Wavelets Compression

Ronald DeVore

What is an Image?

- A function f defined on $\Omega := [0,1]^2$

What is an Image?

- A function f defined on $\Omega := [0,1]^2$
- Digitized Image?

What is an Image?

- A function f defined on $\Omega := [0,1]^2$
- Digitized Image?
- $\mathcal{D}_k$ the set of all dyadic squares of sidelength 2^{-k} contained in Ω.

What is an Image?

- A function f defined on $\Omega := [0,1]^2$
- Digitized Image?
- $\mathcal{D}_k$ the set of all dyadic squares of sidelength 2^{-k} contained in Ω.
- $f_I := \frac{1}{|I|}\int_I f$

What is an Image?

- A function f defined on $\Omega := [0,1]^2$
- Digitized Image?
- $\mathcal{D}_k$ the set of all dyadic squares of sidelength 2^{-k} contained in Ω.
- $f_I := \frac{1}{|I|} \int_I f$
- Pixel values p_I: some quantization of the f_I

What is an Image?

- A function f defined on $\Omega := [0,1]^2$
- Digitized Image?
- $\mathcal{D}_k$ the set of all dyadic squares of sidelength 2^{-k} contained in Ω.
- $f_I := \frac{1}{|I|} \int_I f$
- Pixel values p_I: some quantization of the f_I
- For example the first m binary digits of f_I

Real world images

- The above mathematical definition is too general

Real world images

- The above mathematical definition is too general
- It includes many examples that would not be considered images

Real world images

- The above mathematical definition is too general
- It includes many examples that would not be considered images
- Real world images are far more special

Real world images

- The above mathematical definition is too general
- It includes many examples that would not be considered images
- Real world images are far more special
- If we could give a precise mathematical definition of real world images we would probably be able to find (near) optimal image processing algorithms

Real world images

- The above mathematical definition is too general
- It includes many examples that would not be considered images
- Real world images are far more special
- If we could give a precise mathematical definition of real world images we would probably be able to find (near) optimal image processing algorithms
- Images have an inherent complexity and should be classified according to this complexity

Real world images

- The above mathematical definition is too general
- It includes many examples that would not be considered images
- Real world images are far more special
- If we could give a precise mathematical definition of real world images we would probably be able to find (near) optimal image processing algorithms
- Images have an inherent complexity and should be classified according to this complexity
- This classification will determine how well we can do in such tasks as compression and denoising

Compression: Encoders/Decoders

- An encoder E is a mapping $f \to B$ where $B := B(f)$ is a bitstream

Compression: Encoders/Decoders

- An encoder E is a mapping $f \to B$ where $B := B(f)$ is a bitstream
- We view B as an infinite bitstream: progressive encoding

Compression: Encoders/Decoders

- An encoder E is a mapping $f \to B$ where $B := B(f)$ is a bitstream
- We view B as an infinite bitstream: progressive encoding
- For digitized images B will be finite

Compression: Encoders/Decoders

- An encoder E is a mapping $f \to B$ where $B := B(f)$ is a bitstream
- We view B as an infinite bitstream: progressive encoding
- For digitized images B will be finite
- Define $E_n(f)$ as the first n bits of $B(f)$

Compression: Encoders/Decoders

- An encoder E is a mapping $f \to B$ where $B := B(f)$ is a bitstream
- We view B as an infinite bitstream: progressive encoding
- For digitized images B will be finite
- Define $E_n(f)$ as the first n bits of $B(f)$
- Decoders: D_n maps a bitstream B_n of size n into a function (image) $D_n(B_n)$

Compression: Encoders/Decoders

- An encoder E is a mapping $f \to B$ where $B := B(f)$ is a bitstream
- We view B as an infinite bitstream: progressive encoding
- For digitized images B will be finite
- Define $E_n(f)$ as the first n bits of $B(f)$
- Decoders: D_n maps a bitstream B_n of size n into a function (image) $D_n(B_n)$
- $D_n(E_n(f))$ is our compressed image

The Issues

1. The metric: We shall employ typical compression convention and use least squares metric, i.e. $L_2([0,1]^2)$. This discrete form of this metric is equivalent to PSNR

2. The classes

3. Determine Entropy of Classes

4. Build near optimal Encoders/Decoders

The classes

- K must be compact in $L_2[0,1]^2)$

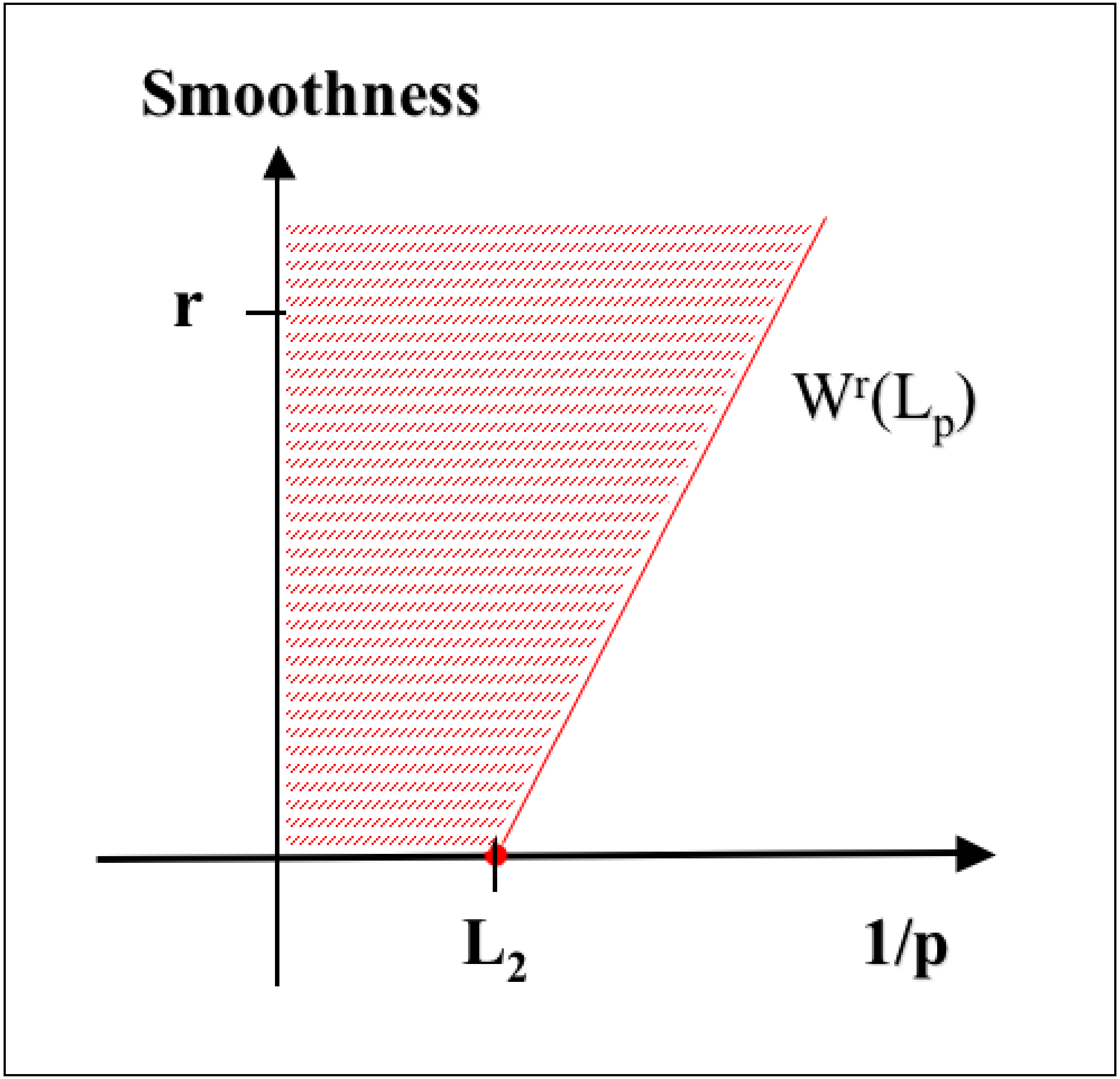
The classes

- K must be compact in $L_2[0,1]^2)$
- Smoothness spaces: Sobolev $W^s(L_p)$and Besov classes $B^s_q(L_p))$

The classes

- K must be compact in $L_2[0,1]^2)$
- Smoothness spaces: Sobolev $W^s(L_p)$and Besov classes $B^s_q(L_p))$
- BV - functions of Bounded Variation: $s = 1\ p = 1$

Sobolev embedding $d = 2$



Wavelets on $\mathbb{R}$

- In $d = 1$, an orthogonal wavelet is a functions ψ whose shifted dilates are a complete orthonormal system for $L_2(\mathbb{R})$

Wavelets on $\mathbb{R}$

- In $d = 1$, an orthogonal wavelet is a functions ψ whose shifted dilates are a complete orthonormal system for $L_2(\mathbb{R})$
- $\psi_I(x) := 2^{j/2}\psi(2^j x - k)$ with $I = 2^{-j}[k, k+1]$

Wavelets on $\mathbb{R}$

- In $d = 1$, an orthogonal wavelet is a functions ψ whose shifted dilates are a complete orthonormal system for $L_2(\mathbb{R})$
- $\psi_I(x) := 2^{j/2}\psi(2^j x - k)$ with $I = 2^{-j}[k, k+1]$
- Simplest example is the Haar function $\psi := \chi_{[0,1/2]} - \chi_{[1/2,1]}$

Wavelets on $\mathbb{R}$

- In $d = 1$, an orthogonal wavelet is a functions ψ whose shifted dilates are a complete orthonormal system for $L_2(\mathbb{R})$
- $\psi_I(x) := 2^{j/2}\psi(2^j x - k)$ with $I = 2^{-j}[k, k+1]$
- Simplest example is the Haar function $\psi := \chi_{[0,1/2]} - \chi_{[1/2,1]}$
- Zillion other examples: most famous Daubechies compactly supported wavelets

Wavelets on $\mathbb{R}$

- In $d = 1$, an orthogonal wavelet is a functions ψ whose shifted dilates are a complete orthonormal system for $L_2(\mathbb{R})$
- $\psi_I(x) := 2^{j/2}\psi(2^j x - k)$ with $I = 2^{-j}[k, k+1]$
- Simplest example is the Haar function $\psi := \chi_{[0,1/2]} - \chi_{[1/2,1]}$
- Zillion other examples: most famous Daubechies compactly supported wavelets
- Scaling function φ and associated mother wavelet ψ

Wavelets on $\mathbb{R}$

- In $d = 1$, an orthogonal wavelet is a functions ψ whose shifted dilates are a complete orthonormal system for $L_2(\mathbb{R})$
- $\psi_I(x) := 2^{j/2}\psi(2^j x - k)$ with $I = 2^{-j}[k, k+1]$
- Simplest example is the Haar function $\psi := \chi_{[0,1/2]} - \chi_{[1/2,1]}$
- Zillion other examples: most famous Daubechies compactly supported wavelets
- Scaling function φ and associated mother wavelet ψ
- For Haar $\varphi = \chi_{[0,1]}$

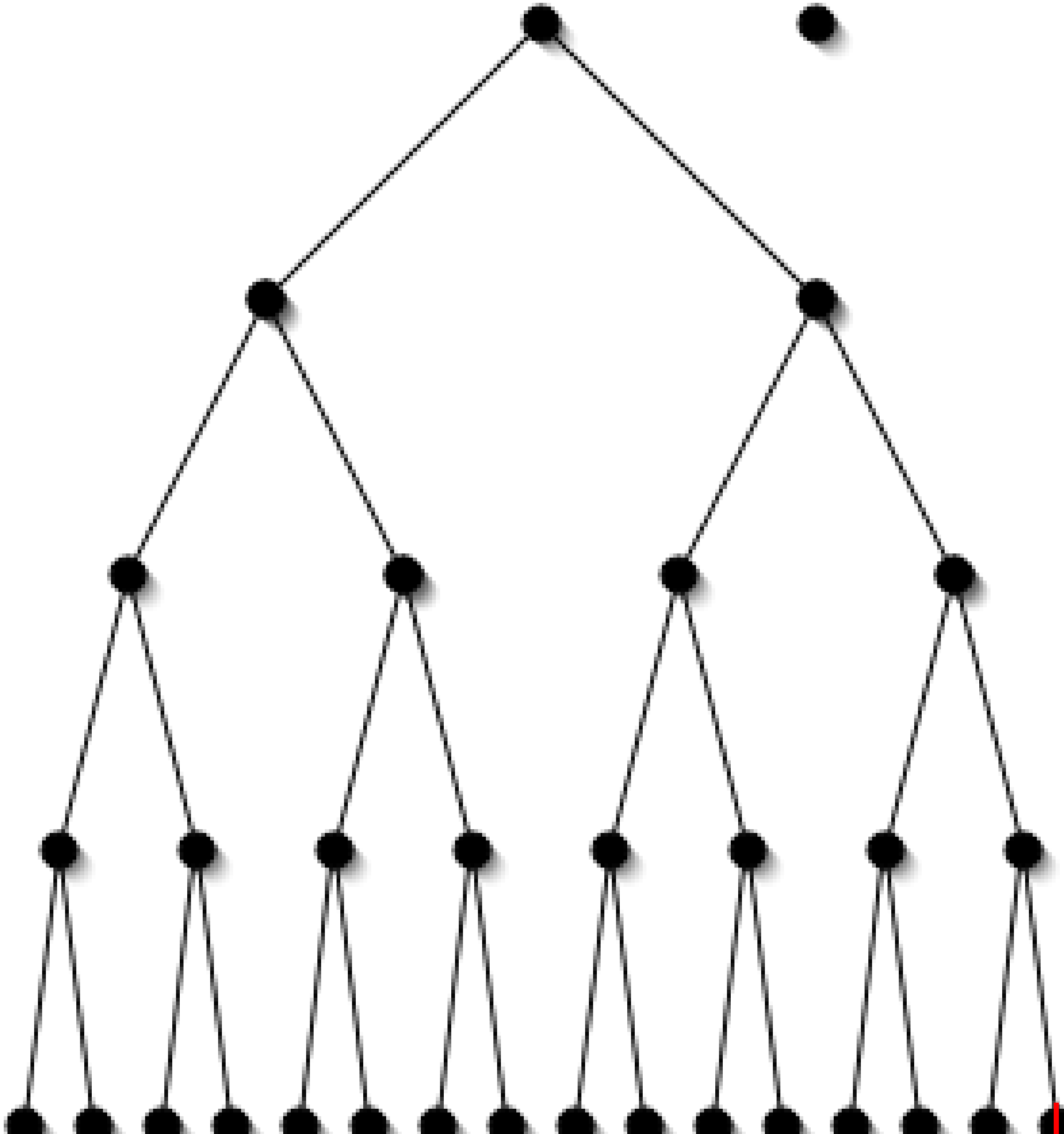
Wavelets on $\mathbb{R}$

- In $d = 1$, an orthogonal wavelet is a functions ψ whose shifted dilates are a complete orthonormal system for $L_2(\mathbb{R})$
- $\psi_I(x) := 2^{j/2}\psi(2^j x - k)$ with $I = 2^{-j}[k, k+1]$
- Simplest example is the Haar function $\psi := \chi_{[0,1/2]} - \chi_{[1/2,1]}$
- Zillion other examples: most famous Daubechies compactly supported wavelets
- Scaling function φ and associated mother wavelet ψ
- For Haar $\varphi = \chi_{[0,1]}$
- On finite domain different basis: scaling functions on level 0 wavelets on level j for each j

Wavelets on $\mathbb{R}$

- In $d = 1$, an orthogonal wavelet is a functions ψ whose shifted dilates are a complete orthonormal system for $L_2(\mathbb{R})$
- $\psi_I(x) := 2^{j/2}\psi(2^j x - k)$ with $I = 2^{-j}[k, k+1]$
- Simplest example is the Haar function $\psi := \chi_{[0,1/2]} - \chi_{[1/2,1]}$
- Zillion other examples: most famous Daubechies compactly supported wavelets
- Scaling function φ and associated mother wavelet ψ
- For Haar $\varphi = \chi_{[0,1]}$
- On finite domain different basis: scaling functions on level 0 wavelets on level j for each j
- Indexed on dyadic intervals: binary tree

Wavelet Tree



Wavelets for $d = 2$

- Wavelet basis from univariate basis

Wavelets for $d = 2$

- Wavelet basis from univariate basis
- Describe how we get a bivariate basis: Haar example

Wavelets for $d = 2$

- Wavelet basis from univariate basis
- Describe how we get a bivariate basis: Haar example
- Let $\psi_I^0 = \varphi_I$ and $\psi_I^1 := \psi_I$

Wavelets for $d = 2$

- Wavelet basis from univariate basis
- Describe how we get a bivariate basis: Haar example
- Let $\psi_I^0 = \varphi_I$ and $\psi_I^1 := \psi_I$
- If $I = I_0 \times I_1$ is a square and $e = (e_0, e_1)$ $e_0, e_1 \in \{0, 1\}$: $I = 2^{-j}[k_0, k_0 + 1] \times [k_1, k_1 + 1]$ a dyadic cube centered at $2^{-j}(k_0, k_1)$

Wavelets for $d = 2$

- Wavelet basis from univariate basis
- Describe how we get a bivariate basis: Haar example
- Let $\psi_I^0 = \varphi_I$ and $\psi_I^1 := \psi_I$
- If $I = I_0 \times I_1$ is a square and $e = (e_0, e_1)$ $e_0, e_1 \in \{0, 1\}$: $I = 2^{-j}[k_0, k_0 + 1] \times [k_1, k_1 + 1]$ a dyadic cube centered at $2^{-j}(k_0, k_1)$
- $\psi_I^e = \Psi_{I_0}^{e_0} \psi_{I_1}^{e_1}$

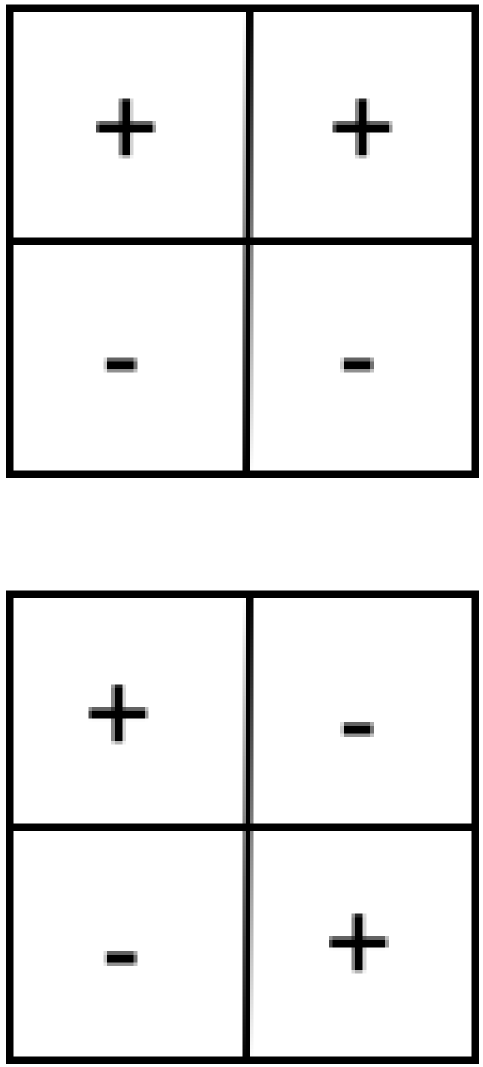
Wavelets for $d = 2$

- Wavelet basis from univariate basis
- Describe how we get a bivariate basis: Haar example
- Let $\psi_I^0 = \varphi_I$ and $\psi_I^1 := \psi_I$
- If $I = I_0 \times I_1$ is a square and $e = (e_0, e_1)$ $e_0, e_1 \in \{0, 1\}$: $I = 2^{-j}[k_0, k_0 + 1] \times [k_1, k_1 + 1]$ a dyadic cube centered at $2^{-j}(k_0, k_1)$
- $\psi_I^e = \Psi_{I_0}^{e_0} \psi_{I_1}^{e_1}$
- There are three wavelets associated to each dyadic square: $e = (1, 0), (0, 1), (1, 1)\}$

Wavelets for $d = 2$

- Wavelet basis from univariate basis
- Describe how we get a bivariate basis: Haar example
- Let $\psi_I^0 = \varphi_I$ and $\psi_I^1 := \psi_I$
- If $I = I_0 \times I_1$ is a square and $e = (e_0, e_1)\ e_0, e_1 \in \{0, 1\}$: $I = 2^{-j}[k_0, k_0 + 1] \times [k_1, k_1 + 1]$ a dyadic cube centered at $2^{-j}(k_0, k_1)$
- $\psi_I^e = \Psi_{I_0}^{e_0} \psi_{I_1}^{e_1}$
- There are three wavelets associated to each dyadic square: $e = (1, 0), (0, 1), (1, 1)\}$
- On a finite domain there are four wavelets at the starting level and three on each other level

Example: Haar Functions



Description of Sobolev-Besov Classes

- Consider classes on Sobolev Embedding Line

Description of Sobolev-Besov Classes

- Consider classes on Sobolev Embedding Line
- W_p^s with $1/p = s/2 + 1/2$

Description of Sobolev-Besov Classes

- Consider classes on Sobolev Embedding Line
- W_p^s with $1/p = s/2 + 1/2$
- Each $f \in W_p^s$ has a wavelet decomposition

Description of Sobolev-Besov Classes

- Consider classes on Sobolev Embedding Line
- W_p^s with $1/p = s/2 + 1/2$
- Each $f \in W_p^s$ has a wavelet decomposition
- $f = \sum_{\mu \in \Lambda} c_\mu(f) \psi_\mu$

Description of Sobolev-Besov Classes

- Consider classes on Sobolev Embedding Line
- W_p^s with $1/p = s/2 + 1/2$
- Each $f \in W_p^s$ has a wavelet decomposition
- $f = \sum_{\mu \in \Lambda} c_\mu(f)\psi_\mu$
- $\|f\|_{W_p^s} \approx \left(\sum_{\mu \in \Lambda} |c_\mu(f)|^p\right)^{1/p}$

Description of Sobolev-Besov Classes

- Consider classes on Sobolev Embedding Line
- W_p^s with $1/p = s/2 + 1/2$
- Each $f \in W_p^s$ has a wavelet decomposition
- $f = \sum_{\mu \in \Lambda} c_\mu(f)\psi_\mu$
- $\|f\|_{W_p^s} \approx \left(\sum_{\mu \in \Lambda} |c_\mu(f)|^p\right)^{1/p}$
- Notice that $p < 2$ so that $(c_\mu) \in \ell_p$ is a sparsity condition.

Description of Sobolev-Besov Classes

- Consider classes on Sobolev Embedding Line
- W^s_p with $1/p = s/2 + 1/2$
- Each $f \in W^s_p$ has a wavelet decomposition
- $f = \sum_{\mu \in \Lambda} c_\mu(f) \psi_\mu$
- $\|f\|_{W^s_p} \approx \left(\sum_{\mu \in \Lambda} |c_\mu(f)|^p \right)^{1/p}$
- Notice that $p < 2$ so that $(c_\mu) \in \ell_p$ is a sparsity condition.
- A sequence n^{-s} is in ℓ_2 if and only if $s > 1/2$

Description of Sobolev-Besov Classes

- Consider classes on Sobolev Embedding Line
- W_p^s with $1/p = s/2 + 1/2$
- Each $f \in W_p^s$ has a wavelet decomposition
- $f = \sum_{\mu \in \Lambda} c_\mu(f) \psi_\mu$
- $\|f\|_{W_p^s} \approx \left(\sum_{\mu \in \Lambda} |c_\mu(f)|^p \right)^{1/p}$
- Notice that $p < 2$ so that $(c_\mu) \in \ell_p$ is a sparsity condition.
- A sequence n^{-s} is in ℓ_2 if and only if $s > 1/2$
- The same sequence is in ℓ_1 if and only if $s > 1$

Wavelets compression: main ideas

- Fix n the number of terms we want to keep.

Wavelets compression: main ideas

- Fix n the number of terms we want to keep.
- Keep the n terms with largest coefficients.

$$S = \sum_{I \in \Lambda} c_I(f)\psi_I, \quad \#(\Lambda) = n$$

Wavelets compression: main ideas

- Fix n the number of terms we want to keep.
- Keep the n terms with largest coefficients.

$$S = \sum_{I \in \Lambda} c_I(f)\psi_I, \quad \#(\Lambda) = n$$

- This is best approximation using n terms

Wavelets compression: main ideas

- Fix n the number of terms we want to keep.
- Keep the n terms with largest coefficients.

$$S = \sum_{I \in \Lambda} c_I(f)\psi_I, \quad \#(\Lambda) = n$$

- This is best approximation using n terms
- Threshold at η: retains terms with coefficients larger than η

$$S_\eta(f) = \sum_{I \in \Lambda_\eta(f)} c_I(f)\psi_I$$

$$\Lambda_\eta(f) := \{I : |c_I(f)| > \eta\}$$

An Example

- Suppose that $f \in U(B_1^1(L_1))$ then wavelet coefficients are in $U(\ell_1)$

An Example

- Suppose that $f \in U(B_1^1(L_1))$ then wavelet coefficients are in $U(\ell_1)$
- Let $c_n^*(f)$ the the rearrangement of the wavelet coefficients in absolute value

An Example

- Suppose that $f \in U(B_1^1(L_1))$ then wavelet coefficients are in $U(\ell_1)$
- Let $c_n^*(f)$ the the rearrangement of the wavelet coefficients in absolute value
- $c_{2n}^*(f) \leq \frac{1}{n} \sum_{k=n+1}^{2n} c_k^*(f) \leq \frac{1}{n}$

An Example

- Suppose that $f \in U(B^1_1(L_1))$ then wavelet coefficients are in $U(\ell_1)$
- Let $c^*_n(f)$ the the rearrangement of the wavelet coefficients in absolute value
- $c^*_{2n}(f) \le \frac{1}{n}\sum_{k=n+1}^{2n} c^*_k(f) \le \frac{1}{n}$
- $\Lambda_n := \{(I,e) : |\, c_{I,e}(f)| \text{ are } n \text{ largest}\}$

An Example

- Suppose that $f \in U(B_1^1(L_1))$ then wavelet coefficients are in $U(\ell_1)$
- Let $c_n^*(f)$ the the rearrangement of the wavelet coefficients in absolute value
- $c_{2n}^*(f) \leq \frac{1}{n}\sum_{k=n+1}^{2n} c_k^*(f) \leq \frac{1}{n}$
- $\Lambda_n := \{(I,e) : |\, c_{I,e}(f)| \text{ are } n \text{ largest}\}$
- The approximation S_{Λ_n} satisfies

$$\|f - S_{\Lambda_n}\|_{L_2}^2 = \sum_{n+1}^{\infty} c_k^*(f)^2 \leq \sum_{n+1}^{\infty} \frac{1}{k^2} \leq \frac{C}{n}$$

An Example

- Suppose that $f \in U(B_1^1(L_1))$ then wavelet coefficients are in $U(\ell_1)$
- Let $c_n^*(f)$ the the rearrangement of the wavelet coefficients in absolute value
- $c_{2n}^*(f) \leq \frac{1}{n}\sum_{k=n+1}^{2n} c_k^*(f) \leq \frac{1}{n}$
- $\Lambda_n := \{(I,e) : |\, c_{I,e}(f)| \text{ are } n \text{ largest}\}$
- The approximation S_{Λ_n} satisfies

$$\|f - S_{\Lambda_n}\|_{L_2}^2 = \sum_{n+1}^{\infty} c_k^*(f)^2 \leq \sum_{n+1}^{\infty} \frac{1}{k^2} \leq \frac{C}{n}$$

- The same result holds if f is of bounded variation

Wavelets Encoding: the problem

- We want to control number of bits not number of terms

Wavelets Encoding: the problem

- We want to control number of bits not number of terms
- positions could potentially occur at arbitrary scales

Wavelets Encoding: the problem

- We want to control number of bits not number of terms
- positions could potentially occur at arbitrary scales
- encoding a position could require arbitrarily large number of bits

Wavelets Encoding: the problem

- We want to control number of bits not number of terms
- positions could potentially occur at arbitrary scales
- encoding a position could require arbitrarily large number of bits
- Practically: for images scales are limited by pixel resolution

Wavelets Encoding: the problem

- We want to control number of bits not number of terms
- positions could potentially occur at arbitrary scales
- encoding a position could require arbitrarily large number of bits
- Practically: for images scales are limited by pixel resolution
- any coefficient could require infinitely many bits

Wavelets Encoding: the problem

- We want to control number of bits not number of terms
- positions could potentially occur at arbitrary scales
- encoding a position could require arbitrarily large number of bits
- Practically: for images scales are limited by pixel resolution
- any coefficient could require infinitely many bits
- Using certain special wavelets and the fact that pixels are quantized the number of bits for complete resolution is finite

Wavelet Encoding: The solution

- Cohen-Dahmen-Daubechies-DeVore (Cohen-Daubechies-Gulleryuz-Orchard)

Wavelet Encoding: The solution

- Cohen-Dahmen-Daubechies-DeVore (Cohen-Daubechies-Gulleryuz-Orchard)
- Assume all coefficient smaller than one

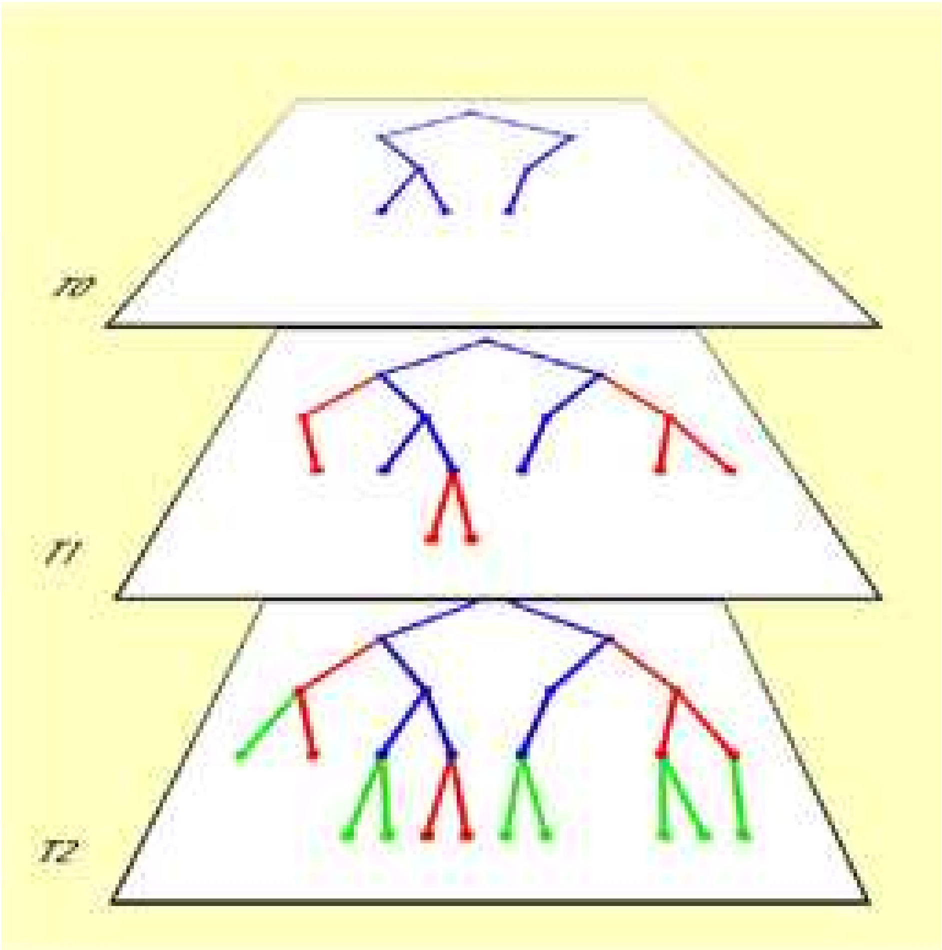
Wavelet Encoding: The solution

- Cohen-Dahmen-Daubechies-DeVore (Cohen-Daubechies-Gulleryuz-Orchard)
- Assume all coefficient smaller than one
- Let $\Lambda_k(f) := \{I : |c_I(f)| > 2^{-k-1}\}$

Wavelet Encoding: The solution

- Cohen-Dahmen-Daubechies-DeVore (Cohen-Daubechies-Gulleryuz-Orchard)
- Assume all coefficient smaller than one
- Let $\Lambda_k(f) := \{I : |c_I(f)| > 2^{-k-1}\}$
- Complete $\Lambda_k(f)$ to a tree $T_k(f)$

Progressive Trees



Wavelet Encoding: The solution

- Cohen-Dahmen-Daubechies-DeVore (Cohen-Daubechies-Gulleryuz-Orchard)
- Assume all coefficient smaller than one
- Let $\Lambda_k(f) := \{I : |c_I(f)| > 2^{-k}\}$
- Complete $\Lambda_k(f)$ to a tree $T_k(f)$
- For each real number c let $s_0(c)$ be the sign bit of c

Wavelet Encoding: The solution

- Cohen-Dahmen-Daubechies-DeVore (Cohen-Daubechies-Gulleryuz-Orchard)
- Assume all coefficient smaller than one
- Let $\Lambda_k(f) := \{I : |c_I(f)| > 2^{-k}\}$
- Complete $\Lambda_k(f)$ to a tree $T_k(f)$
- For each real number c let $s_0(c)$ be the sign bit of c
- Let $b_k(c)$ be the k-th binary bit of $|c|$

Wavelet Encoding: The solution

- Cohen-Dahmen-Daubechies-DeVore (Cohen-Daubechies-Gulleryuz-Orchard)
- Assume all coefficient smaller than one
- Let $\Lambda_k(f) := \{I : |c_I(f)| > 2^{-k}\}$
- Complete $\Lambda_k(f)$ to a tree $T_k(f)$
- For each real number c let $s_0(c)$ be the sign bit of c
- Let $b_k(c)$ be the k-th binary bit of $|c|$
- $c = \pm(b_1(c)2^{-1} + b_2(c)2^{-2} + \dots)$

Wavelet Encoding: The solution

- Cohen-Dahmen-Daubechies-DeVore (Cohen-Daubechies-Gulleryuz-Orchard)
- Assume all coefficient smaller than one
- Let $\Lambda_k(f) := \{I : |c_I(f)| > 2^{-k}\}$
- Complete $\Lambda_k(f)$ to a tree $T_k(f)$
- For each real number c let $s_0(c)$ be the sign bit of c
- Let $b_k(c)$ be the k-th binary bit of $|c|$
- $c = \pm(b_1(c)2^{-1} + b_2(c)2^{-2} + \dots)$
- Note, if $I \in T_{k+1}(f) \setminus T_k(f)$ then $b_1(f) = b_2(f) = \cdots = b_k(f) = 0$

The encoder

- Given f the bitstream $B(f)$ is
  $B(T_0(f)), B(S_0(f)), B(C_0(f)), \ldots,$
  $B(T_k(f)), B(S_k(f)), B(C_k(f)), \ldots$

The encoder

- Given f the bitstream $B(f)$ is
  $B(T_0(f)), B(S_0(f)), B(C_0(f)), \ldots,$
  $B(T_k(f)), B(S_k(f)), B(C_k(f)), \ldots$
- $B(T_k(f))$ bits to identify $T_k(f)$

The encoder

- Given f the bitstream $B(f)$ is $B(T_0(f)), B(S_0(f)), B(C_0(f)), \ldots,$ $B(T_k(f)), B(S_k(f)), B(C_k(f)), \ldots$
- $B(T_k(f))$ bits to identify $T_k(f)$
- $B(S_k(f))$ the sign bits for the new coefficients corresponding to $I \in T_k \setminus T_{k-1}$

The encoder

- Given f the bitstream $B(f)$ is
$B(T_0(f)), B(S_0(f)), B(C_0(f)), \ldots,$
$B(T_k(f)), B(S_k(f)), B(C_k(f)), \ldots$
- $B(T_k(f))$ bits to identify $T_k(f)$
- $B(S_k(f))$ the sign bits for the new coefficients corresponding to $I \in T_k \setminus T_{k-1}$
- $B(C_k(f))$ consists of the lead bit, i.e. $b_k(c_I(f))$ for each $I \in T_k \setminus T_{k-1}$ and one additional bit for each $I \in T_{k-1}(f)$

The encoder

- Given f the bitstream $B(f)$ is
  $B(T_0(f)), B(S_0(f)), B(C_0(f)), \ldots,$
  $B(T_k(f)), B(S_k(f)), B(C_k(f)), \ldots$
- $B(T_k(f))$ bits to identify $T_k(f)$
- $B(S_k(f))$ the sign bits for the new coefficients corresponding to $I \in T_k \setminus T_{k-1}$
- $B(C_k(f))$ consists of the lead bit, i.e. $b_k(c_I(f))$ for each $I \in T_k \setminus T_{k-1}$ and one additional bit for each $I \in T_{k-1}(f)$
- Note that after bits at level k have been received each wavelet coefficient of f is known to accuracy 2^{-k}

Properties of the Tree Encoder

- The encoder is progressive and simultaneously near optimal for all Sobolev and Besov classes (CDDD) and all n

Properties of the Tree Encoder

- The encoder is progressive and simultaneously near optimal for all Sobolev and Besov classes (CDDD) and all n
- Key property: Can encode a binary tree with n nodes using at most $2n$ bits

Properties of the Tree Encoder

- The encoder is progressive and simultaneously near optimal for all Sobolev and Besov classes (CDDD) and all n
- Key property: Can encode a binary tree with n nodes using at most $2n$ bits
- Encoder is similar to other tree based encoders: EZW, Said-Pearlman

Properties of the Tree Encoder

- The encoder is progressive and simultaneously near optimal for all Sobolev and Besov classes (CDDD) and all n
- Key property: Can encode a binary tree with n nodes using at most $2n$ bits
- Encoder is similar to other tree based encoders: EZW, Said-Pearlman
- We know precisely the set of images on which these encoders have decay rate $O(n^{-\alpha})$ for any $\alpha > 0$

Properties of the Tree Encoder

- The encoder is progressive and simultaneously near optimal for all Sobolev and Besov classes (CDDD) and all n
- Key property: Can encode a binary tree with n nodes using at most $2n$ bits
- Encoder is similar to other tree based encoders: EZW, Said-Pearlman
- We know precisely the set of images on which these encoders have decay rate $O(n^{-\alpha})$ for any $\alpha > 0$
- Besov smoothness order s is an exact predictor of tree encoder performance $O(n^{-s/2})$

Drawbacks to the Tree Encoder

- Model Images by Besov smoothness classes

Drawbacks to the Tree Encoder

- Model Images by Besov smoothness classes
- Want better classes more indicative of real images: anisotropic classes

Drawbacks to the Tree Encoder

- Model Images by Besov smoothness classes
- Want better classes more indicative of real images: anisotropic classes
- Wavelet bases are isotropic

Drawbacks to the Tree Encoder

- Model Images by Besov smoothness classes
- Want better classes more indicative of real images: anisotropic classes
- Wavelet bases are isotropic
- Wavelets handle point singularities well

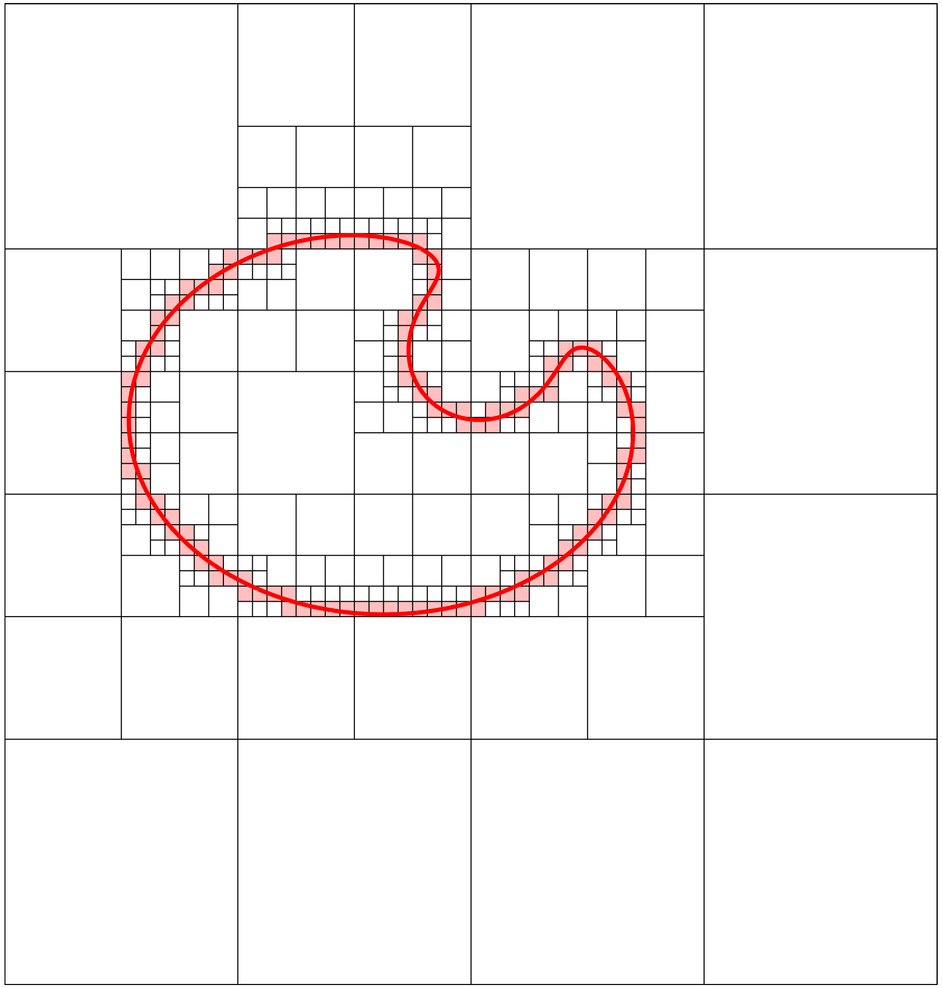
Drawbacks to the Tree Encoder

- Model Images by Besov smoothness classes
- Want better classes more indicative of real images: anisotropic classes
- Wavelet bases are isotropic
- Wavelets handle point singularities well
- Wavelets do not handle singularities along curves

Drawbacks to the Tree Encoder

- Model Images by Besov smoothness classes
- Want better classes more indicative of real images: anisotropic classes
- Wavelet bases are isotropic
- Wavelets handle point singularities well
- Wavelets do not handle singularities along curves
- Toy Problem: Horizon function

Horizon approximation



Possible remedies

- New systems: ridgelets, wedgelets, curvelets, ...

Possible remedies

- New systems: ridgelets, wedgelets, curvelets, ...
- Candes and Donoho

Possible remedies

- New systems: ridgelets, wedgelets, curvelets, ...
- Candes and Donoho
- Frames and redundant systems

Example: Wedgelets

- To each dyadic square I with $|I| = 2^{-j}$ associate a family of wedgelets

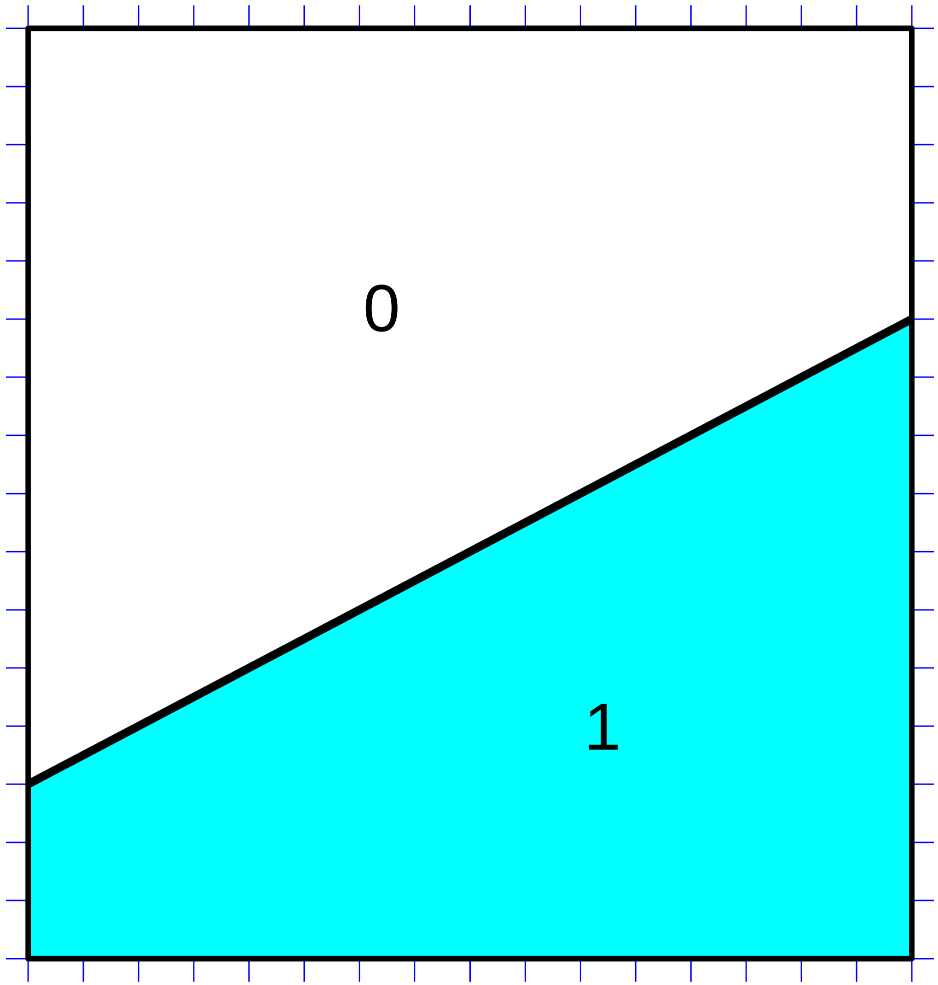
Example: Wedgelets

- To each dyadic square I with $|I| = 2^{-j}$ associate a family of wedgelets
- A wedgelet is a piecewise constant function taking the values $0, 1$

Example: Wedgelets

- To each dyadic square I with $|I| = 2^{-j}$ associate a family of wedgelets
- A wedgelet is a piecewise constant function taking the values $0, 1$
- Demarcation is given by a line connecting grid points on boundary of I with spacing 2^{-2j}

Picture of a wedgelet



Approximating Horizon functions

- Any horizon function with C^2 boundary can be approximated in L_2 to error n^{-1} using n wedgelets

Approximating Horizon functions

- Any horizon function with C^2 boundary can be approximated in L_2 to error n^{-1} using n wedgelets
- By comparison wavelets give error $n^{-1/2}$

Approximating Horizon functions

- Any horizon function with C^2 boundary can be approximated in L_2 to error n^{-1} using n wedgelets
- By comparison wavelets give error $n^{-1/2}$
- Fourier gives error $n^{-1/4}$

Merging systems

- We would like to merge systems like Wavelets, Fourier, Wedgelets, etc.

Merging systems

- We would like to merge systems like Wavelets, Fourier, Wedgelets, etc.
- retain best properties of all systems

Merging systems

- We would like to merge systems like Wavelets, Fourier, Wedgelets, etc.
- retain best properties of all systems
- how to do this in a practical encoder

Example: Merge wavelets and wedgelets

- Baraniuk-Romberg-Wakin

Example: Merge wavelets and wedgelets

- Baraniuk-Romberg-Wakin
- Idea is to decompose domain into regions done by either wavelets or wedgelets

Example: Merge wavelets and wedgelets

- Baraniuk-Romberg-Wakin
- Idea is to decompose domain into regions done by either wavelets or wedgelets
- Ornate wavelet tree with label on how wavelet coefficients are to be computed

Example: Merge wavelets and wedgelets

- Baraniuk-Romberg-Wakin
- Idea is to decompose domain into regions done by either wavelets or wedgelets
- Ornate wavelet tree with label on how wavelet coefficients are to be computed
- interior nodes compute wavelet coefficients in standard way

BRW Continued

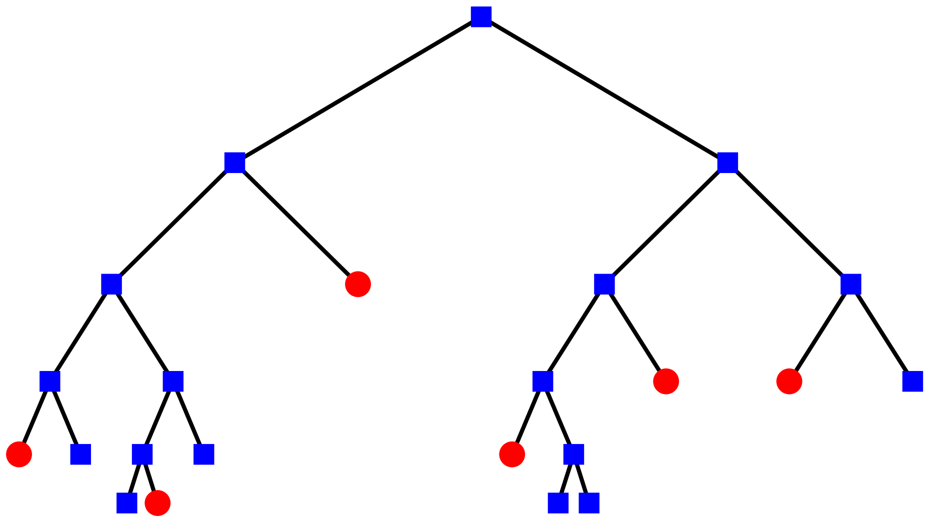
- If leave (final node) of tree is ornated with a wavelet then wavelet coefficients for all nodes below leave are given value zero

BRW Continued

- If leave (final node) of tree is ornated with a wavelet then wavelet coefficients for all nodes below leave are given value zero
- If leave is ornated with wedgelet, all coefficients below this node are computed as wavelet coefficients of that wedgelet (wedgeprint)

Wedgelet-wavelet tree

red = Wedgelets   blue = Wavelets



Final Thoughts on Compression

- We need better model classes for images

Final Thoughts on Compression

- We need better model classes for images
- These classes should model the anisotropies found in real world images

Final Thoughts on Compression

- We need better model classes for images
- These classes should model the anisotropies found in real world images
- Horizon models are too simplistic

Final Thoughts on Compression

- We need better model classes for images
- These classes should model the anisotropies found in real world images
- Horizon models are too simplistic
- Once these new classes are clearly defined then encoders can be designed to perform at near optimal compression rates for these classes