Secure Data Hiding in Wavelet Compressed Fingerprint Images

A paper by N. Ratha, J. Connell, and R. Bolle

1 November, 2006

Matthew Goldfield

<mvg@cs.brandeis.edu>

http://www.cs.brandeis.edu/~mvg/

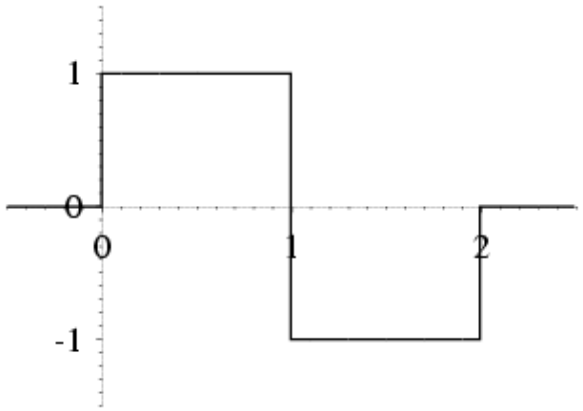
Motivation

- Biological identification is becoming more prevalent as a security measure.

- If someone is able to intercept a fingerprint, they could impersonate the identity of person to whom the fingerprint belongs.

- This paper aims to add security to the transmission of these images by adding hidden information such as a timestamp.

Encoding with the Haar wavelet



The Haar wavelet was the first wavelet to be used. It essentially down-samples an image through averaging sets of values. Note that while it is a wavelet, it is not differentiable, and therefore does not have many desirable properties.

The In-Place Haar Wavelet Transform

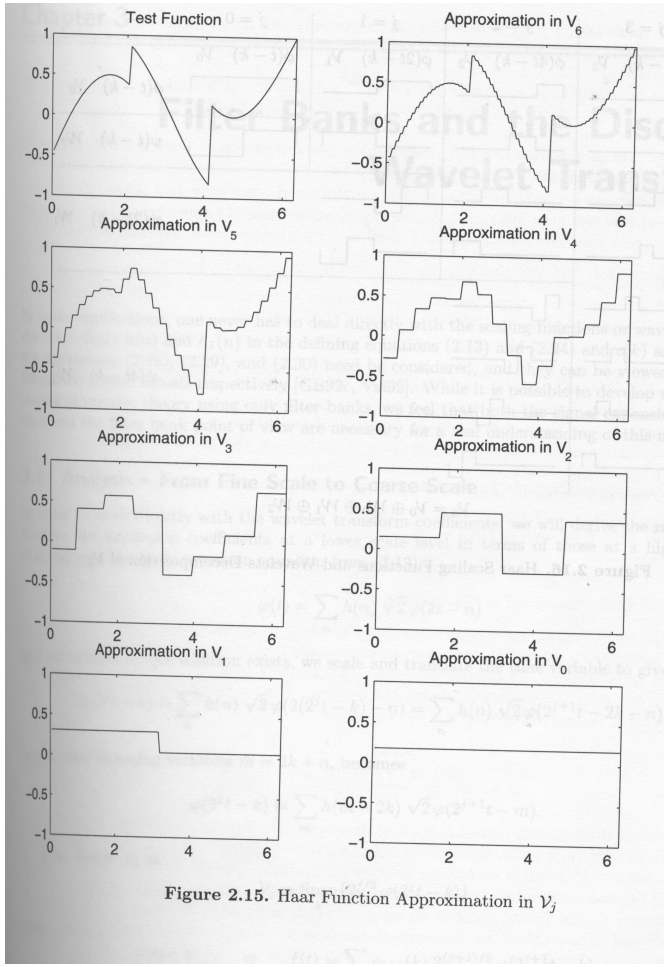
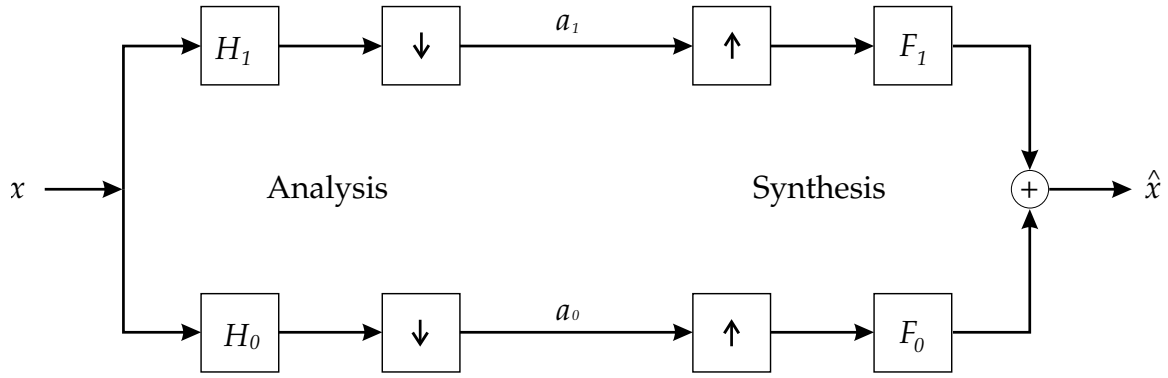


Figure 2.15. Haar Function Approximation in $\mathcal{V}_j$

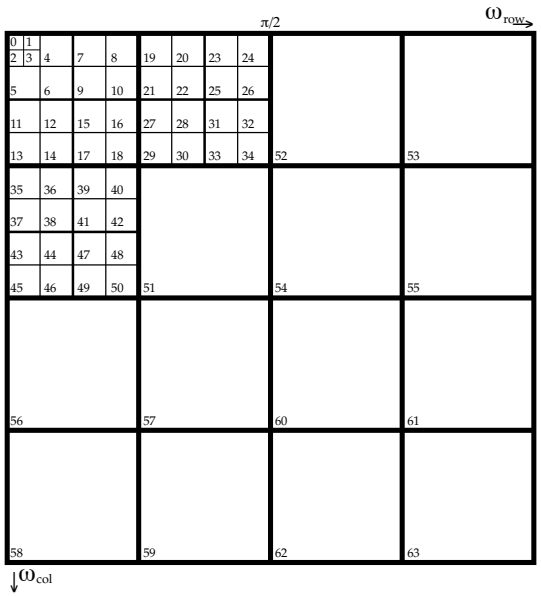
for $(i = 0; i < n; i = i + 2)$
$s_i = (v_i + v_{i+1})/2;$

We save the average of the two quantized pieces and the difference between the average and the original. We can then use the haar transform again on the newly computed averages. Note that for this to work, the data must be of size 2^n for some n. The ususal fix for this is to pad the data with 0's to make it into the correct shape.

Filter Banks Etc.



Filter Bank Transformed Data



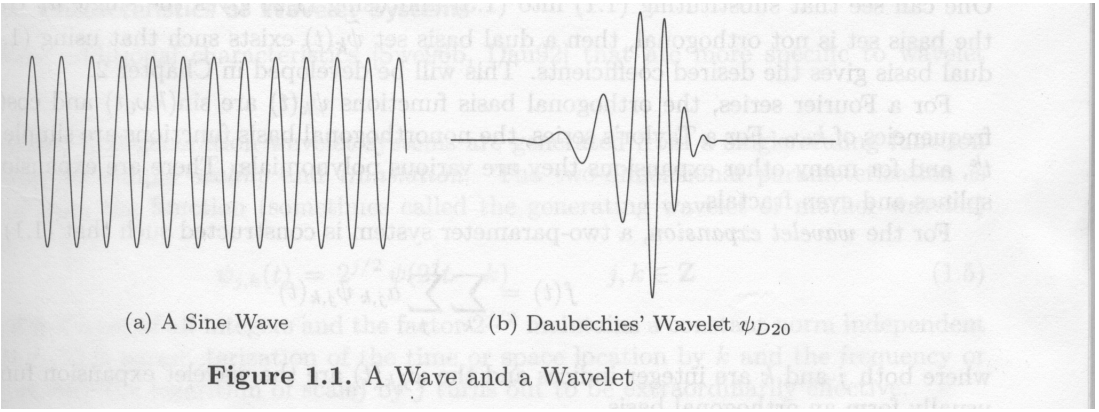
Filter Bank Path

0	00,00,00,00,00	22	00,10,10,11	44	00,01,00,11
1	00,00,00,00,10	23	00,10,00,10	45	00,01,00,00
2	00,00,00,00,01	24	00,10,00,00	46	00,01,00,10
3	00,00,00,00,11	25	00,10,00,11	47	00,01,10,11
4	00,00,00,10	26	00,10,00,01	48	00,01,10,01
5	00,00,00,01	27	00,10,11,01	49	00,01,10,10
6	00,00,00,11	28	00,10,11,11	50	00,01,10,00
7	00,00,10,10	29	00,10,11,00	51	00,11
8	00,00,10,00	30	00,10,11,10	52	10,10
9	00,00,10,11	31	00,10,01,11	53	10,00
10	00,00,10,01	32	00,10,01,01	54	10,11
11	00,00,01,01	33	00,10,01,10	55	10,01
12	00,00,01,11	34	00,10,01,00	56	01,01
13	00,00,01,00	35	00,01,01,00	57	01,11
14	00,00,01,10	36	00,01,01,10	58	01,00
15	00,00,11,11	37	00,01,01,01	59	01,10
16	00,00,11,01	38	00,01,01,11	60	11,11
17	00,00,11,10	39	00,01,11,10	61	11,01
18	00,00,11,00	40	00,01,11,00	62	11,10
19	00,10,10,00	41	00,01,11,11	63	11,00
20	00,10,10,10	42	00,01,11,01		
21	00,10,10,01	43	00,01,00,01		

Figure A.5 Subband Decomposition

What is a wavelet

$$k \in \mathbb{Z} \quad \psi(t) = \psi(t-k)$$
$$\psi_{j,k}(t) = 2^{\frac{j}{2}}\psi(2^j t - k)$$



(a) A Sine Wave (b) Daubechies' Wavelet ψ_{D20}

Figure 1.1. A Wave and a Wavelet

We define a wavelet space $W_j = \overline{span_k\{\psi_{j,k}\}}$

What is an orthonormal basis

Def: **Inner Product**
A mapping from the space^2 to the base field.

In $\mathbb{R}^2$ space:
$< (a, b), (c, d) >= (a * b) + (c * d)$

In function space (or more specifically wavelet space), we define inner products as
$< f(x), g(x) >= \int f(x)g(x)dx.$

What is an orthonormal basis II

A **basis** of a space is a set of elements of the space that, closed over linear combination, make up the entire space.
Example: $\mathbb{R}^2$
$< 1, 0 >< 0, 1 >$

An **Orthonormal Basis** is a basis where the inner product of any two elements is zero. i.e. there is no redundancy in the basis. Example: $\mathbb{R}^2$
$< 1, 0 >< 0, 1 >$ is orthonormal.
$< 1, 1 >< 1, 0 >$ is not orthonormal, but it is still a basis.

A wavelet system is an orthonormal basis

Reminder: $< f(x), g(x) >= \int f(x)g(x)dx$

- For any two wavelets in a wavelet space, their inner product is zero.
- A lossless transform with might take infinitely many wavelets that are infinitely detailed (just like the DCT).
- With no "redundancy", each level of accuracy we add in a wavelet transform returns a maximum amount of information.

Encoding the Houston skyline

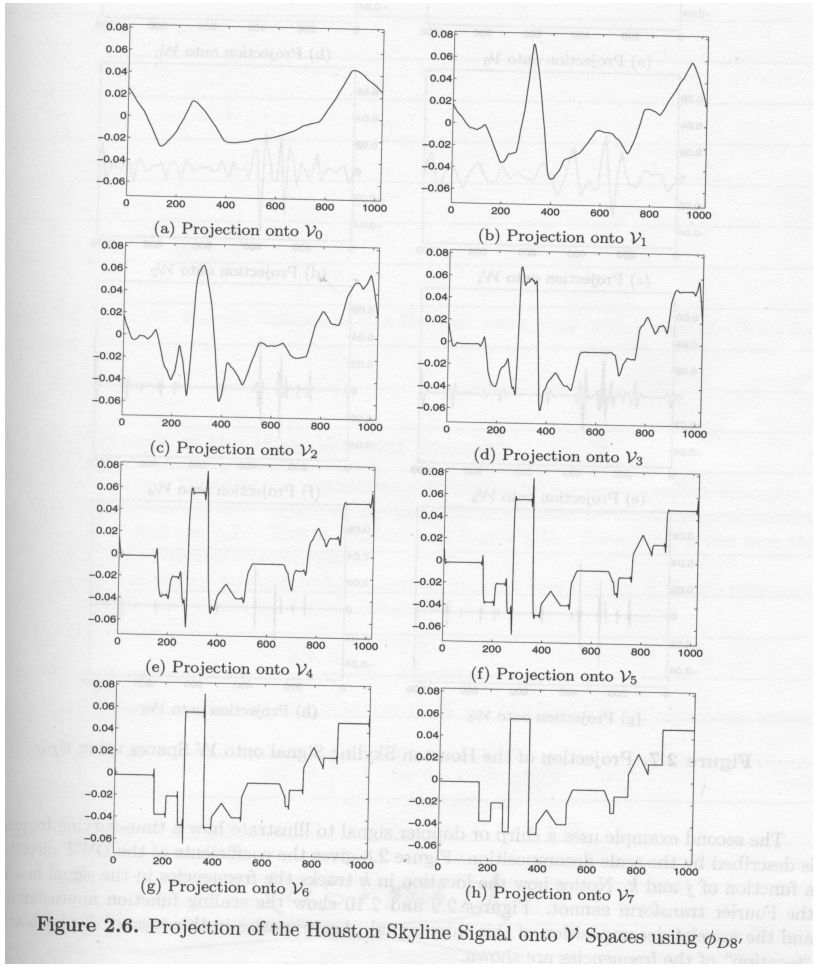


Figure 2.6. Projection of the Houston Skyline Signal onto $\mathcal{V}$ Spaces using $\phi_{D8'}$

The Daubechies is the foundation for most signal processing wavelet applications because if its continuity and differentiability.

Wavelet Resolutions for the Houston Skyline

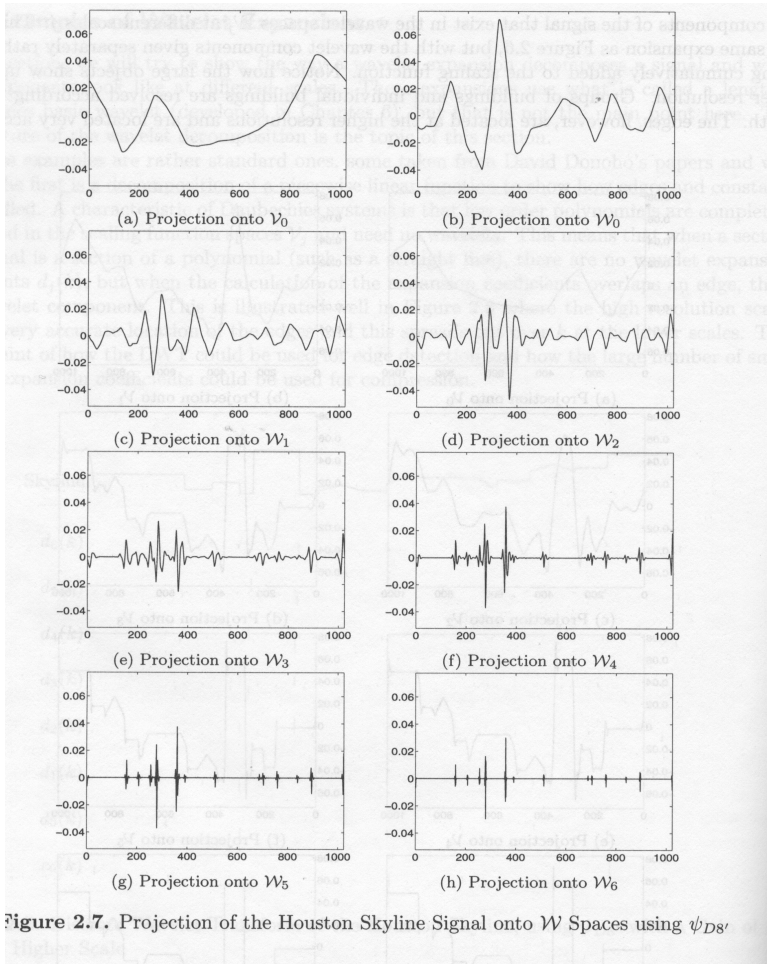
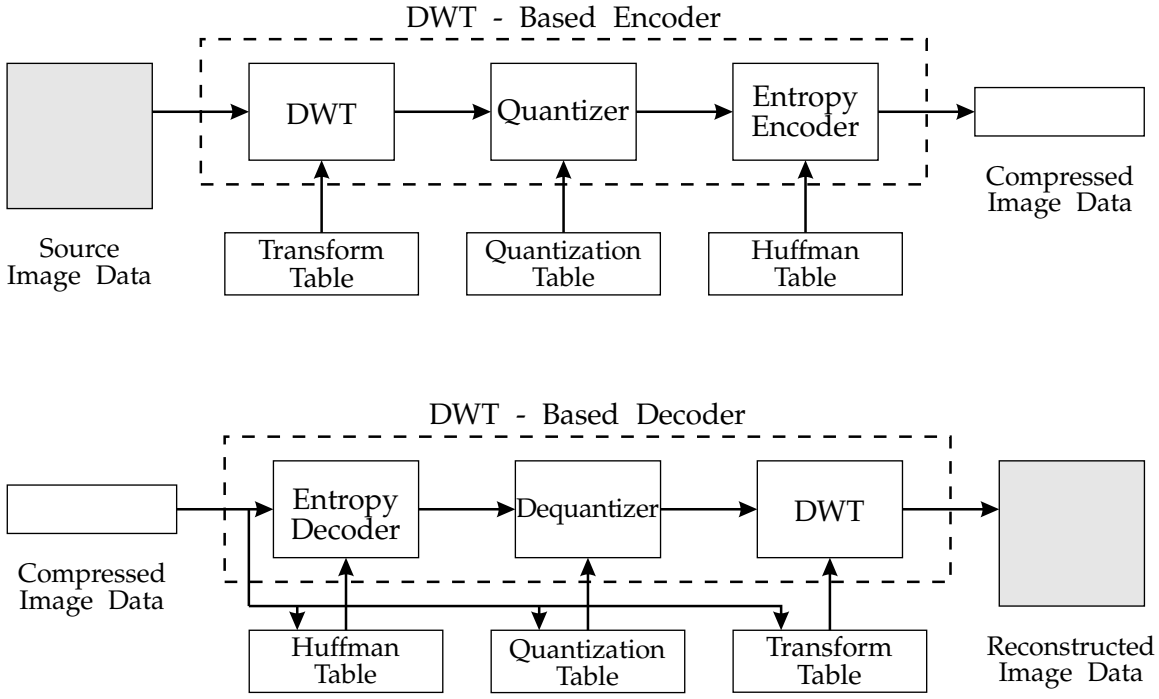


Figure 2.7. Projection of the Houston Skyline Signal onto $\mathcal{W}$ Spaces using $\psi_{D8'}$

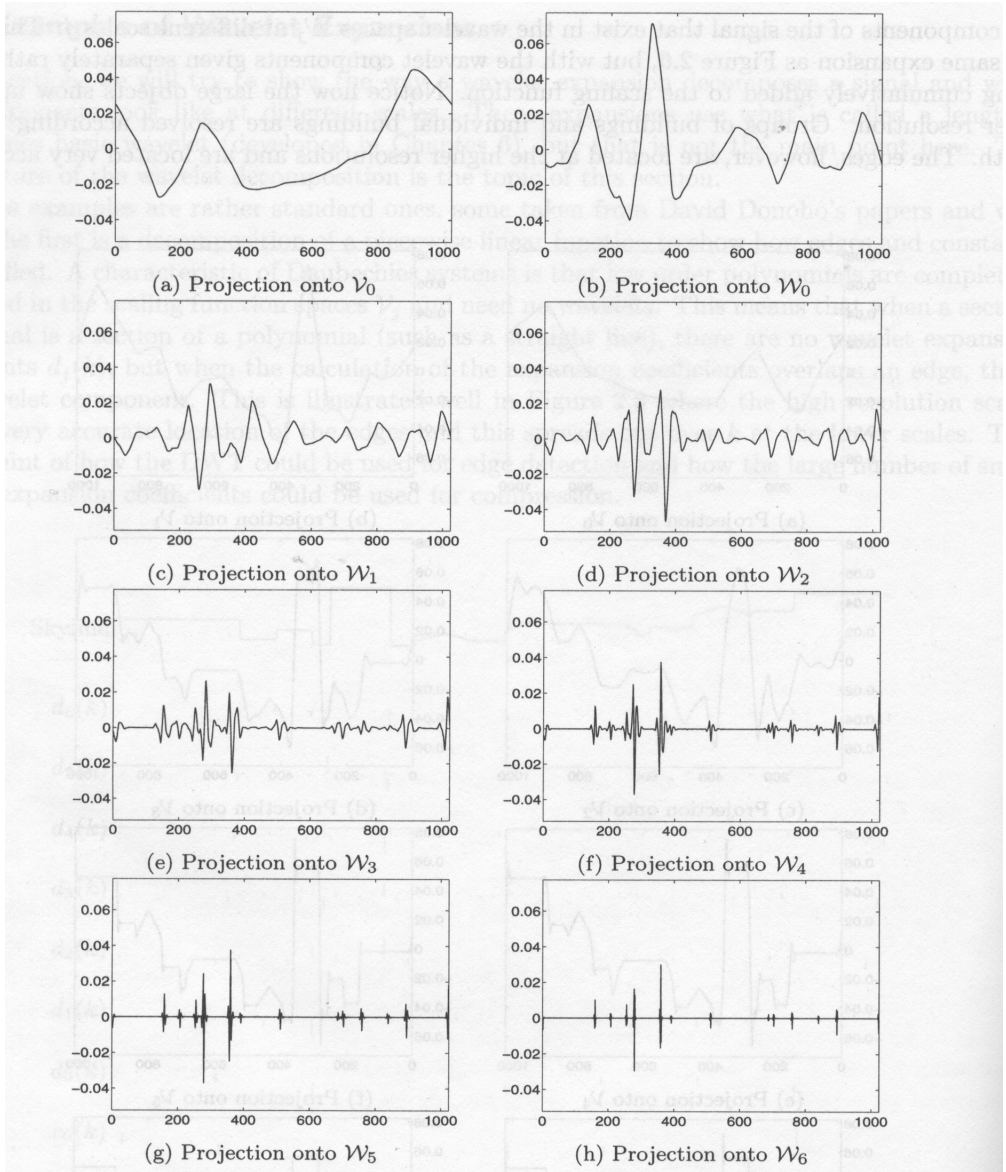
These are the building blocks that made up the images in the last slide.

FBI Wavelet Fingerprint Encoding Spec.



Important

It is assumed that the size of the message to be encoded is on the order of the number of DWT coefficients, i.e. small.



(a) Projection onto $\mathcal{V}_0$

(b) Projection onto $\mathcal{W}_0$

(c) Projection onto $\mathcal{W}_1$

(d) Projection onto $\mathcal{W}_2$

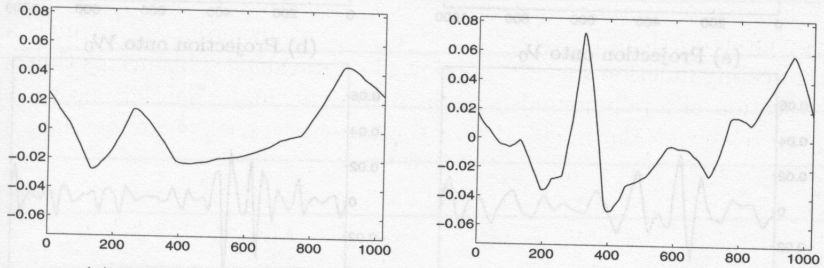
(e) Projection onto $\mathcal{W}_3$

(f) Projection onto $\mathcal{W}_4$

(g) Projection onto $\mathcal{W}_5$

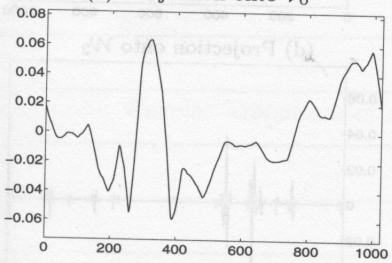
(h) Projection onto $\mathcal{W}_6$

Figure 2.7. Projection of the Houston Skyline Signal onto $\mathcal{W}$ Spaces using $\psi_{D8'}$

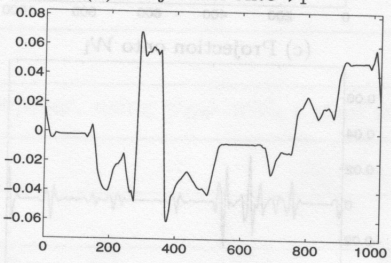


(a) Projection onto $\mathcal{V}_0$

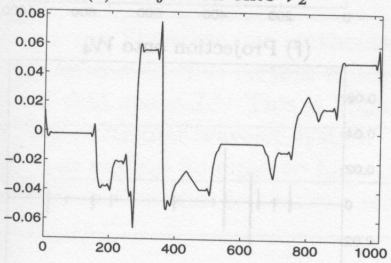
(b) Projection onto $\mathcal{V}_1$



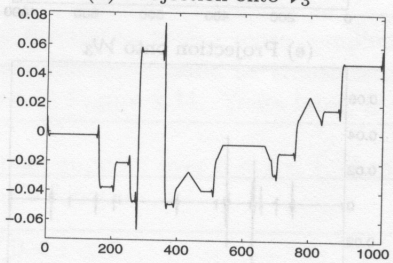
(c) Projection onto $\mathcal{V}_2$



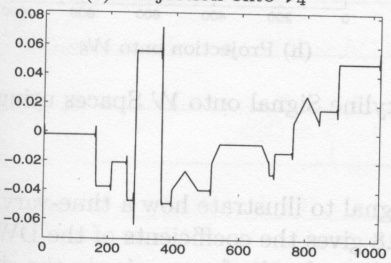
(d) Projection onto $\mathcal{V}_3$



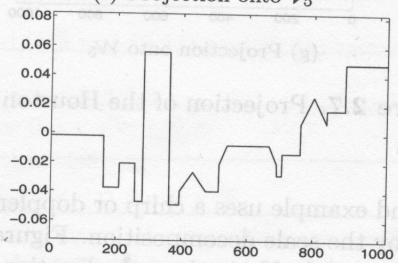
(e) Projection onto $\mathcal{V}_4$



(f) Projection onto $\mathcal{V}_5$



(g) Projection onto $\mathcal{V}_6$



(h) Projection onto $\mathcal{V}_7$

Figure 2.6. Projection of the Houston Skyline Signal onto $\mathcal{V}$ Spaces using $\phi_{D8'}$

The Algorithm

The algorithm systematically identifies low-order bits in high magnitude wavelet components that can be flipped with relatively little effect on the image. The algorithm is run in tandem with the wavelet transform and before any (lossless) compression techniques are applied.

The algorithm has relatively little mathematical basis. The idea behind it is straightforward, and it is verified experimentally.

Step 1: Site Selection

We want to avoid the low-frequency bands, since they have the most overall effect on the image. We set variable $Order$ to be some acceptable cutoff depending on how exactly we have encoded the image. For the implementation in the paper, $Order = 6$

Site Selection Continued

Similarly we would like to only change high-magnitude bits, as any changes in this area will have the least percent of overall effect on the image. Hence we set a variable $Magnitude$ to be a cutoff, only accepting sites of magnitude $\geq$ $Magnitude$. In the implementation from the paper, $Magnitude = 107$ (out of 256).

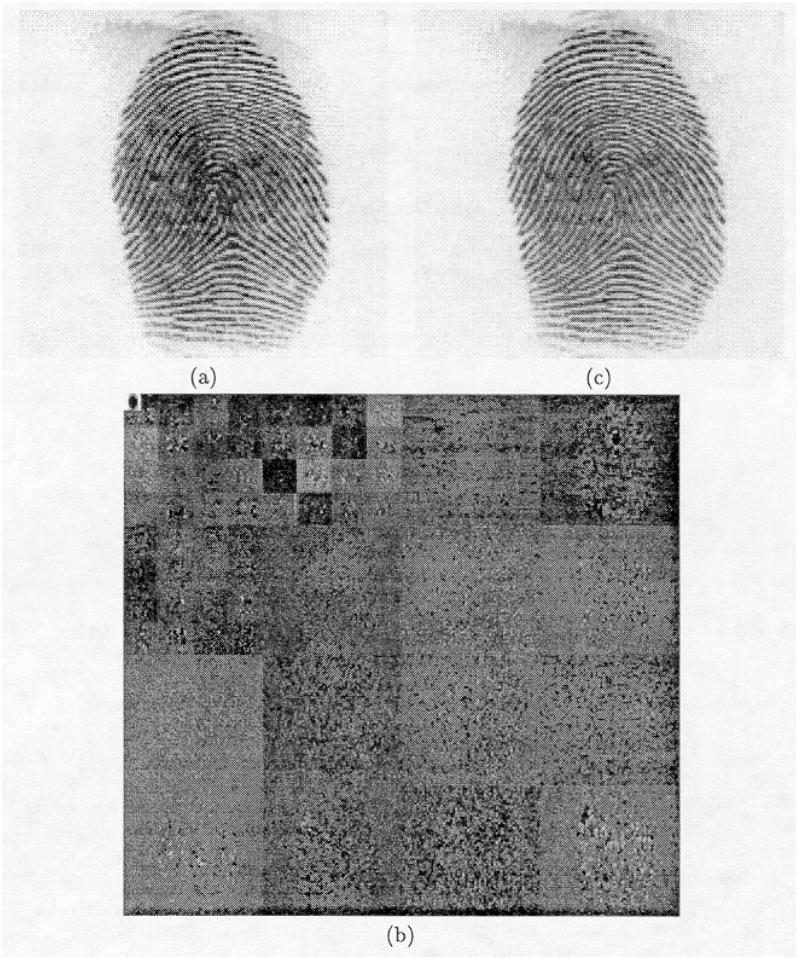
Step 2: Seeding

Sites are then modified in a pseudo-random order. We use a random number generator which generates numbers based on the sub-bands unsuitable for modification. The bits are then changed in suitable sites chosen in that order. If a number comes up that has been already used, the algorithm automatically moves on to the next number.

Step 3: Optional Bit Saving

Because of the pseudo-randomness of this algorithm, one can actually save the changed bits and add them as an appendix to the initial transmission. They are essentially random samples of the image and hence are uncorrelated with the message being transmitted.

Results



(a) The initial fingerprint image
(b) The subbands of the wavelet transform
(c) The reconstructed image

Bibliography

C. Burrus, R. Gopinath, and H. Guo. Wavelets and Wavelet Transforms: A Primer. Prentice Hall, Upper Saddle River, New Jersey, 1998.

N. Ratha and J. Connell and R. Bolle. Secure Data Hiding in Wavelet Compressed Fingerprint Images. In ACM Multimedia Workshop 2000.

WSQ Gray-scale Fingerprint Image Compression Specification.