9.5 APPROXIMATIONS WITH PURSUITS [3]

A music recording often includes notes of different durations at the same time, which means that such a signal is not well represented in a best local cosine basis. The same musical note may also have different durations when played at different times, in which case a best wavelet packet basis is also not well adapted to represent this sound. To approximate musical signals efficiently, the decomposition must have the same flexibility as the composer, who can freely choose the time-frequency atoms (notes) that are best adapted to represent a sound.

Wavelet packet and local cosine dictionaries include $P = N \log_2 N$ different vectors. The set of orthogonal bases is much smaller than the set of non-orthogonal bases that could be constructed by choosing N linearly independent vectors from these P. To improve the approximation of complex signals such as music recordings, we study general non-orthogonal signal decompositions.

Consider the space of signals of size N. Let $\mathcal{D} = \{g_p\}_{0 \leq p < P}$ be a redundant dictionary of $P > N$ vectors, which includes at least N linearly independent vectors. For any $M \geq 1$, an approximation f_M of f may be calculated with a linear

combination of any M dictionary vectors:

$$f_M = \sum_{m=0}^{M-1} a[p_m]\, g_{p_m}.$$

The freedom of choice opens the door to a considerable combinatorial explosion. For general dictionaries of $P > N$ vectors, computing the approximation f_M that minimizes $\|f - f_M\|$ is an *NP hard problem* [151]. This means that there is no known polynomial time algorithm that can solve this optimization.

Pursuit algorithms reduce the computational complexity by searching for efficient but non-optimal approximations. A basis pursuit formulates the search as a linear programming problem, providing remarkably good approximations with $O(N^{3.5}\log_2^{3.5} N)$ operations. For large signals, this remains prohibitive. Matching pursuits are faster greedy algorithms whose applications to large time-frequency dictionaries is described in Section 9.5.2. An orthogonalized pursuit is presented in Section 9.5.3.

9.5.1 Basis Pursuit

We study the construction of a "best" basis $\mathcal{B}$, not necessarily orthogonal, for efficiently approximating a signal f. The N vectors of $\mathcal{B} = \{g_{p_m}\}_{0 \le m < N}$ are selected from a redundant dictionary $\mathcal{D} = \{g_p\}_{0 \le p < P}$ with a pursuit elaborated by Chen and Donoho [119]. Let us decompose f in this basis:

$$f = \sum_{m=0}^{N-1} a[p_m]\, g_{p_m}. \tag{9.76}$$

If we had restricted ourselves to orthogonal bases, Section 9.4.1 explains that the basis choice would be optimized by minimizing

$$C(f,\mathcal{B}) = \sum_{m=0}^{N-1} \Phi\left(\frac{|a[p_m]|^2}{\|f\|^2}\right), \tag{9.77}$$

where $\Phi(u)$ is concave. For non-orthogonal bases, this result does not hold in general.

Despite the absence of orthogonality, a basis pursuit searches for a "best" basis that minimizes (9.77) for $\Phi(u) = u^{1/2}$:

$$C(f,\mathcal{B}) = \frac{1}{\|f\|} \sum_{m=0}^{N-1} |a[p_m]|. \tag{9.78}$$

Minimizing the $\mathbf{l^1}$ norm of the decomposition coefficients avoids diffusing the energy of f among many vectors. It reduces cancellations between the vectors $a[p_m]g_{p_m}$ that decompose f, because such cancellations increase $|a[p_m]|$ and thus increase the cost (9.78). The minimization of an $\mathbf{l^1}$ norm is also related to linear programming, which leads to fast computational algorithms.

Linear Programming Instead of immediately isolating subsets of N vectors in the dictionary $\mathcal{D}$, a linear system of size P is written with all dictionary vectors

$$\sum_{p=0}^{P-1} a[p]\, g_p[n] = f[n], \tag{9.79}$$

while trying to minimize

$$\sum_{p=0}^{P-1} |a[p]|. \tag{9.80}$$

The system (9.79) can be expressed in matrix form with the $P \times N$ matrix $G = \{g_p[n]\}_{0 \le n < N, 0 \le p < P}$

$$Ga = f. \tag{9.81}$$

Although the minimization of (9.80) is nonlinear, it can be reformulated as a linear programming problem.

A standard-form linear programming problem [28] is a constrained optimization over positive vectors of size L. Let $b[n]$ be a vector of size $N < L$, $c[p]$ a non-zero vector of size L and $A[n, p]$ an $L \times N$ matrix. We must find $x[p] \in \mathbb{R}^L$ such that $x[p] \ge 0$, while minimizing

$$\sum_{p=0}^{L-1} x[p]\, c[p] \tag{9.82}$$

subject to

$$Ax = b.$$

To reformulate the minimization of (9.80) subject to (9.81) as a linear programming problem, we introduce "slack variables" $u[p] \ge 0$ and $v[p] \ge 0$ such that

$$a[p] = u[p] - v[p].$$

As a result

$$Ga = Gu - Gv = f \tag{9.83}$$

and

$$\sum_{p=0}^{P-1} |a[p]| = \sum_{p=0}^{P-1} u[p] + \sum_{p=0}^{P-1} v[p]. \tag{9.84}$$

We thus obtain a standard form linear programming of size $L = 2P$ with

$$A = (G, -G) \;\; , \;\; x = \begin{pmatrix} u \\ v \end{pmatrix} \;\; , \;\; b = f \;\; , \;\; c = 1.$$

The matrix A of size $N \times L$ has rank N because the dictionary $\mathcal{D}$ includes N linearly independent vectors. A standard result of linear programming [28] proves

that the vector x has at most N non-zero coefficients. One can also verify that if $a[p] > 0$ then $a[p] = u[p]$ and $v[p] = 0$ whereas if $a[p] \leq 0$ then $a[p] = v[p]$ and $u[p] = 0$. In the non-degenerate case, which is most often encountered, the non-zero coefficients of $x[p]$ thus correspond to N indices $\{p_m\}_{0 \leq m < N}$ such that $\{g_{p_m}\}_{0 \leq m < N}$ are linearly independent. This is the best basis of $\mathbb{R}^N$ that minimizes the cost (9.78).

Linear Programming Computations The collection of feasible points $\{x : Ax = b \ , \ x \geq 0\}$ is a convex polyhedron in $\mathbb{R}^L$. The vertices of this polyhedron are solutions $x[p]$ having at most N non-zero coefficients. The linear cost (9.82) can be minimum only at a vertex of this polyhedron. In the non-degenerate case, the N non-zero coefficients correspond to N column vectors $\mathcal{B} = \{g_{p_m}\}_{0 \leq m < N}$ that form a basis.

One can also prove [28] that if the cost is not minimum at a given vertex then there exists an adjacent vertex whose cost is smaller. The simplex algorithm takes advantage of this property by jumping from one vertex to an adjacent vertex while reducing the cost (9.82). Going to an adjacent vertex means that one of the zero coefficients of $x[p]$ becomes non-zero while one non-zero coefficient is set to zero. This is equivalent to modifying the basis $\mathcal{B}$ by replacing one vector by another vector of $\mathcal{D}$. The simplex algorithm thus progressively improves the basis by appropriate modifications of its vectors, one at a time. In the worst case, all vertices of the polyhedron will be visited before finding the solution, but the average case is much more favorable.

Since the 1980's, more effective interior point procedures have been developed. Karmarkar's interior point algorithm [234] begins in the middle of the polyhedron and converges by iterative steps towards the vertex solution, while remaining inside the convex polyhedron. For finite precision calculations, when the algorithm has converged close enough to a vertex, it jumps directly to the corresponding vertex, which is guaranteed to be the solution. The middle of the polyhedron corresponds to a decomposition of f over all vectors of $\mathcal{D}$, typically with $P > N$ non-zero coefficients. When moving towards a vertex some coefficients progressively decrease while others increase to improve the cost (9.82). If only N decomposition coefficients are significant, jumping to the vertex is equivalent to setting all other coefficients to zero. Each step requires computing the solution of a linear system. If A is an $N \times L$ matrix then Karmarkar's algorithm terminates with $O(L^{3.5})$ operations. Mathematical work on interior point methods has led to a large variety of approaches that are summarized in [252]. The basis pursuit of Chen and Donoho [119] is implemented in WAVELAB with a "Log-barrier" method [252], which converges more quickly than Karmarkar's original algorithm

Wavelet Packet and Local Cosine Dictionaries These dictionaries have $P = N \log_2 N$ time-frequency atoms. A straightforward implementation of interior point algorithms thus requires $O(N^{3.5} \log_2^{3.5} N)$ operations. By using the fast wavelet packet and local cosine transforms together with heuristic computational

rules, the number of operations is considerably reduced [119]. The algorithm still remains relatively slow and the computations become prohibitive for $N \geq 1000$.

Figure 9.11 decomposes a synthetic signal that has two high frequency transients followed by two lower frequency transients and two Diracs for $t < 0.2$. The signal then includes two linear chirps that cross each other and which are superimposed with localized sinusoidal waves. In a dictionary of wavelet packet bases calculated with a Daubechies 8 filter, the best basis shown in Figure 9.11(c) optimizes the division of the frequency axis, but it has no flexibility in time. It is therefore not adapted to the time evolution of the signal components. A basis pursuit algorithm adapts the wavelet packet choice to the local signal structures; Figure 9.11(d) shows that it better reveals its time-frequency properties.

9.5.2 Matching Pursuit

Despite the linear programming approach, a basis pursuit is computationally expensive because it minimizes a global cost function over all dictionary vectors. The matching pursuit introduced by Mallat and Zhang [259] reduces the computational complexity with a greedy strategy. It is closely related to projection pursuit algorithms used in statistics [184] and to shape-gain vector quantizations [27]. Vectors are selected one by one from the dictionary, while optimizing the signal approximation at each step.

Let $\mathcal{D} = \{g_\gamma\}_{\gamma \in \Gamma}$ be a dictionary of $P > N$ vectors, having a unit norm. This dictionary includes N linearly independent vectors that define a basis of the space $\mathbb{C}^N$ of signals of size N. A matching pursuit begins by projecting f on a vector $g_{\gamma_0} \in \mathcal{D}$ and computing the residue Rf:

$$f = \langle f, g_{\gamma_0} \rangle \, g_{\gamma_0} + Rf. \tag{9.85}$$

Since Rf is orthogonal to g_{γ_0}

$$\|f\|^2 = |\langle f, g_{\gamma_0} \rangle|^2 + \|Rf\|^2. \tag{9.86}$$

To minimize $\|Rf\|$ we must choose $g_{\gamma_0} \in \mathcal{D}$ such that $|\langle f, g_{\gamma_0} \rangle|$ is maximum. In some cases, it is computationally more efficient to find a vector g_{γ_0} that is almost optimal:

$$|\langle f, g_{\gamma_0} \rangle| \geq \alpha \sup_{\gamma \in \Gamma} |\langle f, g_\gamma \rangle|, \tag{9.87}$$

where $\alpha \in (0, 1]$ is an optimality factor. The pursuit iterates this procedure by subdecomposing the residue. Let $R^0 f = f$. Suppose that the m^{th} order residue $R^m f$ is already computed, for $m \geq 0$. The next iteration chooses $g_{\gamma_m} \in \mathcal{D}$ such that

$$|\langle R^m f, g_{\gamma_m} \rangle| \geq \alpha \sup_{\gamma \in \Gamma} |\langle R^m f, g_\gamma \rangle|, \tag{9.88}$$

and projects $R^m f$ on g_{γ_m}:

$$R^m f = \langle R^m f, g_{\gamma_m} \rangle \, g_{\gamma_m} + R^{m+1} f. \tag{9.89}$$

The orthogonality of $R^{m+1}f$ and g_{γ_m} implies

$$\|R^m f\|^2 = |\langle R^m f, g_{\gamma_m}\rangle|^2 + \|R^{m+1} f\|^2. \tag{9.90}$$

Summing (9.89) from m between 0 and $M-1$ yields

$$f = \sum_{m=0}^{M-1} \langle R^m f, g_{\gamma_m}\rangle\, g_{\gamma_m} + R^M f. \tag{9.91}$$

Similarly, summing (9.90) from m between 0 and $M-1$ gives

$$\|f\|^2 = \sum_{m=0}^{M-1} |\langle R^m f, g_{\gamma_m}\rangle|^2 + \|R^M f\|^2. \tag{9.92}$$

The following theorem proves that $\|R^m f\|$ converges exponentially to 0 when m tends to infinity.

Theorem 9.10 *There exists $\lambda > 0$ such that for all $m \geq 0$*

$$\|R^m f\| \leq 2^{-\lambda m}\, \|f\|. \tag{9.93}$$

As a consequence

$$f = \sum_{m=0}^{+\infty} \langle R^m f, g_{\gamma_m}\rangle\, g_{\gamma_m}, \tag{9.94}$$

and

$$\|f\|^2 = \sum_{m=0}^{+\infty} |\langle R^m f, g_{\gamma_m}\rangle|^2. \tag{9.95}$$

Proof[3]. Let us first verify that there exists $\beta > 0$ such that for any $f \in \mathbb{C}^N$

$$\sup_{\gamma\in\Gamma} |\langle f, g_\gamma\rangle| \geq \beta \|f\|. \tag{9.96}$$

Suppose that it is not possible to find such a β. This means that we can construct $\{f_m\}_{m\in\mathbb{N}}$ with $\|f_m\| = 1$ and

$$\lim_{m\to+\infty} \sup_{\gamma\in\Gamma} |\langle f_m, g_\gamma\rangle| = 0. \tag{9.97}$$

Since the unit sphere of $\mathbb{C}^N$ is compact, there exists a sub-sequence $\{f_{m_k}\}_{k\in\mathbb{N}}$ that converges to a unit vector $f \in \mathbb{C}^N$. It follows that

$$\sup_{\gamma\in\Gamma} |\langle f, g_\gamma\rangle| = \lim_{k\to+\infty} \sup_{\gamma\in\Gamma} |\langle f_{m_k}, g_\gamma\rangle| = 0 \tag{9.98}$$

so $\langle f, g_\gamma\rangle = 0$ for all $g_\gamma \in \mathcal{D}$. Since $\mathcal{D}$ contains a basis of $\mathbb{C}^N$, necessarily $f = 0$ which is not possible because $\|f\| = 1$. This proves that our initial assumption is wrong, and hence there exists β such that (9.96) holds.

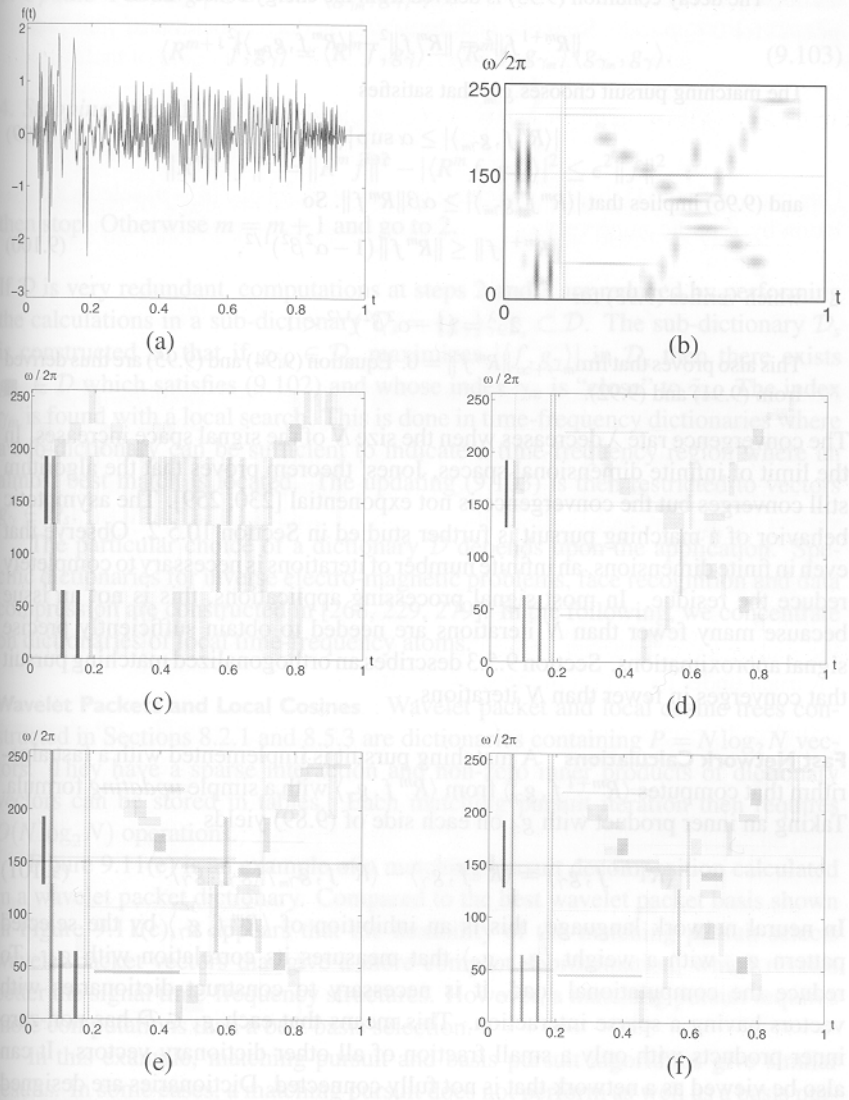

FIGURE 9.11 (a): Signal synthesized with a sum of chirps, truncated sinusoids, short time transients and Diracs. The time-frequency images display the atoms selected by different adaptive time-frequency transforms. The darkness is proportional to the coefficient amplitude. (b): Gabor matching pursuit. Each dark blob is the Wigner-Ville distribution of a selected Gabor atom. (c): Heisenberg boxes of a best wavelet packet basis calculated with Daubechies 8 filter. (d): Wavelet packet basis pursuit. (e): Wavelet packet matching pursuit. (f): Wavelet packet orthogonal matching pursuit.

The decay condition (9.93) is derived from the energy conservation

$$\|R^{m+1}f\|^2 = \|R^m f\|^2 - |\langle R^m f, g_{p_m}\rangle|^2.$$

The matching pursuit chooses g_{γ_m} that satisfies

$$|\langle R^m f, g_{\gamma_m}\rangle| \geq \alpha \sup_{\gamma\in\Gamma} |\langle R^m f, g_\gamma\rangle|, \tag{9.99}$$

and (9.96) implies that $|\langle R^m f, g_{\gamma_m}\rangle| \geq \alpha\beta\|R^m f\|$. So

$$\|R^{m+1}f\| \leq \|R^m f\|\,(1-\alpha^2\beta^2)^{1/2}, \tag{9.100}$$

which verifies (9.93) for

$$2^{-\lambda} = (1-\alpha^2\beta^2)^{1/2} < 1.$$

This also proves that $\lim_{m\to+\infty} \|R^m f\| = 0$. Equation (9.94) and (9.95) are thus derived from (9.91) and (9.92). ■

The convergence rate λ decreases when the size N of the signal space increases. In the limit of infinite dimensional spaces, Jones' theorem proves that the algorithm still converges but the convergence is not exponential [230, 259]. The asymptotic behavior of a matching pursuit is further studied in Section 10.5.2. Observe that even in finite dimensions, an infinite number of iterations is necessary to completely reduce the residue. In most signal processing applications, this is not an issue because many fewer than N iterations are needed to obtain sufficiently precise signal approximations. Section 9.5.3 describes an orthogonalized matching pursuit that converges in fewer than N iterations.

Fast Network Calculations A matching pursuit is implemented with a fast algorithm that computes $\langle R^{m+1}f, g_\gamma\rangle$ from $\langle R^m f, g_\gamma\rangle$ with a simple *updating* formula. Taking an inner product with g_γ on each side of (9.89) yields

$$\langle R^{m+1}f, g_\gamma\rangle = \langle R^m f, g_\gamma\rangle - \langle R^m f, g_{\gamma_m}\rangle \langle g_{\gamma_m}, g_\gamma\rangle. \tag{9.101}$$

In neural network language, this is an inhibition of $\langle R^m f, g_\gamma\rangle$ by the selected pattern g_{γ_m} with a weight $\langle g_{\gamma_m}, g_\gamma\rangle$ that measures its correlation with g_γ. To reduce the computational load, it is necessary to construct dictionaries with vectors having a sparse interaction. This means that each $g_\gamma \in \mathcal{D}$ has non-zero inner products with only a small fraction of all other dictionary vectors. It can also be viewed as a network that is not fully connected. Dictionaries are designed so that non-zero weights $\langle g_\alpha, g_\gamma\rangle$ can be retrieved from memory or computed with $O(1)$ operations. A matching pursuit with a relative precision ϵ is implemented with the following steps.

1. *Initialization* Set $m = 0$ and compute $\{\langle f, g_\gamma\rangle\}_{\gamma\in\Gamma}$.
2. *Best match* Find $g_{\gamma_m} \in \mathcal{D}$ such that

$$|\langle R^m f, g_{\gamma_m}\rangle| \geq \alpha \sup_{\gamma\in\Gamma} |\langle R^m f, g_\gamma\rangle|. \tag{9.102}$$

3. *Update* For all $g_\gamma \in \mathcal{D}$ with $\langle g_{\gamma_m}, g_\gamma \rangle \neq 0$

$$\langle R^{m+1} f, g_\gamma \rangle = \langle R^m f, g_\gamma \rangle - \langle R^m f, g_{\gamma_m} \rangle \langle g_{\gamma_m}, g_\gamma \rangle. \tag{9.103}$$

4. *Stopping rule* If

$$\|R^{m+1} f\|^2 = \|R^m f\|^2 - |\langle R^m f, g_{\gamma_m} \rangle|^2 \leq \epsilon^2 \|f\|^2$$

then stop. Otherwise $m = m+1$ and go to 2.

If $\mathcal{D}$ is very redundant, computations at steps 2 and 3 are reduced by performing the calculations in a sub-dictionary $\mathcal{D}_s = \{g_\gamma\}_{\gamma \in \Gamma_s} \subset \mathcal{D}$. The sub-dictionary $\mathcal{D}_s$ is constructed so that if $g_{\tilde{\gamma}_m} \in \mathcal{D}_s$ maximizes $|\langle f, g_\gamma \rangle|$ in $\mathcal{D}_s$ then there exists $g_{\gamma_m} \in \mathcal{D}$ which satisfies (9.102) and whose index γ_m is "close" to $\tilde{\gamma}_m$. The index γ_m is found with a local search. This is done in time-frequency dictionaries where a sub-dictionary can be sufficient to indicate a time-frequency region where an almost best match is located. The updating (9.103) is then restricted to vectors $g_\gamma \in \mathcal{D}_s$.

The particular choice of a dictionary $\mathcal{D}$ depends upon the application. Specific dictionaries for inverse electro-magnetic problems, face recognition and data compression are constructed in [268, 229, 279]. In the following, we concentrate on dictionaries of local time-frequency atoms.

Wavelet Packets and Local Cosines Wavelet packet and local cosine trees constructed in Sections 8.2.1 and 8.5.3 are dictionaries containing $P = N \log_2 N$ vectors. They have a sparse interaction and non-zero inner products of dictionary vectors can be stored in tables. Each matching pursuit iteration then requires $O(N \log_2 N)$ operations.

Figure 9.11(e) is an example of a matching pursuit decomposition calculated in a wavelet packet dictionary. Compared to the best wavelet packet basis shown in Figure 9.11(c), it appears that the flexibility of the matching pursuit selects wavelet packet vectors that give a more compact approximation, which reveals better the signal time-frequency structures. However, a matching pursuit requires more computations than a best basis selection.

In this example, matching pursuit and basis pursuit algorithms give similar results. In some cases, a matching pursuit does not perform as well as a basis pursuit because the greedy strategy selects decomposition vectors one by one [159]. Choosing decomposition vectors by optimizing a correlation inner product can produce a partial loss of time and frequency resolution [119]. High resolution pursuits avoid the loss of resolution in time by using non-linear correlation measures [195, 223] but the greediness can still have adverse effects.

Translation Invariance Section 5.4 explains that decompositions in orthogonal bases lack translation invariance and are thus difficult to use for pattern recognition. Matching pursuits are translation invariant if calculated in translation invariant

dictionaries. A dictionary $\mathcal{D}$ is *translation invariant* if for any $g_\gamma \in \mathcal{D}$ then $g_\gamma[n-p] \in \mathcal{D}$ for $0 \leq p < N$. Suppose that the matching decomposition of f in $\mathcal{D}$ is

$$f[n] = \sum_{m=0}^{M-1} \langle R^m f, g_{\gamma_m} \rangle \, g_{\gamma_m}[n] + R^M f[n]. \tag{9.104}$$

One can verify [151] that the matching pursuit of $f_p[n] = f[n-p]$ selects a translation by p of the same vectors g_{γ_m} with the same decomposition coefficients

$$f_p[n] = \sum_{m=0}^{M-1} \langle R^m f, g_{\gamma_m} \rangle \, g_{\gamma_m}[n-p] + R^M f_p[n].$$

Patterns can thus be characterized independently of their position. The same translation invariance property is valid for a basis pursuit. However, translation invariant dictionaries are necessarily very large, which often leads to prohibitive calculations. Wavelet packet and local cosine dictionaries are not translation invariant because at each scale 2^j the waveforms are translated only by $k2^j$ with $k \in \mathbb{Z}$.

Translation invariance is generalized as an invariance with respect to any group action [151]. A frequency translation is another example of a group operation. If the dictionary is invariant under the action of a group then the pursuit remains invariant under the action of the same group.

Gabor Dictionary A time and frequency translation invariant Gabor dictionary is constructed by Qian and Chen [287] as well as Mallat and Zhong [259], by scaling, translating and modulating a Gaussian window. Gaussian windows are used because of their optimal time and frequency energy concentration, proved by the uncertainty Theorem 2.5.

For each scale 2^j, a discrete window of period N is designed by sampling and periodizing a Gaussian $g(t) = 2^{1/4}\exp(-\pi t^2)$:

$$g_j[n] = K_j \sum_{p=-\infty}^{+\infty} g\left(\frac{n-pN}{2^j}\right).$$

The constant K_j is adjusted so that $\|g_j\| = 1$. This window is then translated in time and frequency. Let Γ be the set of indexes $\gamma = (p,k,2^j)$ for $(p,k) \in [0,N-1]^2$ and $j \in [0, \log_2 N]$. A discrete Gabor atom is

$$g_\gamma[n] = g_j[n-p] \exp\left(\frac{i2\pi kn}{N}\right). \tag{9.105}$$

The resulting Gabor dictionary $\mathcal{D} = \{g_\gamma\}_{\gamma \in \Gamma}$ is time and frequency translation invariant modulo N. A matching pursuit decomposes real signals in this dictionary by grouping atoms g_{γ^+} and g_{γ^-} with $\gamma^\pm = (p, \pm k, 2^j)$. At each iteration, instead

of projecting $R^m f$ over an atom g_γ, the matching pursuit computes its projection on the plane generated by $(g_{\gamma^+}, g_{\gamma^-})$. Since $R^m f[n]$ is real, one can verify that this is equivalent to projecting $R^m f$ on a real vector that can be written

$$g_\gamma^\phi[n] = K_{j,\phi}\, g_j[n-p] \cos\left(\frac{2\pi k n}{N} + \phi\right).$$

The constant $K_{j,\phi}$ sets the norm of this vector to 1 and the phase ϕ is optimized to maximize the inner product with $R^m f$. Matching pursuit iterations yield

$$f = \sum_{m=0}^{+\infty} \langle R^m f, g_{\gamma_m}^{\phi_m} \rangle\, g_{\gamma_m}^{\phi_m}. \tag{9.106}$$

This decomposition is represented by a time-frequency energy distribution obtained by summing the Wigner-Ville distribution $P_V g_{\gamma_m}[n,k]$ of the complex atoms g_{γ_m}:

$$P_M f[n,k] = \sum_{m=0}^{+\infty} |\langle R^m f, g_{\gamma_m}^{\phi_m} \rangle|^2\, P_V g_{\gamma_m}[n,k]. \tag{9.107}$$

Since the window is Gaussian, if $\gamma_m = (p_m, k_m, 2^{j_m})$ then $P_V g_{\gamma_m}$ is a two-dimensional Gaussian blob centered at (p_m, k_m) in the time-frequency plane. It is scaled by 2^{j_m} in time and $N2^{-j_m}$ in frequency.

Example 9.1 Figure 9.11(b) gives the matching pursuit energy distribution $P_M f[n,k]$ of a synthetic signal. The inner structures of this signal appear more clearly than with a wavelet packet matching pursuit because Gabor atoms have a better time-frequency localization than wavelet packets, and they are translated over a finer time-frequency grid.

Example 9.2 Figure 9.12 shows the Gabor matching pursuit decomposition of the word "greasy", sampled at 16 kHz. The time-frequency energy distribution shows the low-frequency component of the "g" and the quick burst transition to the "ea". The "ea" has many harmonics that are lined up. The "s" is noise whose time-frequency energy is spread over a high-frequency interval. Most of the signal energy is characterized by a few time-frequency atoms. For $m = 250$ atoms, $\|R^m f\|/\|f\| = .169$, although the signal has 5782 samples, and the sound recovered from these atoms is of excellent audio-quality.

Matching pursuit calculations in a Gabor dictionary are performed with a subdictionary $\mathcal{D}_s$. At each scale 2^j, the time-frequency indexes (p,k) are subsampled at intervals $a2^j$ and $aN2^{-j}$ where the sampling factor $a < 1$ is small enough to detect the time-frequency regions where the signal has high energy components. The step 2 of the matching pursuit iteration (9.102) finds the Gabor atom in $g_{\tilde{\gamma}_m} \in \mathcal{D}_s$ which best matches the signal residue. This match is then improved by searching

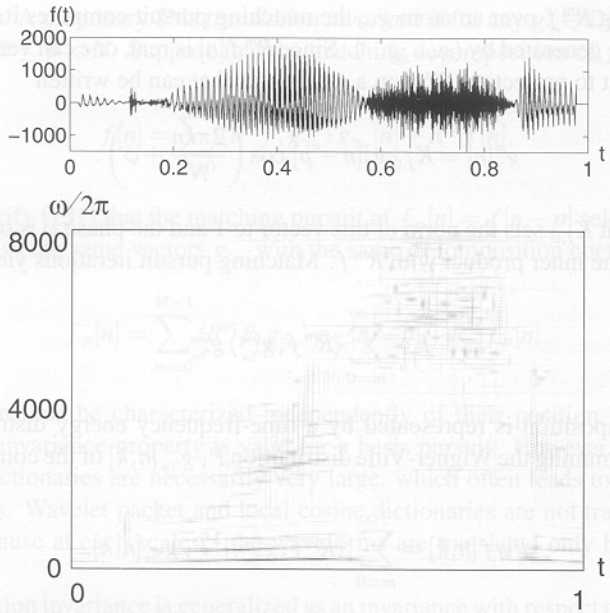

FIGURE 9.12 Speech recording of the word "greasy" sampled at $16kHz$. In the time-frequency image, the dark blobs of various sizes are the Wigner-Ville distributions of a Gabor functions selected by the matching pursuit.

for an atom $g_{\gamma_m} \in \mathcal{D}$ whose index γ_m is close to $\tilde{\gamma}_m$ and which locally maximizes the correlation with the signal residue. The updating formula (9.103) is calculated for $g_\gamma \in \mathcal{D}_s$. Inner products between two Gabor atoms are computed with an analytic formula [259]. Since $\mathcal{D}_s$ has $O(N \log_2 N)$ vectors, one can verify that each matching pursuit iteration is implemented with $O(N \log_2 N)$ calculations.

9.5.3 Orthogonal Matching Pursuit

The approximations of a matching pursuit are improved by orthogonalizing the directions of projection, with a Gram-Schmidt procedure proposed by Pati et al. [280] and Davis et al. [152]. The resulting orthogonal pursuit converges with a finite number of iterations, which is not the case for a non-orthogonal pursuit. The price to be paid is the important computational cost of the Gram-Schmidt orthogonalization.

The vector g_{γ_m} selected by the matching algorithm is a priori not orthogonal to the previously selected vectors $\{g_{\gamma_p}\}_{0 \le p < m}$. When subtracting the projection of $R^m f$ over g_{γ_m} the algorithm reintroduces new components in the directions of $\{g_{\gamma_p}\}_{0 \le p < m}$. This is avoided by projecting the residues on an orthogonal family $\{u_p\}_{0 \le p < m}$ computed from $\{g_{\gamma_p}\}_{0 \le p < m}$.

Let us initialize $u_0 = g_{\gamma_0}$. For $m \ge 0$, an orthogonal matching pursuit selects

g_{γ_m} that satisfies

$$|\langle R^m f, g_{\gamma_m}\rangle| \geq \alpha \sup_{\gamma\in\Gamma} |\langle R^m f, g_\gamma\rangle|. \tag{9.108}$$

The Gram-Schmidt algorithm orthogonalizes g_{γ_m} with respect to $\{g_{\gamma_p}\}_{0\leq p<m}$ and defines

$$u_m = g_{\gamma_m} - \sum_{p=0}^{m-1} \frac{\langle g_{\gamma_m}, u_p\rangle}{\|u_p\|^2}\, u_p. \tag{9.109}$$

The residue $R^m f$ is projected on u_m instead of g_{γ_m}:

$$R^m f = \frac{\langle R^m f, u_m\rangle}{\|u_m\|^2}\, u_m + R^{m+1} f. \tag{9.110}$$

Summing this equation for $0 \leq m < k$ yields

$$\begin{aligned} f &= \sum_{m=0}^{k-1} \frac{\langle R^m f, u_m\rangle}{\|u_m\|^2}\, u_m + R^k f \\ &= P_{\mathbf{V}_k} f + R^k f, \end{aligned} \tag{9.111}$$

where $P_{\mathbf{V}_k}$ is the orthogonal projector on the space $\mathbf{V}_k$ generated by $\{u_m\}_{0\leq m<k}$. The Gram-Schmidt algorithm ensures that $\{g_{\gamma_m}\}_{0\leq m<k}$ is also a basis of $\mathbf{V}_k$. For any $k \geq 0$ the residue $R^k f$ is the component of f that is orthogonal to $\mathbf{V}_k$. For $m = k$ (9.109) implies that

$$\langle R^m f, u_m\rangle = \langle R^m f, g_{\gamma_m}\rangle. \tag{9.112}$$

Since $\mathbf{V}_k$ has dimension k there exists $M \leq N$ such that $f \in \mathbf{V}_M$, so $R^M f = 0$ and inserting (9.112) in (9.111) for $k = M$ yields

$$f = \sum_{m=0}^{M-1} \frac{\langle R^m f, g_{\gamma_m}\rangle}{\|u_m\|^2}\, u_m. \tag{9.113}$$

The convergence is obtained with a finite number M of iterations. This is a decomposition in a family of orthogonal vectors so

$$\|f\|^2 = \sum_{m=0}^{M-1} \frac{|\langle R^m f, g_{\gamma_m}\rangle|^2}{\|u_m\|^2}. \tag{9.114}$$

To expand f over the original dictionary vectors $\{g_{\gamma_m}\}_{0\leq m<M}$, we must perform a change of basis. The triangular Gram-Schmidt relations (9.109) are inverted to expand u_m in $\{g_{\gamma_p}\}_{0\leq p\leq m}$:

$$u_m = \sum_{p=0}^{m} b[p,m]\, g_{\gamma_p}. \tag{9.115}$$

Inserting this expression into (9.113) gives

$$f = \sum_{p=0}^{M-1} a[\gamma_p]\, g_{\gamma_p} \tag{9.116}$$

with

$$a[\gamma_p] = \sum_{m=p}^{M-1} b[p,m]\, \frac{\langle R^m f, g_{\gamma_m}\rangle}{\|u_m\|^2}.$$

During the first few iterations, the pursuit often selects nearly orthogonal vectors, so the Gram-Schmidt orthogonalization is not needed. The orthogonal and non-orthogonal pursuits are then nearly the same. When the number of iterations increases and gets close to N, the residues of an orthogonal pursuit have norms that decrease faster than for a non-orthogonal pursuit.

Figure 9.11(f) displays the wavelet packets selected by an orthogonal matching pursuit. A comparison with Figure 9.11(e) shows that the orthogonal and non-orthogonal pursuits selects nearly the same wavelet packets having a high amplitude inner product. These wavelet packets are selected during the first few iterations, and since they are nearly orthogonal the Gram-Schmidt orthogonalization does not modify much the pursuit. The difference between the two algorithms becomes significant when selected wavelet packet vectors have non-negligible inner products, which happens when the number of iterations is large.

The Gram-Schmidt summation (9.109) must be carefully implemented to avoid numerical instabilities [29]. Orthogonalizing M vectors requires $O(NM^2)$ operations. In wavelet packet, local cosine and Gabor dictionaries, M matching pursuit iterations are calculated with $O(MN\log_2 N)$ operations. For M large, the Gram-Schmidt orthogonalization increases very significantly the computational complexity of the pursuit. The non-orthogonal pursuit is thus more often used for large signals.

9.6 PROBLEMS

9.1. [1] Prove that for any $f \in \mathbf{L}^2[0,1]$, if $\|f\|_V < +\infty$ then $\|f\|_\infty < +\infty$. Verify that one can find an image $f \in \mathbf{L}^2[0,1]^2$ such that $\|f\|_V < +\infty$ and $\|f\|_\infty = +\infty$.

9.2. [1] Prove that if $f \in \mathbf{W}^s(\mathbb{R})$ with $s > p + 1/2$ then $f \in \mathbf{C}^p$.

9.3. [1] The family of discrete polynomials $\{p_k[n] = n^k\}_{0\le k<N}$ is a basis of $\mathbb{C}^N$.

(a) Implement in WAVELAB a Gram-Schmidt algorithm that orthogonalizes $\{p_k\}_{0\le k<N}$.

(b) Let f be a signal of size N. Compute the polynomial f_k of degree k which minimizes $\|f - f_k\|$. Perform numerical experiments on signals f that are uniformly smooth and piecewise smooth. Compare the approximation error with the error obtained by approximating f with the k lower frequency Fourier coefficients.

9.4. [1] If f has a finite total variation $\|f\|_V$ on $[0,1]$, prove that its linear approximation in a wavelet basis satisfies $\epsilon_l[M] = O(\|f\|_V^2 M^{-1})$ (Hint: use Theorem 9.6). Verify that $\epsilon_l[M] \sim \|f\|_V^2 M^{-1}$ if $f = C\,\mathbf{1}_{[0,1/2]}$.