

# A METHOD FOR APPROXIMATE RECONSTRUCTION FROM FILTER BANKS

MATT CUSHMAN

**ABSTRACT.** We consider the problem of reconstructing an analog signal from discrete samples of from a known analysis filter bank. We reformulate the problem in terms of calculating the best approximation to the signal within a given closed subspace  $W$  of  $L^2$ , where  $W$  is expressed as a sum of finite dimensional subspaces  $W_i$ . The case where  $f \in W$  corresponds to perfect reconstruction. The procedure easily handles perturbations of the sampling points (nonuniform sampling) and variation in the number of output filters at different points.

By reinterpreting the sampling procedure as a projection to these smaller subspaces, we give an effective iterative algorithm for this reconstruction problem. The algorithm converges under certain assumptions on the spaces  $W_i$ . We introduce the notion of “near orthogonality” of the family of subspaces  $W_i$ ; this condition guarantees the convergence of this iterative algorithm with error decreasing exponentially in time. Several examples of the use of the algorithm applied to Hermite derivative filters are given.

The original motivation of this work was in image compression, where the task was to reconstruct an image given samples of certain filters at points on the 2-dimensional hexagonal lattice. The algorithm is dual in some sense to von Neumann’s method of alternating projections, or MAP, but converges faster and requires less memory.

## 1. INTRODUCTION

One of the central themes in signal processing is the study of when and how analog signals can be reconstructed from their discrete samples. The seminal result in this area, proved originally by Whittaker, rediscovered independently by Kotel’nikov and popularized and used to great effect by Shannon, states that if a function  $f(t)$  has Fourier transform supported in  $[-\pi, \pi]$ , then its values (or “samples”) at the points which are integer multiples of  $\pi$  determine it completely. Moreover, it can be explicitly constructed from these samples by the formula:

$$f(t) = \sum_{k=-\infty}^{\infty} f(k\pi) \frac{\sin(\pi t)}{\pi t}$$

Whittaker called this the cardinal series expansion, and the important function  $\sin(t)/t$  which appears (with a dilation) is called the cardinal sine, or sinc. This basic theorem has been analyzed and generalized very heavily over the past 50 years, see [8] for an excellent recent summary.

A few of the generalizations have focused on what occurs when the sampling points are non-uniform. This means that, instead of knowing the function’s values at integer multiples of  $\pi$ , we instead have its values on some other set. These results usually state that, if the set  $S$  on which we know the signal is close in some

appropriate sense to  $\pi\mathbb{Z}$ , then it is a “set of stable sampling”. Roughly, these means that  $f$  is determined by its values on this set, and there are functions which we can use to form an expansion like 1. We call these functions “synthesis functions”.

Another set of generalizations studies what happens when, instead of having the samples of the function itself, we are given samples of  $f$  after having been passed through some operators. Usually, these are shift-invariant linear operators, or filters. Shannon himself observed that knowing the value of the signal and its derivative at every other multiple of  $\pi$  sufficed to reconstruct a  $\pi$ -bandlimited signal. A central result in this direction was proved by Papoulis in [5], who showed that for most filters  $H_1, H_2, \dots, H_n$  (the “analysis filters”), it suffices to sample  $H_i(f)$  at integer multiples of  $n\pi$ . Furthermore, the original signal is the sum of translates of some collection of functions  $G_j(t)$  for  $j = 1 \dots n$ . These  $G_j(t)$  are usually called synthesis functions (or filters). This area has found many applications to communications, image processing, voice compression, and other fields. See [6] for an excellent account of this area.

All of these results fall under the rubric of sampling theorems. This paper is concerned with half of this story: the part concerning the synthesis filters. We approach the problem from the following perspective: suppose we are given a function  $f$  with some condition on its Fourier transform. It is passed through some collection of filters, and the outputs are sampled at certain points (which need not be uniform). How can one synthesize the original signal as best as possible? We give a fast iterative algorithm for constructing the best approximation to  $f$  from a space spanned by different translates of some synthesis functions, given samples of  $f$  from certain filters. In particular, if an “abstract” sampling theorem implies that the original signal  $f$  lies in the space spanned by the synthesis filters, this algorithm converges to  $f$ . This algorithm is proved to be stable under certain assumptions on the synthesis filters. This algorithm deals with non-uniform sampling (within limits), multiple synthesis filters and even omitted samples. Finally, the algorithm is generalized to higher dimensional signals.

## 2. ABSTRACT VECTOR SPACE PROJECTIONS

Let  $V$  be a Hilbert space, with inner product given by  $\langle -, - \rangle$ . Suppose  $W_i, i \in I$  is a family of closed orthogonal subspaces of  $V$ . Denote the orthogonal projection to any subspace  $W$  by  $\pi_W$ , and set  $\pi_i = \pi_{W_i}$ . As is well-known, the projection to the closure of a sum of orthogonal subspaces is the sum of the projections:

$$\pi_{\overline{\bigoplus W_i}} = \sum_{i \in I} \pi_{W_i}$$

This fact can be seen as a generalization of Parseval’s equality. Here, we consider the case of non-orthogonal subspaces  $W_i$ . In order to do this, we need to consider iterated composites of projections  $\pi_{W_{i_1}} \circ \pi_{W_{i_2}} \circ \dots \circ \pi_{W_{i_m}}$ . In the orthogonal case, the composite of any two distinct projections is 0, so they give no additional information. In the non-orthogonal case, knowledge of these composites can be used to calculate the projection to the sum of all subspaces. One algorithm for this was discovered by von Neumann ([7]), and applied to image reconstruction in ([9]).

This algorithm is actually dual to von Neumann’s well-known method of alternating projections, or MAP. The purpose of MAP is to determine the projection of a point  $x$  in a Hilbert space  $H$  to the intersection of a family of subspaces  $W_i$  for  $i \in I$ . It is assumed that it is easy to calculate the projection of an arbitrary

element of  $H$  to any one of the subspaces. Von Neumann described the algorithm for the case of two subspaces, and it was generalized by Halperin to finitely many subspaces.

The basic idea is geometrically clear if we consider the case of two subspaces, for example two planes  $W_1$  and  $W_2$  in 3 space. First, project  $x$  to  $W_1$ , then project that to  $W_2$ , project back to  $W_1$ , and iterate. It's clear that the speed of convergence is influenced by the angle between  $W_1$  and  $W_2$ . In the case of multiple subspaces  $W_1$  through  $W_n$ , the first step is to project  $x$  first to  $W_1$ , then this projection onto  $W_2$  and so on up to  $W_n$ . Then the whole process is repeated again cyclicly. A few observations are warranted at this point: first, this can only be performed for finitely many subspaces, since we need to loop back to the first at some stage. Secondly, the algorithm depends in an arbitrary way on the order of the spaces  $W_1$  up to  $W_n$ . The series obtained will differ for different orderings, and even the speed of convergence depends on this order.

The algorithm we give is for the dual problem: given a set of subspaces  $W_i$  for  $i \in I$  and  $x \in V$ , calculate the projection of  $x$  to the sum  $\sum_{i \in I} W_i$ . One method to approach this problem is do a straightforward dualization of the MAP algorithm. Using the standard dictionary between a Hilbert space and its dual, projecting to a sum of subspaces is the same as projecting to the orthogonal complement of the intersection of the orthogonal complements. This does yield an algorithm which will converge, but suffers from some disadvantages in terms of speed and memory requirements compared to the algorithm proposed here.

We will now describe our algorithm. We begin with the following notation. Let  $\mathcal{I}$  be set of finite sequences of elements from  $I$  of the form  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$  with  $i_k \neq i_{k+1}$  for each  $k = 1 \dots m - 1$ . The length of  $\alpha$  is  $|\alpha| = m$ . We denote the iterated projection corresponding to  $\alpha$  by  $\pi_\alpha$ . This is precisely given by the following formula:

$$\pi_\alpha = \pi_{\alpha_1} \circ \pi_{\alpha_2} \circ \dots \circ \pi_{\alpha_m}$$

**Theorem 1.** *Let  $W_i, i \in I$  be a family of subspaces of a Hilbert space  $V$ . Let  $v \in V$ . Then*

$$\pi_{\bigoplus W_i} v = \sum_{\alpha \in \mathcal{I}} (-1)^{|\alpha|+1} \pi_\alpha v$$

*whenever the sum on the right hand side absolutely converges.*

*Proof.* Let  $w$  be value of the sum on the right hand side. It suffices to show that  $w \in \overline{\bigoplus W_i}$  and  $v - w \in \overline{\bigoplus W_i}^\perp$ .

The first assertion is clear, since  $w$  is the limit of a sequence lying in  $\bigoplus W_i$ . For the second, observe that  $\overline{\bigoplus W_i}^\perp$  is the intersection of the kernels of  $\pi_i$  as  $i$  varies over  $I$ . Thus, after passing the continuous operator  $\pi_i$  through the absolutely convergent sum, it suffices to show

$$\pi_i v = \sum_{\alpha \in \mathcal{I}} (-1)^{|\alpha|+1} \pi_i \pi_\alpha v$$

Note that  $\pi_i v$  is one of the summands on the right hand side, so we need to show that the other summands sum to 0. This sum telescopes to 0, however, once we observe the fact that  $\pi_i \pi_\alpha = \pi_\alpha$  if  $\alpha_1 = i$ .  $\square$

The following theorem furnishes an criterion for this convergence which can be verified in special cases:

**Theorem 2.** *Suppose that  $v \in V$  is such that the sum  $\sum_{i \in I} \pi_i v$  converges absolutely. Suppose further that there is a  $\lambda < 1$  such that*

$$\sum_{i \neq j} \|\pi_i \pi_j\| \leq \lambda$$

*for every  $j \in I$  (note that  $\lambda$  is independent of  $j$ ). Then the sum in 1 converges absolutely. Here,  $\| - \|$  denotes the standard operator norm.*

*Proof.* Set

$$C_n = \sum_{|\alpha|=n} |\pi_\alpha v|$$

which (as far as we know) may or may not be finite.

We must show

$$\sum_{\alpha \in \mathcal{I}} |\pi_\alpha v| = \sum_n C_n$$

is finite.

The first assumption in the theorem states that  $C_1$  is finite. Using the second statement we find

$$\begin{aligned} C_{m+1} &= \sum_{|\alpha|=m+1} |\pi_\alpha v| \\ &= \sum_{|\alpha|=m} \sum_{i \neq \alpha_1} |\pi_i \pi_\alpha v| \\ &\leq \sum_{|\alpha|=m} \sum_{i \neq \alpha_1} |\pi_i \pi_{\alpha_1}| |\pi_\alpha v| \\ &\leq \sum_{|\alpha|=m} \lambda |\pi_\alpha v| \\ &= \lambda C_m \end{aligned}$$

Hence, the sequence  $C_m$  is bounded above by a geometric sequence with ratio  $\lambda < 1$ , and thus is summable.  $\square$

Roughly speaking, this last proposition says that if the subspaces  $W_i$  are not too far from being orthogonal, then the projection to the sum of the subspaces can be readily calculated from the projection to each summand. We encapsulate this in the following definition

**Definition 1.** *A collection of subspaces  $W_i$ ,  $i \in I$  is called nearly orthogonal if there is a  $\lambda < 1$  such that*

$$\sum_{i \neq j} \|\pi_i \pi_j\| \leq \lambda$$

*for every  $j \in I$ , and the sum  $\sum_i \pi_i$  converges pointwise.*

Of course, every collection of orthogonal subspaces is automatically nearly orthogonal. A collection of nearly orthogonal subspaces automatically satisfies the conditions of the above theorem for every  $v \in V$ .

Examples of nearly orthogonal subspaces can be found in signal processing applications, and are described below. The basic cases occur when one has a fixed set of filters which are used to analyze a signal. In other words, the signal is passed through these filters and the output is sampled. If the sample rate is sufficiently small and the impulse responses of the cascade of two of these filters is sufficiently localized in time, then the problem of synthesizing the original signal from these samples can be solved by this algorithm

### 3. FREQUENCY DOMAIN PROJECTIONS

Although this algorithm could be applied to many different areas, we only explore applications to signal processing. The main problem we consider is the reconstruction of an analog signal from its sampled output from filter banks. We will also allow the sampling points be non-uniform, which is in contrast to standard multi-rate filter bank theory.

Fix a positive weight function  $\rho$  on the interval  $(a, b)$ , where either  $a$  or  $b$  could be  $\pm\infty$ . We consider the space  $L^2_{\rho^{-1}}$  of functions  $f$  satisfying

$$\int_a^b |f|^2 \rho^{-1} < \infty$$

which is a Hilbert space using this as the corresponding quadratic form. We write the corresponding inner product as  $\langle f, g \rangle_{\rho^{-1}} = \int_a^b f \bar{g} \rho^{-1}$  to distinguish it from the usual  $L^2$  inner product. We will also consider the dual inner product  $\langle f, g \rangle_{\rho}$ , which is defined in the analogous manner.

Let  $p_k$  for  $k \in S$  be a complete orthonormal family of functions with respect to  $\langle -, - \rangle_{\rho}$ . Dually,  $p_k \rho$  is an orthonormal family with respect to  $\langle -, - \rangle_{\rho^{-1}}$ .

The following proposition is a consequence of standard facts (Parseval's equality) about approximation in Hilbert spaces

**Proposition 1.** *Let  $f \in L^2_{\rho^{-1}}$ ,  $g_t(\omega) = e^{i\omega t}$  (with  $t \in \mathbb{R}$  fixed). The orthogonal projection  $\pi_{t,T}(f)$  of  $f$  to the subspace of  $L^2_{\rho^{-1}}$  generated by  $g_t p_k$  for  $k \in T \subset S$  is given by*

$$\begin{aligned} \pi_{t,T}(f) &= \sum_{k \in T} \langle \bar{g}_t f, p_k \rho \rangle_{\rho^{-1}} g_t p_k \rho \\ &= \sum_{k \in T} \langle f, g_t p_k \rangle_{\rho} g_t p_k \rho \end{aligned}$$

Of course, the "orthogonality" of the projection is with respect to the inner product  $\langle -, - \rangle_{\rho^{-1}}$ .

### 4. FILTERS AND PROJECTIONS

If we regard the results of the previous section as statements about the frequency domain, translating this into the time domain yields interesting results. Let  $f \in L^2(\mathbb{R})$  be function such that  $\hat{f} \in L^2_{\rho^{-1}}$ . Let  $H_i$  for  $i \in T \subset S$  denote the filters with frequency response  $p_i$ , and let  $B_i = \widehat{p_i \rho}$ . Let  $(-, -)$  be the inner product defined by  $(f, g) = \int \hat{f} \hat{g} \rho^{-1}$ . A restatement of Proposition 1 is

**Proposition 2.** Fix a time  $t_0$ . The orthogonal projection with respect to the inner product  $(-, -)$  of  $f$  to the subspace  $W_{t_0}$  generated by  $B_i(t - t_0)$  for  $i \in T$  is

$$\sum_{i \in T} H_i(f)(t_0) B_i(t - t_0)$$

Thus, the samples of the output of certain filter banks can be seen as the coefficients of the projection of the input signal to a subspace. As the sampling moment  $t_0$  changes, the subspace shifts in time as well.

This is exactly the type of situation to which Theorem 1 applies. Let  $K_{i,j}$  denote the output of signal  $B_j$  through filter  $H_i$ . Let  $W_{t_j}$  denote the subspace generated by  $B_i(t - t_j)$  for  $j = 1, 2$ . Then,

$$\pi_1 B(t - t_0)(t) = \sum_j K_{i,j}(t_1 - t_0) B_j(t - t_1)$$

The following theorem is a consequence of this fact and Theorem 1.

**Theorem 3.** Assume  $I$  is a discrete subset of  $\mathbb{R}$  (thought of as being a set of sampling times), and let  $T_{t_0}$  be a finite subset of  $S$  for each  $t_0 \in S$  (we think  $T_{t_0}$  as indexing the set of filters whose output we sample at time  $t_0$ ). For each  $t_0 \in I$ , let  $W_{t_0}$  be the vector space generated by the functions  $B_i(t - t_0)$  as  $i$  varies over  $J_{T_0}$ . Assume that there is a  $\lambda < 1$  such that for every  $t_0 \in I$  the sum

$$\sum_{t_1 \neq t_0} \sum_{i \in J_{t_0}} \sum_{j \in J_{t_1}} K_{i,j}(t_1 - t_2)^2$$

is bounded above by  $\lambda$ . Assume further that  $\sum_{t_0 \in S} \pi_{t_0} v$  is absolutely convergent (which is automatically satisfied, for example, when the sum  $\sum_{t_0 \in S} \pi_{t_0}$  is pointwise convergent). For each pair  $\beta = (t_0, i) \in I \times T$ , set  $v_\beta = (v, B_i(t - t_0)) = H_i(v)(t_0)$ . Then the best approximation to  $v$  (with respect to the inner product  $(-, -)$ ) in the closed subspace spanned by all of the  $W_{t_0}$  is given by

$$\sum_{\alpha \in I} (-1)^{|\alpha|+1} \pi_\alpha v = \sum_{\beta} (-1)^{|\beta|+1} v_\beta \prod_{i=1}^{|\beta|-1} K_{\beta_{2,i+1}, \beta_{2,i}}(\beta_{1,i+1} - \beta_{1,i})$$

where the sum is over all finite sequences  $\beta$  of pairs of elements  $(t, i)$  with  $t \in I$  and  $i \in T_t$ , with  $\beta_{1,i} \neq \beta_{1,i+1}$  for every  $i$ .

The statement of this theorem looks very complicated, but it is really just unwinding the definition of the spaces  $W_i$  and the projections  $\pi_i$ , together with Theorem 1.

It is worth noting that the theorem is very general. The set of sampling times  $I$  admits a great deal of flexibility; it need not be uniform, or have any sort of near periodicity. We need not have the output of the same filters at each sampling time. These specific requirements are replaced with a general requirement that the subspaces in question be nearly orthogonal.

Although we state this in the context of one dimensional signals (in both domain and range), it immediately generalizes to signals defined on subsets of  $\mathbb{R}^m$ , with range in  $\mathbb{R}^n$ . In fact, the inspiration for these results came from an image reconstruction problem. In this problem, one has the output of certain circularly symmetric filters sampled at the points of the lattice  $\mathbb{Z} + \frac{\sqrt{3}}{2}\mathbb{Z}$ , which is the familiar hexagonal lattice. Since this lattice is not the product of two one dimensional

lattices, one cannot lift a one-dimensional reconstruction scheme in any straightforward manner. The algorithm based upon the above theorem gives a good solution, though.

## 5. AN EFFICIENT ALGORITHM

If  $W_i$ ,  $i \in I$  is a nearly orthogonal collection of subspaces, we can use the expansion in Theorem 1 to calculate its projection. By truncating the sum at some finite stage (meaning summing over some finite subset  $\mathcal{J} \subset \mathcal{I}$ ) one can approximate the entire sum as closely as we wish. In fact, if we examine the proof of Theorem 2, we can see that the partial sums over sequences  $\alpha$  with  $|\alpha| \leq n$  converge with error decreasing exponentially in  $n$ . Unfortunately, even when  $I$  is finite the size of these subsets increases exponentially in  $n$ , and so this does not yield a particularly efficient algorithm. We will give a simpler algorithm which converges with exponentially small error in linear time and fixed memory requirements.

We assume throughout this section that the  $W_i$  are nearly orthogonal. Let  $\mathcal{I}_n = \{\alpha \in \mathcal{I} \mid |\alpha| = n\}$ . Then by Theorem 1

$$v = \sum_n (-1)^{n+1} \sum_{|\alpha|=n} \pi_\alpha v$$

Notice the the inner summand for each  $n$  is an element of the space  $\sum_{i \in I} W_i$ . We call such an element a *page*. Thus,  $v$  is expressed as an alternating sum of pages. Denote by  $e^n$  the  $n$ -th page in this sequence. Thus,  $e^1$  is just the sum of the projections of  $v$  to each subspace, and  $e^n$  is the sum of compositions of projections of length  $n$  applied to  $v$ .

The main observation is that  $e^{n+1}$  can be calculated solely from  $e^n$  and knowledge of the projections  $\pi_{i,j} = W_j \rightarrow W_i$  for  $i \neq j$ . The equation if given by

$$e_i^{n+1} = \sum_{j \neq i} \pi_{i,j} e_j^n$$

It is easy to see that this is  $e^{n+1}$ . Thus, to calculate the sum  $\sum_{n \leq N} e^n$  we can first calculate  $e^1$ , which we regard as the first partial sum of the series. We then use  $e^1$  to get  $e^2$ , and subtract  $e^2$  from  $e^1$  yielding the second partial sum. Then use  $e^2$  to calculate  $e^3$ , and add  $e^3$  to the previous partial sum. Iterating this process, we can calculate the  $n$ -th partial sum in time proportional to  $n$ . Furthermore, we never need to store more than 2 pages simultaneously: the current page  $e^n$ , and the current alternating partial sum of pages up to page  $n$ .

In order to calculate the linear maps  $\pi_{i,j}$ , we need to perform a matrix multiplication. If we let  $d_i$  be the dimension of  $W_i$ , then this takes  $O(d_i d_j)$  many operations. Thus, calculating  $e^{n+1}$  from  $e^n$  takes on the order of  $\sum_{i \neq j} d_i d_j$  operations, which is approximately  $(\sum_i d_i)^2$ . This step of the iteration dominates the accumulation step, which is just a simple summation. Let  $d = \sum_i d_i$ , which is the dimension of the space  $\bigoplus_i W_i$ . The algorithm takes  $O(nd^2)$  time to approximate the projection to within an error bound which decreases exponentially in  $n$ .

The error can be explicitly bounded from knowledge of  $\lambda$  and knowledge of the  $L^2$  norm of the final page. By examining the proof of Theorem 2 more closely, one can see that the error term of truncating at the  $n$ -th stage is bounded above by  $\frac{\lambda}{1-\lambda} \|e^n\|$ , where  $\|e^n\|$  is the square root of the sum of the coefficients in the page

$e^n$ . This gives a simple and practical method for determining when to cease further iterations when an error tolerance is specified ahead of time.

In many practical situations, we do not need all of the projections  $\pi_{i,j}$  in the sum. Many of these will be so small as to be negligible. The error introduced by these omissions can be explicitly bounded if one knows the norms of  $\pi_{i,j}$ , but we omit that analysis here.

## 6. COMPARISON WITH MAP

This algorithm computes the project of the original signal  $f$  to a sum of subspaces  $W_i$ , which as noted above could be done by applying the method of alternating projections. There are several advantages to SMAP over MAP, some of which are qualitative (e.g. reduced implementation complexity) and some quantitative (measured, for example, by the number of iterations required to achieve a certain level of accuracy). In this section we show that in many instances the algorithm presented here converges more faster than MAP, and indicate why the implementation of SMAP can be simpler than MAP.

First, we address the convergence rate issue. Of course, if the subspaces  $W_i$  are not nearly orthogonal, then SMAP need not converge at all. Since MAP always converges, this shows that SMAP doesn't always beat MAP in convergence rate. In general, it may be very difficult to give a good theoretical basis to compare the convergence rates in general, so we tested the behavior of SMAP and MAP by a Monte Carlo simulation.

We begin by choosing three 1-dimensional subspaces of  $\mathbb{R}^3$  by adding a normal random variable with standard deviation  $\sigma$  to each of the entries of  $I_3$ , the  $3 \times 3$  identity matrix. Each row of the resulting matrix spans one of the corresponding subspaces. Generally speaking, these subspaces will be very close to orthogonal for small values of  $\sigma$ , and more skew for larger values of  $\sigma$ . These subspaces are the  $W_i$  used for both the MAP and SMAP algorithms. The initial vector we want to approximate is chosen from a 3-dimensional standard normal distribution.

For various values of  $\sigma$  we generated this data and ran the SMAP and MAP algorithms for 8 iterations. Each value of  $\sigma$  was run 1000 times. The proportion of trials for which SMAP yielded a closer approximation after those 8 trials is shown in Figure 6. As is clear from this graph, SMAP outperforms MAP for  $\sigma < 1/2$ , with the advantage of SMAP over MAP becoming very pronounced for smaller  $\sigma$ . This shows that, broadly speaking, SMAP performs very well when the  $W_i$  are reasonably well separated.

Another advantage of SMAP over MAP comes from the ease with which SMAP can be implemented. It is easiest to see this by example, so we focus on the case of the signal reconstruction algorithm described above. In order to implement this using MAP, one needs to iterate through the projection to each  $W_i^\perp$  in sequence. Thus, each stage in the iteration is different from the other stages. Furthermore, even though the end result is to calculate the projection of  $f$  to the sum of the  $W_i$ , at each step of the iteration one needs to store an element of  $W_i^\perp$ , which is not necessarily in the space  $\sum W_i$  (and even if it were, we wouldn't know how to write it in terms of the bases we were given for  $W_i$ ). Thus, simply storing this value itself requires storing the signal itself as a "full-fledged" element of the signal space, however we choose to do this.

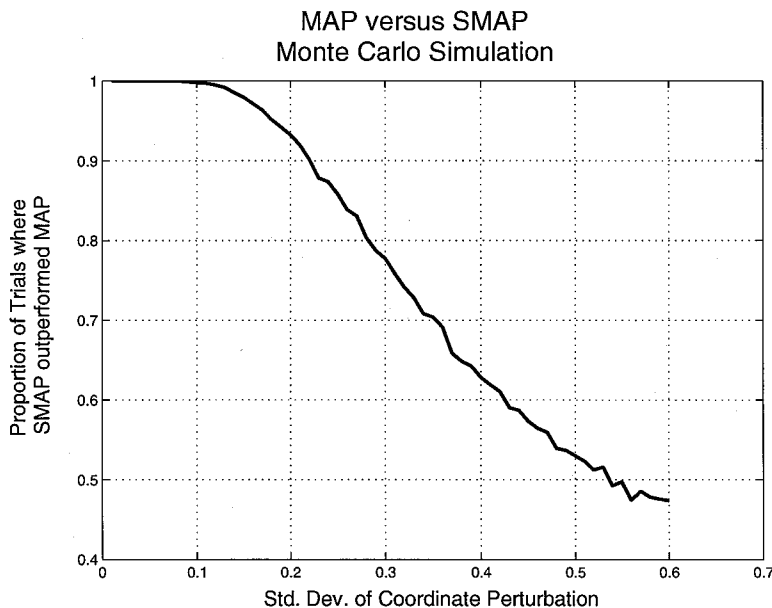


FIGURE 1. Monte Carlo Simulation of Algorithms

On the other hand, each partial result obtained along the way for the SMAP algorithm is an element of  $\sum W_i$ , and can easily be expressed as a linear combination of the given bases of the  $W_i$ . Furthermore, the calculations for moving from one stage of the iteration to the next are always the same.

## 7. EXAMPLES

An excellent application of this technique is reconstructing a signal from its Hermitian chromatic derivatives. We begin with a review of this theory.

Let  $\rho(\omega) = e^{-\omega^2}$ . The family of polynomials which are orthonormal with respect to this weight is well-known. These are the normalized Hermite polynomials  $h_k(\omega)$  given by the following equations:

$$H_k(\omega) = e^{\omega^2} \frac{d^{(k)}}{d\omega^{(k)}} [e^{-\omega^2}]$$

$$h_k(\omega) = (2^n n!)^{-1/2} H_k(\omega)$$

Here,  $H_k(\omega)$  are the standard Hermite polynomials, which are orthogonal but not orthonormal. Let  $CD_k$  denote the filter corresponding to  $h_k(\omega)$ . Formally,

$$CD_k(f)(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} h_k(\omega) \hat{f}(\omega) e^{i\omega t} d\omega$$

This is a differential operator of order  $k$ . There is an explicit differential recurrence relation which allows one to calculate  $CD_k(f)$  explicitly, and this family enjoys many other nice properties ([2]).

The corresponding time domain functions are given by the formula

$$B_k(t) = (2^n n!)^{-1/2} t^n e^{-t^2/4}$$

From the general theory of chromatic derivatives, for any signal  $f$  for which  $\int \hat{f}(\omega) e^{\omega^2} d\omega$  converges, we have the expansion

$$f(t) = \sum_{k=0}^{\infty} CD_k(f, t_0) B_k(t - t_0)$$

which is called the chromatic expansion of  $f$  at the point  $t_0$ . Furthermore, truncating this expansion at a finite stage  $M$  produces the projection of  $f$  to the finite dimensional vector space  $W_{t_0, M}$  generated by the functions  $B_k(t - t_0)$  for  $k = 0 \dots M$ . This projection is along the inner product defined by  $\rho^{-1}$  in the frequency domain. In particular, the projection of  $B_k(t - t_0)$  to  $W_{M_1, t_1}$   $t_1 \neq t_0$  can be readily calculated by evaluating the chromatic derivatives of  $B_k$ . In this manner, we find that the projection  $W_{M_0, t_0} \rightarrow W_{M_1, t_1}$  is given by the  $(M_1 + 1) \times (M_0 + 1)$  matrix

$$\begin{bmatrix} CD_0(B_0)(t_1 - t_0) & CD_0(B_1)(t_1 - t_0) & \cdots & CD_0(B_{M_0})(t_1 - t_0) \\ CD_1(B_0)(t_1 - t_0) & CD_1(B_1)(t_1 - t_0) & \cdots & CD_1(B_{M_0})(t_1 - t_0) \\ \vdots & \vdots & \ddots & \vdots \\ CD_{M_1}(B_0)(t_1 - t_0) & CD_{M_1}(B_1)(t_1 - t_0) & \cdots & CD_{M_1}(B_{M_0})(t_1 - t_0) \end{bmatrix}$$

The chromatic expansion gives the expression for  $f$  in terms of all of the chromatic derivatives of  $f$  at a single point. Frequently, however, this is not sufficient. One would like to be able to reconstruct  $f$  from finitely many chromatic derivatives at a family of different sampling points. Using this method, we can give an efficient algorithm to solve this problem. One simply takes the known CDs of  $f$ , which may be viewed as the coefficients of the projections of  $f$  to various linear subspaces of the space of  $L^2$  functions with finite energy in the frequency domain weighted by  $\rho^{-1}$ , and plugs it into the algorithm in the previous section.

Figure 1 is a concrete example illustrating the convergence. Our test signal is  $f(t) = e^{-(t-13)^2/4}$ , which is a shift of the Gaussian function itself. We start with the the first 5 chromatic derivatives of  $f$  (i.e. CDs 0 through 4) of  $f$  evaluated at the points 5,10,15,20 and 25. The plots show both  $f$  and the approximation of  $f$  given by the algorithm after 1, 5 and 36 iterations. Notice that after 36 iterations, the approximation is very accurate.

Figure 4 is another example. The function is a sum of 26 Gaussian monopulses centered at integers 0 through 25. The sampling times are not uniform, of the form  $5k \pm \delta$ , where  $\delta$  is a random number in  $[-1/2, 1/2]$ . This example shows the flexibility of the algorithm to deal with nonuniform samples.

We give one more example in Figure 5. Here, instead of using the Hermite family of filters, we use the family of hyperbolic secant derivatives discussed in [1]. This shows the approximations given by the algorithm at various stages converging to  $\text{sech}(\pi(t - 1))$  in the subspaces centered at  $1/2$  and  $3/2$  generated by  $\text{sech}(\pi t) \tanh(\pi t)^k$  for  $k = 0 \dots 15$ .

## 8. CALCULATING SYNTHESIS FILTERS

A general problem in filter bank theory is to determine the synthesis filters from a given set of analysis filters. For example, the classic Shannon sampling theorem

states that we can reconstruct a  $\pi$ -bandlimited signal perfectly from its samples at integer points by using the sinc function

$$f(t) = \sum_{k=-\infty}^{\infty} f(k) \frac{\sin(\pi(t-k))}{\pi(t-k)}$$

In this theorem, one can think of sampling as being the “analysis” filter, and the sinc function as being the “synthesis” filter.

Papoulis [5] gave a wide-reaching generalization of this theorem, which states that for most families of analysis filters  $H_1, \dots, H_m$ , there is a family of synthesis filters  $G_1, \dots, G_m$  such that one can reconstruct  $f$  by taking linear combinations of the  $G_k$  using the samples of  $H_i(f)$  as coefficients. The rate at which one must sample an given filter is decreased by a factor of  $m$ , so that the overall sampling rate is the same. These types of filters have been extensively studied, and are now known as perfect reconstruction systems. See [6].

One can use the techniques developed above to approximate the synthesis filters corresponding to certain analysis filters. This is done by applying the algorithm, using as input “faked” filter output information. Instead of starting with a signal  $f$ , we suppose that we had  $f$  with the property that  $H_j(f)(0) = 1$  for some specific  $j \in \{1, \dots, m\}$ , but  $H_i(f)(t) = 0$  whenever  $i \neq j$ , or  $t \neq 0$  (of course, we only require this for  $t$  which correspond to sampling times). We input this data as the first page of the algorithm, and iterate. Assuming that the subspaces are nearly orthogonal, it will converge to the corresponding synthesis filter.

We give two examples of this procedure, in Figures 2 and 3. Figure 3 shows the convergence of the algorithm to the synthesis filter  $g$  corresponding to sampling the first 5 Hermite chromatic derivatives at the points  $\{5, 10, \dots, 45\}$ , with  $CD_4(g)(25) = 1$ , and  $CD_i(g)(t) = 0$  for all other combinations of  $t$  and  $i$ . Notice that  $f$  becomes very flat around the sampling points other than 25; this is because it has a zero of order 5 at each of these points. It is also interesting to note how the information begins localized around the point 25, and slowly leaks outward to a wider range of times. The impulse response is quite long after 36 iterations.

Figure 3 is an example of the same algorithm, but using non-uniform sampling as well. The various CDs of the output signal (after 64 iterations) at the sampling times are summarized in the following table, Table 1. This shows that this signal is a very accurate approximation to the ideal dual filter table, corresponding to the analysis filter which samples  $CD_4$  at time 25. The times listed in the left-most column are the non-uniform sampling times. The number of CDs listed for a given time corresponds to the number of analysis filters assumed at the corresponding time. The spaces at the different sampling moments have a variety of different dimensions, which demonstrates the flexibility of this algorithm. We simply omit the CDs which were not used at a give time, as they are not relevant. Notice that  $CD_4(f)(25)$  is within  $3 \cdot 10^{-5}$  of 1, and the largest error for the other values is  $3 \cdot 10^{-4}$ .

#### REFERENCES

- [1] J. Byrnes, T. Herron, Families of Chromatic Derivatives, Kromos Technology Technical Report.
- [2] A. Ignjatovic, Numerical Differentiation and Signal Processing, Kromos Technology Technical Report.

Time	CD(0)	CD(1)	CD(2)	CD(3)	CD(4)	CD(5)
5	-9e-7	-3e-6	-7e-6	-1e-5	-2e-5	-3e-5
11	4e-8	8e-8	9e-8	7e-8		
15	1e-4	-2e-4	4e-4	-3e-4	3e-4	
20	3e-9	-1e-8	1e-8	-3e-8		
25	-2e-6	6e-6	-1e-5	1e-5	.999978	
32	4e-9	1e-8	-1e-7	5e-6	6e-5	-5e-4
37	2e-5	2e-5	2e-5			
41	2e-7	3e-7	7e-7	6e-7	1e-5	
45	3e-5	-5e-5	6e-5	-4e-5		

TABLE 1. Chromatic Derivatives at Nonuniform sampling moments

- [3] M. Narasimha, A. Ignjatovic, P.P. Vaidyanathan "Chromatic Derivative Filter Banks Part I: Infinite Channels," submitted.
- [4] M. Narasimha, M. Cushman, P.P. Vaidyanathan "Chromatic Derivative Filter Banks Part I: Infinite Channels," preprint.
- [5] A. Papoulis "Generalized Sampling Expansion," IEEE Trans. on Circuits and Sys., pp. 652-654, Nov. 1977b.
- [6] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- [7] J. von Neumann, *The Geometry of Orthogonal Spaces*. vol II, Princeton, NJ: Princeton University Press, 1950.
- [8] M. Unser, "Sampling-50 Years after Shannon", Proceedings of the IEEE, vol. 88, no. 4, pp. 569-587, April 2000.
- [9] D.C. Youla "Generalized Image Restoration by the Method of Alternating Projections," IEEE Trans. on Circuits and Sys., pp. 694-702, Sept. 1978.

KROMOS TECHNOLOGY, 4984 EL CAMINO REAL, LOS ALTOS, CA 94022  
*E-mail address:* matt@kromos.com