Diffusion Wavelets for Natural Image Analysis

Tyrus Berry

December 16, 2011

Contents

1 Project Description

A recent construction by Maggioni and Coifman [2], provides a promising generalization of wavelet analysis which may provide the answer. Thus we first verify the construction of Maggioni and Coifman by attempting to recover standard wavelet analysis on the interval $[-1/2, 1/2]$ as a special case of their general technique. This construction is very helpful in understanding the generalization. We then apply the general analysis in cases such as non-uniform sampling and generalized dilation operators which are still easy to understand and illustrate the power and limitations of the generalized technique. Finally, we apply the technique to the highly complex case of natural images of faces taken from the Labeled Faces in the Wild database [6].

2 Introduction to Diffusion Wavelets

Diffusion wavelets seek to generalize the construction of wavelet bases to smooth manifolds where there is still a natural notion of translation given by the local coordinate maps. However, since there is no dilation operator on a manifold, Maggioni and Coifman substitute a diffusion operator. For a smooth manifold Ω, the prototype of a diffusion operator is the heat kernel of the laplacian on a manifold given by

$$T\phi = \int_{\Omega} K(t, x, y)\phi(y)dy = e^{t\triangle}\phi$$

which is always a smoothing operator. Thus, instead of dilations on a Euclidean domain, diffusion wavelets are adapted to a diffusion operator such as T. In this section we will define a multiresolution adapted to the operator T in analogy to the classical wavelet construction. We then give the process for constructing scaling functions and wavelets from any operator T. In the next section we will demonstrate that taking $\Omega = [-1/2, 1/2]$ and T to be dilation, we can numerically recover classical wavelets and scaling functions.

2.1 Diffusion Multiresolution

Let Ω be a smooth manifold and $\{T^t\}_{t\geq 0}$ be a semigroup of operators $T^t : L^2(\Omega) \to L^2(\Omega)$, such that $T^{t_1+t_2} = T^{t_1}T^{t_2}$. We call T^t a diffusion semigroup and $T = T^1$ a diffusion operator if

1. $||T^t|| \leq 1$ for all $t \geq 0$ (contractive)
2. T^t is self-adjoint (symmetry)
3. For every $f \in L^2(\Omega)$ with $f \geq 0$ we have $T^t f \geq 0$ (positive)

Note that in a discretization of the domain Ω, T will become a matrix and these conditions correspond to T being positive semi-definite with eigenvalues bounded above by 1. In the next section we will give several relevant ways to construct such a matrix on a discretized domain.

Classical wavelet analysis constructs the multiresolution using the span of translates of a scaling function at a certain scale. For diffusion wavelets, we use the span of all eigenfunctions of T at a certain scale. Let $\sigma(T) \subset [0,1]$ denote the spectrum of T and $\xi(T)$ the associated eigenfunctions. Then $\sigma(T)^t$ is the spectrum of T^t so we can decompose the spectrum into dyadic scales by defining

$$\sigma_j(T) = \{\lambda \in \sigma(T) : \lambda^{t_j} \geq \epsilon\} \qquad t_j = 2^{j+1} - 1$$

and

$$\xi_j(T) = \{f_\lambda \in \xi(T) : \lambda \in \sigma_j(T)\}$$

Which gives a natural decomposition of $L^2(\Omega)$ as

$$\begin{aligned} V_{-1} &= L^2(\Omega) \\ V_j &= \overline{\text{span}}\{\xi_j(T)\} \end{aligned}$$

Note that by definition, each V_j is spanned by $\xi_j(T)$ and since each $\xi_j(T)$ is made up of eigenfunctions of T it is a basis for V_j. Moreover, since $\sigma_{j+1}(T) \subset \sigma_j(T)$ we have $V_{j+1} \subset V_j \subset V_{-1}$ for each scale j. Thus the collection $\{V_j\}_{j \geq -1}$ will play the role of the scaling multiresolution for our diffusion wavelet analysis. Moreover, we can easily define the detail spaces W_j to be the orthogonal complement of V_{j+1} in V_j just as in classical wavelet analysis.

The key to recovering a wavelet analysis from the multiresolution $\{V_j\}$ is finding a basis for V_j which is made up of translates of a single function. Since there is no reason to expect the eigenfunctions in $\xi_j(T)$ to have this property we will have to construct such a basis.

3 Recovering Classical Wavelet Analysis

While the above construction is for a self-adjoint operator T, this is only a helpful property for certain algorithms and is not required for the basic constructions. Thus if we set $T = D_2$ the dilation operator given by $D_2 f(x) = 2^{-1/2} f(x/2)$ we should recover classical wavelet analysis. We first examine this problem numerically to test our numerical procedure and gain insight in a more simple domain. Thus we discretize the domain $[-1/2, 1/2]$ into N equal intervals of length $\delta = 1/N$. We can represent the discrete

version of $L^2([-1/2, 1/2])$ as vectors $\overline{y} \in \mathbb{R}^N$ where $y_i = f(i/N - 1/2)$. Thus y_i is the representation of f with respect to the basis (for the discrete space of functions) of delta functions $\delta_i = 1_{\{x=i/N-1/2\}}$. Note that we cannot define D_2 exactly on this domain, since $D_2(y_i) = D_2(f(i/N - 1/2) = f(i/N/2 - 1/4)$ but there is no value in our vector $\overline{y}$ which represents the point $i/N/2 - 1/4$. Thus we must choose an interpolation scheme, and the discrete version of D_2 will actually represent interpolation followed by dilation. We now introduce two interpolation schemes which will be used below.

3.1 First Order Interpolation

For a first order interpolation, we simply interpolate linearly between the discrete points $i/N - 1/2$. Thus to find $D_2(y_i) = f(i/N/2 - 1/4)$ choose $k/N - 1/2$ be the closest point to the left of $i/N/2 - 1/4$. Then, letting $m = (i/2 - k + N/4)$ we can approximate

$$D_2(y_i) = f(i/N/2 - 1/4) \cong \frac{y_{k+1} - y_k}{1/N}(i/N/2 - 1/4) + y_k = my_{k+1} + (1 - m)y_k$$

Thus in our matrix representation of D_2 we set the (i, k) entry to $(1-m)$ and the $(i, k+1)$ entry to m. Repeating this for each i gives us a matrix representation for D_2 which is shown graphically in Figure 1. Thus for any function represented discretely as a vector $\overline{y}$ we can compute interpolation followed by dilation simply by multiplying $\overline{y}$ by the matrix D_2. Note that the rows of D_2 sum to 1 which guarantees that the eigenvalues are bounded above by 1 and D_2 is positive semi-definite but not symmetric.

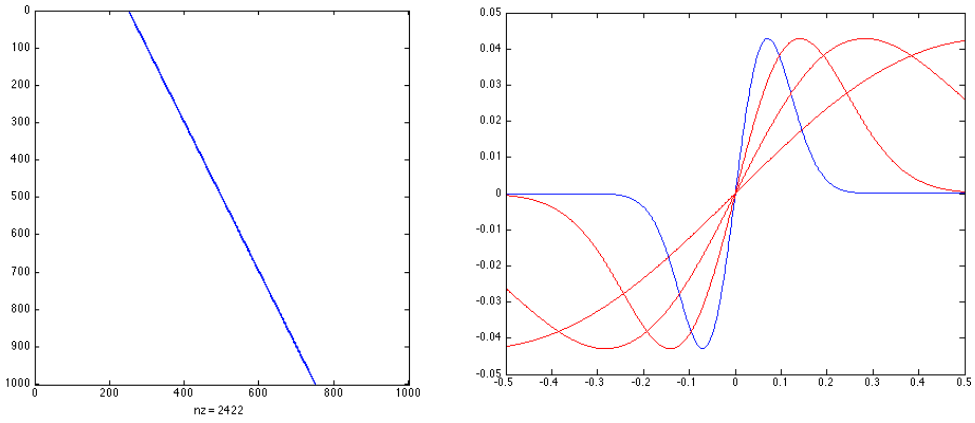

Figure 1: Left: Image of the matrix D_2 for the first order interpolation, Right: D_2 applied to vector representing a simple function.

To find the scaling functions, we compute the QR-decomposition of D_2, which is shown in Figure 2. The matrix Q represents a change of basis matrix that takes the initial basis

of delta functions δ_i to a basis for V_0 which consists of translates of a single function. Note that since D_2 is not full rank, only the left half of the Q matrix in figure 2 is meaningful. This can also be seen by noticing that R is only non-zero in the upper half of the matrix as shown in figure 2. Notice that each column of Q represents a scaling function at scale 1 which are all translates of each other and this is reflected in the structure of Q, each column is simply a shift of the previous column.

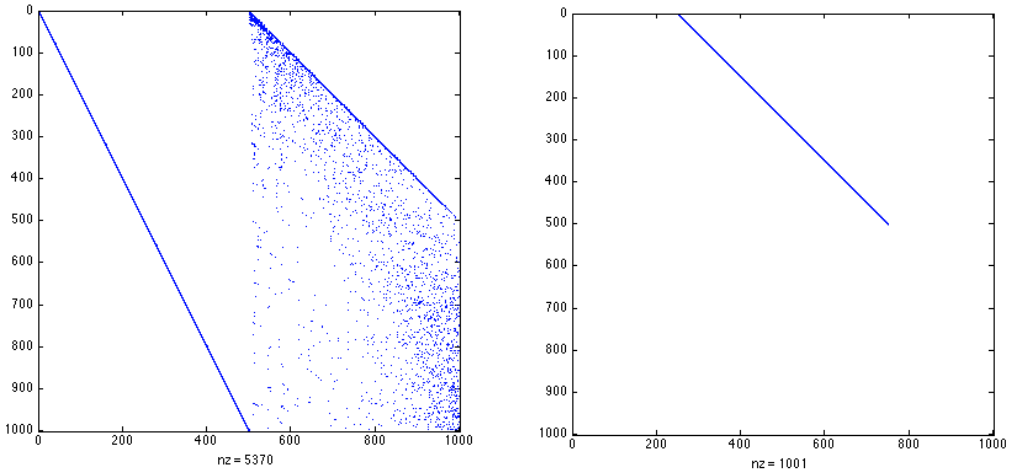

Figure 2: The scaling QR decomposition of the operator D_2 in figure 1, Left: Q matrix, Right: R matrix.

To find the wavelet functions we need to repeat the above analysis, but on the orthogonal complement of the space V_0 in V_{-1}. This is reflected in the fact that the matrix Q is not full rank, and hence $I - QQ^T$ spans the space W_0 and we can find the wavelet functions by performing a QR-decomposition

$$I - QQ^T = \overline{Q_1}\,\overline{R_1}$$

and the columns of $\overline{Q_1}$ will contain the wavelet functions. This is shown in the left image of figure 3. As we can see the columns from this matrix are not well localized, which is due to the right half of Q which is meaningless (shown in figure 2). By setting $\hat{Q}$ equal to Q with the right half set to zero we achieved much better results with

$$I - \hat{Q}\hat{Q}^T = \overline{Q}\,\overline{R}$$

In all our results we use wavelet functions coming from the columns of $\overline{Q}$. Note that in figure 3 we can see that $\overline{Q}$ is well localized and each wavelet function is approximately a simple translate of a single function. The results in Figure 4 show examples of the first order scaling functions and wavelets at scale V_0. The left image in figure 4 shows that

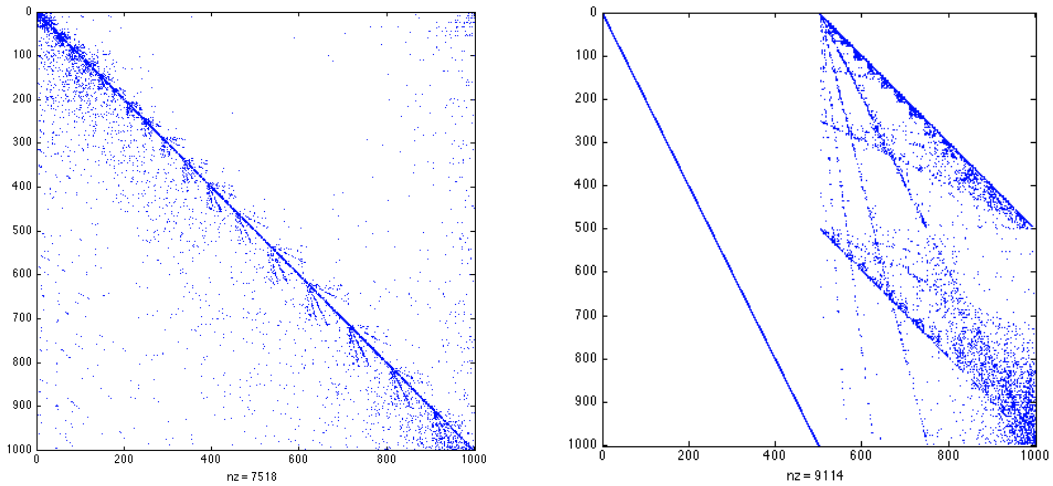

Figure 3: The wavelet decomposition, $\overline{Q_1}$ (left) such that $\overline{Q_1 R_1} = I - QQ^T$ and the matrix $\overline{Q}$ (right) such that $\overline{QR} = 1 - \hat{Q}\hat{Q}^T$

each wavelet is approximately a translate of a single function (note some are inverted, which can be accounted for by the sign of the corresponding entry in the R matrix from the QR-decomposition). In the right image in figure 4 we show examples of a scaling function (red) along with its FFT (red, dotted) and also a wavelet (blue) and its FFT (blue, dotted).

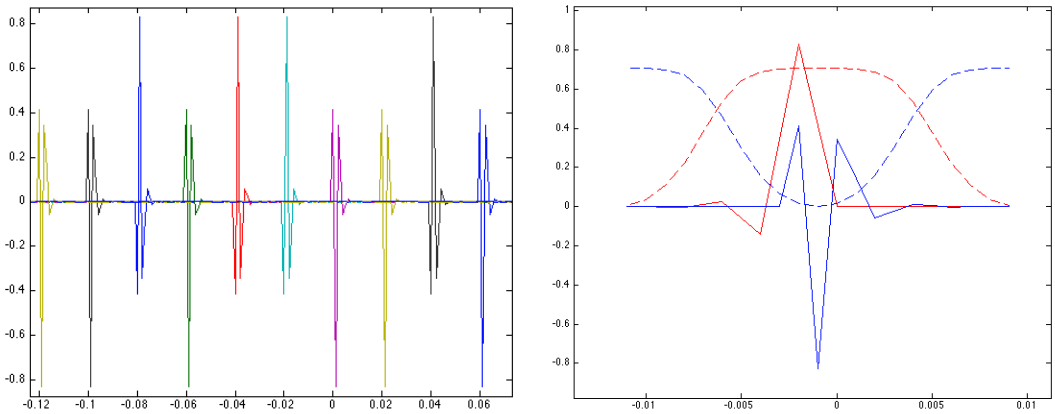

Figure 4: Left: First Order Wavelets, Right: First Order Wavelet (blue) and FFT (blue dotted) and Scaling function (red) and FFT (red dotted)

3.2 Second Order Interpolation

The second order interpolation version of the dilation matrix D_2 is constructed using 3 values of the original function and interpolating quadratically. Thus to find $D_2(y)_i = f(i/N/2 - 1/4)$ we again find k such that $k/N - 1/2$ is the closest point to the left of $i/N/2 - 1/4$. We then define

$$\begin{aligned} m &= (i/2 - k + N/4) \\ c &= m^2/2 \\ D(i, k+2) &= c \\ D(i, k+1) &= m - 2c \\ D(i, k) &= 1 - m + c \end{aligned}$$

So that the matrix D represents quadratic interpolation followed by dilation by D_2. The above analysis can be repeated on this new version and a slightly different scaling function and wavelet function are found as shown in Figure 5.

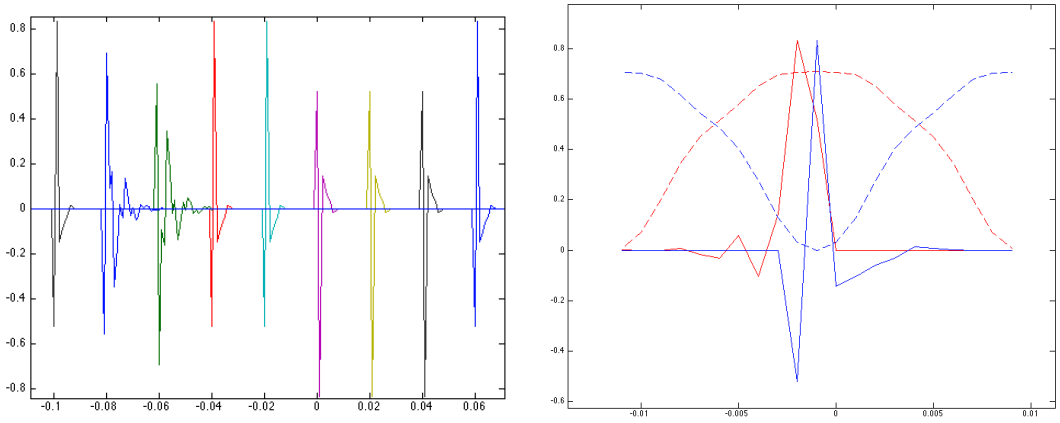

Figure 5: Left: Second Order Wavelets, Right: Second Order Wavelet (blue) and FFT (blue dotted) and Scaling function (red) and FFT (red dotted)

3.3 Heat Kernel Interpolation

The generality of the Diffusion Wavelet analysis now allows us to analyze a completely different operator T which is a discrete version of the Heat Kernel on the domain $[-1/2, 1/2]$. To construct the operator T we simply specify a number of neighbors K and then define

the diffusion distance from a discrete point to each of its K nearest neighbors as

$$T(i,k) = \exp\left\{-(i/N - k/N)^2/\sigma\right\}$$

Where k is the index of one of the K nearest neighbors of the point indexed by i and σ is a tuning parameter which is related to the diffusion rate and may need to be adjusted as the number K of nearest neighbors changes. Note that this distance could be defined for all points, however it decays very quickly and so a sparse representation of T using the K nearest neighbors is much more efficient. Finally the columns of T are normalized to sum to 1.

The above analysis was repeated for the matrix T in order to construct the scaling and wavelet functions. Again they were found to be approximately translates of a single function and the results are shown in Figure 6.

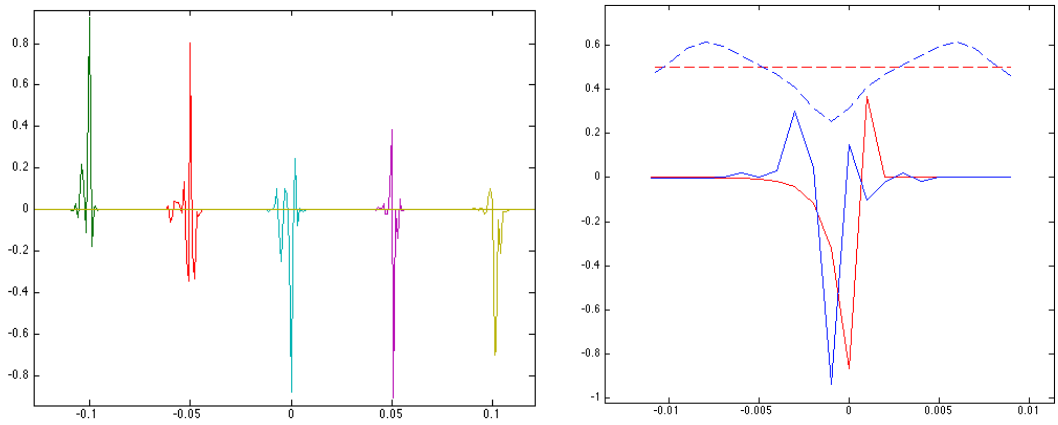

Figure 6: Left: Heat Kernel Wavelets, Right: Heat Kernel Wavelet (blue) and FFT (blue dotted) and Scaling function (red) and FFT (red dotted)

4 Image Analysis

Images are very high dimensional, however, restricted to a particular domain, such as images of faces, only a small portion of the high dimensional space contains data. A common personal camera can now produce 2048-pixel by 1024-pixel images, with at least 3 vales (red, green, and blue) per pixel. If we assume that these values lie in $I = [0,1]$ then a priori such an image is a point in $I^{6,291,456}$. Of course we are not interested in the entire set of images, but a subset. For example if we are only studying images of faces, we can consider the space $\mathcal{A} \subset I^{6,291,456}$ of all face images. Moreover, a smooth perturbation of a face image will usually still look like a face and thus should lie very near the space

$\mathcal{A}$. So we may hope there is actually a smooth manifold $\overline{\mathcal{A}}$ which lies 'near' the space of interest, $\mathcal{A}$. In such a case, we can consider our data set to be intrinsically m-dimensional, where m is the dimension of the manifold $\overline{\mathcal{A}}$. If m is reasonably small we may be able to construct an efficient basis or frame for our data set using images which are concentrated on $\overline{\mathcal{A}}$.

However, even the space $\overline{\mathcal{A}}$ is difficult to work with, for example, how could we interpolate between two images using their representation as elements of $\overline{\mathcal{A}}$? To solve this problem it is advantageous to interpret images as functions $f : X \to I$ on the set X of pixels. This space is naturally identified with $I^{6,291,456}$ and we can identify the set of all face images as a subset of functions $\mathcal{F} \subset L^2(X, I)$. If we could find a basis β for $\mathcal{F}$ then we could easily interpolate between images of faces by writing each image in the basis β and then linearly interpolating.

In the next section we introduce the frame of sub-images, which gives a data driven approach to find a frame for the space $\mathcal{F}$.

4.1 Frames of Sub-Images

Let $\mathcal{A} = \{A_i\}$ be a family of images for $i = 1, ..., M$. For simplicity assume that all images have the same number, p, of pixels where separate color values are counted as separate pixels. Thus in the above example of a 2048x1024 image with RGB values, we would take $p = 6,291,456$. Let X be the set of pixels so $|X| = p$ and identify each image A_i with a function $f_i : X \to [0, 1]$ so that for each pixel $x \in X$ the value of $f(x)$ is the value of the pixel x in the image A_i. Since f_i is a function on a discrete set X we will naturally identify f_i with a column vector $[f_i]$ in I^p.

A simple basis for the set of all images is given by the delta functions $\delta_x(y) = 0$ for y not equal to x and $\delta_x(x) = 1$. This basis for the set of all images is actually a frame for $\mathcal{F}$, since when $\mathcal{A}$ has dimension less than p the set of delta functions is overcomplete. Note that for any collection of functions $G = \{g_k : X \to I\}$ we can define the analysis operator as a matrix $[G]$ which has row k given by g_k and the synthesis operator $[H] = [G]^T$. Note that G is a frame for $\mathcal{F}$ if and only if the frame operator $[S] = [H][G] = [G]^T[G]$ is invertible on $\mathcal{F}$. Thus another trivial frame for $\mathcal{F}$ is simply given by the set $\mathcal{F}$.

So we trivially have two frames for $\mathcal{F}$. The set of deltas has too many elements and is computationally intractable, and the full set $\mathcal{F}$ has too few elements, and cannot take advantage of any redundancies in the images. A frame of sub-images lies between these extremes. Pick a window $\gamma : X \to I$ which is supported on a strict subset of X and let $T_{a,b}\gamma$ be a two-dimensional periodic translation of the image represented by γ by a pixels to the right and b pixels down. Define $\mathcal{F}_\gamma = \{fT_{a,b}\gamma : f \in \mathcal{F}, a, b \in \mathbb{Z}\}$, then $\mathcal{F}_\gamma$ is also a frame for $\mathcal{F}$, we call this a frame of sub-images.

We now give an example of a frame of sub-images and the frame reconstruction on a space $\mathcal{A}$ of images of faces. We take a window function γ which is 1 on a 4-pixel by 4-pixel square in the middle of the image and zero everywhere else. Thus the product $fT_{a,b}\gamma$ is literally a sub-image of the full image f. Each image $A_i \in \mathcal{A}$ will be a simple 32x32 RGB image of a face a shown in Figure 7. Thus each image can be represented as a vector of length 3072 (32x32 with 3 values at each location).

Figure 7: Left: Example images of faces from $\mathcal{A}$, Right: left column shows original face images, middle column shows the frame operator applied to the image, right column shows the reconstructed face image (this format is used repeatedly below).

In this example we only took 16 images for our space $\mathcal{A}$ and instead of forming the full frame $\mathcal{F}_\gamma$ we chose 4096 random translations. Since the images are at most 3072-dimensional it was reasonably certain that 4096 images would give a frame, and indeed the frame operator was invertible. In Figure 7 this is clearly shown as images had very good reconstruction after the applying the frame operator. Even more impressive, this seems to be a good frame for the set of all such face images since Figure 8 shows that even faces which were not included in the initial set $\mathcal{A}$ could still be reconstructed after the frame operator was applied.

Next we applied the same analysis with only 1024 random translations of the window. Since the full image space is 3072-dimensional this cannot be a frame for the full image space, however it may still be considered a frame for the space of facial images. Note that images of faces in Figure 9 are not perfectly reconstructed, viewed as an image they are noisy, however viewed as a face this reconstruction may be considered exact depending upon the precise definition of a face. Note, also in Figure 9, that on images of non-faces the performance degrades significantly.

Figure 8: Frame operator and reconstruction applied to out-of-sample faces.

Figure 9: Left: Good reconstruction for faces, Right: Poor reconstruction for non-faces.

Finally, we attempted a reconstruction using only 512 random translations and performance degraded considerably as shown in Figure 10 but faces are still somewhat recognizable.

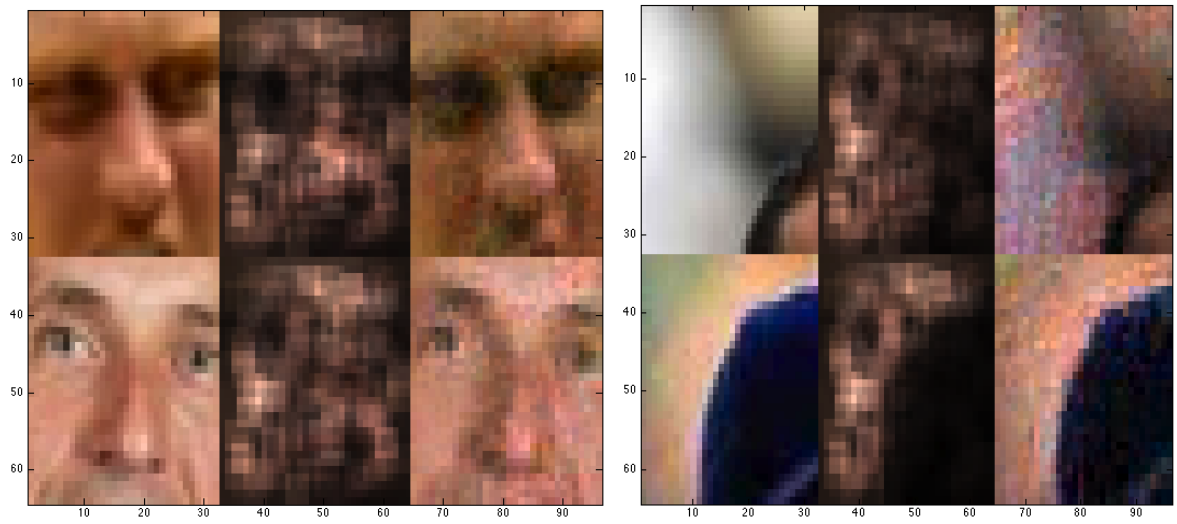

Figure 10: Left: Good reconstruction for faces, Right: Poor reconstruction for non-faces.

Note, in Figure 11, that even partial faces are reconstructed, while the non-face parts of the image are drastically changed.

This pragmatic approach to constructing a frame for a space of images uses sub-images (defined by the translates of the window γ) to build a frame. It is clear that if enough such sub-images are chosen the result will be a frame, however our experiments show that even for a relatively small number of randomly chosen sub-images the behavior is similar to a frame in that approximate reconstruction is possible. In face these small sets are frames for some space (their span) which is a reasonable approximation of the space $\mathcal{A}$ which we wish to represent.

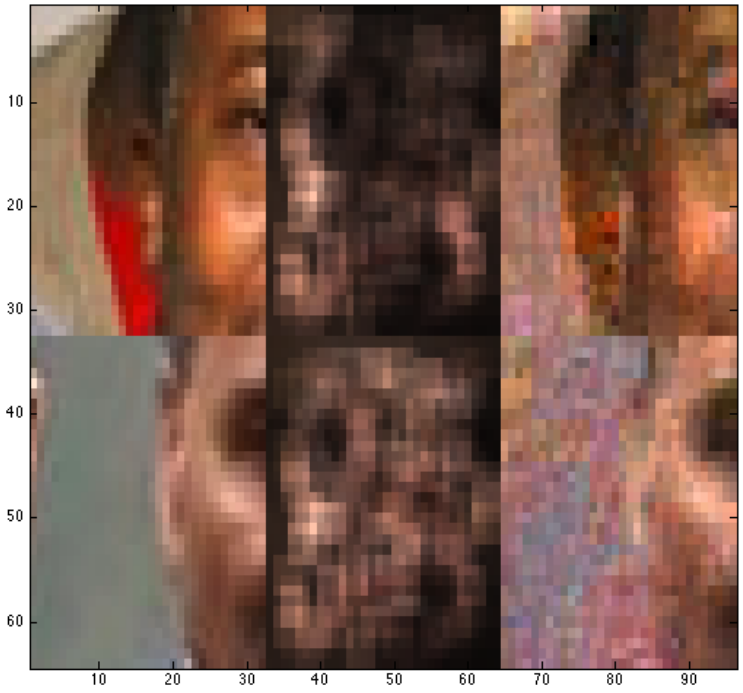

Figure 11: Reconstruction of partial face images.

4.2 Multiscale Image Distance

Let $\mathcal{A} = \{A_i\}$ be a family of images for $i = 1, ..., M$. Our first goal is to produce a multi-scale distance on images following [2]. We begin by defining a distance on the sub-images. Let $\{X_{ij}\}_{j=1}^{N}$ be the collection of all 5-pixel by 5-pixel sub-images of A_i. Note that $\mathcal{F}_\gamma = \{X_{ij}\}_{i,j}$ as defined in Section 4.1. For simplicity we assume that all the images are the same size so N does not depend on i.

Each sub-image has 16 pixels with 3 color values and so we can represent it as a 48 dimensional vector. Each sub-image also has a 2-coordinate location in it's source image, which we denote (x_{ij}, y_{ij}). Following [2] we let $Y_{ij} = (X_{ij}, x_{ij}, y_{ij})$ be a 50-dimensional vector created by appending the sub-image location to the 50-dimensional representation of each respective sub-image. Next we define a distance on the space of sub images by

$$d(Y_{ij}, Y_{kl}) = ||X_{ij} - X_{kl}||_2 + \sigma\sqrt{(x_{ij} - x_{kl})^2 + (y_{ij} - y_{kl})^2}$$

We will explore the role of the parameter σ in later sections. For convenience, we choose an ordering of the sub-images $\{Y_r\}$ using a single index.

To define a diffusion operator on the space of images $\mathcal{A}$ we first define the operator on the set of sub-images $\mathcal{F}_\gamma$. Essentially we define a random walk on the space of sub-images, so the the translation probability between any two sub-images is proportional

to $\exp\{-d(Y_r, Y_s)/\kappa\}$. Note that when the distance $d(Y_{ij}, Y_{kl})$ is large the probability is essentially zero, thus we can save memory by defining a sparse matrix T such that $T_{rs} = \exp\{-d(Y_r, Y_s)/\kappa\}$ for s ranging over the k-nearest neighbors of Y_r and $T_{rs} = 0$ otherwise.

Note that as long as the set of sub-images $\{Y_s\}$ is a frame for the set of images, we can use T to define a multi-scale distance on the set of all images by setting

$$d_s(A_i, A_k) = ||T^q G f_i - T^q G f_k||_2$$

where f_i, f_j the function representations of the images A_i and A_j respectively and G is the frame operator as defined in Section 4.1. In the next section we use the diffusion operator on the space of images to build Diffusion Wavelets using the technique from Section 2.

4.3 Wavelets for Spaces of Images

With the above construction of a diffusion operator T we can proceed according to the construction in Section 2 to find Diffusion Wavelets. In general these wavelets will depend on the size of the sub-images (or more generally on the window γ defined in Section 4.1) and also on the parameter σ defined in Section 4.2. In Figures 12 and 13 we show examples of scaling functions and wavelets.

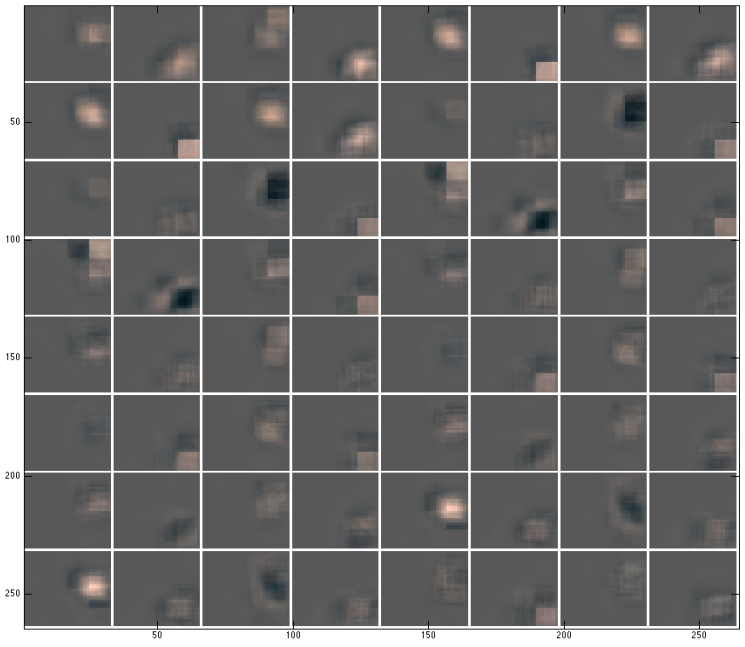

Figure 12: Diffusion Scaling Functions on the space of face images.

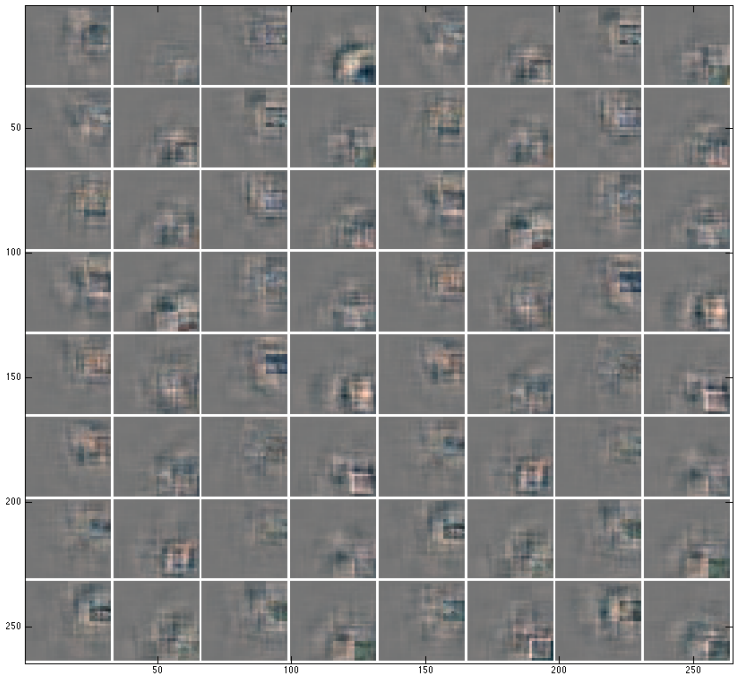

Figure 13: Diffusion Wavelets on the space of face images.

5 Bibliography

[1] R. Coifman and S. Lafon, *Diffusion Maps*, Applied and Computational Harmonic Analysis **21** (2006), 5-30.

[2] R. Coifman and M. Maggioni, *Diffusion Wavelets*, Applied and Computational Harmonic Analysis **21** (2006), 53-94.

[3] R. Coifman, J. Bremer, M. Maggioni, and A. Szlam, *Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs*, Proc. SPIE 5914 (2005).

[4] A. Szlam, R. Coifman, J. Bremer, and M. Maggioni, *Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions*, Proc. SPIE 5914 (2005).

[5] V. Garcia and E. Debreuve and M. Barlaud, *Fast k nearest neighbor search using GPU*, CVPR Workshop on Computer Vision on GPU, 2008.

[6] Gary B. Huang and Manu Ramesh and Tamara Berg and Erik Learned-Miller, *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments* **07-49** (2007).