Representing Coastlines with Linear Transforms

A course project for

CSC 2508S Information : Quantification,
Specification, and Manipulation

TEH Yee Whye
009753429

January 18, 2000

Contents

1 Introduction

Figure 1a shows an islands coastline. Figure 1b to 1d show views of the coastline as seen from different locations and altitudes. When exploring the coastline, sometimes one may need to have a high level view of the whole coastline where most of the details are missing, as in figure 1a. Sometimes, one might "zoom" in and explore certain parts of the coastline in detail, as in figures 1b to 1d.

In this project we explore ways of representing and storing coastline information for efficient retrieval. By retrieval we mean getting sufficient information to reconstruct the coastline on a computer screen with sufficient detail and accuracy for simple viewing purposes. The views of the coastline can be from a range of altitudes and locations as in figure 1, so that the reconstruction can range from high resolution (low altitude) to low resolution (high altitude), and can be for a restricted section of the coastline instead of the whole coastline. This is akin to the many web-based street mapping utilities but applies to coastlines instead of streets.

For the purposes of this project we assume that the coastline is a discretized curve on the plane represented as a sequence of coordinates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. The number of discretizations n can potentially be very large, so that naive algorithms will not apply. We

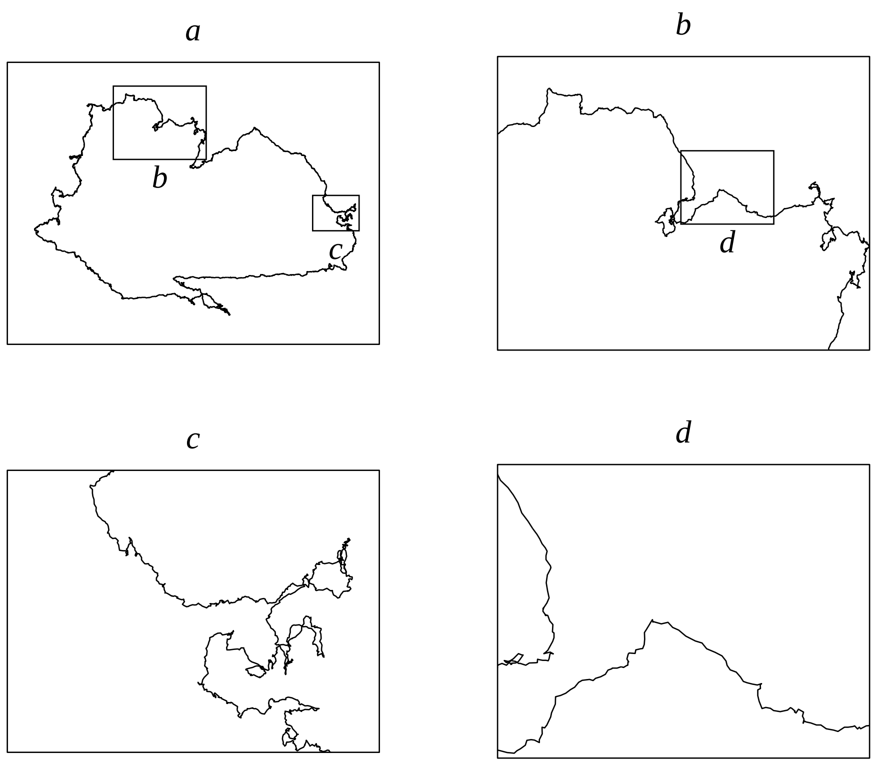


Figure 1: a : The coastline of an island. b, c and d : higher resolution views of the coastline. Note in d that there are more than one contiguous sections of the coastline visible.

assume that the correlations between the x-coordinates and the y-coordinates are sufficiently small, so that encoding them separately as two sequences of real numbers will not incur much loss in performance.

Sometimes from a single view many non-contiguous sections of the coastline are visible, as in figure 1d. It is nontrivial to efficiently determine the visible sections of the coastline. In this project we only deal with the case in which the required reconstructions are contiguous sections of the coastline, i.e. we want $(x_i, y_i), \ldots, (x_j, y_j)$ where $1 \leq i \leq j \leq n$. We define a low resolution view of a section of the coastline as an approximation to the true coastline, where the error of the approximation is bounded above by a tolerance value θ. θ determines whether the view is a high resolution (low θ) or low resolution (high θ) view. We shall defer defining the error measure we used to section 3.1.

The sequences of x and y coordinates are discretely sampled signals. Signal processing is a well established field for the encoding of signals, and we shall be using a broad signal processing technique called linear transforms for the representation of the coastlines. This includes Fourier and wavelet analyses. Fourier analysis assumes periodic signals. In coastline terms, this means that we should assume the coastlines are coastlines of islands that form closed loops. There are transforms related to the Fourier transform that does not assume periodicity, for example the discrete cosine transform. To keep things simple, we shall assume periodic signal, i.e. the coastlines are of islands which form closed loops. We also assume that the signals are dyadic – i.e. $n = 2^l$ for some l where n is the number of coordinates. This is because the discrete wavelet transforms we will be using and the fast Fourier transform (FFT) and fast wavelet transform (FWT) algorithms assume dyadic signals.

1.1 How to generate coastlines

We could not get examples of coastlines based on real islands and land masses. However there are many methods of generating fractal curves that look similar to coastlines. Here we introduce one method and use it to generate toy "coastlines" for analysis in the rest of the project.

First decide on a probability distribution P over $\mathbb{R}^2$ with zero mean and an adjustable variance σ^2. Here we use a zero mean circular Gaussian with variance σ^2. The method is recursive. Take a line segment, as in figure 2a, and determine it's midpoint $m = (m_x, m_y)$ (figure 2b). Generate a sample (δ_x, δ_y) from P and perturb the midpoint by $m' = (m_x + \delta_x, m_y + \delta_y)$, as in figure 2$c$. Now recurse on the two resulting line segments, but with the variance of the perturbation decreased by a factor λ to $\lambda\sigma^2$ (figure 2d). Stopping after l recursions give us an approximately fractal curve consisting of 2^l line segments.

Normally we take λ to be approximately $\frac{1}{2}$, with larger values giving more wiggly curves (higher fractal dimensions), and smaller values smoother curves (lower fractal dimensions). We used a value of $\lambda = 0.54$ as this gives a measured fractal dimension of around 1.14, which is close to the measured fractal dimensions of the world's coastlines [1]$^{(*)}$.

The resulting fractal looks very similar to real coastlines, except that it often crosses itself. However this is not an important problem because our techniques will not be assuming that

$^{(*)}$For example, Portugal's coastline has fractal dimension 1.12, Australia's 1.13, Germany's 1.12 and Great Britain's a wiggly 1.24.

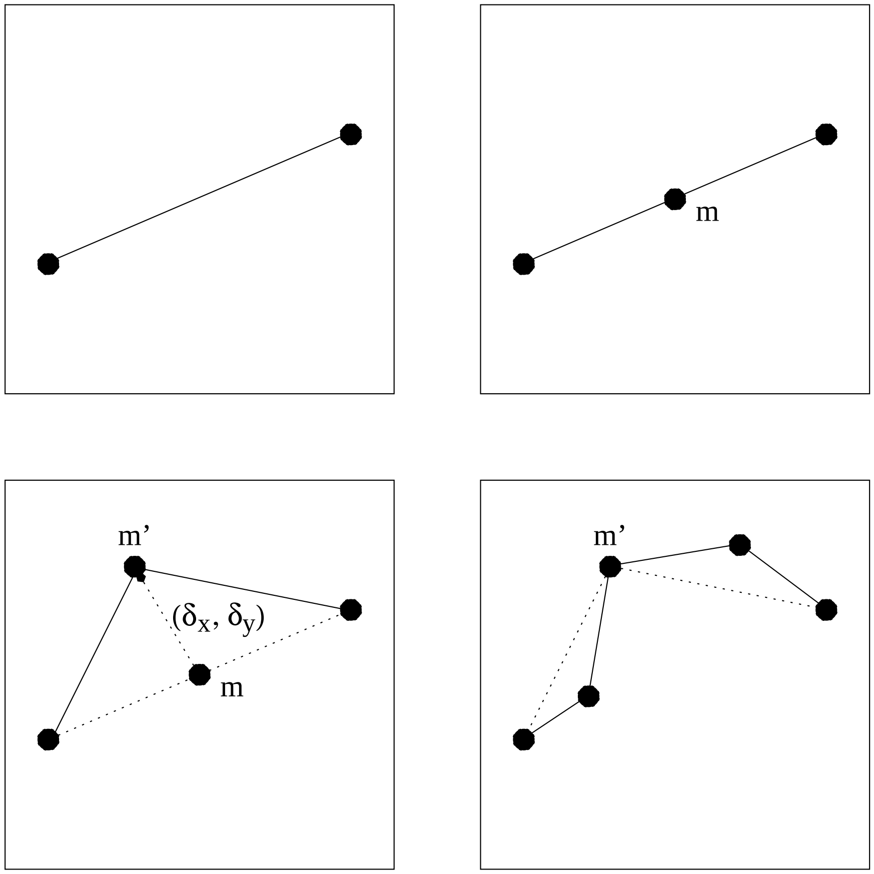


Figure 2: Generating a fractal by perturbing a line segment recursively.

coastlines do not cross themselves. However for visual appeal we chose one that did not cross itself too often at a high level (figure 3). This curve consists of 2^{16} segments and has a fractal dimension of 1.14. It will be used as the test coastline for the rest of the project.

1.2 Overview of the project

In section 2 we introduce linear transforms and how they can be useful in coding coastlines as signals. We describe the Fourier transform and the frequency domain of signals, and introduce wavelets as a trade-off between resolution in the frequency and spatial domains. In section 3 we apply linear transforms to the representation of coastlines. We compare the various transforms with respect to a number of criteria. In section 4 we discuss perceptual differences in coastlines and methods to decrease the perceived differences between approximate coastlines and the true coastlines. In section 5 we conclude.

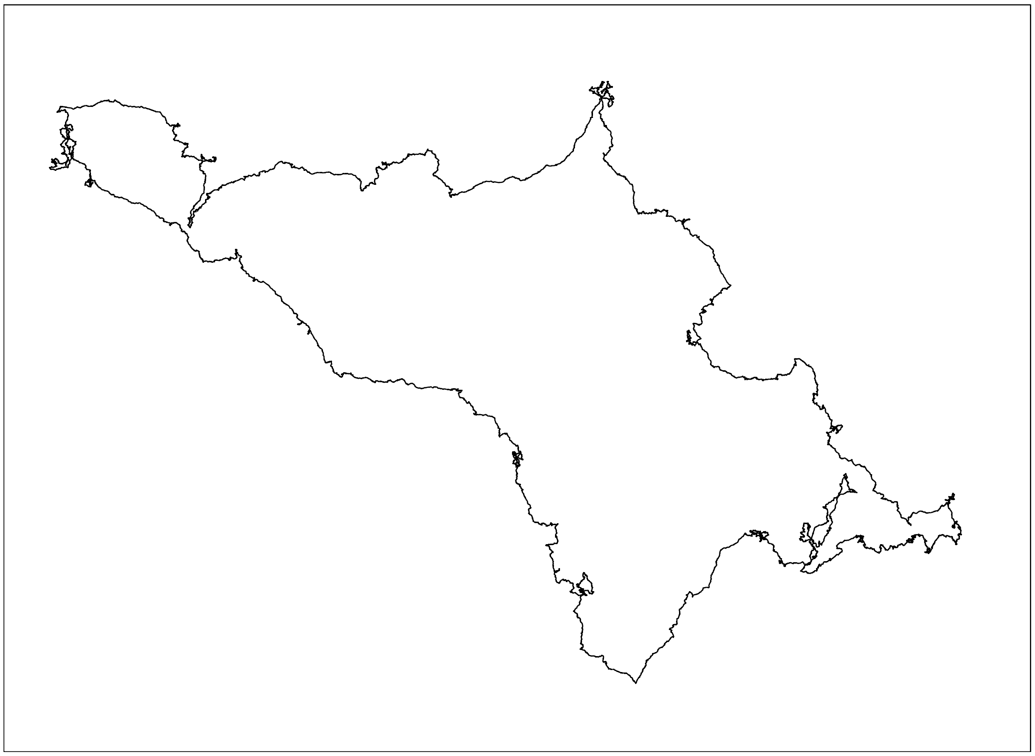


Figure 3: The example "coastline" we shall be using for the rest of the project.

2 Linear transforms

2.1 Nomenclature

For our purposes, a signal s is a sequence of real numbers, which we can write in vector form $s = [s_1, \ldots, s_n]^T$. If $B = C^{-1}$ is an invertible $n \times n$ matrix, then we can represent s as a linear combination of the columns B_i of B,

$$s = \sum_i r_i B_i = Br \qquad \text{(sp1)}$$

where $r = [r_1, \ldots, r_n]^T$. The B_i's form a basis for $\mathbb{R}^n$ and each B_i is called a basis vector. Since B is invertible, we have the inverse relation $r = Cs$ so that

$$r_i = C^i s \qquad \text{(sp2)}$$

where C^i is the i^{th} row of C for all i. The application of C to s is called a (linear) transform, and the r_i's the outputs or responses of the transform. The rows C^i's are called projection vectors, since r_i is the projection of s onto C^i. Note that applying B to r to get $s = Br$ is also a transform – it is the inverse to the C transform. We identify matrices B and their corresponding transforms to be the same thing and refer to them interchangeably.

We say B is orthonormal if $B^* = B^{-1} = C$ where M^* is the complex conjugate transpose of a matrix M. In this case, we have $C^i = B_i^*$, so that the basis vectors coincide with the projection vectors up to conjugation. Orthonormal transforms are useful because the same set of n vectors can be used for both processing the signal ($r = Cs$) and reconstructing the signal from the response ($s = Br$). The power of a signal is simply the squared length : $P(s) = \|s\|^2$. In physical terms, this is the amount energy transmitted by the signal. Another useful property of orthonormal transforms is that they conserve the power of a signal, i.e. $P(Cs) = P(s)$.

2.2 Fourier transform

An important linear transform is the Fourier transform. The Fourier transform is an orthonormal complex $n \times n$ matrix F with

$$F_{ij} = \frac{1}{\sqrt{n}} e^{-\frac{2\iota\pi}{n}(i-1)(j-1)} \tag{1}$$

where $\iota = \sqrt{-1}$[†]. Because F is complex the responses can be complex valued too. Complex numbers take twice as much space to store than real numbers. However it turns out that only half of the responses r_i are needed as $r_i = r^*_{n+2-i}$ for $i = 2, \ldots, \frac{n}{2}$ while both r_1 and $r_{\frac{n}{2}+1}$ are real-valued, so really no extra space is required to store the Fourier responses.

The projection vector F^i for $i \leq \frac{n}{2}$ is a complex vector whose real component is a discretized sine curve that oscillates for i times. The complex component is a discretized cosine curve oscillating for i times. This means that the response r_i for $i \leq \frac{n}{2}$ is the response of the i^{th} frequency component of the signal. The i^{th} frequency component is the component of the signal that oscillates for i times during the duration of the signal. The maximum frequency we can determine from the signal is $\frac{n}{2}$. This is called the Nyquist limit of the signal.

Linear transforms normally require $O(n^2)$ time to transform signals into responses (a matrix-vector multiplication). Due to the structure of the Fourier transform, $O(n \log n)$ algorithms called the fast Fourier transform (FFT) and the inverse fast Fourier transform (IFFT) exist to transform dyadic signals to responses and back. This allows for the wide spread application of Fourier transforms to large amounts of data.

This little subsection does little justice to Fourier transforms. For further information there are countless texts devoted to this subject.

2.3 The frequency and spatial resolution trade-off

With the Fourier transform, Each response r_i (along with r_{n+2-i}) encodes all the information related to a frequency component, and not any information about other frequency components. On the other hand, each r_i encodes just a little information about the value of each s_i, and hence one cannot determine the value of any s_i accurately from just one r_i. This can be seen in the fact that the basis vector F^i is global and not localized to some subset of the s_i's.

[†]Actually, the form of the Fourier transform that is generally known and used differs from (1) by a factor of $\sqrt{n}$. This is not important for the discussion here, and the form of (1) makes the transform orthonormal instead "orthonormal up to multiplying by n", which is troublesome.

This is simply a special case of the famous Heisenberg uncertainty principle, which is a formal statement along the following sense : there is a constant c so that if v is a vector confined to an interval of width Δx (v is zero or decreases quickly to zero outside of this interval), and Δf is the uncertainty in its frequency, then $\Delta x \Delta f \geq c$. The uncertainty principle essential tells us that we cannot at the same time get lots of information both in the spatial and frequency domains.

The trade-off between having a small Δf and a small Δx is the frequency versus spatial resolution trade-off. Fourier responses have a small Δf but large Δx, while the original signal components have small Δx but large Δf.

In coastline representation terms, note that low frequency components actually encode large scale structures in the coastline, while high frequency components encode small scale structures in the coastline. When reconstructing a low resolution view of the coastline, only the large scale structures are important, so only the low frequency components are required. In a linear transform with good frequency resolution, the components are stratified, ranging from low frequency components to high frequency components. Since only the low frequency components are needed, the reconstruction can be more efficiently carried out than if all components of the transform are needed. So using a linear transform with good frequency resolution is advantageous.

On the other hand, while reconstructing a section the coastline a component is only required if the interval the component is confined to falls within the section of the coastline. In a linear transform with good spatial resolution, the components are confined to small intervals, i.e. they are localized in spatial extent. So using a linear transform with good spatial resolution is advantageous also, since less components are required to reconstruct the segment well.

Hence the frequency versus spatial resolution trade-off translates to a trade-off between being able to reconstruct low resolution views efficiently and being able to reconstruct subsections of the coastline efficiently.

2.4 Wavelet transforms

Wavelet transforms arise as a compromise between the frequency and spatial resolution trade-off, where both Δx and Δf are reasonably small. That is, wavelet transforms are orthonormal linear transforms that are both localized in space and frequency. So we can expect wavelet transforms to fair well over a range of reconstructions of our coastlines.

Another important property of wavelets is that the basis vectors are all translations and dilations of a single vector, called the mother wavelet[‡]. This is shown in figure 4 for the Haar wavelet. Note that the translations and dilations can be arranged into a binary tree as shown in figure 4, where the two children of each wavelet each occupy half the interval where the wavelet occupies. There are $l = \log n$ levels, with level k for $0 \leq k \leq l-1$ having 2^k wavelets and each wavelet occupying 2^{-k} of the whole interval.

Another important property of wavelet transforms is that there exists an algorithm called the fast wavelet transform (FWT) that can perform wavelet transforms in $O(n)$ time and

[‡]Except for one of them, called the father wavelet, which is the wavelet equivalent of the DC component of the Fourier transform.

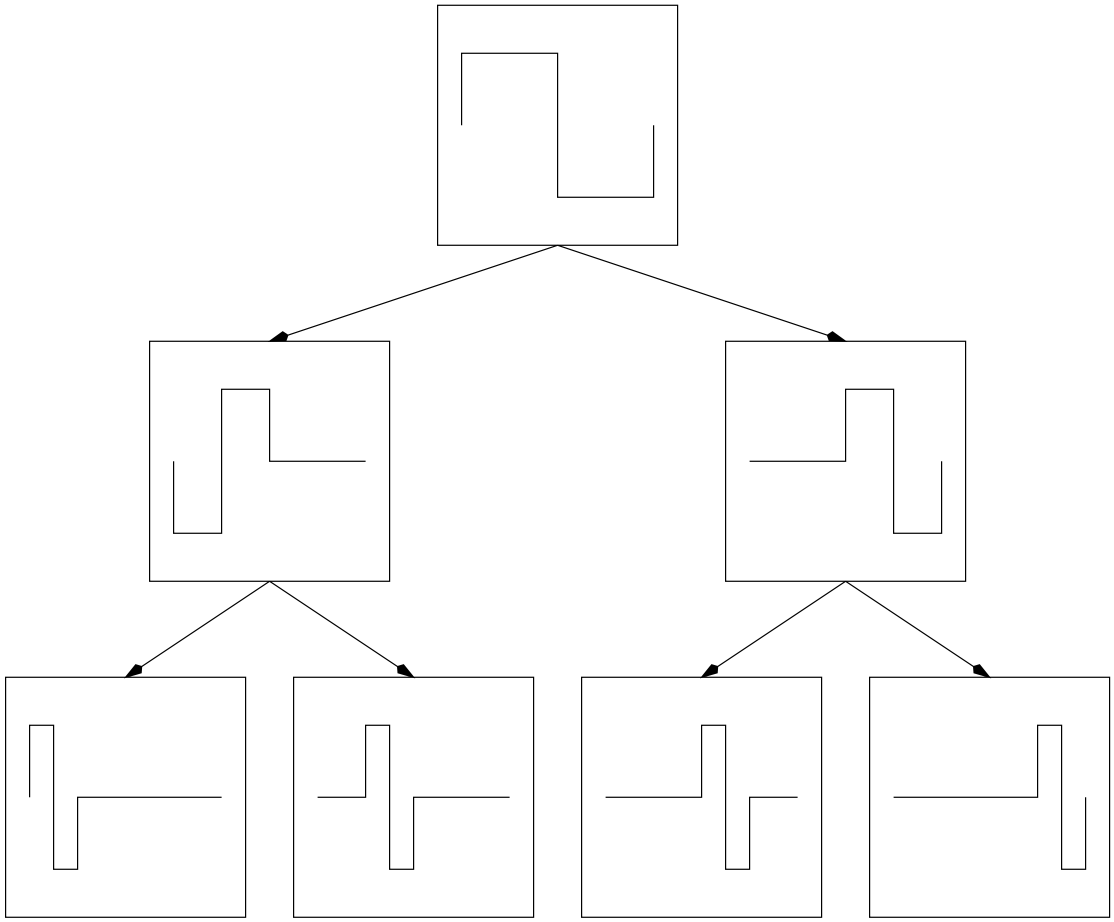


Figure 4: The wavelet basis tree for the Haar wavelet.

an algorithm called the inverse wavelet transform (IWT) that can perform inverse wavelet transforms in $O(n)$ time [10].

Figure 5 shows some examples of wavelets. The Haar wavelet is the original wavelet and is simple and useful for educational purposes, but is not often used in practice due to poor approximation properties. The other three wavelets each belong to a class of wavelets (indexed by the number following the names of the classes). The Daubechies wavelets are fractals and are highly non-smooth.

For a simple introduction to wavelets, the reader is referred to Graps [4] and Wickerhauser [11]. For an introduction to the mathematics of wavelets, including why the dilation/translation property is important, the reader is referred to Strang [7]. For all kinds of information on wavelets on the web, the reader can go to [9] and [5].

2.5 Signal representation and linear transforms

Linear transforms change the basis with respect to which we represent a signal. The basis vectors form a representation of the signal. The responses r are encodings of s with respect to the representation. Different basis vectors give different representations. In the standard

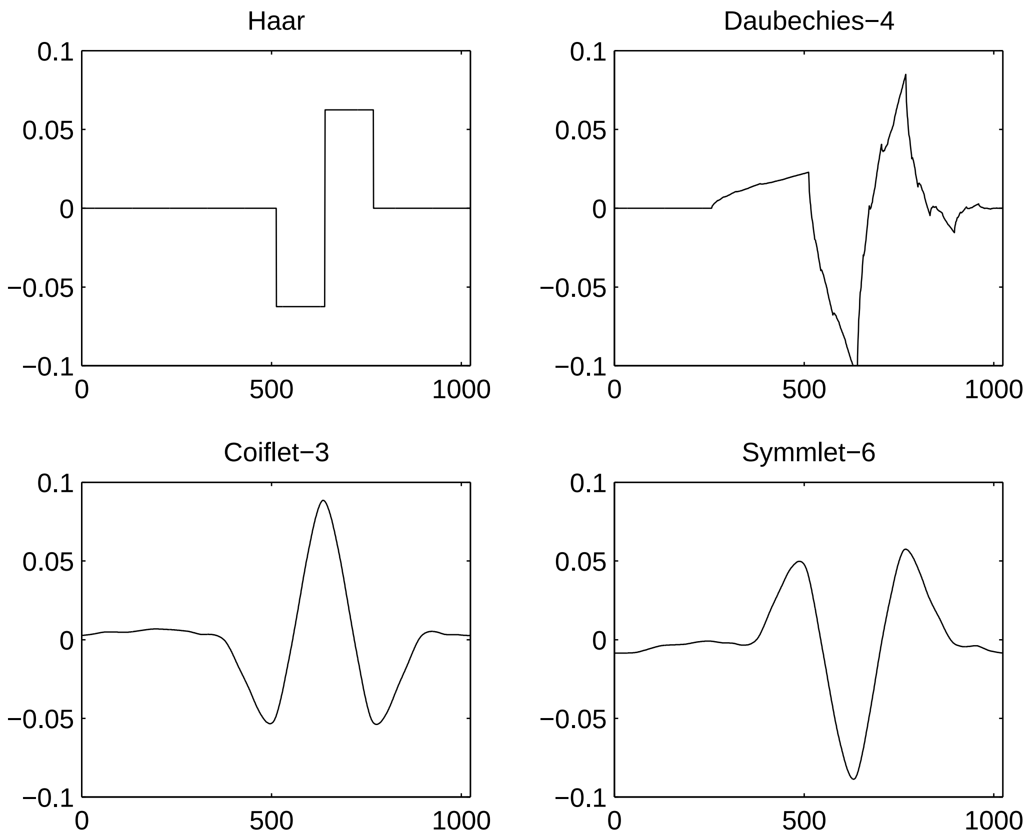


Figure 5: Four families of wavelets.

basis (given by the identity matrix), each response r_i in the encoding describes a component s_i of the signal. In the basis for Fourier transform, each response describes a frequency component of the signal, although it tells us nothing about what any of the s_i's are. In wavelet bases, each response gives partial information about a range of frequencies and a range of s_i's.

Each response contains some information about the original signal. If we can collect most of the required information to reconstruct the signal s accurately in a few responses, then we can compress the signal by representing it only with the important responses and pruning the others away (by setting to 0 and ignoring). For example, if we know in advance that signals s can be well represented[§] as

$$s = \sum_{i=1}^{m} r_i B_i + \epsilon \tag{2}$$

where $m \ll n$ and $\epsilon = \sum_{i=m+1}^{n} r_i B_i$ is some small zero mean additive noise, then in encoding s it is sufficient just to use r_i for $i = 1, \ldots, m$ and ignore r_i for $i > m$. This lets us compress the signal and even clean up the noise.

[§] If we do not know what the best representation is in advance, there are statistical techniques available allowing us to "learn" a good representation given the probability distribution over signals. Some examples are PCA, ICA and factor analysis [3].

The viewing of coastlines is another example. When viewing the coastline at large scales, we really only need information about that coastline that is visible at large scales, and not information at smaller scales. If we can represent the coastline signals using a transform that partitions the information into bands of different scales, we can then obtain a large scale view of the coastline efficiently using only the information in the large scale bands.

3 Representing coastlines with linear transforms

Recall that a coastline consists of the sequence of x-coordinates $x = [x_1, \dots, x_n]^T$ and the sequence of y-coordinates $y = [y_1, \dots, y_n]^T$. We assume that $n = 2^l$ for some integer l.

We used MATLAB [6] as our test bed. We used the WaveLab [8] toolbox for manipulating wavelets and for DWT and IWT. We shall compare the various linear transforms on a typical generated coastline, as shown in figure 3.

The linear transforms we shall be comparing are : the identity transform (i.e. the transform associated with the identity matrix, i.e. representing the coastline directly), the Fourier transform, and the Haar, Daubechies-6, Coiflet-3 and Symmlet-6 wavelet transforms.

There are many ways of comparing the relative merits of the various linear transforms. In the following subsections we compare the various transforms mentioned earlier with respect to two criteria – the reconstruction error and computational efficiency.

3.1 Reconstruction error

When reconstructing the coastline from the transform responses it is often not necessary to use all the available responses to obtain a reasonably good reconstruction (i.e. "close enough" to the naked eye). A criteria to consider when we have approximate reconstructions is how close the approximation is to the true coastline.

One way of quantifying the error or difference from the true coastline is by the total area bounded between the approximation and the true coastline. Another way is by the maximum displacement of the approximation from the true coastline. Even though they make more intuitive sense, the previous two distance measures are hard to compute. We resort to a simpler distance measure. Suppose the real coastline has coordinates $x = [x_1, \dots, x_n]^T$, $y = [y_1, \dots, y_n]^T$, and the approximation has coordinates $x' = [x'_1, \dots, x'_n]^T$, $y' = [y'_1, \dots, y'_n]^T$, then we define the distance between the two as

$$\sum_{i=1}^{n} (x_i - x'_i)^2 + (y_i - y'_i)^2 = \|x - x'\|^2 + \|y - y'\|^2 \tag{3}$$

To compare the various transforms we did a simple experiment. We compared the reconstruction errors of the various transforms while varying the number of responses used for the reconstructions.

For the identity transform, every $\frac{n}{m}$ response is used, where n is the total length of the coastline, and m is the number of responses used. Then linear interpolation is used to approximate the coordinates of the points in between every two responses. For the Fourier and wavelet transforms, the m lowest frequency responses are used. The reconstructions for

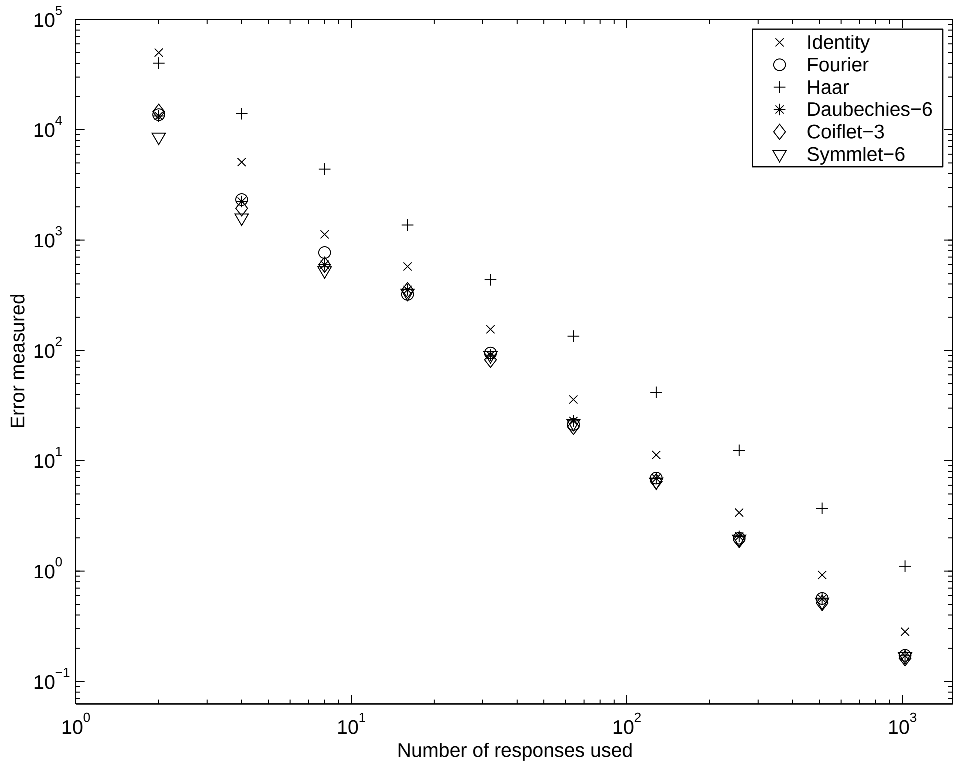


Figure 6: Error measured versus number of responses used on a log-log plot.

$m = 16, 64, 256$ and 1024 for each transform are given in the appendix. Note that with only 1024 or even 256 components, all the transforms are already giving very good reconstructions of the coastline.

Figure 6 shows the reconstruction errors versus the fraction of responses used for the various linear transforms. Figure 6 shows that the Haar transform is the worst, while the Fourier and other wavelet transforms are better than both the Haar and the identity transform. The Haar transform does the worst because it has the worst approximation properties among the wavelets. Using every few responses of the identity transform and linearly interpolating between them can be seen as an approximation to a non-orthonormal wavelet transform whose basis vectors are "Mexican hat" vectors, as shown in figure 7. This has better approximation properties than the Haar transform(¶), although it is seldom used as a wavelet because it is non-orthonormal. Besides having worse reconstructions than the Fourier and wavelet transforms (except Haar transform), using every few responses of the identity transform has another problem. It is not guaranteed to be a good reconstruction. For example, if our signal consists of alternating 0's and 1's then, using just the odd responses, we would believe that the signal is just 0. With Fourier and wavelet transforms we can prove theorems bounding the reconstruction errors.

(¶)The basis vectors are at least continuous, as opposed to the Haar basis vectors.

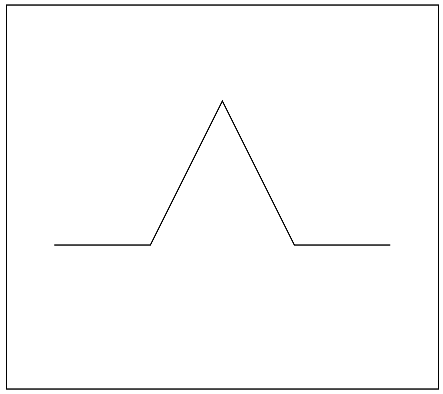


Figure 7: The Mexican Hat "non-orthonormal wavelet".

3.2 Computational efficiency

Due to the potentially large amounts of data the methods might need to handle, the amount of processing required to reconstruct the coastlines is an important criteria.

The amount of storage required is not a concern for us. As a matter of fact every response should be stored because the user can zoom in to a part of the coastline and view it at the finest resolution. If instead our aim was to compress the whole coastline for viewing purposes at the coarsest resolutions, then the amount of storage required will be an issue, and will be related to the reconstruction errors incurred as a result of compressing the responses.

There are two things we need to consider : the amount of processing time required and the number of responses used to reconstruct the coastline. The number of responses required is an important consideration when retrieving the responses might take a much longer amount of time than the processing, for example reading from a slow secondary storage or retrieving through the internet.

First consider the case when we wish to achieve the highest resolution reconstruction on a section of the coastline, say between indices i to $i+j-1$. With the identity transform, this requires j responses and $O(j)$ processing time just to retrieve the responses. With the Fourier transform, as every basis vector is global, every response is required to reconstruct the section, hence n responses are required and at least $O(n \log j)$ processing time is required (FFT requires $O(n \log n)$ time but since we only require j signal values this may be improved to $O(n \log j)$). With wavelet transforms, the only responses required are those whose corresponding basis vectors are non-zero on the section of the reconstruction. On level k of the wavelet tree, each wavelet is confined to an interval of length 2^{l-k}, where $l = \log n$, hence at most $\lceil \frac{j}{2^{l-k}} \rceil + 1$ basis vectors are relevant to reconstructing the coastline section, so in total at most

$$1 + \sum_{k=0}^{\log n - 1} \lceil \frac{j}{2^{l-k}} \rceil + 1 \leq 1 + \sum_{k=0}^{\log n - 1} \frac{j}{2^{l-k}} + 2 \leq j + 2 \log n + 1 \tag{4}$$

responses are required. Further, the IWT can easily be adapted to ignore irrelevant computations and run in $O(j + \log n)$ time. With $j \ll n$, the Fourier transform is clearly at a

Transform	Relation
Identity	$\log m = -0.7986 \log E + 9.0168$
Fourier	$\log m = -0.8236 \log E + 8.6063$
Haar	$\log m = -0.8363 \log E + 10.0346$
Daubechies-6	$\log m = -0.8342 \log E + 8.6317$
Coiflet-3	$\log m = -0.8315 \log E + 8.5916$
Symmlet-6	$\log m = -0.8516 \log E + 8.5894$

Table 1: Relation between error (E) and number of responses used (m).

disadvantage, while the wavelet transforms are only worse than the identity transform by an additive factor of $O(\log n)$.

Now consider the case when we wish to achieve a low resolution reconstruction of the whole coastline. That is, we want an approximate coastline $(x'_1, y'_1), \ldots, (x'_n, y'_n)$ such that the error measured is within some tolerance θ of the true coastline. Given a tolerance θ, one can estimate the number of responses required from figure 6. First, using linear regression we fit straight lines to the log-log plot, giving equations relating the error to the number of responses required for each linear transform. This is shown in table 1. We see that in general the gradients of the lines are all approximately 0.83, while the additive constants differ more significantly. This can be seen from figure 6 directly. In particular the additive constants of the Haar transform is 10.0346 and that of the identity transform is 9.0168, both being larger than the other transforms, which are all approximately 8.6. The differences in additive constants shows that the number of responses required by the Haar transform is about $\frac{10.03}{8.6} = 1.166$ times the number required by the Fourier and other wavelet transforms, while the identity transform needs about $\frac{9.02}{8.6} = 1.0485$ times as many responses. As for the processing time required, both the identity transform and wavelet transforms require only $O(n)$ processing time, while the Fourier transform requires $O(n \log n)$ (again this might be improved to $O(n \log m)$ where m is the number of responses used.). All in all, in this case, the Haar and identity transform require slightly more responses to achieve the same quality of reconstruction, while the Fourier transform needs more processing time. However, with the identity transform, there is no guarantee that it will perform as well as it did here, as the way we pick the responses is only a heuristics.

Finally consider the case when an approximate reconstruction of a segment of the coastline is required. With the Fourier transform, we first determine the responses required to approximate the whole coastline within the given tolerance. But as the corresponding basis vectors are global, all of the responses are required, even though we only need to reconstruct a segment of the coastline. However it may be possible to spend less processing time reconstructing just the segment instead of the whole coastline. With the identity and wavelet transforms, after determining the responses required to reconstruct the whole coastline to the given tolerance, instead of using all of them, we only use those that are relevant to the segment of the coastline we are interested in. This can be done efficiently.

While Fourier transforms are inefficient at reconstructing just a subsection of the coastline, there are no guarantees on the performance of the identity transform at reconstructing coastlines at low resolutions. On the other hand, wavelet transforms (except Haar transform) can efficiently reconstruct subsections of coastlines at various resolutions, and there are bounds on the reconstruction errors incurred.

4 Perceptual importance

4.1 Perceptual differences

Although the error measures we suggested in section 3.1 are mathematically quantifiable, they do not necessarily correspond to how different we actually perceive the approximation to be from the true coastline. To reduce the perceptual difference between the approximation and the true coastline, it is important that the perceptual characteristics of the true coastline, like the degree of wiggliness (fractal dimension) and shapes of bays and river mouths, are matched as closely as possible. A drawback of this criteria is that it is very hard to quantify perceptual difference. Also, perceptual differences may depend on outside factors.

4.2 Perceptual importance

As a first step towards measuring perceptual difference, we can introduce an importance factor to each location on the coastline. Often, some parts of the coastline are more important than others. For example, the coastline close to a city might be more important than the surrounding areas simply because users might pay more attention to the cities rather than the country side. For another example, given the east coast of the United States, it is important for approximations to get the shape and location of Florida right, as the peninsula of Florida is a salient feature of the east coast. To handle differences in importance, we can weigh the errors in section 3.1 in each part of the coastline by how important that part is, so that deviations from more important parts of the coastline are penalized more, so the more important sections will be better approximated.

Another way of using the importance of sections of the coastline is related to the computational criteria. When we have limited computational capabilities or time limits, it is important to concentrate the available computations on more important parts of the coastline, so that these parts are better approximated.

One way to assignment importances is to look at the shape of the coastline. Normally parts of the coastline with high curvature are more salient hence important than parts with low curvature, i.e. straight lines etc. However the problem is that it is hard to determine what is the curvature of a coastline. This is because coastlines are fractals, and are highly irregular and non-smooth. What looks like a smooth curve at one scale will turn out to have intricate structure at lower scales. This makes analyzing the shape of coastlines a challenge.

An interesting line of research is to determine some ways of quantifying perceptual importance based on user input. This is along the same line of thought as Cutzu and Tarr [2] where the idea is to let users choose good views of a 3-dimensional object, and based on those views, determine which parts of the object are salient (important). In our case, we can define the importance of a part of the coastline as a positive number. Start off the program with all parts of the coastline equally important, and let users zoom in to parts of the coastline at will. Presumably the parts of the coastline which users zoom in at are more important, so we can increase the importance of those parts, and decrease the importance of other parts. If we make sure that the sum of the importances is a fixed constant after every update, this procedure will converge to the perceived importance of each part of the coastline. Further, this procedure can also adapt to changing user preferences so long as user preferences do not change abruptly often.

4.3 Perceptual characteristics of linear transforms

As seen in the reconstructions in the appendix, the characteristics of reconstructed coastlines depend critically on the types of linear transforms used. The reconstructions based on the identity and Haar transforms seem to be made up of straight line segments, just as the basis vectors of the Haar transform and the Mexican hat vectors in figure 7 do. The reconstructions based on the Fourier, Coiflet-3 and Symmlet-6 transforms look like smooth curves, just as the corresponding basis vectors do. The reconstructions based on the Daubechies-6 transform are somewhat wiggly, as the basis vectors of the Daubechies-6 transform are fractals.

If the basis vectors of a transform have perceptual characteristics that differ significantly from coastlines, then more responses will be required to match the characteristics of the coastline better. An interesting idea is to find a fractal orthonormal wavelet transform whose perceptual characteristics like wiggliness etc match as closely as possible to those of coastlines, while at the same time can be shown to have good approximating properties. Such a transform will require less responses to reconstruct coastlines at an acceptable level of accuracy, as measured both by the error measure and by perceptual differences.

5 Conclusion

We show that wavelet transforms except the Haar transform are better at efficiently approximating coastlines and subsections of coastlines at a range of resolutions. The Fourier transform is not suitable at reconstructing subsections of the coastline, while the identity transform is not guaranteed to work well at low resolutions.

Our techniques developed for coastline representation can potentially be applied to other domains, for example, elevation maps, images, and audio signals.

Two interesting lines of research were pointed out in section 4. One is to find some way of determining the perceptual importance of parts of the coastline. The second is to find a wavelet transform whose perceptual characteristics match as closely as possible to those of coastlines.

References

[1] M. F. Barnsley. *Fractals Everywhere.* Morgan Kaufmann Publishers, 1988.

[2] F. Cutzu and M. J. Tarr. Inferring perceptual saliency fields from viewpoint-dependent recognition data. *Neural Computation*, 11:1331–1348, 1999.

[3] Z. Ghahramani and S. Roweis. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999.

[4] A. Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2), 1995.

[5] Mathsoft wavelet resources. `http://www.mathsoft.com/wavelets.html`.

[6] Matlab. `http://www.mathworks.com/products/matlab/`.

[7] G. Strang. Wavelets and dilation equations : a brief introduction. *SIAM Review*, 31:613–627, 1989.

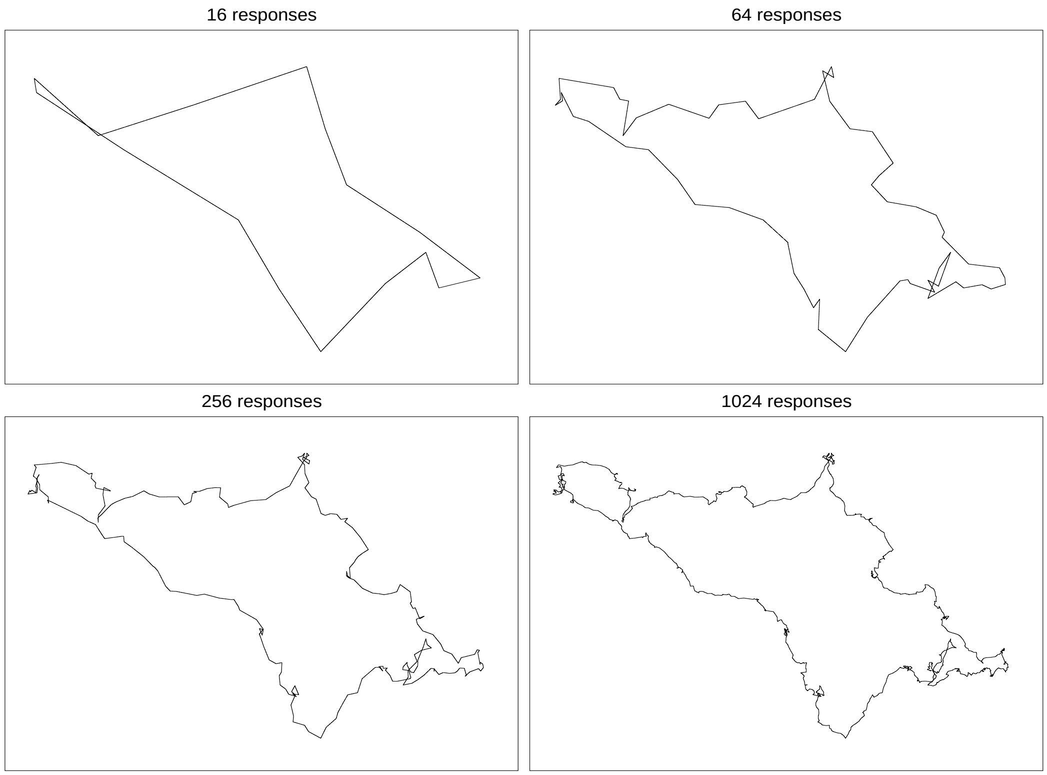
[8] Wavelab. `http://www-stat.stanford.edu/~wavelab/`.

[9] Wavelet digest. `http://www.wavelet.org/wavelet`.

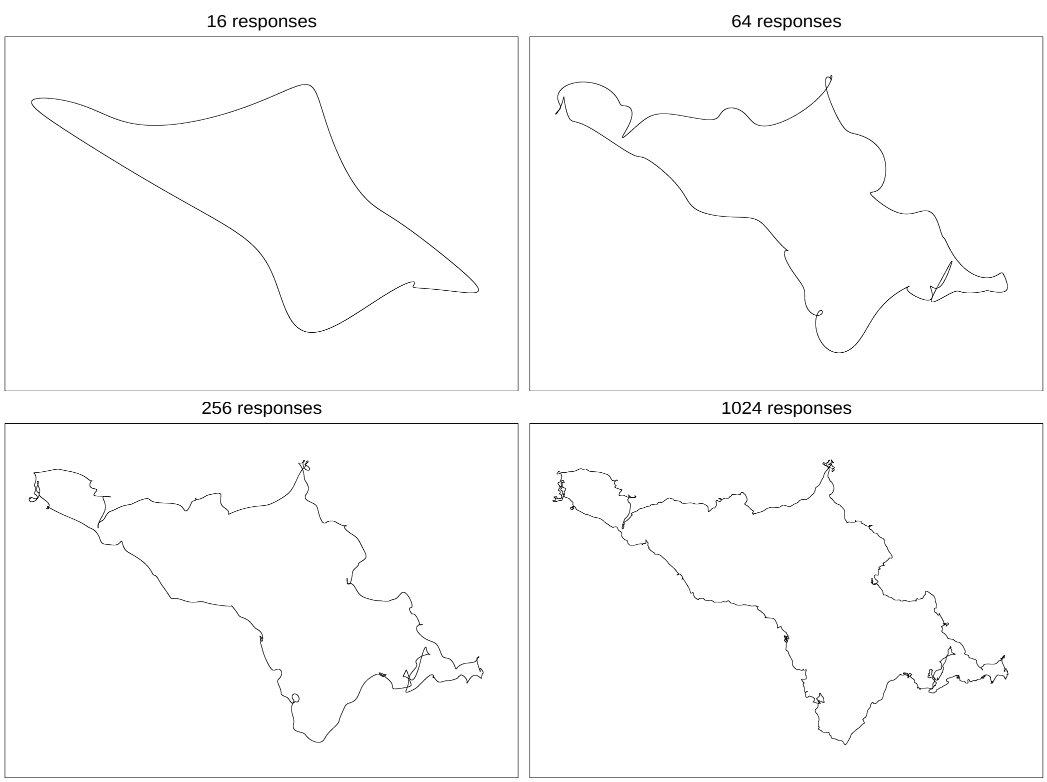
[10] M. V. Wickerhauser. *Adapted Wavelet Analysis : from Theory to Software.* AK Peters Ltd, Wellesley, Massachusetts, 1994.

[11] M. V. Wickerhauser. Wavelet transforms. In P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, Henry F. Schaeffer III, and P. R. Schreiner, editors, *Encyclopedia of Computational Chemistry*, volume 5, pages 3214–3222. John Wiley & Sons, Limited, 1998.

A Reconstruction using Identity transform

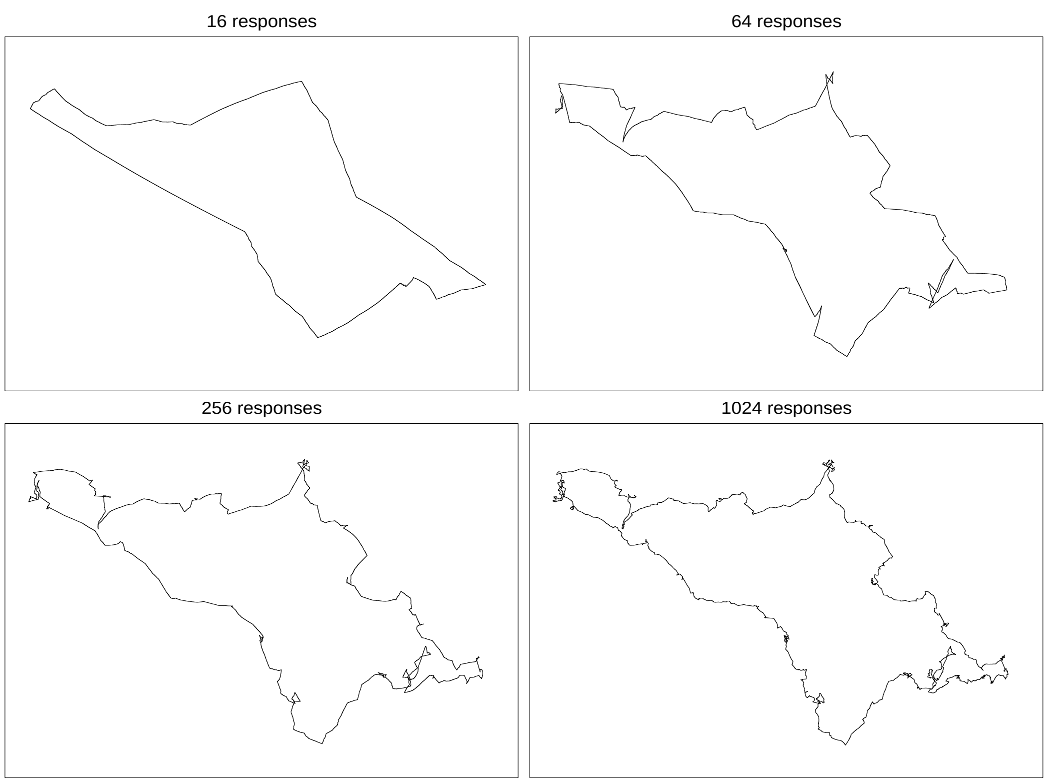


B Reconstruction using Fourier transform

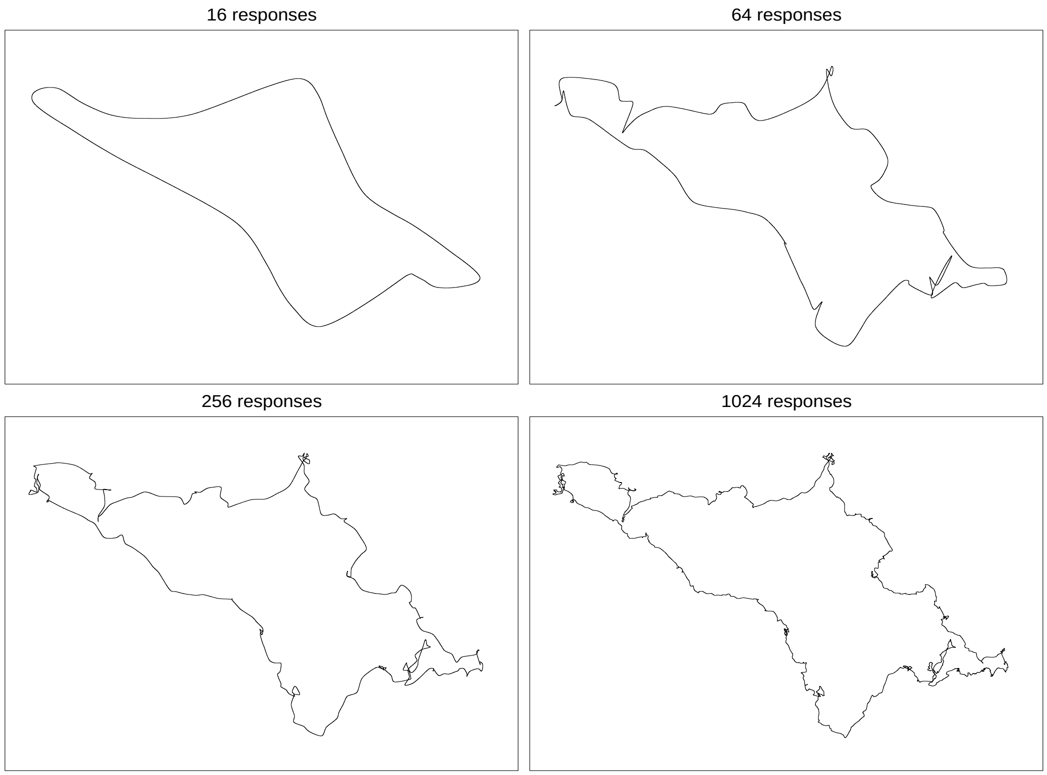


C Reconstruction using Haar transform

D Reconstruction using Daubechies-6 transform



E Reconstruction using Coiflet-3 transform



F Reconstruction using Symmlet-6 transform