packet dictionary based on a bell 16 samples wide. It is evident that BOB cannot resolve the nonorthogonality between the sinusoid and the FM signal. Neither can MP. However, BP yields a clean representation of the two structures.

3.2.4. Gong. Figure 3.3a displays the `Gong` signal, which vanishes until time t_0 and then follows a decaying sinusoid for $t > t_0$.

In Figures 3.3c–3.3d, we analyze it with the cosine packet dictionary based on a bell 16 samples wide. BP gives the finest representation of the decay structure, which is visually somewhat more interpretable than the BOB and MP results.

3.3. Comparisons. We briefly compare BP with the three main methods introduced in section 2.3.

3.3.1. Matching Pursuit. At first glance MP and BP seem quite different. MP is an iterative *algorithm*, which does not explicitly seek any overall goal but merely applies a simple rule repeatedly. In contrast, BP is a *principle* of global optimization without any specified algorithm. The contrast of orthogonal MP with a specific algorithm, BP-simplex, may be instructive. Orthogonal matching pursuit starts from an “empty model” and builds up a signal model an atom at a time, at each step adding to the model only the most important new atom among all those not already in the model. In contrast, BP-simplex starts from a “full” model (i.e., a representation of the

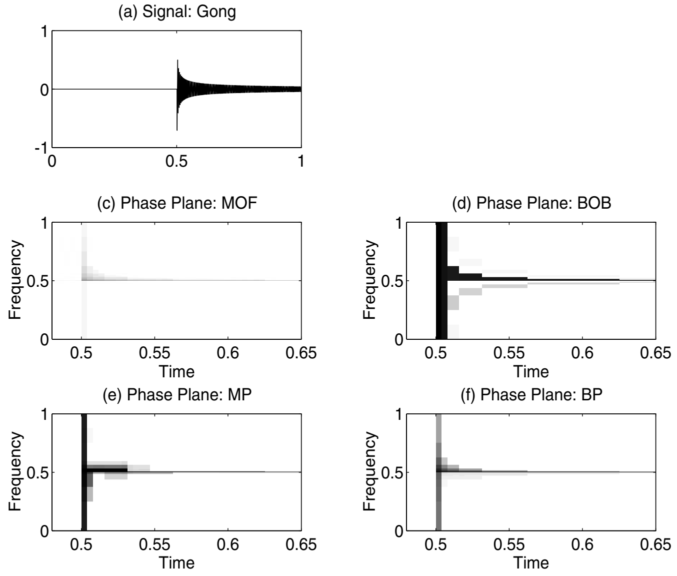

Fig. 3.3 *Analyzing the signal* `Gong` *with a cosine packet dictionary.*

object in a basis) and then iteratively improves the "full" model by taking relatively useless terms out of the model and swapping them for useful new ones. Hence MP is a sort of build-up approach, while BP-simplex is a sort of swap-down approach.

3.3.2. Best Orthogonal Basis. To make BP and BOB most comparable, suppose that they are both working with a cosine packet dictionary, and note that the ℓ^1 norm of coefficients is what Coifman and Wickerhauser [7] called an "additive measure of information." So suppose we apply the Coifman–Wickerhauser best basis algorithm with entropy $\mathcal{E} = \ell^1$. Then the two methods compare as follows: in BOB, we are optimizing $\mathcal{E}$ only over *orthogonal bases* taken from the dictionary, while in BP we are optimizing $\mathcal{E}$ over *all bases* formed from the dictionary.

This last remark suggests that it might be interesting to apply the BOB procedure with the ℓ^1 norm as entropy in place of the standard Coifman–Wickerhauser entropy. In Figure 2.6c we try this on the `WernerSorrows` example of section 2.3.3. The signal is analyzed in a cosine packet dictionary, with primitive bell width 16. The ℓ^1 entropy results in a time-varying basis that reveals clearly some of the underlying signal structure. The ℓ^1 entropy by itself improves the performance of BOB, but BP does better still (Figure 2.6d).

This connection between BP and BOB suggests an interesting algorithmic idea. In the standard implementation of the simplex method for LP, one starts from an initial basis and then iteratively improves the basis by swapping one term in the basis for one term not in the basis, using the swap that best improves the objective function. Which initial basis will be used? It seems natural in BP-simplex to use the Coifman–Wickerhauser algorithm and employ as a start the BOB.

With this choice of starting basis, BP can be seen as a method of refining BOB by swapping nonorthogonal atoms with orthogonal ones whenever this will improve the objective.

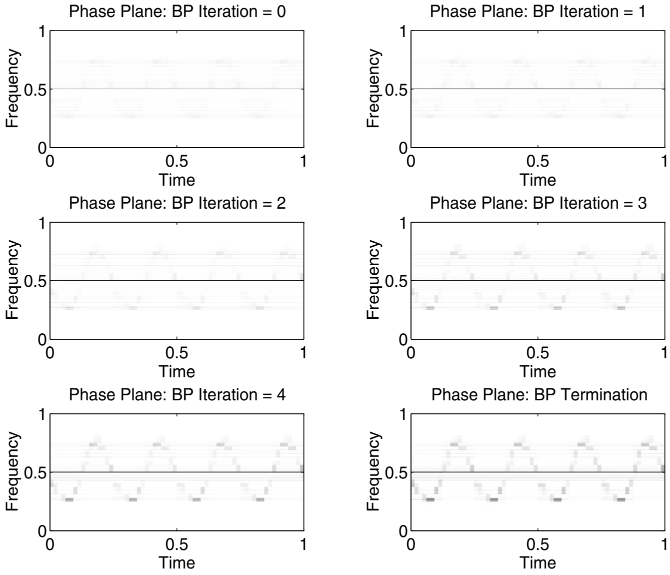

Fig. 3.4 *Phase plane evolution at BP-interior iteration.*

3.3.3. MOF. As already discussed, MOF and BP differ in the replacement of an l^2 objective function by an l^1 objective. BP-interior has an interesting relation to the MOF. BP-interior initializes with the MOF solution. Hence one can say that BP sequentially "improves" on the MOF. Figure 3.4 shows a "movie" of BP-interior in action on the `FM-Cosine` example, using a cosine packet dictionary. Six stages in the evolution of the phase plane are shown, and one can see how the phase plane improves in clarity, step by step.

4. Variations.

The recent development of time-frequency dictionaries motivates most of what we have done so far. However, the methods we have developed are general and can be applied to other dictionaries, with interesting results.

4.1. Stationary Smooth Wavelets. The usual (orthonormal) dictionaries of (periodized) smooth wavelets consist of wavelets at scales indexed by $j = j_0, \dots, \log_2(n) - 1$; at the jth scale, there are 2^j wavelets of width $n/2^j$. The wavelets at this scale are all circulant shifts of each other, the shift being $n/2^j$ samples. Some authors [45] have suggested that this scheme can be less than satisfactory, essentially because the shift between adjacent wavelets is too large. They would say that *if* the important "features" of the signal are (fortuitously) "aligned with" the wavelets in the dictionary, then the dictionary will provide a sparse representation of the signal; however, because there are so few wavelets at level j, then most likely the wavelets in the dictionary are not "precisely aligned" with features of interest, and the dictionary may therefore provide a very diffuse representation.

The stationary wavelet dictionary has, at the jth level, n (not 2^j) wavelets; these are all the circulant shifts of the basic wavelet of width $\approx n/2^j$. Since this dictionary always contains wavelets "aligned with" any given feature, the hope is that such a dictionary provides a superior representation.

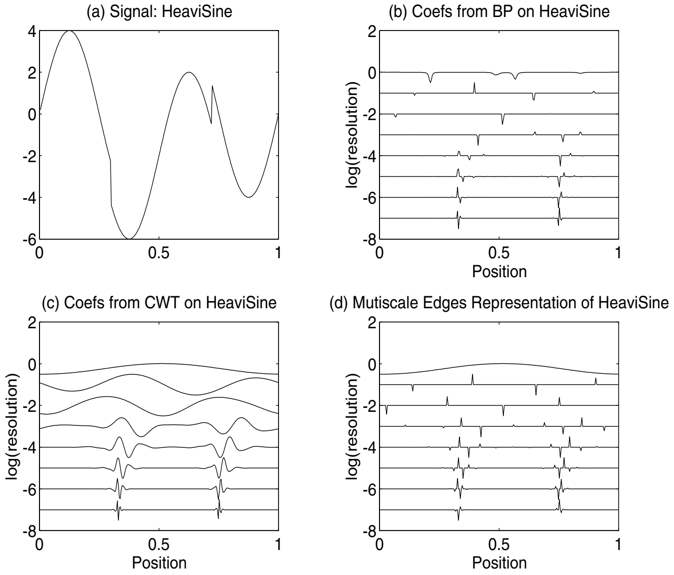

Fig. 4.1 *Analyzing the signal* `HeaviSine` *with a stationary wavelet dictionary.*

Figure 4.1a shows the signal `HeaviSine`, and Figure 4.1b shows the result of BP with the stationary symmlet-8 dictionary mentioned in section 2.1; the coefficients are displayed in a multiresolution fashion, where at level j all the coefficients of scale $2^j/n$ are plotted according to spatial position.

There is a surprisingly close agreement of the BP representation in a stationary wavelet dictionary with ideas about signal representation associated with the "multiscale edges" ideas of Mallat and Hwang [28] and Mallat and Zhong [30]. The multiscale edge method analyzes the continuous wavelet transform (CWT) at scale 2^{-j} and identifies the maxima of this transform. Then it selects maxima that are "important" by thresholding based on amplitude. These "important" maxima identify important features of the signal. Mallat and Zhong proposed an iterative method that reconstructs an object having the same values of the CWT at "maxima." This is almost (but not quite) the same thing as saying that one is identifying "important" wavelets located at the corresponding maxima and reconstructing the object using just those maxima.

Figure 4.1c shows a CWT of `HeaviSine` based on the same symmlet-8 wavelet, again in multiresolution fashion; Figure 4.1d shows the maxima of the CWT. At fine scales, there is virtually a one-to-one relationship between the maxima of the transform and the wavelets selected by BP; compare Figure 4.1b. So in a stationary wavelet dictionary, the global optimization principle BP yields results that are close to certain heuristic methods.

As an important contrast, Meyer has a counterexample to multiscale edge approaches, which shows that the Mallat–Zhong approach may fail in certain cases [34], but there can be no such counterexamples to BP.

4.2. Dictionary Mergers. An important methodological tool is the ability to combine dictionaries to make bigger, more expressive dictionaries. We mention here two possibilities. Examples of such decompositions are given in section 5 below.

Jump+sine. Merge the heaviside dictionary with a Fourier dictionary. Either dictionary can efficiently represent objects that the other cannot; for example, heavisides have difficulty representing sinusoids, while sinusoids have difficulty representing jumps. Their combination might therefore be able to offer the advantages of both.

Jump+wavelet. For similar reasons, one might want to merge heavisides with wavelets. In fact, we have found it sometimes preferable instead to merge "tapered heavisides" with wavelets; these are step discontinuities that start at 0, jump at time t_0 to a level one unit higher, and later decay to the original 0 level.

5. Denoising.

We now adapt BP to the case of noisy data. We assume data of the form

$$\mathbf{y} = \mathbf{s} + \sigma \mathbf{z},$$

where (z_i) is a standard white Gaussian noise, $\sigma > 0$ is a noise level, and $\mathbf{s}$ is the clean signal. In this setting, $\mathbf{s}$ is unknown, while $\mathbf{y}$ is known. We don't want to get an exact decomposition of $\mathbf{y}$, so we don't apply BP directly. Instead decompositions like (1.2) become relevant.

5.1. Proposal.

Basis pursuit denoising (BPDN) refers to the solution of

$$\min_{\alpha} \ \frac{1}{2}\|\mathbf{y} - \Phi\alpha\|_2^2 + \lambda\|\alpha\|_1. \tag{5.1}$$

The solution $\alpha^{(\lambda)}$ is a function of the parameter λ. It yields a decomposition into signal-plus-residual:

$$\mathbf{y} = \mathbf{s}^{(\lambda)} + \mathbf{r}^{(\lambda)},$$

where $\mathbf{s}^{(\lambda)} = \Phi\alpha^{(\lambda)}$. The size of the residual is controlled by λ. As $\lambda \to 0$, the residual goes to zero and the solution behaves exactly like BP applied to $\mathbf{y}$. As $\lambda \to \infty$, the residual gets large; we have $\mathbf{r}^{(\lambda)} \to \mathbf{y}$ and $\mathbf{s}^{(\lambda)} \to 0$.

As we have noted in [4], (5.1) is equivalent to the following perturbed linear program:

$$\min_{\mathbf{x},\mathbf{p}} \ \mathbf{c}^T\mathbf{x} + \frac{1}{2}\|\mathbf{p}\|^2 \quad \text{subject to} \quad A\mathbf{x} + \delta\mathbf{p} = \mathbf{b}, \quad \mathbf{x} \geq 0, \quad \delta = 1,$$

where $A = (\Phi, -\Phi)$, $\mathbf{b} = \mathbf{y}$, $\mathbf{c} = \lambda(1;1)$, $\mathbf{x} = (\mathbf{u};\mathbf{v})$, $\alpha = \mathbf{u} - \mathbf{v}$. Perturbed LP is really quadratic programming, but it retains a structure similar to LP [20]. Hence we can have a similar classification of algorithms into BPDN-simplex and BPDN-interior-point types. (In quadratic programming, "simplex-like" algorithms are usually called active set algorithms, so our label is admittedly nonstandard.)

5.2. Choice of λ.

Assuming the dictionary is normalized so that $\|\phi_\gamma\|_2 = 1$ for all γ, we set λ to the value

$$\lambda_p = \sigma\sqrt{2\log(p)},$$

where p is the cardinality of the dictionary.

This can be motivated as follows. In the case of a dictionary that is an orthonormal basis, a number of papers [13, 18] have carefully studied an approach to denoising by so-called soft thresholding in an orthonormal basis. In detail, suppose that Φ is an orthogonal matrix, and define empirical Φ-coefficients by

$$\tilde{\mathbf{y}} = \Phi^T\mathbf{y}.$$

Define the soft threshold nonlinearity $\eta_\lambda(y) = \mathrm{sgn}(y) \cdot (|y| - \lambda)_+$ and define the thresholded empirical coefficients by

$$\hat{\alpha_\gamma} = \eta_{\lambda_n}(\tilde{y}_\gamma), \qquad \gamma \in \Gamma.$$

This is soft thresholding of empirical orthogonal coefficients. The papers just cited show that thresholding at λ_n has a number of optimal and near-optimal properties regarding mean-squared error.

We claim that (again in the case of an ortho basis) the thresholding estimate $\hat{\alpha}$ is also the solution of (5.1). Observe that the soft-thresholding nonlinearity solves the scalar minimum problem

$$\eta_\lambda(y) = \frac{1}{2} \arg\min_\xi \, (y - \xi)^2 + \lambda|\xi|. \tag{5.2}$$

Note that, because of the orthogonality of Φ, $\|\mathbf{y} - \Phi\alpha\|_2 = \|\tilde{\mathbf{y}} - \alpha\|_2$, and so we can rewrite (5.1) in this case as

$$\min_\alpha \; \frac{1}{2} \sum_\gamma (\tilde{y}_\gamma - \alpha_\gamma)^2 + \lambda \sum_\gamma |\alpha_\gamma|. \tag{5.3}$$

Now applying (5.2) coordinatewise establishes the claim.

The scheme we have suggested here—to be applied in overcomplete as well as orthogonal settings—therefore includes soft thresholding in ortho bases as a special case. Formal arguments similar to those in [17] can be used to give a proof that mean-squared error properties of the resulting procedure are near optimal under certain conditions.

5.3. Examples. We present two examples of BPDN in action with time-frequency dictionaries. We compare BPDN with three other denoising methods adapted from MOF, MP, and BOB. Method of frames denoising (MOFDN) refers to minimizing the squared l^2 error plus an l^2 penalizing term

$$\min_\alpha \; \|\mathbf{s} - \Phi\alpha\|_2^2 + \lambda\|\alpha\|_2^2,$$

where λ is a penalizing parameter; we chose λ in these examples to be $\sigma\sqrt{2\log(p)}$. Matching pursuit denoising (MPDN) runs MP until the coefficient associated with the selected atom gets below the threshold $\sigma\sqrt{2\log(p)}$. Best orthogonal basis denoising (BOBDN) is a thresholding scheme in the basis chosen by the BOB algorithm with a special entropy [16].

5.3.1. Gong. Figure 5.1 displays denoising results on the signal `Gong`, at signal to noise ratio 1, using a cosine packet dictionary. Figure 5.1a displays the noiseless signal and Figure 5.1b displays a noisy version. Figures 5.1c–5.1f display denoising results for MOF, BOB, MP, and BP, respectively. BP outperforms the other methods visually.

5.3.2. TwinSine. Figure 5.2 employs the signal `TwinSine`, described earlier, to investigate superresolution in the noisy case. Figures 5.2a and 5.2b give the noiseless and noisy `TwinSine`, respectively. Using a fourfold overcomplete discrete cosine dictionary, reconstructions by the MOF, MP, and BPDN are given. MOF gives a reconstruction that is inherently resolution limited and oscillatory. As in the noiseless case, MP gives a reconstruction that goes wrong at step 1—it selects the average of the two frequencies in the `TwinSine` signal. BP correctly resolves the nonnegative doublet structure.

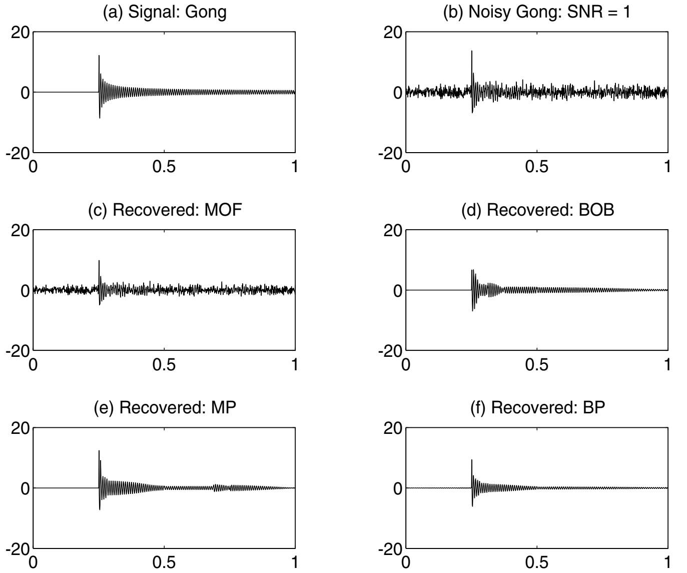

Fig. 5.1 *Denoising noisy* `Gong` *with a cosine packet dictionary.*

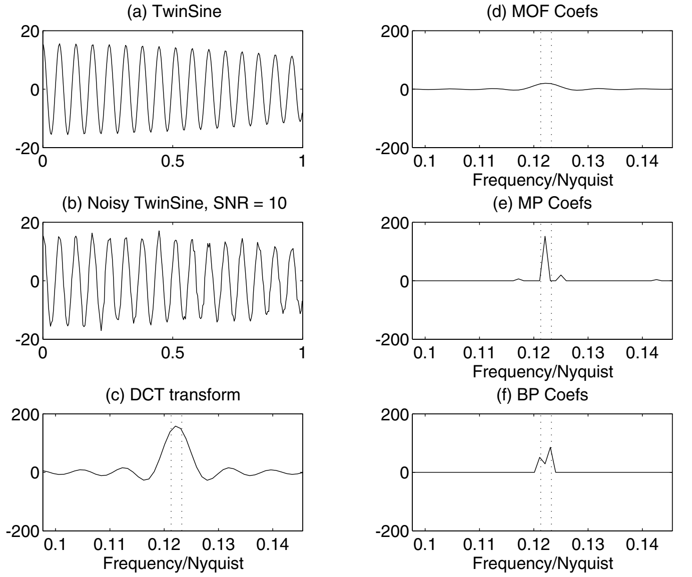

Fig. 5.2 *Denoising noisy* `TwinSine-2` *with a fourfold overcomplete discrete cosine dictionary.*

5.4. Total Variation Denoising. Recently, Rudin, Osher, and Fatemi [41] have called attention to the possibility of denoising images using total variation penalized least squares. More specifically, they proposed the optimization problem

$$\min_{\mathbf{g}} \; \frac{1}{2}\|\mathbf{y}-\mathbf{g}\|_2^2 + \lambda \cdot TV(\mathbf{g}), \tag{5.4}$$

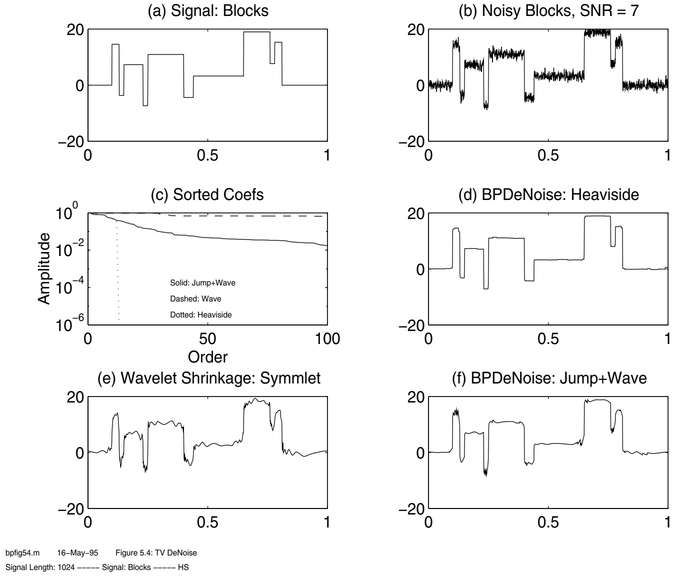

Fig. 5.3 *Denoising noisy* `Blocks`*.*

where $TV(\mathbf{g})$ is a discrete measure of the total variation of $\mathbf{g}$. A solution of this problem is the denoised object. Li and Santosa [26] have developed an alternative algorithm for this problem based on interior-point methods for convex optimization.

For the one-dimensional case (signals rather than images) it is possible to implement what amounts to total variation denoising by applying BPDN with a heaviside dictionary. Indeed, if $\mathbf{s}$ is an arbitrary object, it has a unique decomposition in heavisides (recall (2.1)). Suppose that the object is 0 at $t = 0$ and $t = n - 1$ and that the decomposition is $\mathbf{s} = \sum_i \alpha_i H_{t_i}$; then the total variation is given by

$$TV(\mathbf{s}) = \sum_{i \neq 0} |\alpha_i|.$$

Moreover, to get approximate equality even for objects not obeying zero-boundary conditions, one has only to normalize ϕ_0 appropriately. Consequently, total variation denoising is essentially a special instance of our proposal (5.1).

We have studied BPDN in the heaviside dictionary, thereby obtaining essentially a series of tests of total variation denoising. For comparison, we also considered soft thresholding in orthogonal wavelet dictionaries based on the S8-symmlet smooth wavelet. We also constructed a new dictionary, based on the *jump+wave* merger of S8-symmlet wavelets with "smoothly tapered heavisides," which is to say atoms ϕ_γ that jump at a given point γ and then decay smoothly away from the discontinuity. For comparability with the heaviside dictionary, we normalized the *jump+wave* dictionary so that every $\|\phi_\gamma\|_{TV} \approx 1$.

A typical result for the object `Blocky` is presented in Figure 5.3. From the point of view of visual appearance, total variation reconstruction (Figure 5.3d) far outperforms the other methods.

Of course, the object `Blocky` has a very sparse representation in terms of heavisides. When we consider an object like `Cusp`, which is piecewise smooth rather than

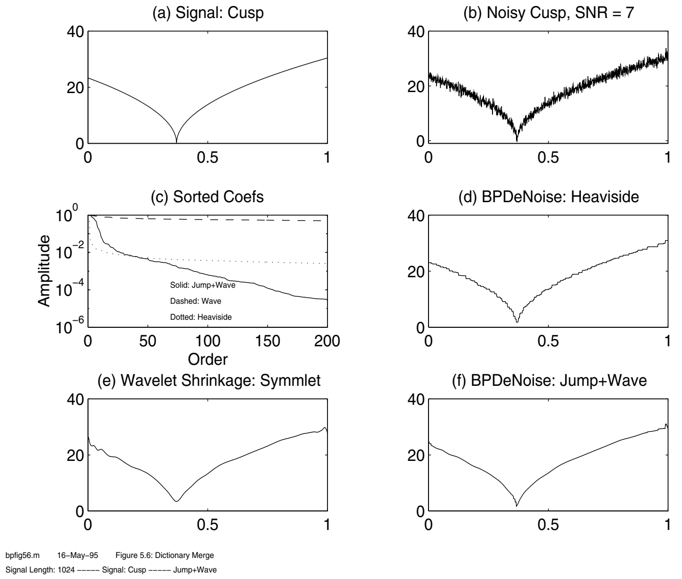

Fig. 5.4 *Denoising noisy* `Cusp`.

piecewise constant, the object will no longer have a sparse representation. On the other hand, using the *jump+wave* dictionary based on a merger of wavelets with tapered heavisides will lead to a sparse representation—see Figure 5.4c. One can predict that a heaviside dictionary will perform less well than this merged dictionary.

This completely obvious comment, translated into a statement about TV denoising, becomes a surprising prediction. One expects that the lack of sparse representation of smooth objects in the heaviside dictionary will translate into worse performance of total variation denoising than of BPDN in the merged *jump+wave* dictionary.

To test this, we conducted experiments. Figure 5.4 compares total variation denoising, wavelet denoising, and BPDN in the merged *jump+wave* dictionary. Total variation denoising now exhibits visually distracting stairstep artifacts; the dictionary *jump+wave* seems to us to behave much better.

6. Solutions of Large-Scale Linear Programs. As indicated in section 3.1, the optimization problem (3.1) is equivalent to a linear program (3.2). Also, as in section 5.1, the optimization problem (5.1) is equivalent to a perturbed linear program (5.3). The problems in question are large scale; we have conducted decompositions of signals of length $n = 8192$ in a wavelet packet dictionary, leading to a linear program of size 8192 by 212,992.

Over the last ten years there has been a rapid expansion in the size of linear programs that have been successfully solved using digital computers. An overview of the rapid progress in this field is afforded by the article of Lustig, Marsten, and Shanno [27] and the accompanying discussions by Bixby [1], Saunders [43], Todd [46], and Vanderbei [47]. Much of the rapid expansion in the size of linear programs solved is due to the "interior-point revolution" initiated by Karmarkar's proof that a pseudo-polynomial-time algorithm could be based on an interior-point method [24]. Since then a wide array of interior-point algorithms have been proposed and considerable practical [25, 27, 32, 50] and theoretical [49, 35, 40] understanding is now available. In this section we describe our algorithm and our experience with it.

6.1. Duality Theory. We consider the linear program in the standard form

$$\min\ \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq 0. \tag{6.1}$$

This is often called the *primal* linear program. The primal linear program is equivalent to the *dual* linear program

$$\max\ \mathbf{b}^T\mathbf{y} \quad \text{subject to} \quad A^T\mathbf{y} + \mathbf{z} = \mathbf{c}, \quad \mathbf{z} \geq 0. \tag{6.2}$$

$\mathbf{x}$ is called the *primal* variable; $\mathbf{y}$ and $\mathbf{z}$ are called the *dual* variables. For any $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ with $\mathbf{x} \geq 0$ and $\mathbf{z} \geq 0$, the term *primal infeasibility* refers to the quantity $\|\mathbf{b} - A\mathbf{x}\|_2$; the term *dual infeasibility* refers to $\|\mathbf{c} - \mathbf{z} - A^T\mathbf{y}\|_2$; the term *duality gap* refers to the difference between the primal objective and the dual objective: $\mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{y}$.

A fundamental theorem of LP states that $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ solves the linear program (6.1) if and only if the primal infeasibility, the dual infeasibility, and the duality gap are all zero. Therefore, when $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ are nearly primal feasible and nearly dual feasible, the duality gap offers a good description about the accuracy of $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ as a solution: the smaller the duality gap is, the closer $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ are to the optimal solution.

6.2. A Primal-Dual Log-Barrier LP Algorithm. Mathematical work on interior-point methods over the last ten years has led to a large variety of approaches with names like *projective scaling*, *(primal/dual) affine scaling*, *(primal/dual) logarithmic barrier*, and *predictor corrector*. We cannot summarize all these ideas here; many of them are mentioned in [49, 27, 50, 40], for example.

Our approach is based on a *primal-dual* log-barrier algorithm. In order to regularize standard LP, Gill et al. [20] proposed solving the following *perturbed* linear program:

$$\min\ \mathbf{c}^T\mathbf{x} + \frac{1}{2}\|\gamma\mathbf{x}\|^2 + \frac{1}{2}\|\mathbf{p}\|^2 \quad \text{subject to} \quad A\mathbf{x} + \delta\mathbf{p} = \mathbf{b}, \quad \mathbf{x} \geq 0, \tag{6.3}$$

where γ and δ are normally small (e.g., 10^{-4}) regularization parameters. (We comment that such a perturbed linear program with $\delta = 1$ solves the BPDN problem (5.1).) The main steps of the interior-point algorithm are as follows.

1. Set parameters: the feasibility tolerance `FeaTol`, the duality gap tolerance `PDGapTol`, the two regularization parameters γ and δ.
2. Initialize $\mathbf{x} > 0$, $\mathbf{y} = 0$, $\mathbf{z} > 0$, $\mu > 0$.
3. Loop
   (a) Compute residuals and diagonal matrix D:

$$\begin{aligned} \mathbf{t} &= \mathbf{c} + \gamma^2\mathbf{x} - \mathbf{z} - A^T\mathbf{y}, \\ \mathbf{r} &= \mathbf{b} - A\mathbf{x} - \delta^2\mathbf{y}, \\ \mathbf{v} &= \mu\mathbf{e} - Z\mathbf{x}, \\ D &= (X^{-1}Z + \gamma^2 I)^{-1}, \end{aligned}$$

   where X and Z are diagonal matrices composed from $\mathbf{x}$ and $\mathbf{z}$; $\mathbf{e}$ is a vector of 1s.
   (b) Solve

$$(ADA^T + \delta^2 I)\Delta\mathbf{y} = \mathbf{r} + AD(\mathbf{t} - X^{-1}\mathbf{v}) \tag{6.4}$$

for $\Delta\mathbf{y}$ and set

$$\Delta\mathbf{x} = D(A^T\Delta\mathbf{y} + X^{-1}\mathbf{v} - \mathbf{t}), \quad \Delta\mathbf{z} = X^{-1}(\mathbf{v} - Z\Delta\mathbf{x}).$$

(c) Calculate the primal and dual step sizes ρ_p, ρ_d and update the variables:

$$\begin{aligned}
&\rho_p = .99\max\{\rho : \mathbf{x} + \rho\Delta\mathbf{x} \geq 0\},\\
&\rho_d = .99\max\{\rho : \mathbf{z} + \rho\Delta\mathbf{z} \geq 0\},\\
&\mathbf{x} = \mathbf{x} + \rho_p\Delta\mathbf{x}, \quad \mathbf{y} = \mathbf{y} + \rho_d\Delta\mathbf{y}, \quad \mathbf{z} = \mathbf{z} + \rho_d\Delta\mathbf{z},\\
&\mu = (1 - \min(\rho_p,\ \rho_d,\ .99))\mu.
\end{aligned}$$

4. Terminate if the following three conditions are satisfied:
   (a) Primal infeasibility $= \frac{\|\mathbf{r}\|_2}{1+\|\mathbf{x}\|_2} < \texttt{FeaTol}$.
   (b) Dual infeasibility $= \frac{\|\mathbf{t}\|_2}{1+\|\mathbf{y}\|_2} < \texttt{FeaTol}$.
   (c) Duality gap $= \frac{\mathbf{z}^T\mathbf{x}}{1+\|\mathbf{z}\|_2\|\mathbf{x}\|_2} < \texttt{PDGapTol}$.

For fuller discussions of this and related algorithms, again see [20, 49, 27, 50, 40].

Note that when $\delta > 0$, the central equation (6.4) may be written as the least-squares problem

$$\min_{\Delta\mathbf{y}} \left\| \begin{pmatrix} D^{1/2}A^T \\ \delta I \end{pmatrix} \Delta\mathbf{y} - \begin{pmatrix} D^{1/2}(\mathbf{t} - X^{-1}\mathbf{v}) \\ \mathbf{r}/\delta \end{pmatrix} \right\|_2, \tag{6.5}$$

which may be better suited to numerical solution if δ is not too small.

While in principle we could have based our approach on other interior-point schemes, the primal-dual approach naturally incorporates several features we found useful. First, the iterates $\mathbf{x}, \mathbf{y}, \mathbf{z}$ do not have to be feasible. We are only able to choose a starting point that is *nearly* feasible and remain *nearly* feasible throughout the sequence of iterations. Second, after both primal and dual feasibility have been nearly achieved, it is easy to check for closeness to the solution value; at the limiting solution $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$, and the duality gap $\mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{y} \approx \mathbf{x}^T\mathbf{z}$ quantifies the distance from this ideal.

6.3. Implementation Heuristics. The primal-dual log barrier algorithm we just described works in a fashion similar to other interior-point methods [27]. It starts from an initial feasible (or nearly feasible) solution located at or near the "center" of the feasible region and iteratively improves the current solution until the iterates $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ achieve the desired accuracy. It requires a relatively small number of iterations; for example, a few dozen iterations would be common. Each iteration requires the solution of a system of equations involving A, A^T, and other problem data like $\mathbf{x}, \mathbf{y}, \mathbf{z}$. In the primal-dual log barrier method, the system is (6.4). Thus the numerical solution to a linear program by interior-point methods amounts to a sequence of several dozen solutions of special systems of linear equations. This leads to a slogan: *if* those systems can be solved rapidly, *then* it is possible to solve the LP rapidly.

Of course, in general, solving systems of equations is not rapid: a general n-by-n system $B\mathbf{w} = \mathbf{h}$ takes order $O(n^3)$ time to solve by standard elimination methods or by modern stable factorization schemes [22, 21]. In order for practical algorithms to be based on the interior-point heuristic, it is necessary to be able to solve the systems of equations much more rapidly than one could solve general systems. In the current state of the art of linear programming [27], one attempts to do this by exploiting *sparsity* of the underlying matrix A.

However, the optimization problems we are interested in have a key difference from the successful large-scale applications outlined in [26, 1]. The matrix A we deal with is not at all sparse; it is generally completely dense. For example, if A is generated from a Fourier dictionary, most of the elements of A will be of the same order of magnitude. Because of this density, it is unlikely that existing large-scale interior-point computer codes could be easily applied to the problems described in this paper.

In our application we have a substitute for sparsity. We consider only dictionaries that have fast implicit algorithms for $\Phi\alpha$ and $\Phi^T\mathbf{s}$ and therefore lead to linear programs where the A matrix admits fast implicit algorithms for both $A\mathbf{u}$ and $A^T\mathbf{v}$. (Compare section 2.2.2.) Now whenever one has fast implicit algorithms, it is natural to think of solving equations by conjugate-gradient methods; such methods allow one to solve equations $B\mathbf{w} = \mathbf{h}$ using only products $B\mathbf{v}$ with various strategically chosen vectors $\mathbf{v}$. Adapting such ideas, one develops fast implicit algorithms for $(ADA^T + \delta^2 I)\mathbf{v}$ and attempts to solve the central equations (6.4) iteratively, avoiding the costly step of explicitly forming the matrices $(ADA^T + \delta^2 I)$.

Similarly, the algorithms for $A\mathbf{u}$ and $A^T\mathbf{v}$ can be used directly in conjugate-gradient methods such as LSQR [36, 37] for solving the least-squares problem (6.5).

In our application, we do not really need an exact solution of the optimization problem. Moreover, we have a natural initial solution—from MOF—that would be viewed by some researchers as already an acceptable method of atomic decomposition. By starting from this decomposition and applying a strategy based on a limited number of iterations of our algorithm, we get what we view as an iterative improvement on MOF. (Compare Figure 3.4.) We stress that our strategy is to "pursue an optimal basis." While we would like to reach the optimal basis, we make no specific claims that we can always reach it in reasonable time; perhaps the "pursuit" language will help remind one of this fact. We do believe that the pursuit process, carried out for whatever length of time we are willing to invest in it, makes a useful improvement over the MOF.

6.4. Routine Settings for BP. Our strategy for routine signal processing by BP is as follows.

- We employ the "primal-dual logarithmic barrier method" for perturbed LP [20], as described in section 6.2.
- We assume fast implicit algorithms for $A\mathbf{u}$ and $A^T\mathbf{v}$.
- We only aim to reach an approximate optimum. `FeaTol` $= 10^{-1}$ and `PDGapTol` $= 10^{-1}$ would usually suffice for this.
- Each barrier iteration involves *approximate* solution of the central equations (6.4) using the conjugate-gradient method, e.g., with `CGAccuracy` $= 10^{-1}$.

We refer the reader to [4] for a more detailed discussion of our implementation.

6.5. Complexity Analysis. Table 6.1 displays the CPU times spent running various atomic decomposition techniques in our experiments; all computation was done on a Sun Sparc20 workstation. We employ a conjugate-gradient solver for the generalized inverse in the MOF solution (2.4); the resulting algorithm for MOF has a complexity of $O(n \log(n))$. (Note that it would be numerically preferable to apply Craig's method or LSQR to problem (2.3); see [36].) We implement Coifman and Wickerhauser's BOB algorithm [7], which also has complexity $O(n \log(n))$. We observe that BP is typically slower than MOF and BOB. BP is also slower than MP (which has a quasi-linear complexity, depending on the number of chosen atoms) except on the `FM-Cosine` signal in Figure 3.2.

Table 6.1 *CPU running times of the examples.*

Figure	Signal	Problem size	CPU running time in seconds			
			MOF	BOB	MP	BP
Figure 2.4	`TwinSine`	256	.3500	–	.6667	7.517
Figure 2.6	`WernerSorrows`	1024	–	.9500	–	158.2
Figure 3.1	`Carbon`	1024	.2000	2.617	2.650	11.70
Figure 3.2	`FM-Cosine`	1024	1.050	.9333	182.9	150.2
Figure 3.3	`Gong`	1024	1.433	5.683	50.63	448.2
Figure 4.1	`HeaviSine`	256	–	–	–	26.92
Figure 5.1	Noisy `Gong`	1024	2.117	6.767	8.600	142.2
Figure 5.2	Noisy `TwinSine`	256	.4167	–	.6833	5.717

Several factors influence the running time of BP.

1. *Problem sizes.* The complexity goes up *quasi-linearly* as the problem size increases [4]. By this we mean merely that the innermost computational step—a conjugate-gradient iteration—has a complexity that scales with problem size like $O(n)$ or $O(n \log(n))$ depending on the type of dictionary we are using. We generally run the algorithm using parameters set so that the number of invocations of this innermost step increases only gradually with problem size.

2. *Parameter settings.* The complexity of our primal-dual logarithmic barrier interior-point implementation depends on both the accuracy of the solution and the accuracy of the conjugate-gradient solver. The accuracy of the solution is determined by the two parameters `FeaTol`, `PDGapTol` controlling the number of barrier iterations, and the parameter `CGAccuracy`, which decides the accuracy of the conjugate-gradient solver and consequently the number of conjugate-gradient iterations. As the required solution accuracy goes up, the complexity goes up drastically. We recommend setting `FeaTol`, `PDGapTol`, and `CGAccuracy` at 10^{-1} for routine signal processing; we recommend 10^{-2} or 10^{-3} when one is interested in superresolution. We used the setting 10^{-1} for the computational experiments presented in Figures 2.6, 3.1–3.3, 5.1, and 5.3. In Figures 2.5, 3.2, and 5.2, we attempted to superresolve two cosines with close frequencies; thus we used the setting 10^{-2}. In Figure 4.1, we used the setting 10^{-3}.

3. *Signal complexity.* When the signal has a very sparse representation, the algorithm converges quickly. The signal `Carbon`, which contains only six atoms from a wavelet packet dictionary, takes about 10 seconds, whereas it takes about seven minutes for the signal `Gong`, which is much more complex.

4. *BP versus BPDN.* We employ the same interior-point implementation for BP and BPDN, except for a difference in the value of the regularization parameter δ: δ is small, e.g., 10^{-4} for BP, while $\delta = 1$ for BPDN. The choice $\delta = 1$ helps: it regularizes the central equations to be solved at each barrier iteration. Thus the BPDN implementation (BPDN_Interior.m on the Atomizer web site; see section 7.3) seems to converge more quickly than BP_Interior.m. For example, according to our experiments [4], it takes only three minutes to perform BPDN on the noisy `Gong` signal of length 1024 with a cosine packet dictionary at the parameter setting 10^{-3}; it takes about eight hours to perform BP on the signal `Gong` at the same parameter setting.

5. *Alternative implementations.* We have recently developed BP_Interior2.m and BPDN_Interior2.m, in which pdsco.m [44] is used to solve the perturbed LP problem (6.3), with a MATLAB form of LSQR being applied to the least-squares problem (6.5) to compute $\Delta\mathbf{y}$. pdsco.m has a more elaborate strategy for adjusting the barrier

parameter μ, and LSQR incorporates reliable stopping rules for its conjugate-gradient-type method. We wish to explore more fully the effect of `BPAccuracy` and `CGAccuracy` using these codes. Comparisons will be reported on the Atomizer web site.

6.6. Alternative Algorithms for BPDN. For certain dictionaries, Sardy et al. [42] showed how to minimize the BPDN function (5.1) using a block coordinate relaxation (BCR) method. They assumed that the columns of Φ are the union of (perhaps many) orthonormal complete matrices $\Phi^{(1)}, \ldots, \Phi^{(M)}$, and they took advantage of the closed form solution for ortho bases (see section 5.2).

For low-accuracy solutions, the BCR method may be significantly faster than our primal-dual conjugate-gradient approach. Further, the BCR approach extends naturally to complex signals (again assuming ortho union-complete bases).

7. Concluding Comments.

7.1. The Phenomenon of Ideal Atomic Decomposition. Empirically one often observes that BP provides a kind of *ideal atomic decomposition*. We mean that, in synthetic examples where "everything is known" and where by design there is a truly sparse solution to the atomic decomposition problem, BP will typically find exactly that sparse solution. We saw numerous examples of this phenomenon in preparing this paper and in Chen's thesis [4].

Recently, Donoho and Huo [15] have given a theoretical explanation. They have proven a number of results showing that mathematically exact solution of BP in overcomplete dictionaries can exhibit precisely the phenomenon of ideal atomic decomposition. For example, suppose we have a combined dictionary of sinusoids and Diracs and that the underlying object $\mathbf{y}$, a discrete-time signal of length n, is truly a superposition of fewer than $\sqrt{n}/2$ sinusoids and Diracs. Then (a) there is only one way a signal can be made up of so few sinusoids and Diracs, and (b) BP in such a dictionary will recover exactly that solution, with the frequencies and spike locations correctly determined, along with the amplitudes and signs.

Related results are obtained for dictionaries of wavelets and sinusoids and for dictionaries of ridgelets and wavelets, and applications are given to robust speech scrambling.

7.2. BP Image Processing. BP has been discussed here in the context of treating one-dimensional signals, although in fact it can be used to decompose multidimensional arrays, such as the two-way arrays of the type that represent images. However, such higher dimensional arrays typically lead to much larger optimization problems than in the one-dimensional case. When the original work for this paper was done (1993–1995), the linear programs and quadratic programs attacked here, with sizes of tens of thousands by hundreds of thousands, were among the very largest that had been attempted. We would not have dared then to consider the solution of the even larger problems that would arise in an image processing context.

Since that time, however, Moore's law and other related phenomena (such as declining memory prices and multicomputers) have made it possible to consider image processing experiments, at least on a limited scale. Huo's thesis [23] considered decompositions in an overcomplete dictionary consisting of wavelets and so-called edgelets [14]. This allowed both dictionaries to compete on an equal footing to use exactly the terms that best "explain" the image data.

In one experiment, Huo analyzed a digitized image and found that the humanly interpretable information was really carried by the edgelet component of the decomposition. This surprising finding shows that, in a certain sense, images are *not* made

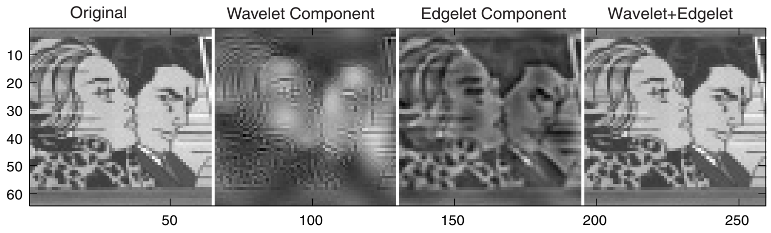

Fig. 7.1 *Decomposition of an image in an overcomplete dictionary composed of wavelets and edgelets; note the extent to which edgelets carry the visually interpretable component of the solution. From Huo* [23].

of wavelets, but instead, the perceptually important components of the image are carried by edgelets. This contradicts the frequent claim that wavelets are the optimal basis for image representation, which may stimulate discussion.

Figure 7.1 illustrates this finding; see [23] for more details. (Note: Huo's work did not actually apply the BP ideas and software developed here, but instead developed a modified approach inspired by BP but specifically adapted to the needs of large-scale image processing.)

A valuable role for BP has thus emerged. BP extracts a basis from a large dictionary according to objective principles, in a setting where we want to understand what the "right basis" for a given kind of data might be, rather than imposing our own opinion.

7.3. Reproducible Research. This paper has been written following the discipline of *reproducible research* [3]. As a complement to this article, we are releasing the underlying software environment via the WaveLab and Atomizer web sites:

http://www-stat.stanford.edu/˜wavelab/ http://www-stat.stanford.edu/˜atomizer/

REFERENCES

[1] R. E. Bixby, *Commentary: Progress in linear programming*, ORSA J. Comput., 6 (1994), pp. 15–22.

[2] P. Bloomfield and W. Steiger, *Least Absolute Deviations: Theory, Applications, and Algorithms*, Birkhäuser, Boston, 1983.

[3] J. Buckheit and D. L. Donoho, *WaveLab and reproducible research*, in Wavelets and Statistics, A. Antoniadis, ed., Springer-Verlag, Berlin, New York, 1995.

[4] S. S. Chen, *Basis Pursuit*, Ph.D. Thesis, Department of Statistics, Stanford University, Stanford, CA, 1995; see also http://www-stat.stanford.edu/˜atomizer/.

[5] S. Chen, S. A. Billings, and W. Luo, *Orthogonal least squares methods and their application to non-linear system identification*, Internat. J. Control, 50 (1989), pp. 1873–1896.

[6] R. R. Coifman and Y. Meyer, *Remarques sur l'analyze de Fourier à Fenêtre*, C. R. Acad. Sci. Paris (A), 312 (1991), pp. 259–261.

[7] R. R. Coifman and M. V. Wickerhauser, *Entropy-based algorithms for best-basis selection*, IEEE Trans. Inform. Theory, 38 (1992), pp. 713–718.

[8] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

[9] I. Daubechies, *Time-frequency localization operators: A geometric phase space approach*, IEEE Trans. Inform. Theory, 34 (1988), pp. 605–612.

[10] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.

[11] G. Davis, S. Mallat, and Z. Zhang, *Adaptive time-frequency decompositions*, Optical Engrg., 33 (1994), pp. 2183–2191.

[12] R. A. DeVore and V. N. Temlyakov, *Some remarks on greedy algorithms*, Adv. Comput. Math., 5 (1996), pp. 173–187.

[13] D. L. Donoho, *De-Noising by soft thresholding*, IEEE Trans. Inform. Theory, 41 (1995), pp. 613–627.

[14] D. L. Donoho, *Wedgelets: Nearly-minimax estimation of edges*, Ann. Statist., 27 (1999), pp. 859–897.

[15] D. L. Donoho and X. Huo, *Uncertainty Principles and Ideal Atomic Decomposition*, Technical Report 99-13, Department of Statistics, Stanford University, Stanford, CA, 1999; IEEE Trans. Inform. Theory, to appear.

[16] D. L. Donoho and I. M. Johnstone, *Ideal de-noising in an orthonormal basis chosen from a library of bases*, C. R. Acad. Sci. Paris Sér. I Math., 319 (1994), pp. 1317–1322.

[17] D. L. Donoho and I. M. Johnstone, *Empirical Atomic Decomposition*, manuscript, 1995.

[18] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, *Wavelet shrinkage: Asymptopia?* J. Roy. Statist. Soc. Ser. B, 57 (1995), pp. 301–369.

[19] D. Gabor, *Theory of communication*, J. Inst. Elect. Eng., 93 (1946), pp. 429–457.

[20] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Solving Reduced KKT Systems in Barrier Methods for Linear and Quadratic Programming*, Report SOL 91-7, Stanford University, Stanford, CA, July 1991.

[21] P. E. Gill, W. Murray, and M. H. Wright, *Numerical Linear Algebra and Optimization*, Addison-Wesley, Redwood City, CA, 1991.

[22] G. Golub and C. V Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[23] X. Huo, *Sparse Image Decomposition via Combined Transforms*, Ph.D. Thesis, Department of Statistics, Stanford University, Stanford, CA, 1999; see also http://www-stat.stanford.edu/research/abstracts/99-18.ps.

[24] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 375–395.

[25] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, Springer-Verlag, New York, 1989.

[26] Y. Li and F. Santosa, *A computational algorithm for minimizing total variation in image restoration*, IEEE Trans. Image Proc., 5 (1996), pp. 987–995.

[27] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Interior point methods for linear programming: Computational state of the art*, ORSA J. Comput., 6 (1994), pp. 1–14.

[28] S. Mallat and W. L. Hwang, *Singularity detection and processing with wavelets*, IEEE Trans. Inform. Theory, 38 (1992), pp. 617–643.

[29] S. Mallat and Z. Zhang, *Matching pursuit in a time-frequency dictionary*, IEEE Trans. Signal Proc., 41 (1993), pp. 3397–3415.

[30] S. Mallat and S. Zhong, *Wavelet transform maxima and multiscale edges*, in Wavelets and Their Applications, M. B. Ruskai, G. Beylkin, and R. Coifman, eds., Jones and Bartlett, Boston, 1992.

[31] MATLAB, The MathWorks, Inc., Natick, MA.

[32] N. Megiddo, *On finding primal- and dual-optimal bases*, ORSA J. Comput., 3 (1991), pp. 63–65.

[33] Y. Meyer, *Ondelettes sur l'intervalle*, Rev. Mat. Iberoamericana, 7 (1991), pp. 115–134.

[34] Y. Meyer, *Wavelets: Algorithms and Applications*, SIAM, Philadelphia, 1993.

[35] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.

[36] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[37] C. C. Paige and M. A. Saunders, *Algorithm* 583; *LSQR: Sparse linear equations and least-squares problems*, ACM Trans. Math. Software, 8 (1982), pp. 195–209.

[38] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*, in Proc. 27th Asilomar Conference on Signals, Systems and Computers, A. Singh, ed., IEEE Comput. Soc. Press, Los Alamitos, CA, 1993.

[39] S. Qian and D. Chen, *Signal representation using adaptive normalized Gaussian functions*, Signal Process., 36 (1994), pp. 1–11.

[40] C. Roos, T. Terlaky, and J.-Ph. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, Wiley, Chichester, UK, 1997.

[41] L. J. Rudin, S. Osher, and E. Fatemi, *Nonlinear total-variation-based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[42] S. Sardy, A. G. Bruce, and P. Tseng, *Block coordinate relaxation methods for nonparametric wavelet denoising*, J. Comput. Graph. Statist., 9 (2000), pp. 361–379.

[43] M. A. Saunders, *Commentary: Major Cholesky would feel proud*, ORSA J. Comput., 6 (1994), pp. 23–27.

[44] M. A. Saunders, *pdsco.m,* Matlab *code for minimizing convex separable objective functions subject to $Ax = b$, $x \geq 0$*, http://www-stat.stanford.edu/~atomizer/.

[45] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, *Shiftable multiscale transforms*, IEEE Trans. Inform. Theory, 38 (1992), pp. 587–607.

[46] M. J. Todd, *Commentary: Theory and practice for interior point methods*, ORSA J. Comput., 6 (1994), pp. 28–31.

[47] R. J. Vanderbei, *Commentary: Interior point methods: Algorithms and formulations*, ORSA J. Comput., 6 (1994), pp. 32–34.

[48] L. F. Villemoes, *Best approximation with Walsh atoms*, Constr. Approx., 13 (1997), pp. 329–355.

[49] M. H. Wright, *Interior methods for constrained optimization*, Acta Numerica, 1992, pp. 341–407.

[50] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1996; see also http://www.siam.org/books/swright/.