Multiscale Geometric Image Compression using Wavelets and Wedgelets

Richard Baraniuk

Rice University
dsp.rice.edu

Joint work with *Hyeokho Choi, Justin Romberg, Mike Wakin*

RICE UNIVERSITY

Multiscale Geometric *Compression* of *Piecewise Smooth Images* using Wavelets and Wedgelets

Textures **Cartoons**

Computational Harmonic Analysis

- Representation $f = \sum_k \underbrace{a_k}_{\text{coefficients}} \underbrace{\mathbf{b}_k}_{\text{basis, frame}}$

- **Analysis** study f through *structure* of $\{a_k\}$
  $\{\mathbf{b}_k\}$ should *extract features* of interest

- **Approximation** $\widehat{f}_N$ uses just a few terms N
  exploit *sparsity* of $\{a_k\}$

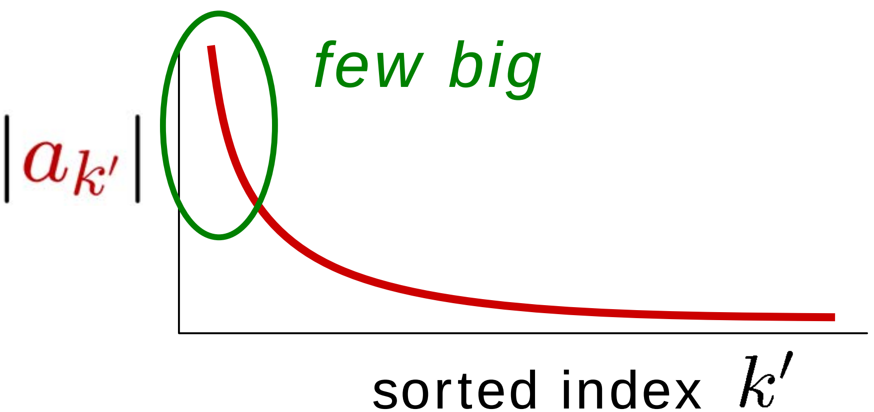
Nonlinear Approximation

$$f = \sum_k a_k \mathbf{b}_k$$

- N-*term approximation*: use *largest* a_k *independently*

$$\widehat{f}_N := \sum_{k'=1}^{N} a_{k'} \mathbf{b}_{k'}$$

- Greedy / *thresholding*



Error Approximation Rates

$$f \;=\; \sum_k a_k \, \mathbf{b}_k$$

$$\widehat{f}_N \;=\; \sum_{k'=1}^{N} a_{k'} \, \mathbf{b}_{k'}$$

$$\|f - \widehat{f}_N\|_2^2 \;<\; C\, N^{-\alpha} \qquad \text{as } N \to \infty$$

- Optimize asymptotic *error decay rate* α

From Approximation to Compression

- *Quantize* coefficients $\{a_k\}$, approximate with R bits

$$f = \sum_k a_k \mathbf{b}_k$$

$$\widehat{f}_R = \sum_k a_k^q \mathbf{b}_k$$

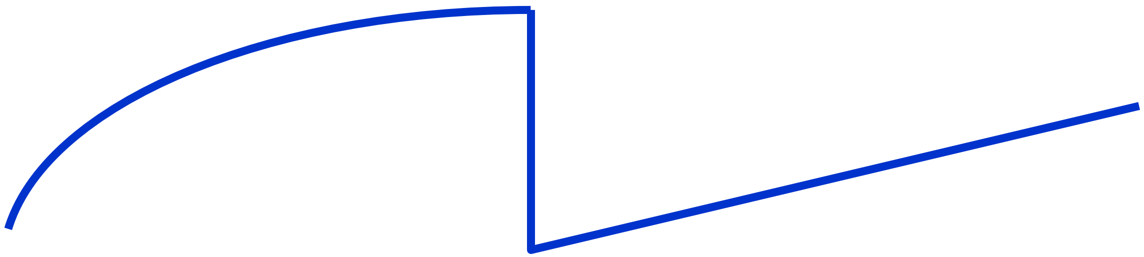
- R bits must specify both *value* a_k^q and *location* k

- Optimize Rate/Distortion decay rate α

$$D(R) = \|f - \widehat{f}_R\|_2^2 < C\, R^{-\alpha} \quad \text{as} \quad R \to \infty$$

1-D Piecewise Smooth Signals

- f *smooth* except for *singularities* at a finite number of 0-D *points*

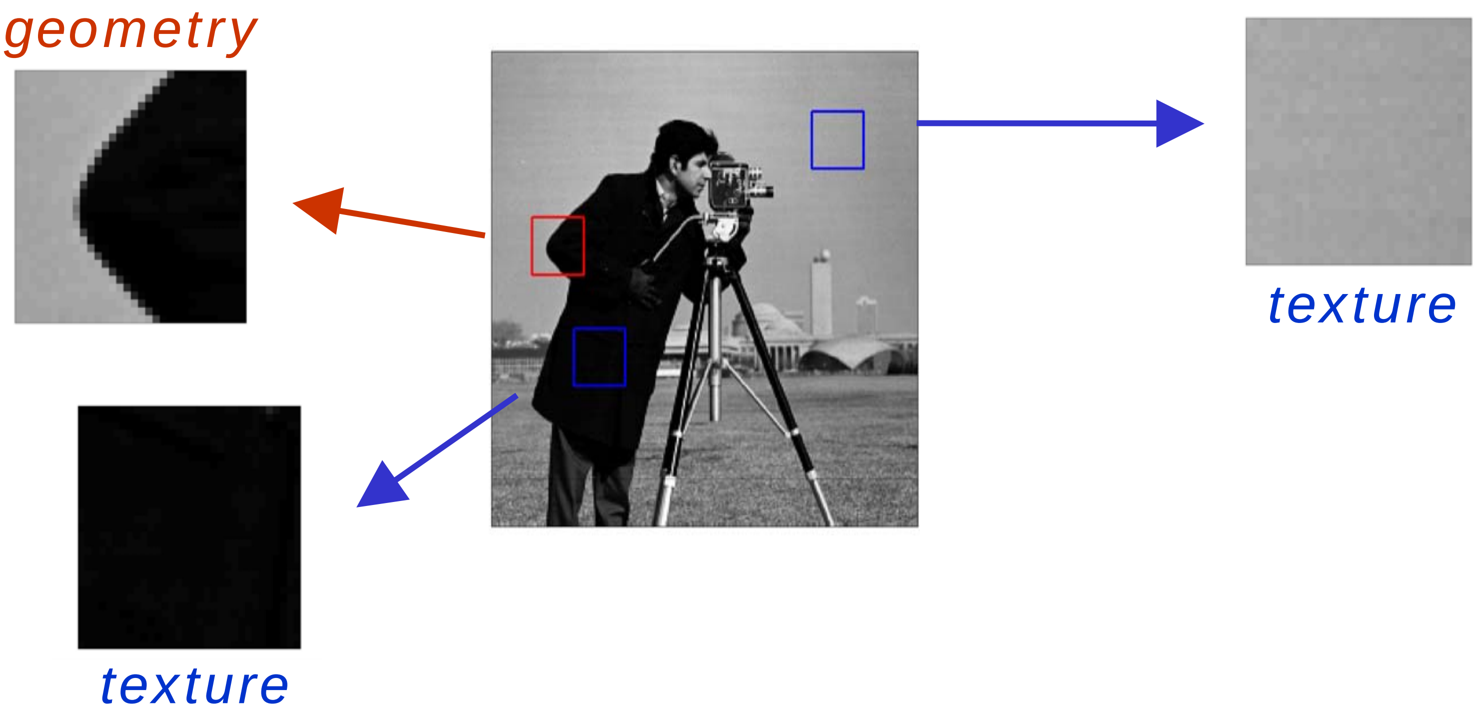


Fourier sinusoids: suboptimal greedy approximation and extraction

wavelets: *optimal* greedy approximation extract singularity structure

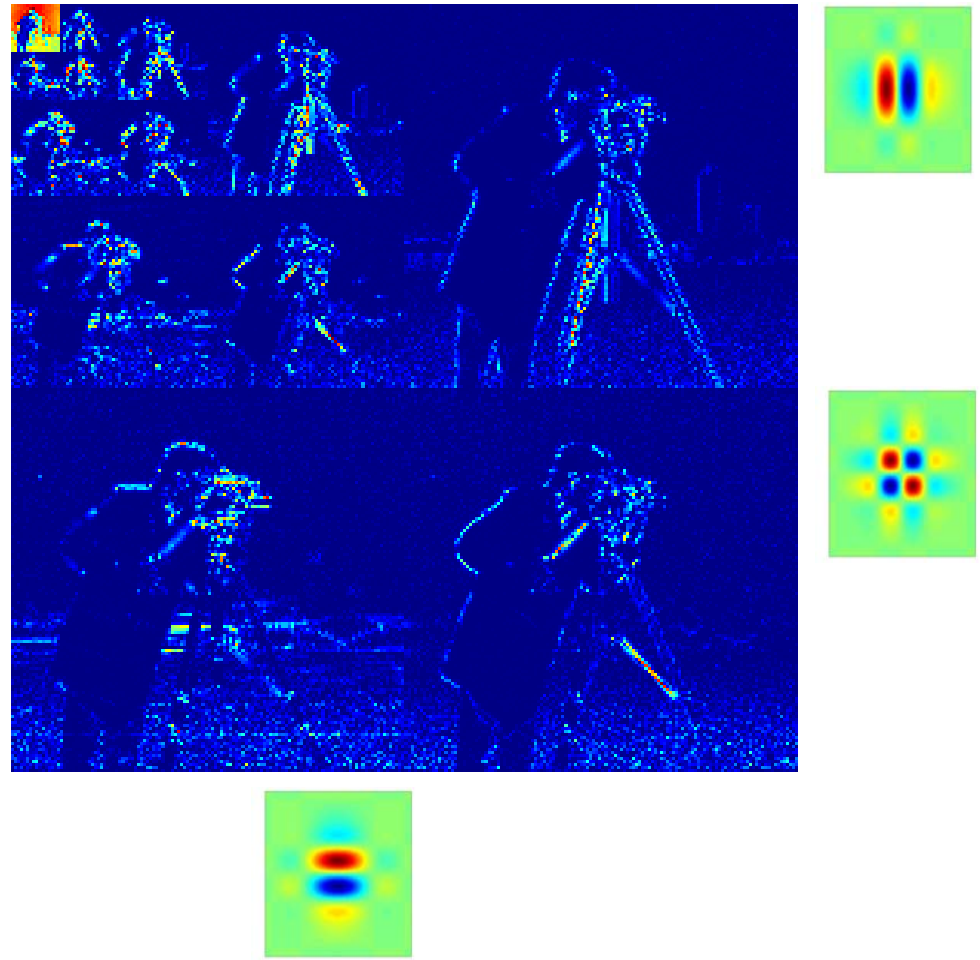
2-D Piecewise Smooth Signals

- f *smooth* except for *singularities* along a finite number of smooth 1-D *curves*



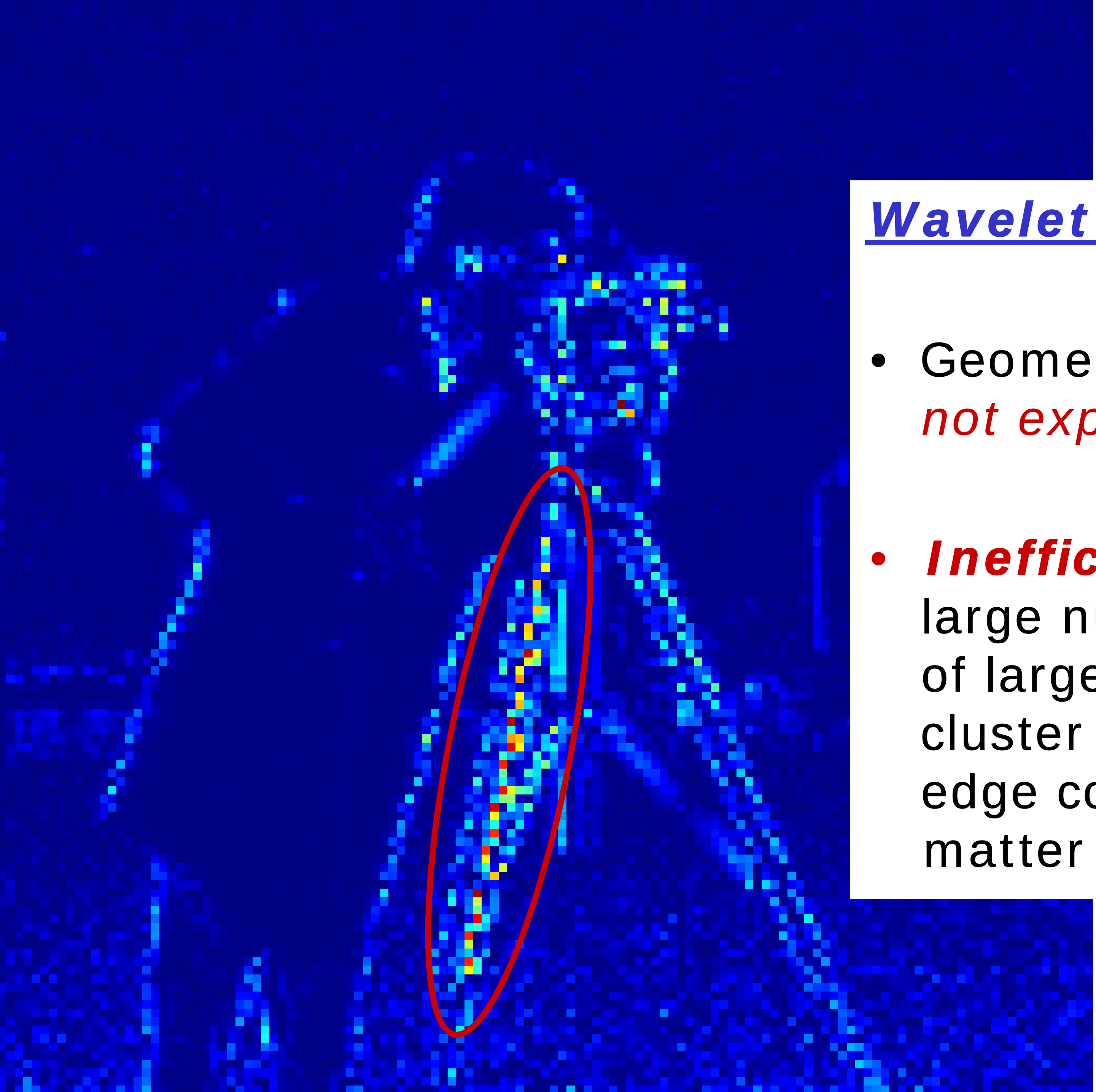
- Challenge: analyze/approximate ***geometric*** *structure*

Wavelet-based Image Processing



- Standard 2-D tensor product *wavelet transform*

$$f = \sum_k a_k \mathbf{b}_k$$



Wavelet Challenges

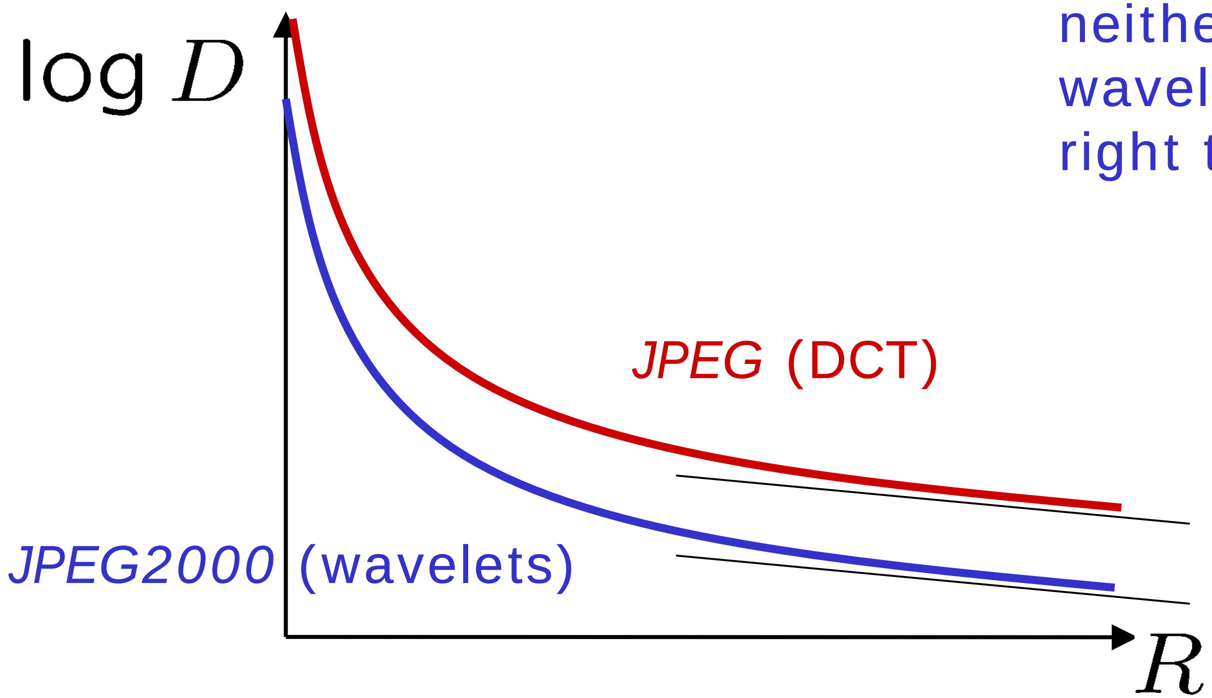
- Geometrical info *not explicit*
- ***Inefficient*** - large number of large wc's cluster around edge contours, no matter how smooth

The JPEG2000 Disappointment

- Current wavelet methods do *not* improve on decay rate of JPEG!

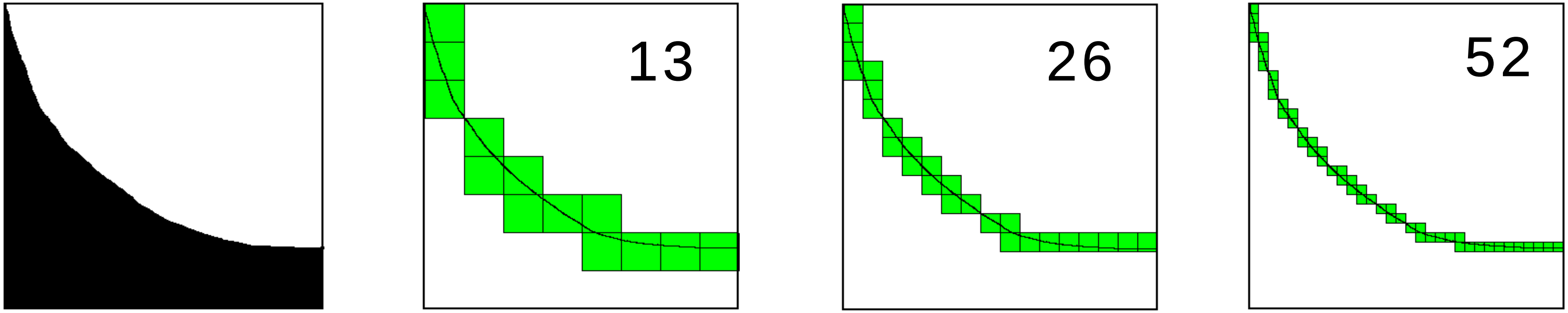
$$D(R) = \|f - \widehat{f}_R\|_2^2 \;<\; C\,R^{-\alpha}$$

- WHY?
  neither DCT nor wavelets are the right transform



2-D Wavelets: Poor Approximation

- Even for a smooth C^2 contour, which straightens at fine scales...



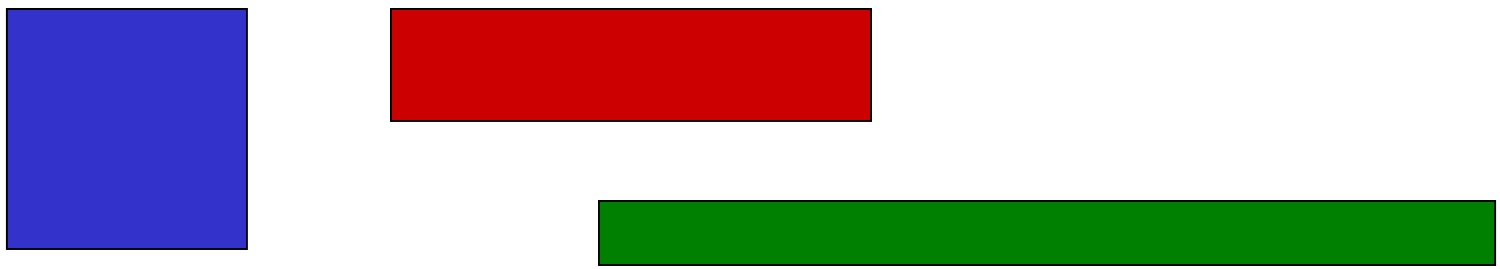
- *Too many wavelets required!*

$$\widehat{f}_N \quad := \quad N\text{-term wavelet approximation}$$

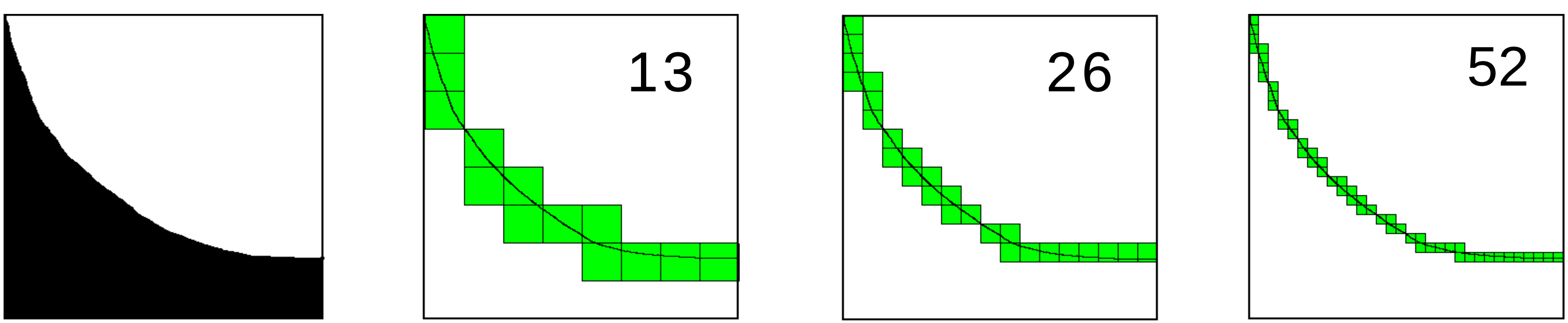
$$\|f - \widehat{f}_N\|_2^2 < C\,N^{-1} \quad \textit{not} \quad N^{-2}$$

Solution 1: Upgrade the *Transform*

- Introduce *anisotropic transform*
  - curvelets, ridgelets, contourlets, ...

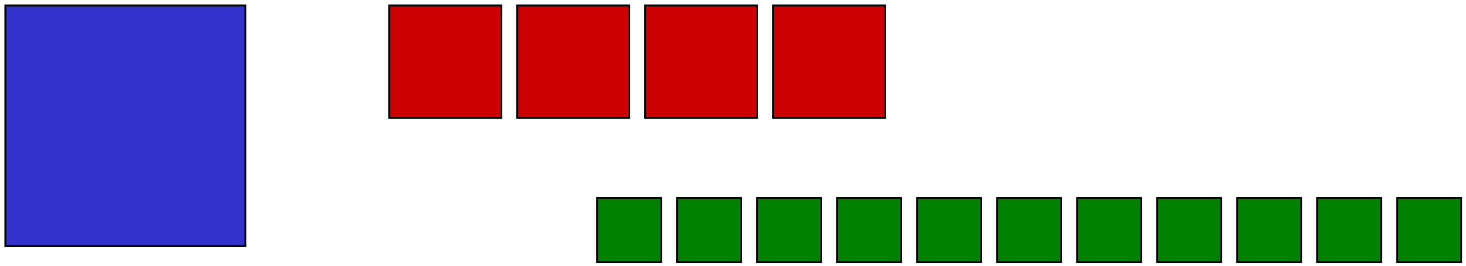


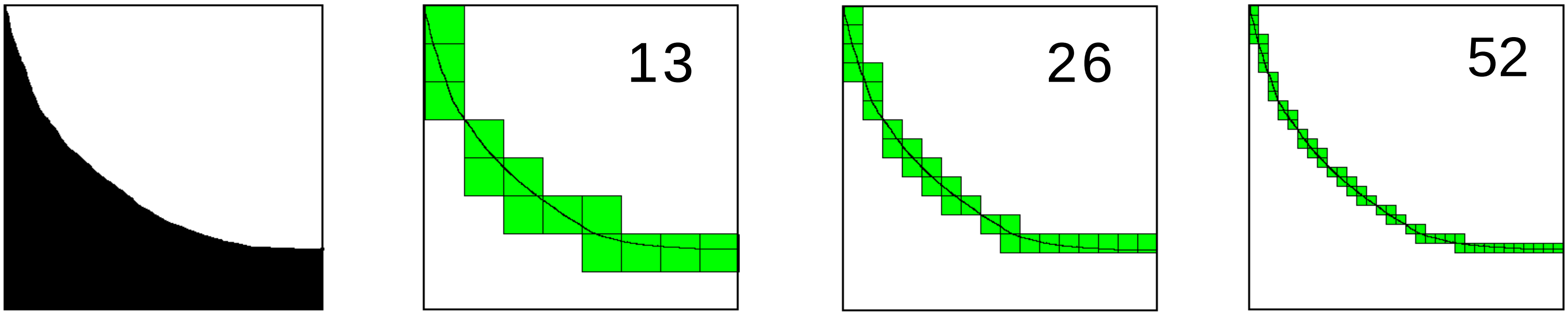
- Optimal error decay rates for cartoons +



Solution 2: Upgrade the *Processing*

- Replace coefficient thresholding by a new wc *model* that captures *anisotropic spatial correlations*

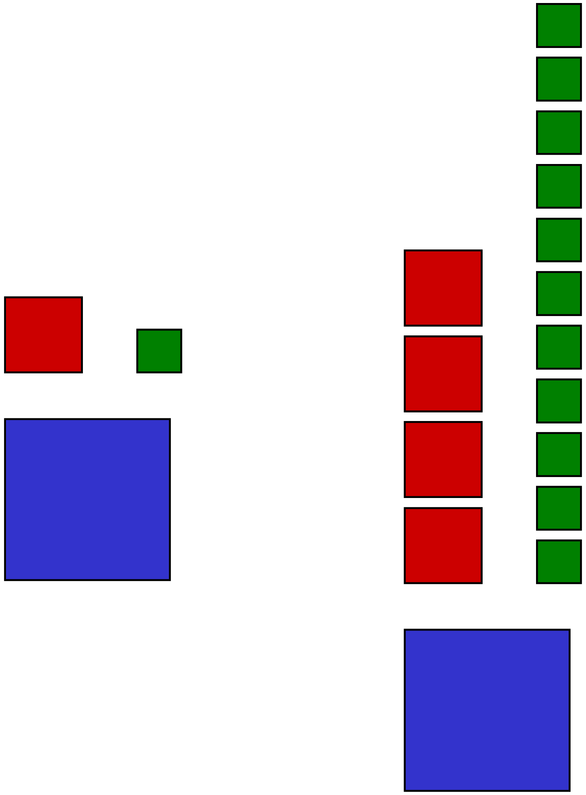




Our Goal: New Image Models

- Wavelet coefficient models that capture both *geometric* and *textural* aspects of natural images

- Optimal error decay rates for some image class

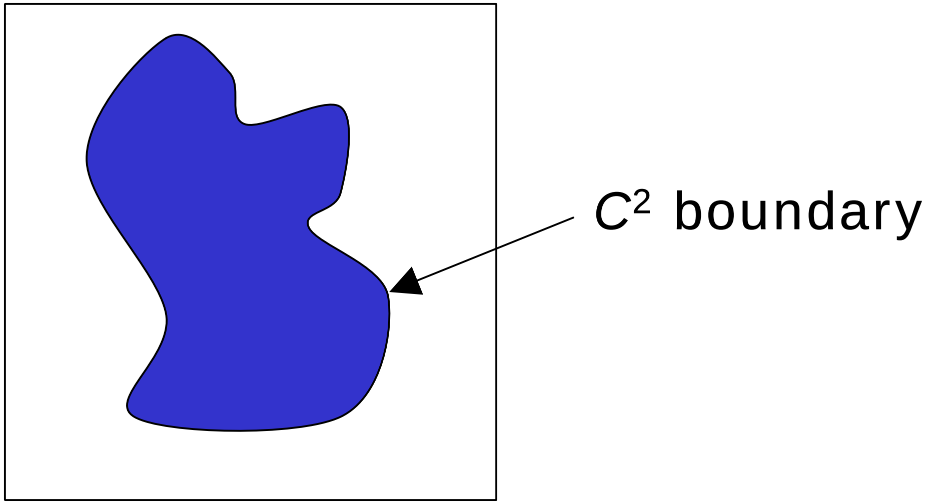


Part I

Geometric Modeling of *Cartoons* with *Wedgelets*

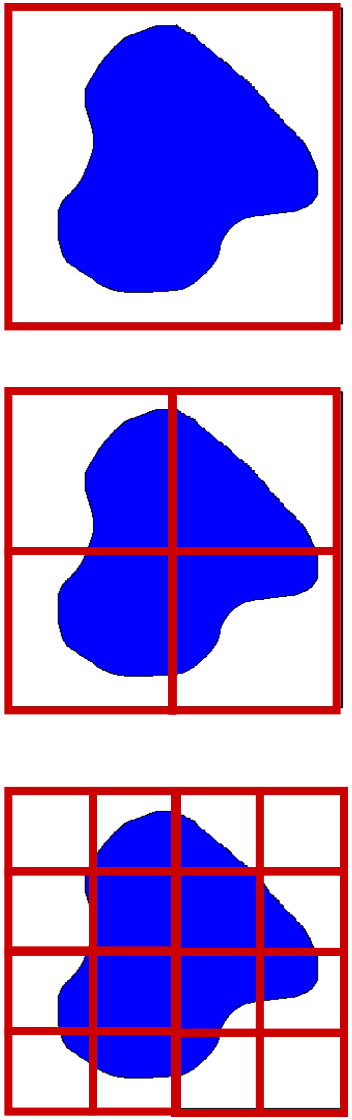
Geometry Model for *Cartoons*

- Toy model:
  flat regions separated by smooth contours



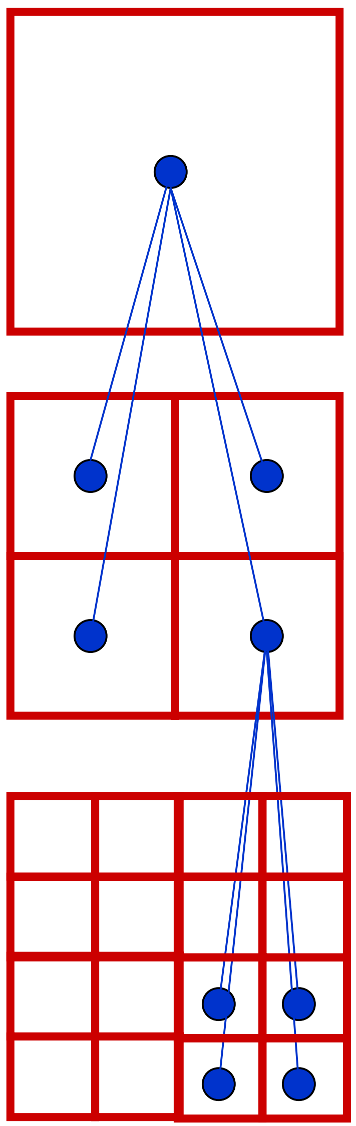
- Goal: representation that is
  - sparse
  - simply modeled
  - efficiently computed
  - extensible to texture regions

2-D Dyadic Partition



- *Multiscale* analysis
- *Partition*; not a basis/frame
- Zoom in by factor of 2 each scale

2-D Dyadic Partition = *Quadtree*

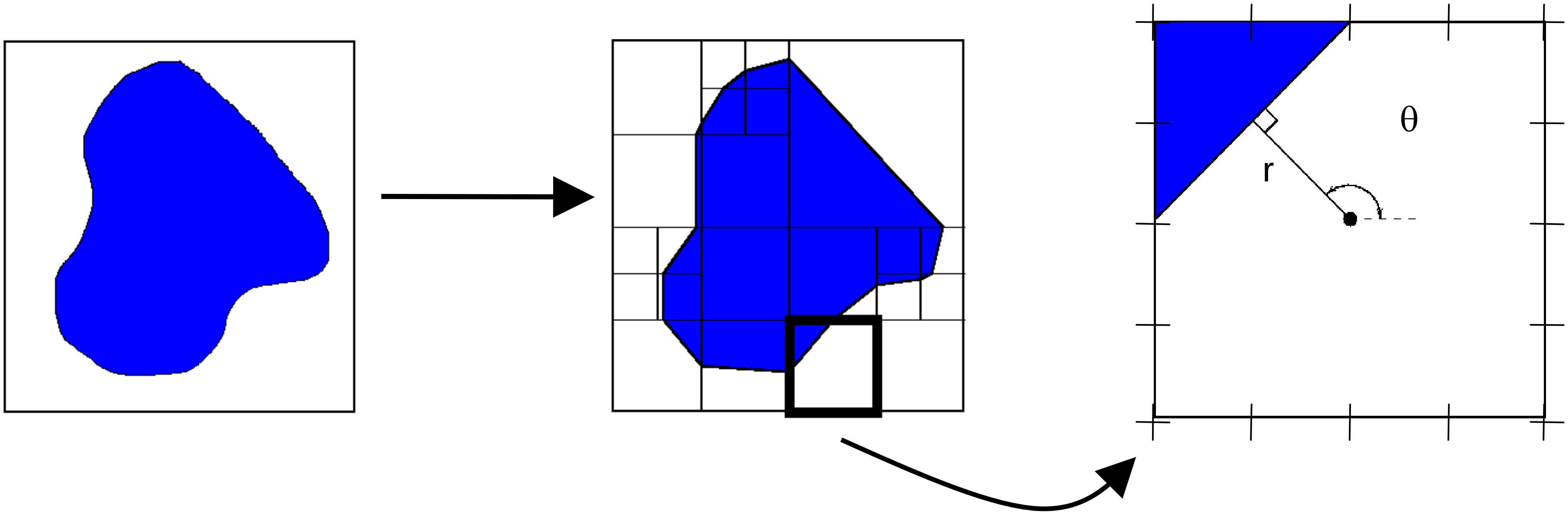


- *Multiscale* analysis
- *Partition*;
  not a basis/frame
- Zoom in by factor
  of 2 each scale
- Each *parent* node
  has *4 children* at
  next finer scale

Wedgelet Representation [Donoho]

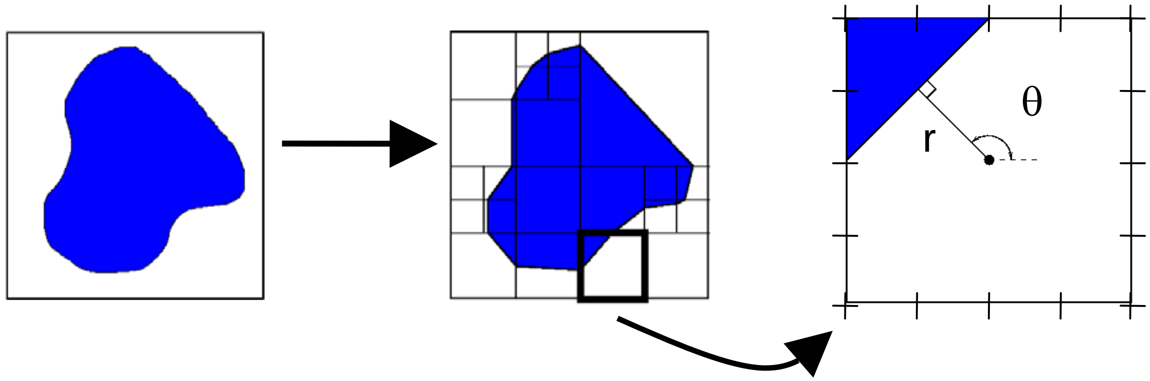
- Build a cartoon using *wedgelets* on dyadic squares

wedgelet = atomic geometric element



Wedgelet Representation

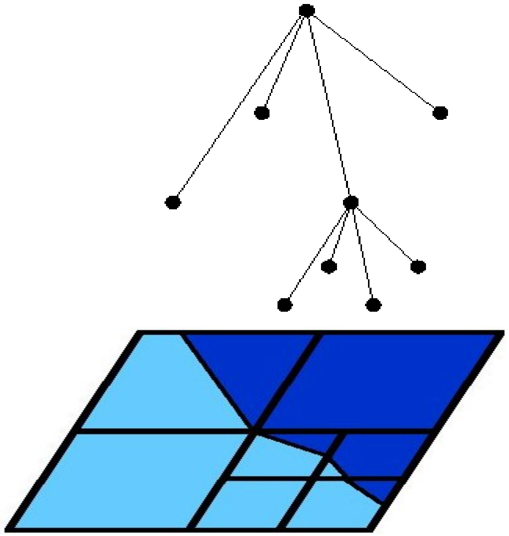
- Build a cartoon using *wedgelets* on dyadic squares



- Choose orientation (r,θ) from finite dictionary (toroidal sampling)

- *Quad-tree* structure

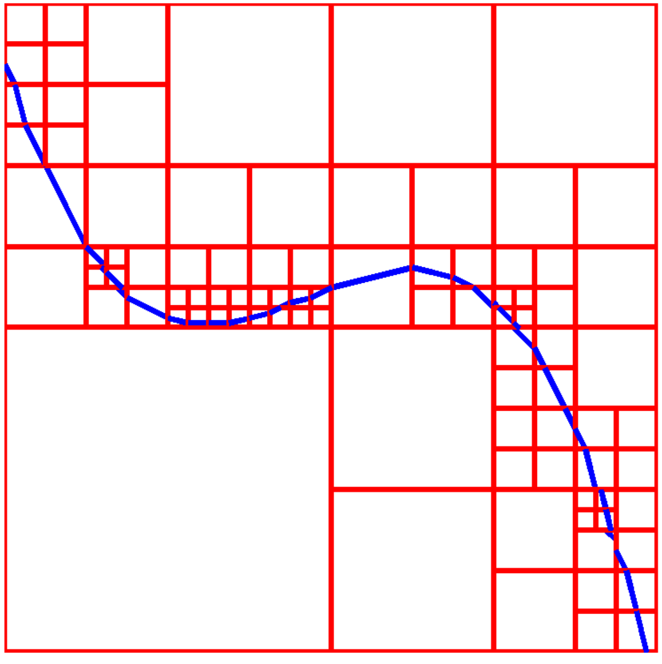
deeper in tree $\Rightarrow$
finer curve approximation

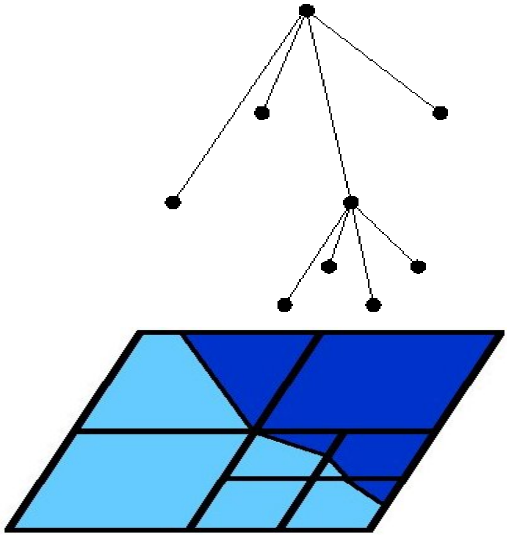


Wedgelet Representation

- *Prune* wedgelet quadtree to approximate local geometry (adaptive)

- *Decorate* leaves with (r, θ) parameters





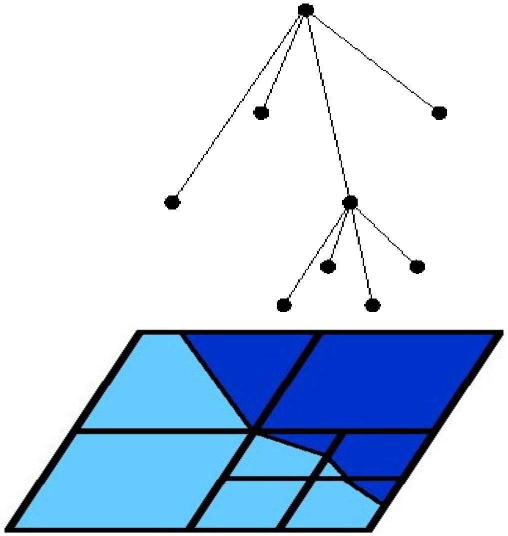
Wedgelet Inference

- Find representation / prune tree to balance a fidelity vs. complexity trade-off

$$\min_{W} \underbrace{\|f - \widehat{f}_W\|_2^2} + \lambda \underbrace{\mathrm{Comp}(W)}$$

- For Comp(W) – need a ***model*** for the wedgelet representation W (quadtree + (r,θ))

- Donoho: Comp(W) = #leaves



Wedgelet Inference

- Find representation / prune tree to balance a fidelity vs. complexity trade-off

$$\min_{W} \underbrace{\|f - \widehat{f}_W\|_2^2}_{} + \lambda \underbrace{[\#\text{leaves}]}_{}$$

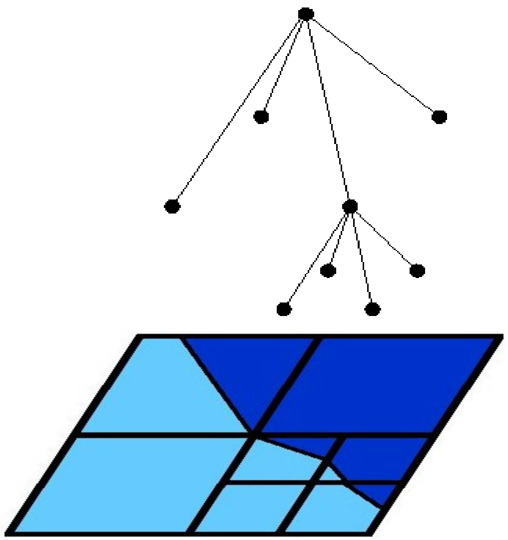
- *O*(*N*) *dynamic programming* solution (bottom-up recursion)

- Optimal *approximation*

$$L_2 \text{ error} \sim (\#\text{leaves})^{-2}$$

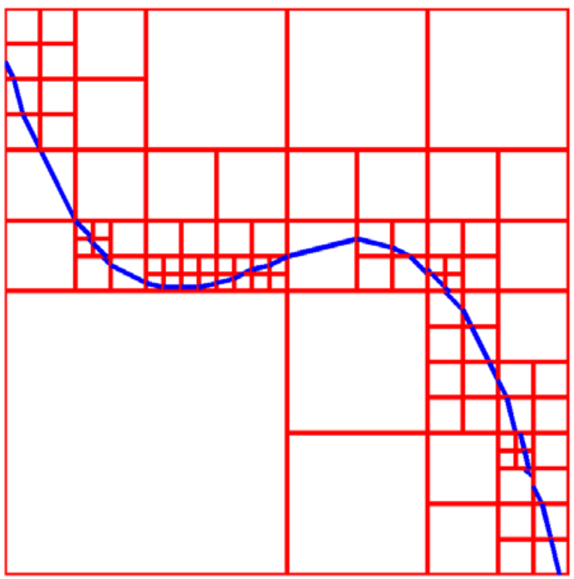
- *Near-optimal* rate/distortion decay

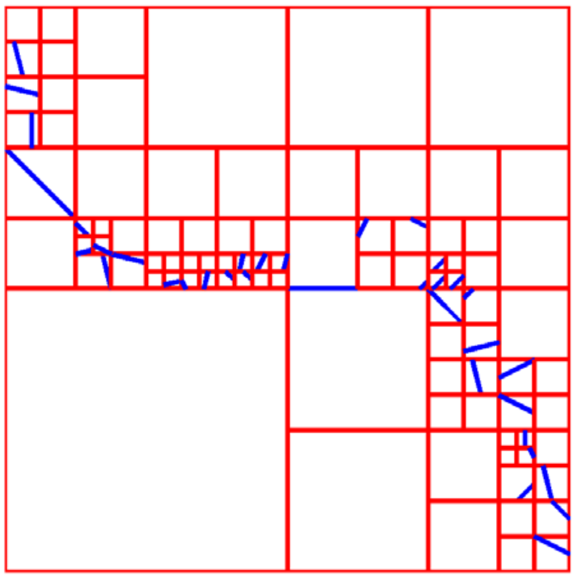
$$D(R) \sim (\log R)^2 \, R^{-2}$$



#Leaves Complexity Penalty

- Accounts for wedgelet partition ***size***,
  but not wedgelet ***orientation***



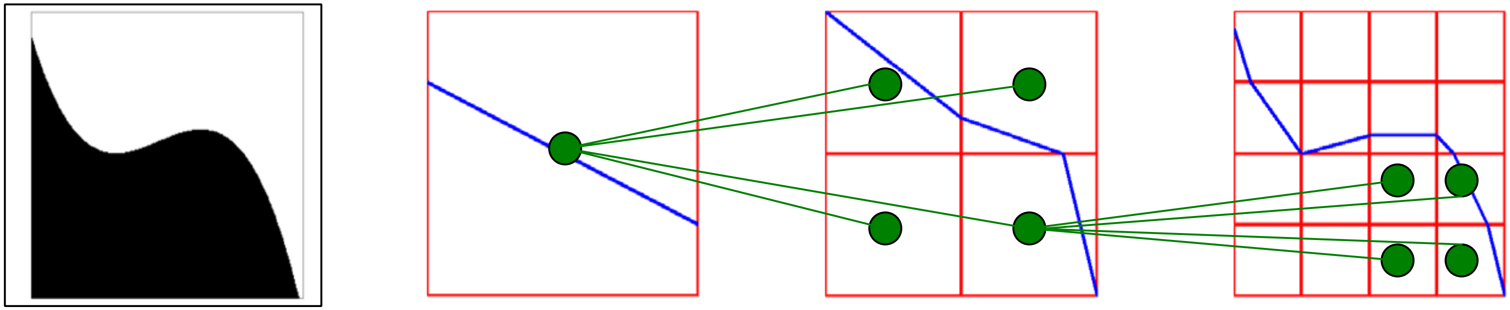


#leaves = #leaves

“smooth”
“simple”

“rough”
“complex”

Multiscale Geometry Model (MGM)



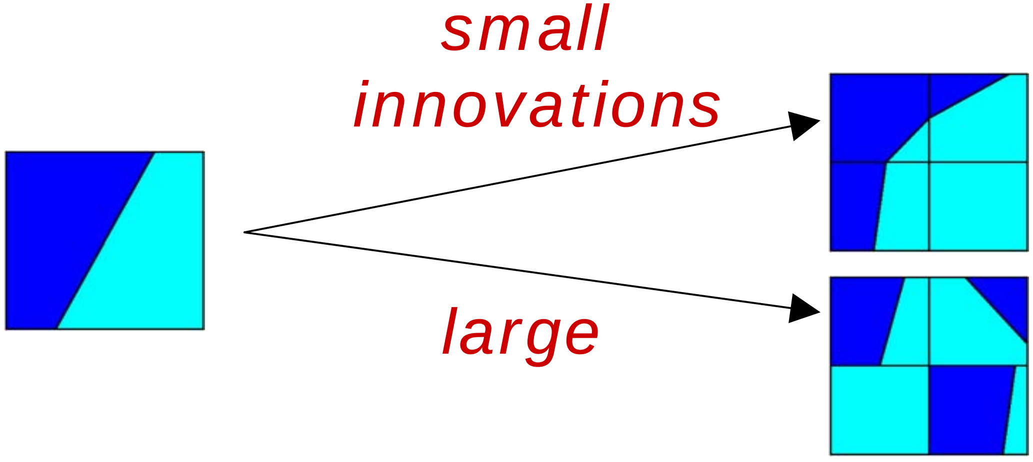
- Decorate *each* tree node with orientation (r,θ) and then *model dependencies thru scale*

- Insight: Smooth curve $\Rightarrow$ *Geometric innovations small at fine scales*

- Model: Favor small innovations over large innovations (*statistically*)

Multiscale Geometry Model (MGM)

- Wavelet-like geometry model: *coarse-to-fine prediction*
  - model parent-to-child transitions of orientations



MGM

- Wavelet-like geometry model:

  coarse-to-fine prediction

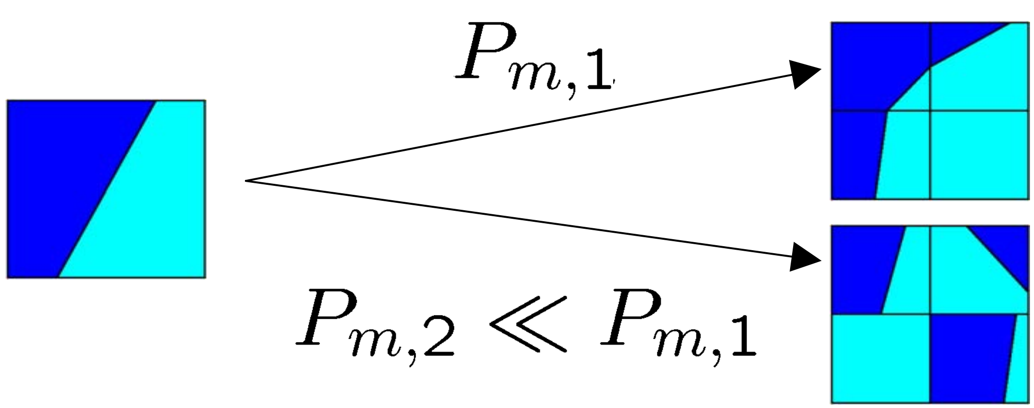
  – model parent-to-child transitions of orientations

- Markov-1 statistical model

  – state = (r, θ) orientation of wedgelet

  – parent-to-child state transition matrix

$$P_{mn} = P(\text{child orientation} = n | \text{parent orientation} = m)$$



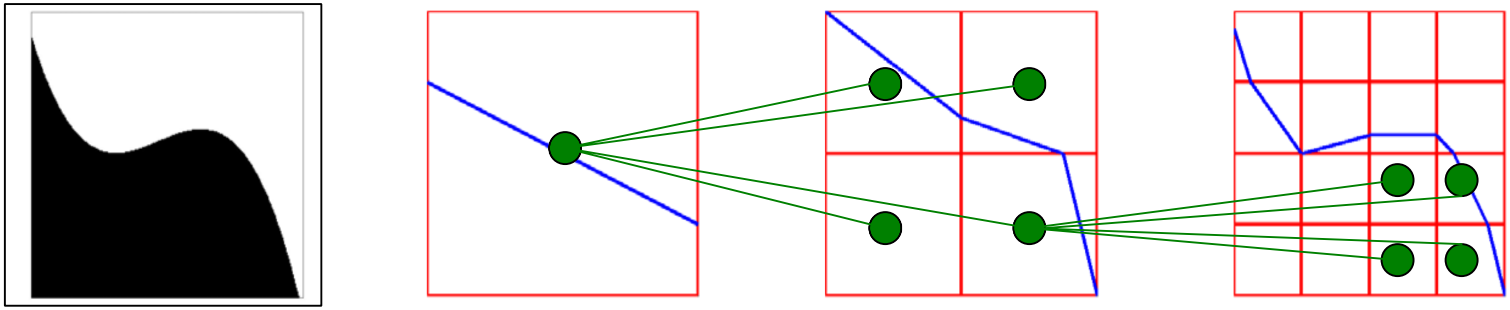
MGM

- Markov-1 statistical model

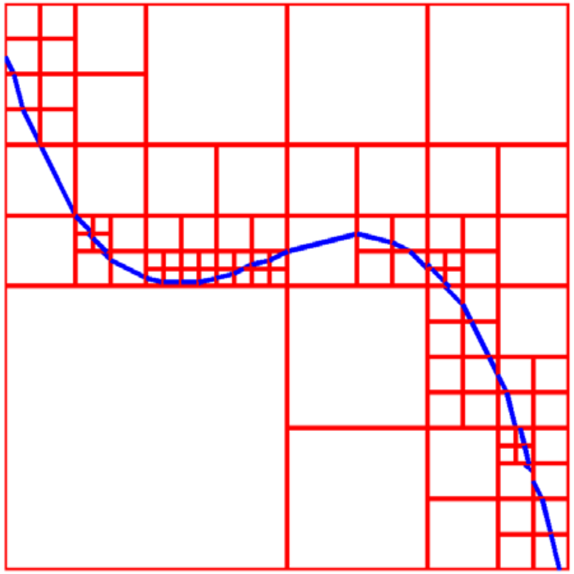
$$P_{mn} = P(\text{child orientation} = n | \text{parent orientation} = m)$$

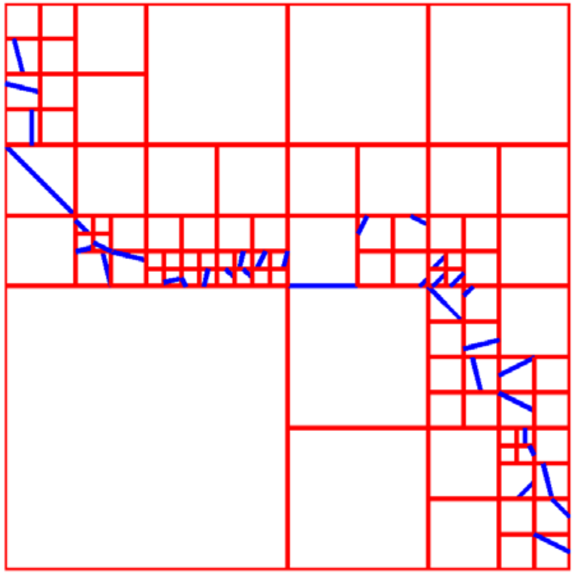
- *Joint wedgelet Markov probability model:* $P(W)$

- **Complexity** = *Shannon codelength* = $-\log P(W)$
  = number of bits to encode W



MGM and Edge Smoothness





$$P(W_1) \gg P(W_2)$$

$$-\log P(W_1) \ll -\log P(W_2)$$

"smooth"
"simple"

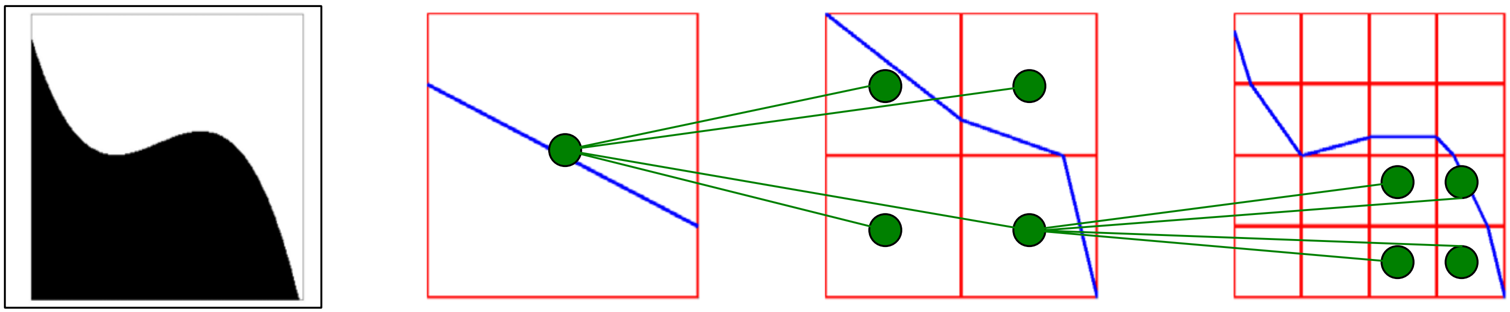
"rough"
"complex"

MGM Inference

- Find representation / prune tree to balance the fidelity vs. complexity trade-off

$$\min_{W} \underbrace{\|f - \widehat{f}_W\|_2^2}_{} + \lambda \underbrace{[-\log P(W)]}_{}$$

- Efficient $O(N)$ solution via *dynamic programming*



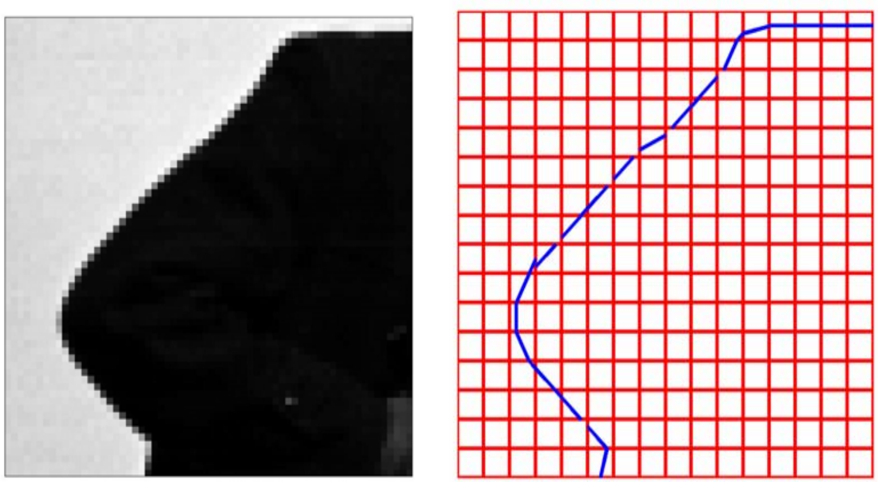
MGM Approximation

- Find representation / prune tree to balance the fidelity vs. complexity trade-off

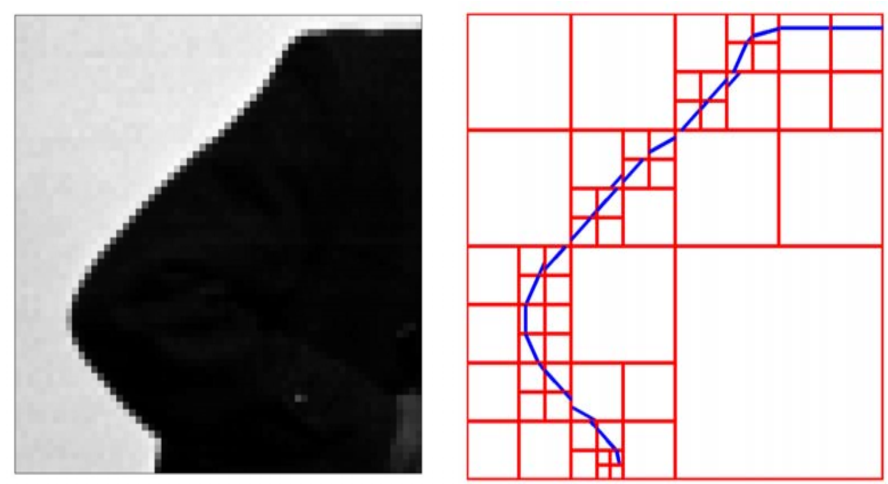
$$\min_{W} \underbrace{\|f - \widehat{f}_W\|_2^2}_{} + \lambda \underbrace{[-\log P(W)]}_{}$$

- *Optimal* L^2 error decay rate for cartoons $(\# W)^{-2}$

single scale



multiscale



Wedgelet *Coding* of Cartoon Images

- Choosing wedgelets = rate-distortion optimization

$$\min_{W} \underline{\|f - \widehat{f}_W\|_2^2} + \lambda \underline{[-\log P(W)]}$$

Shannon code length

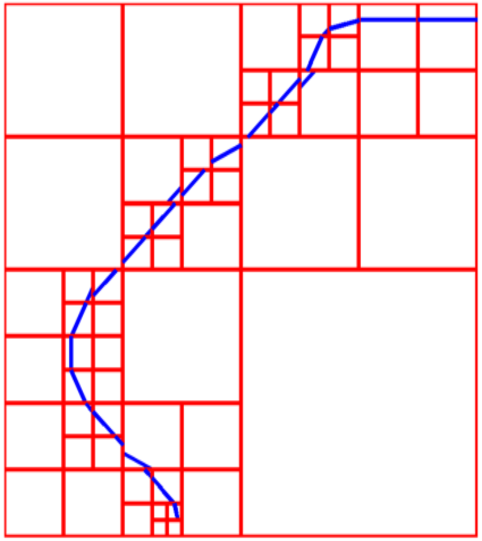
$$\min_{W} \underline{\text{Distortion}} + \lambda \underline{\text{Rate}\,(\text{bits})}$$

to encode (r, θ, m_1, m_2)

Wedgelet Coding of Cartoon Images

- Choosing wedgelets = rate-distortion optimization

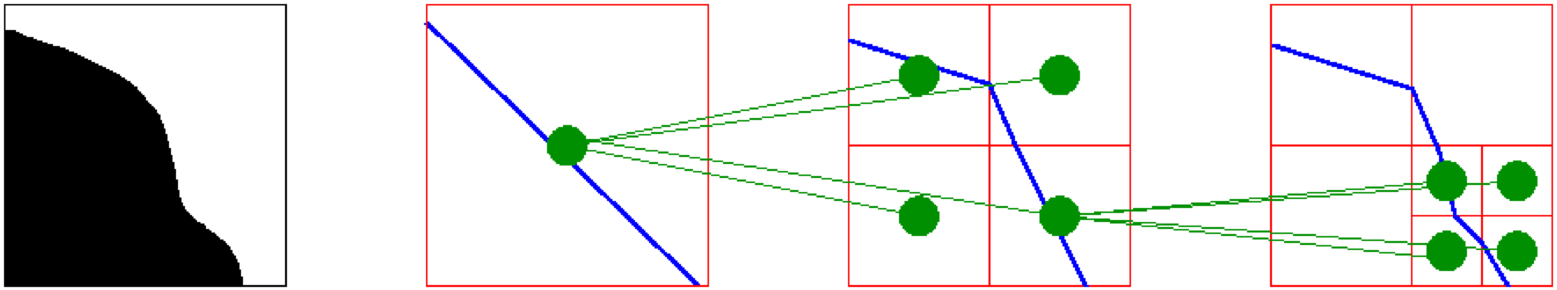
$$\underbrace{\min_{W} \|f - \widehat{f}_W\|_2^2}\quad \text{s.t.} \quad \underbrace{(\#\text{bits}) \le R^\star}$$



~130 bits coded independently

~25 bits coded with MGM

Predictive Wedgelet Coding is Optimal



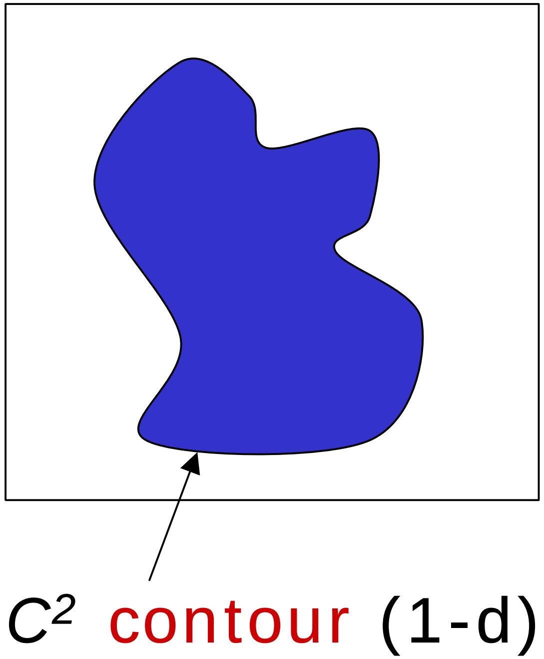
Optimal rate-distortion performance

$$D(R) \sim R^{-2}$$

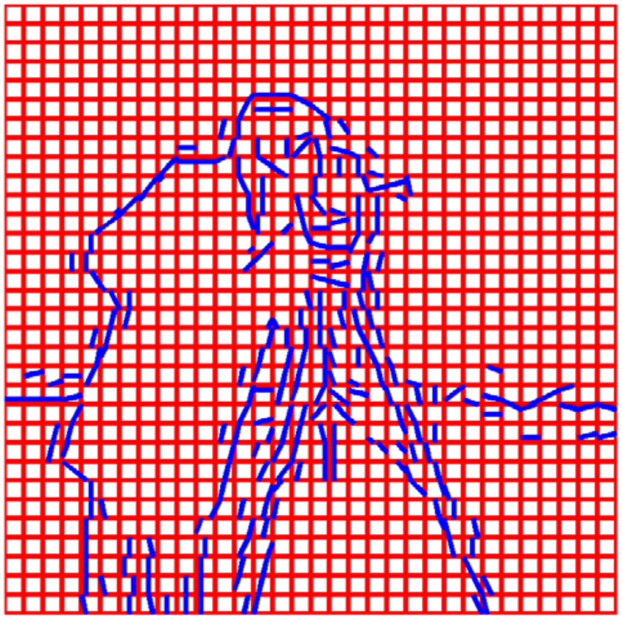
compared to

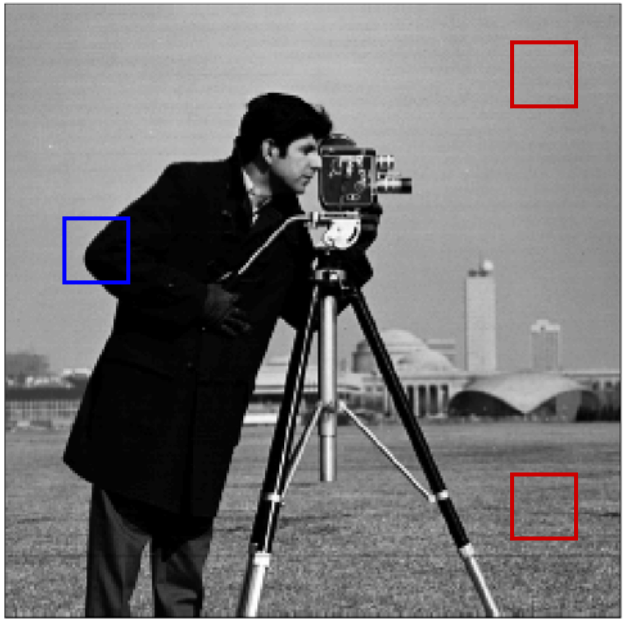
$$(\log R)^2 \; R^{-2}$$

for leaf-only encoding



Joint Texture/Geometry Modeling





- Dictionary *D* = {wavelets} U {wedgelets}

- Representation tradeoff:
  texture vs. geometry

- Test case: *approximation / compression*

Subtracting Doesn't Work

wedgelet approximation

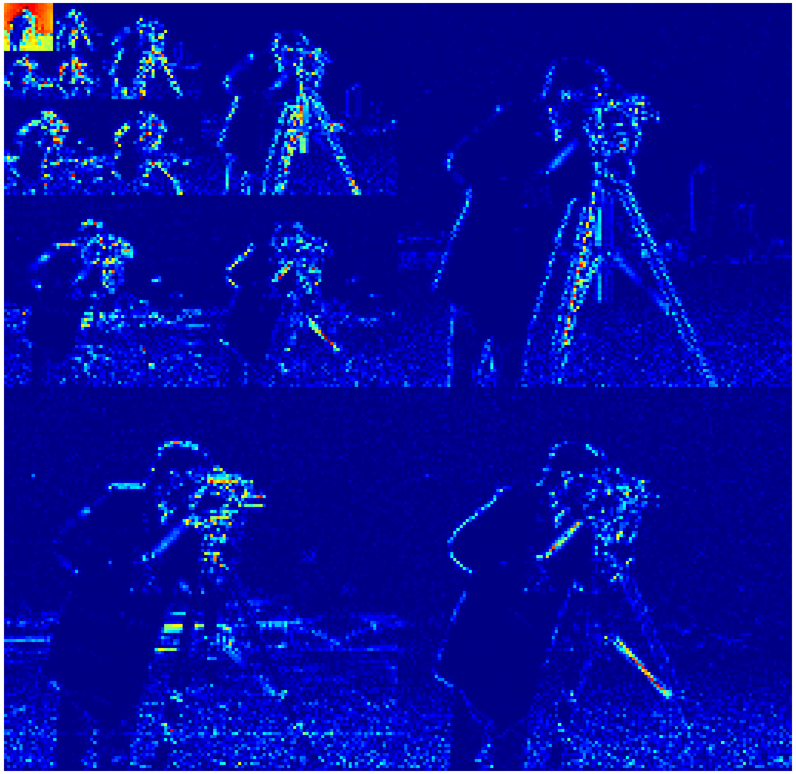
residual = f – w.a.

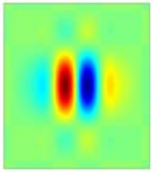
* *ridge artifacts just as hard to approx/code as edges*

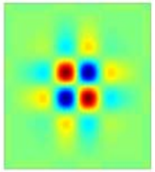
Part II

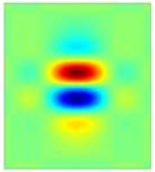
Wedding Wedgelets with Wavelets

Wavelet Representation







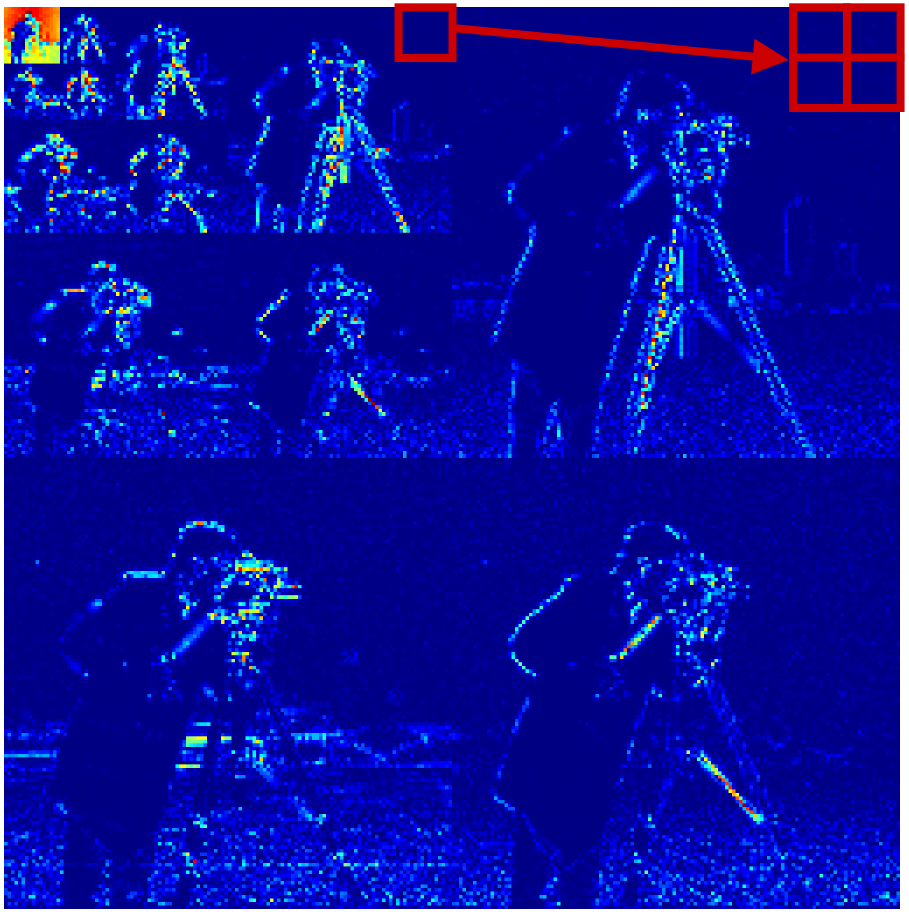


- Standard 2-D tensor product *wavelet transform*

$$f = \sum_k a_k \mathbf{b}_k$$

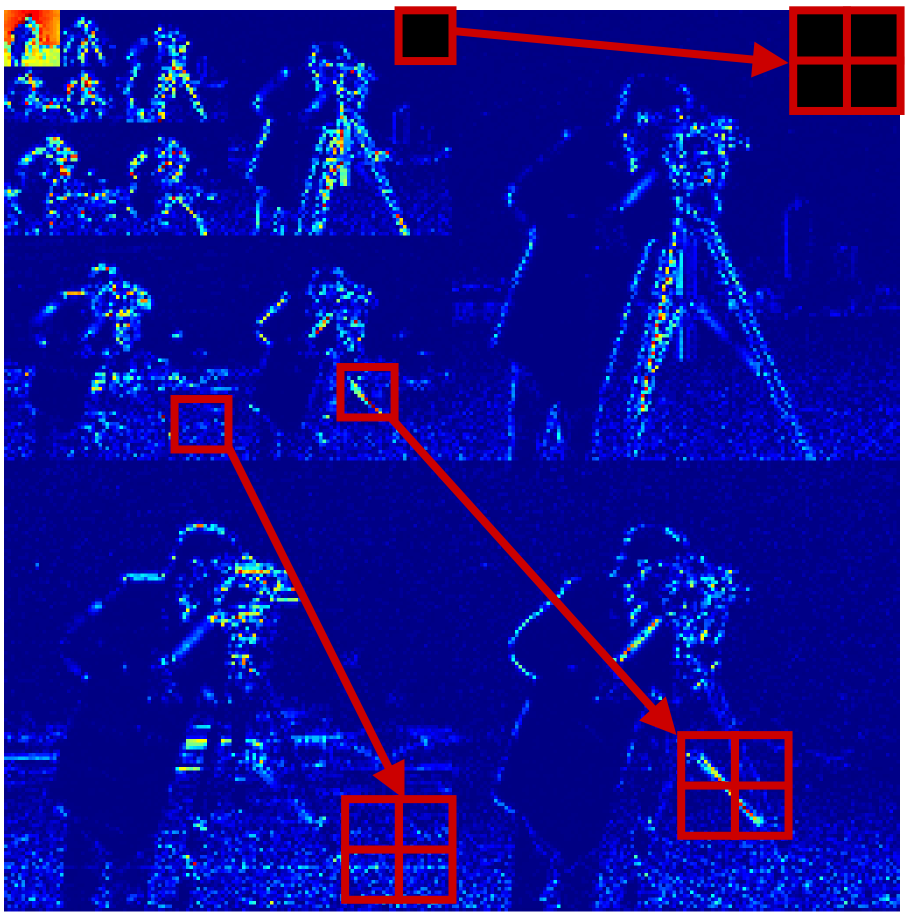
Wavelet Quadtrees

- Wavelet coefficients structured on *quadtree*
  - each *parent* has *4 children* at next finer scale



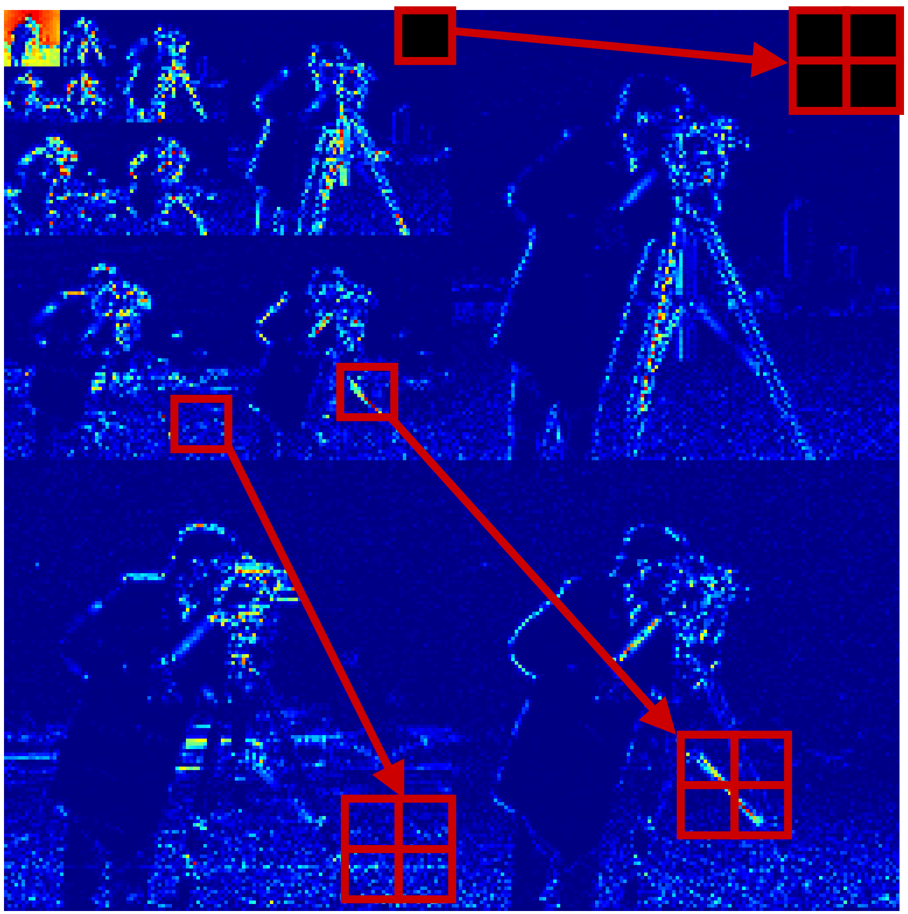
Wavelet Persistence

- *Smooth* region - *small* values down tree
- *Singularity*/*texture* - *large* values down tree



Zero Tree Approximation

- Idea: *Prune* wavelet subtrees in smooth regions
  - *tree-structured thresholding*



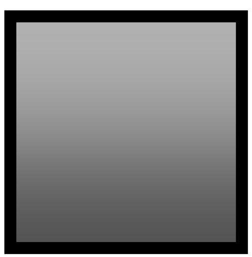
Zero Tree Approximation

- Label pruned wavelet quadtree with 2 states

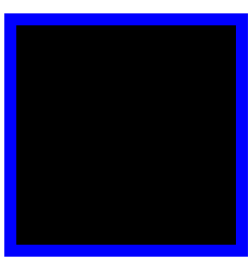
zero-tree	*- smooth region*	(prune)
significant	*- edge/texture region*	(keep)

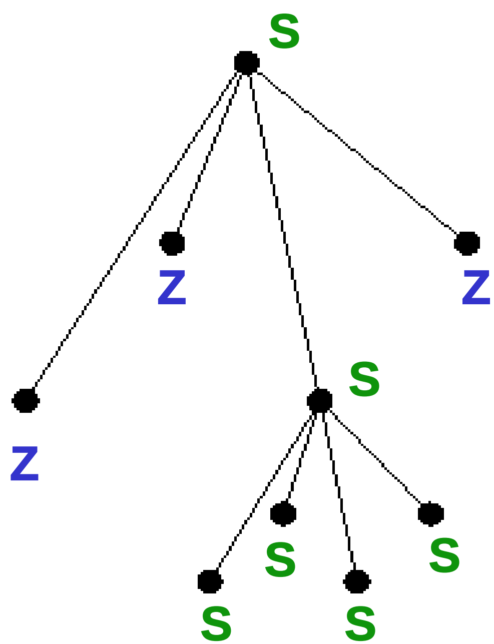
Z: all wc’s below=0

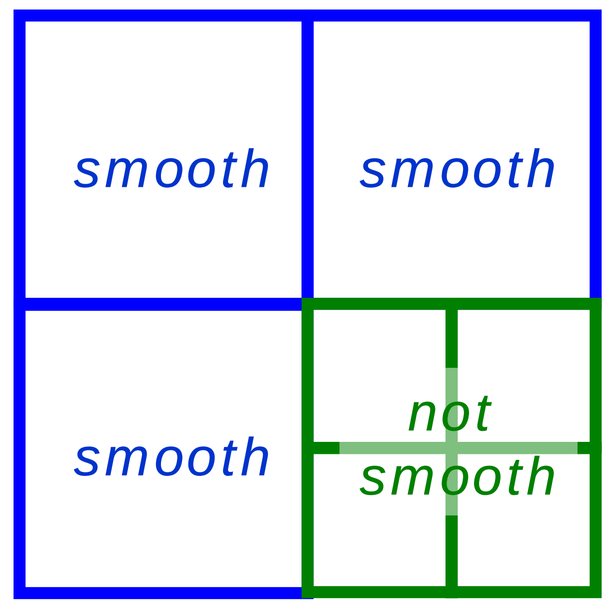
ie: wc’s of



+*atom*





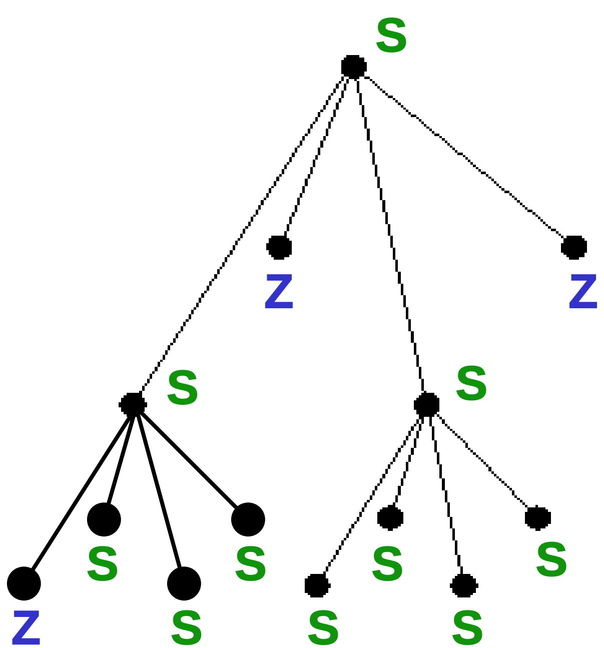


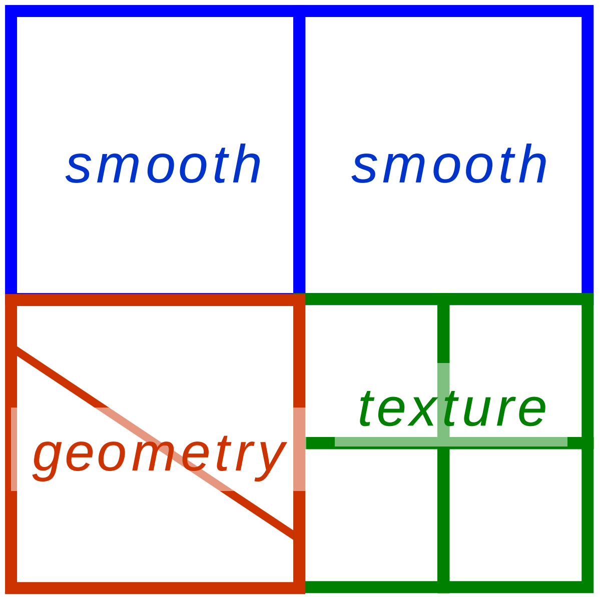
Enter Geometry

- Label pruned wavelet quadtree with 2 states

zero-tree	*- smooth region*	(prune)
significant	*- edge/texture region*	(keep)

- *Suboptimal* NLA/R-D decay w/ edges $D(R) \sim R^{-1}$



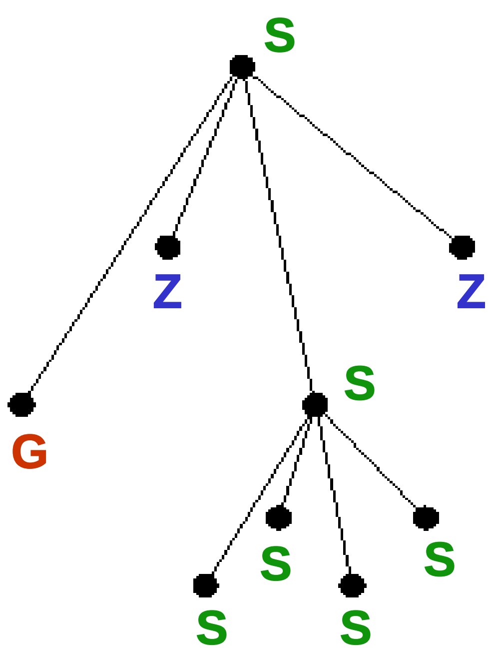


Wedgelets for Geometry

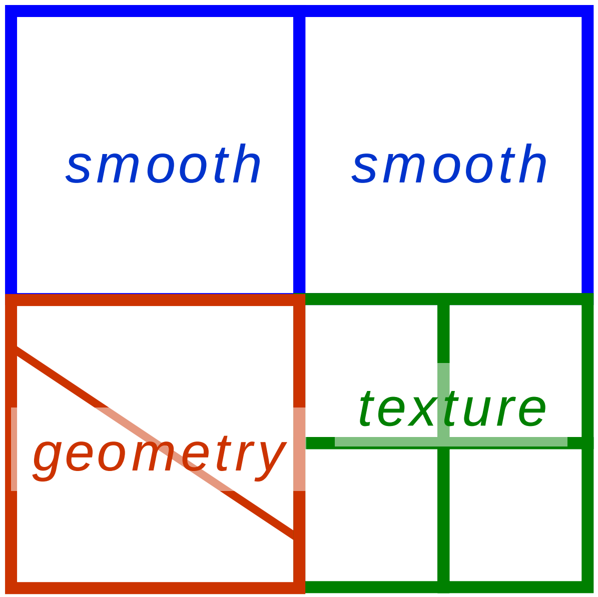
- Label pruned wavelet quadtree with 2 states

zero-tree	- *smooth region*	(prune)
significant	- *edge/texture region*	(keep)

- Idea: use wedgelets in *geometric* edge squares



G: wc's below
are wc's of a
wedgelet

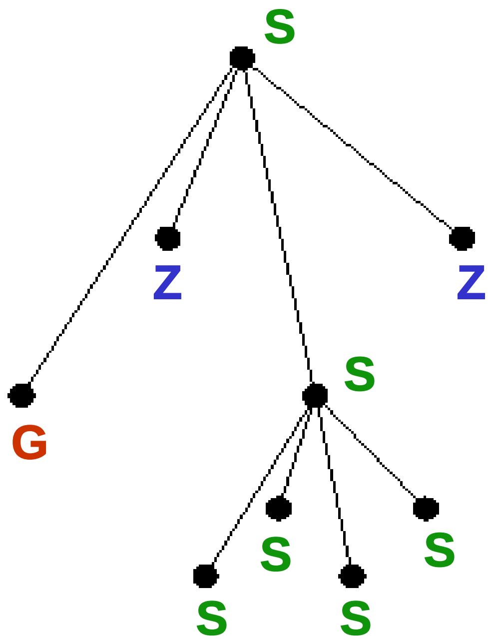


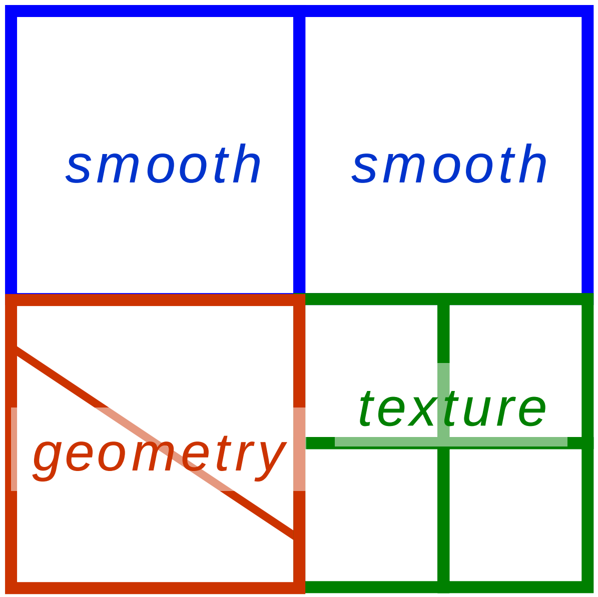
Wedgelets for Geometry

- Label pruned wavelet quadtree with 3 states

zero-tree	- *smooth region*	(prune)
geometry	- *edge region*	(prune)
significant	- *texture region*	(keep)

G: wc's below are wc's of a *wedgelet*



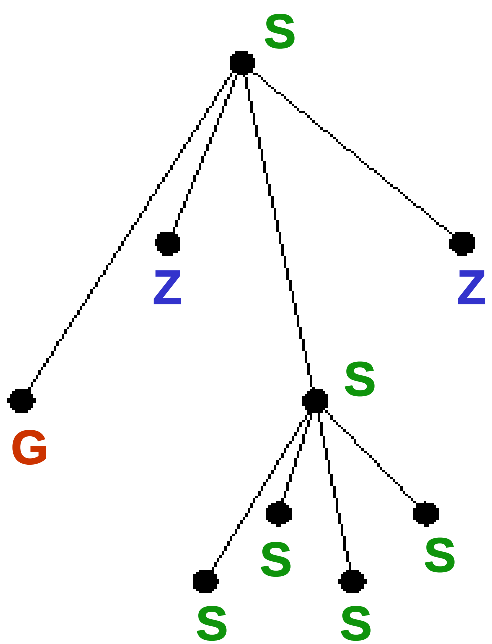


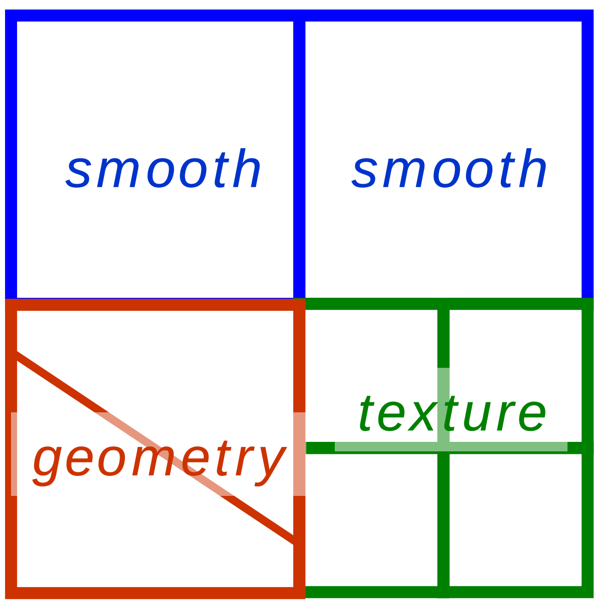
Wedgelets for Geometry

- Label pruned wavelet quadtree with 3 states

zero-tree	- *smooth region*	(prune)
geometry	- *edge region*	(prune)
significant	- *texture region*	(keep)

- Optimize placement of **Z**, **G**, **S** by dyn. programming





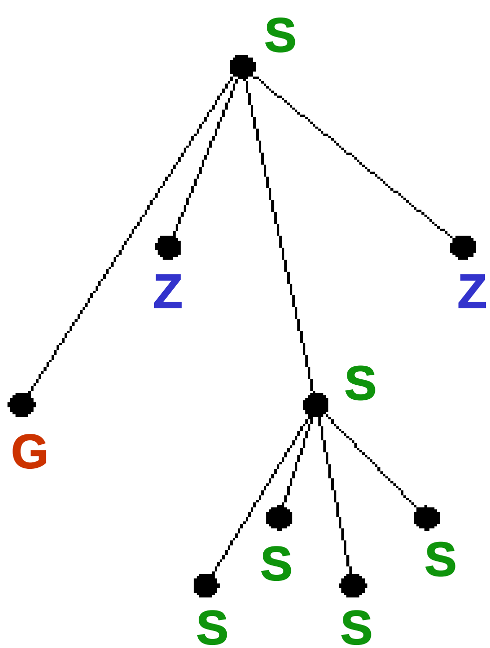
G: wc's below are wc's of a *wedgelet*

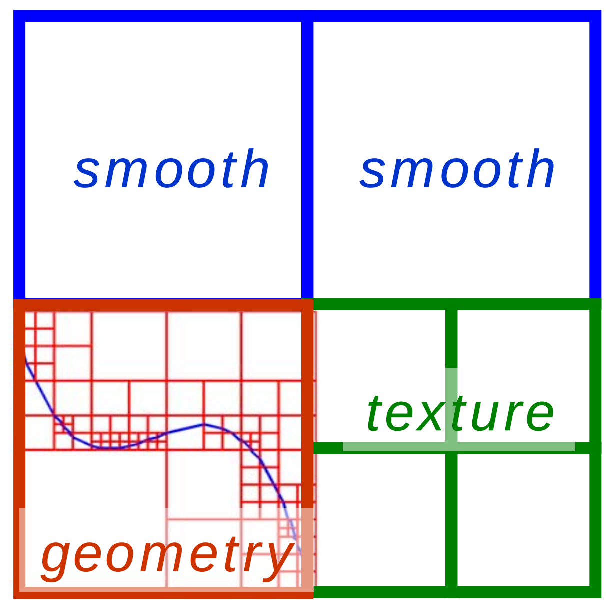
Wedgelet Trees for Geometry

- Label pruned wavelet quadtree with 3 states

zero-tree	- *smooth region*	(prune)
geometry	- *edge region*	(prune)
significant	- *texture region*	(keep)

- Optimize placement of **Z**, **G**, **S** by dyn. programming



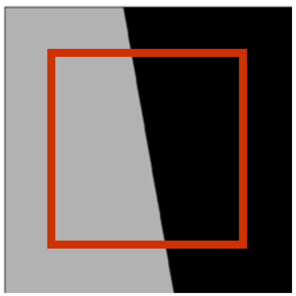


G: wc's below are wc's of a *wedgelet tree*

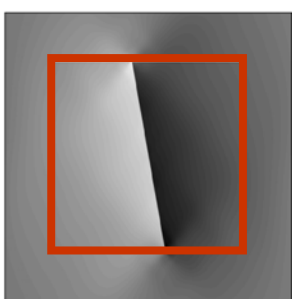
Image Approximation

G: wc's below
are wc's of a
wedgelet

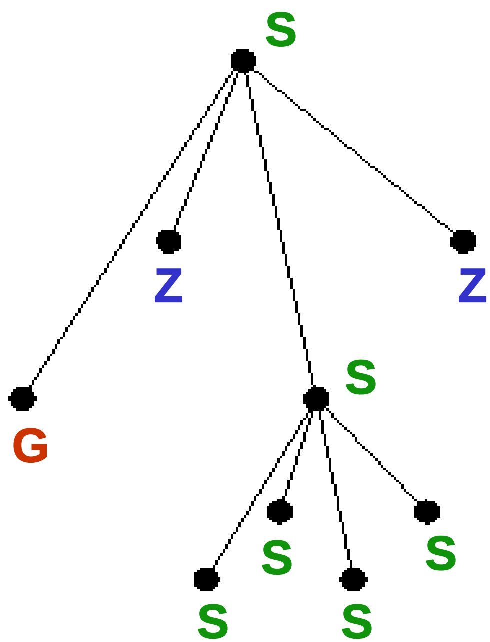
ie: wc's of

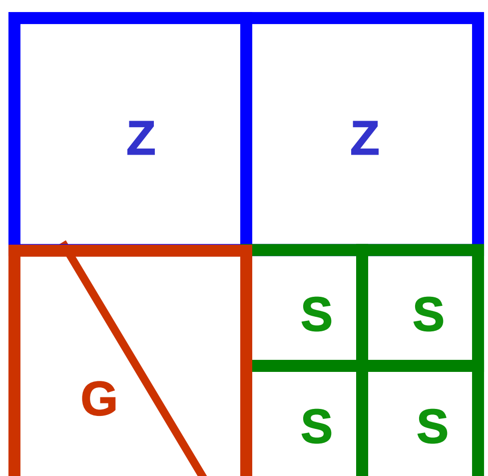


+*atom*



"wedgeprint"





Z: all wc's below=0

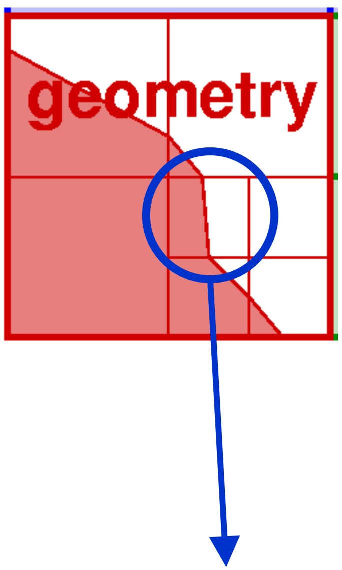
ie: wc's of

+*atom*

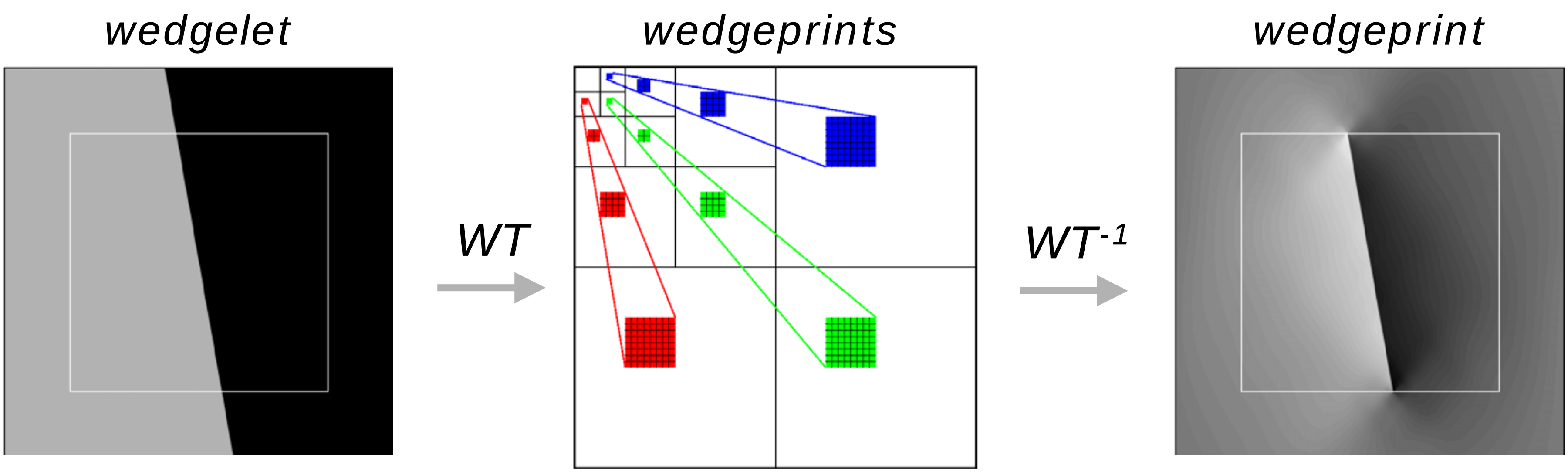
"smoothprint"

S: describe
(code) wc's

Wedgeprints



- Wedgelet projected onto wavelet subtree yields an *adaptive atom* matched to local *edge geometry*
- *Wedgeprint* collapses many wc's (entire subtree) into a *single* oriented atom
- Akin to wavelet *vector quantization*



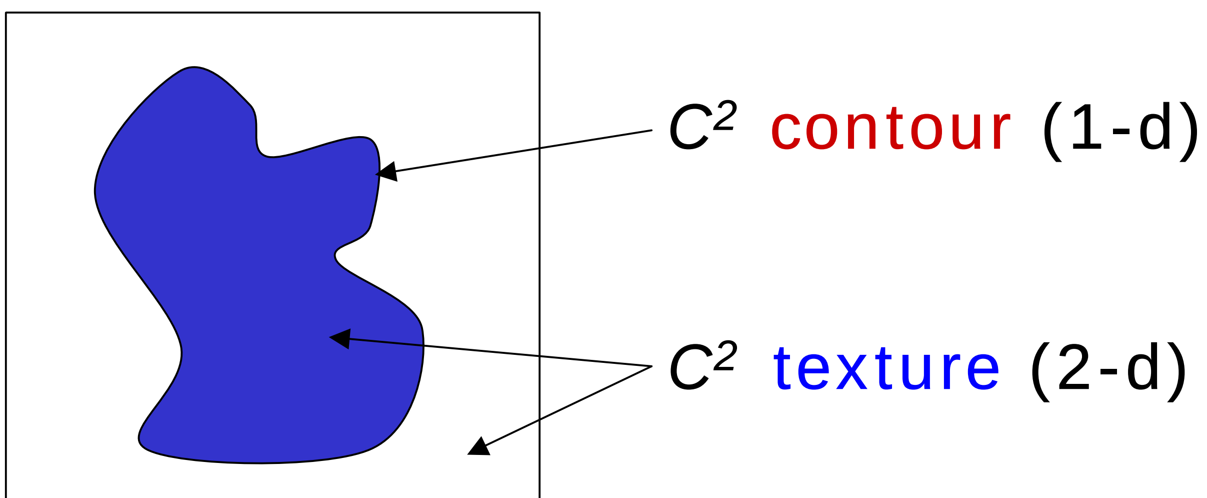
Optimality of Wedgeprints

- **Theorem** For C^2 / C^2 images, *optimal* asymptotic L^2 error decay

$$\|f - \widehat{f}_N\|_2^2 \; < \; C\, N^{-2}$$

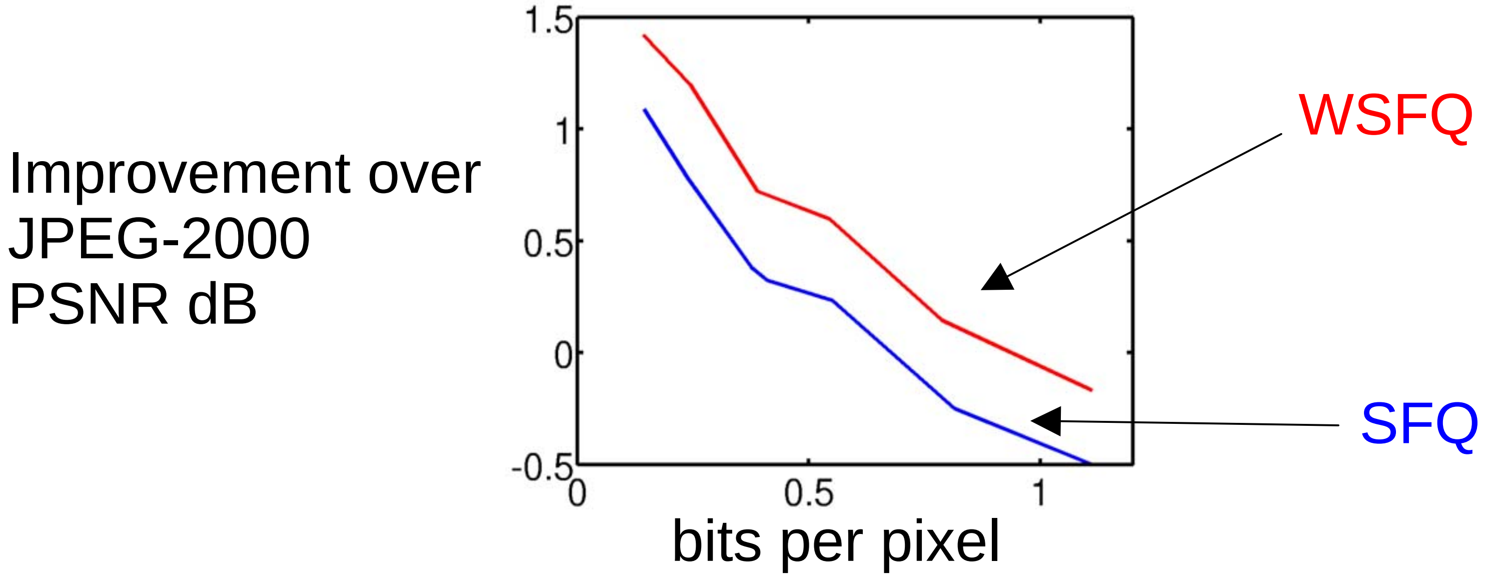
and *near-optimal* rate-distortion

$$D(R) = \log(R)^2\, R^{-2}$$



Practical Image Coder

- *Wedgelet-SFQ (WSFQ)* coder
  builds on SFQ coder [Xiong, Ramchandran, Orchard]



- At low bit rates, often significant improvement in visual quality over SFQ and JPEG-2k (much sharper edges)

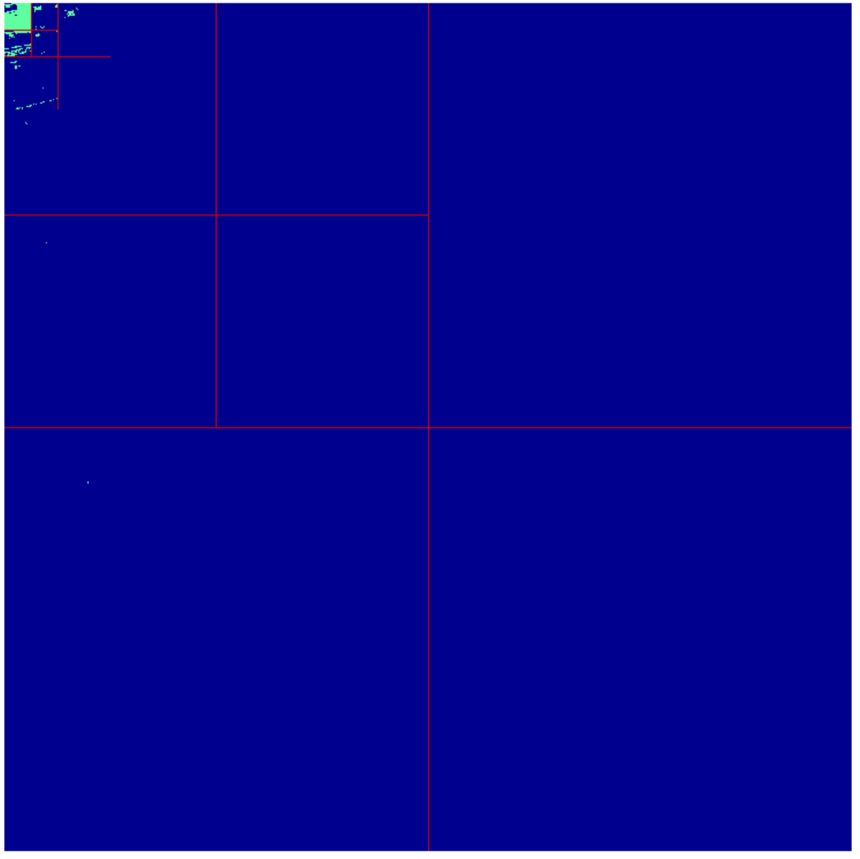
- **Bonus**: WSFQ representation contains *explicit geometry information*

Wet Paint Test Image

SFQ Compressed

PSNR = 29.77dB @ 0.0103bpp

SFQ WC Code Map



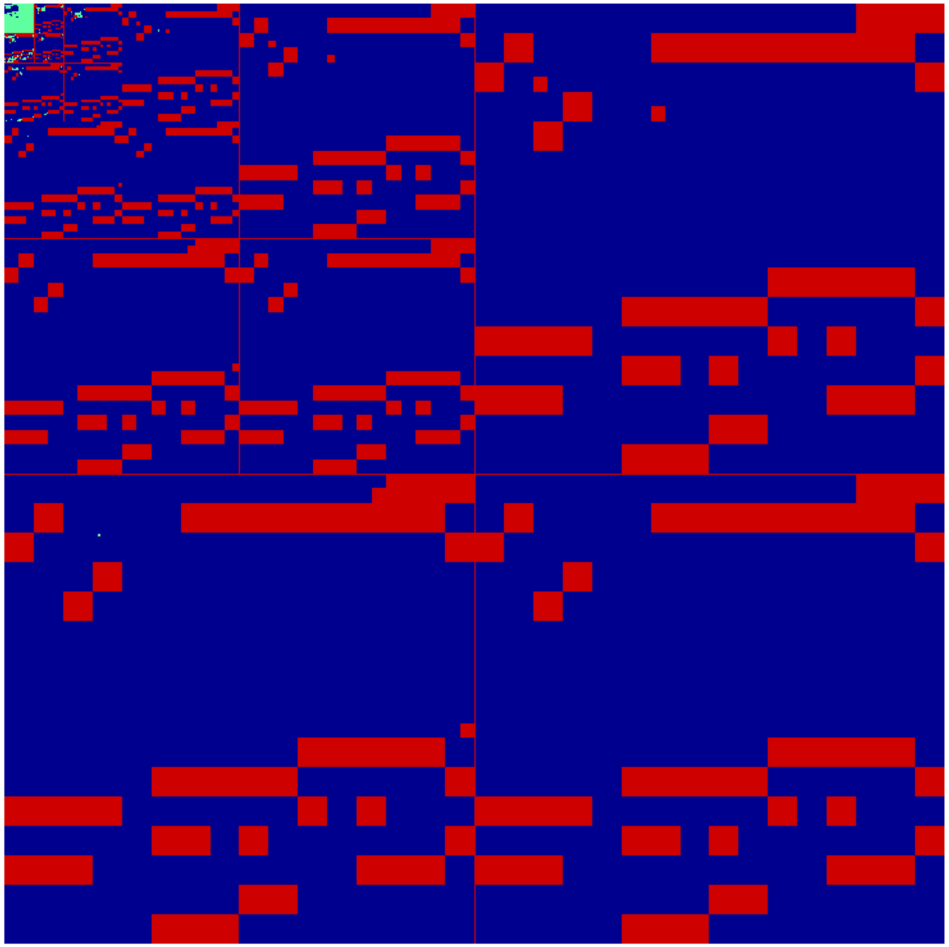
green = significant   blue = zero tree

SFQ Zoom

WSFQ Compressed

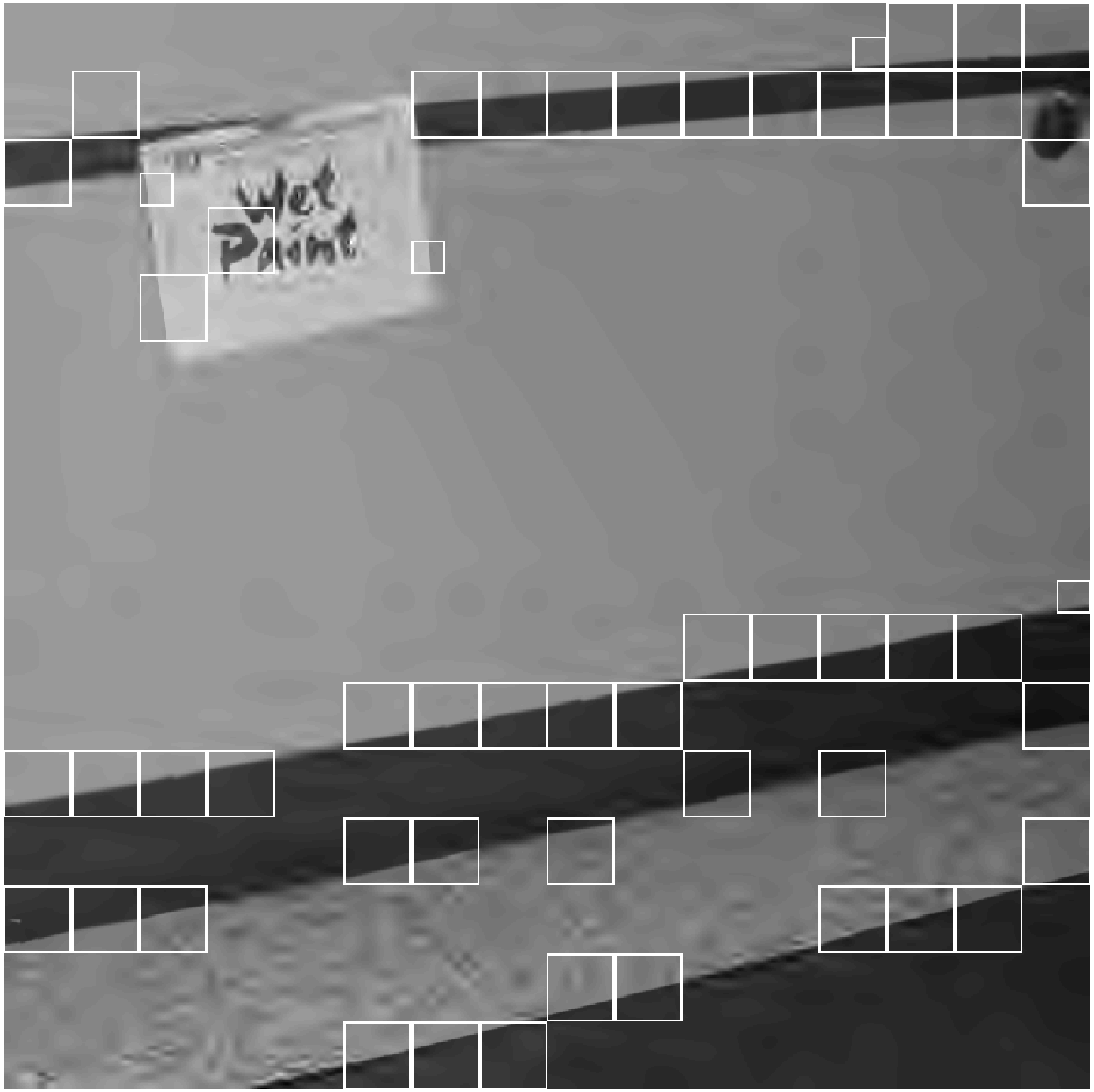
PSNR = 30.19dB @ 0.0102bpp

WSFQ WC Code Map

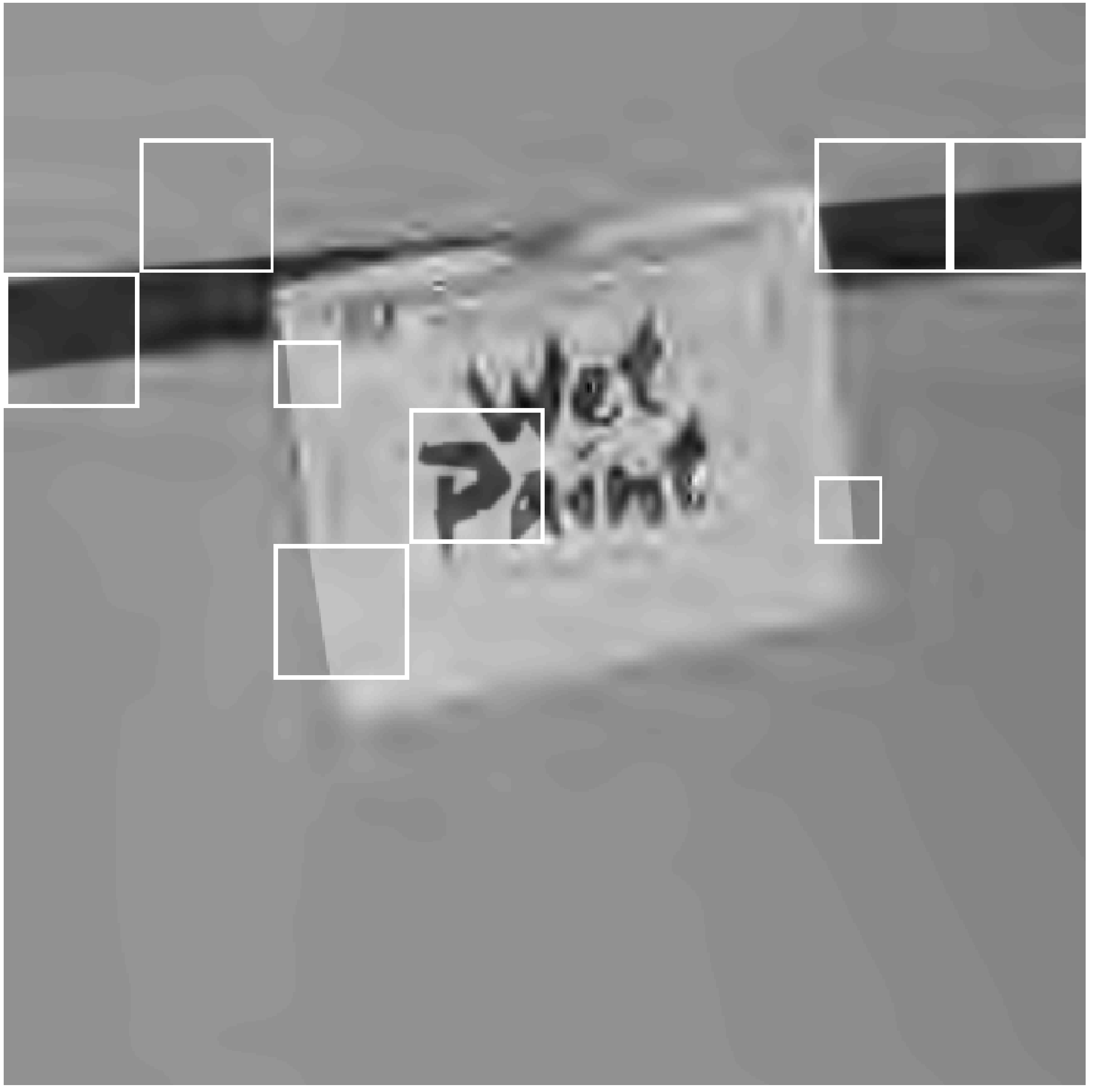


green=significant   blue=zero tree   red=contour tree

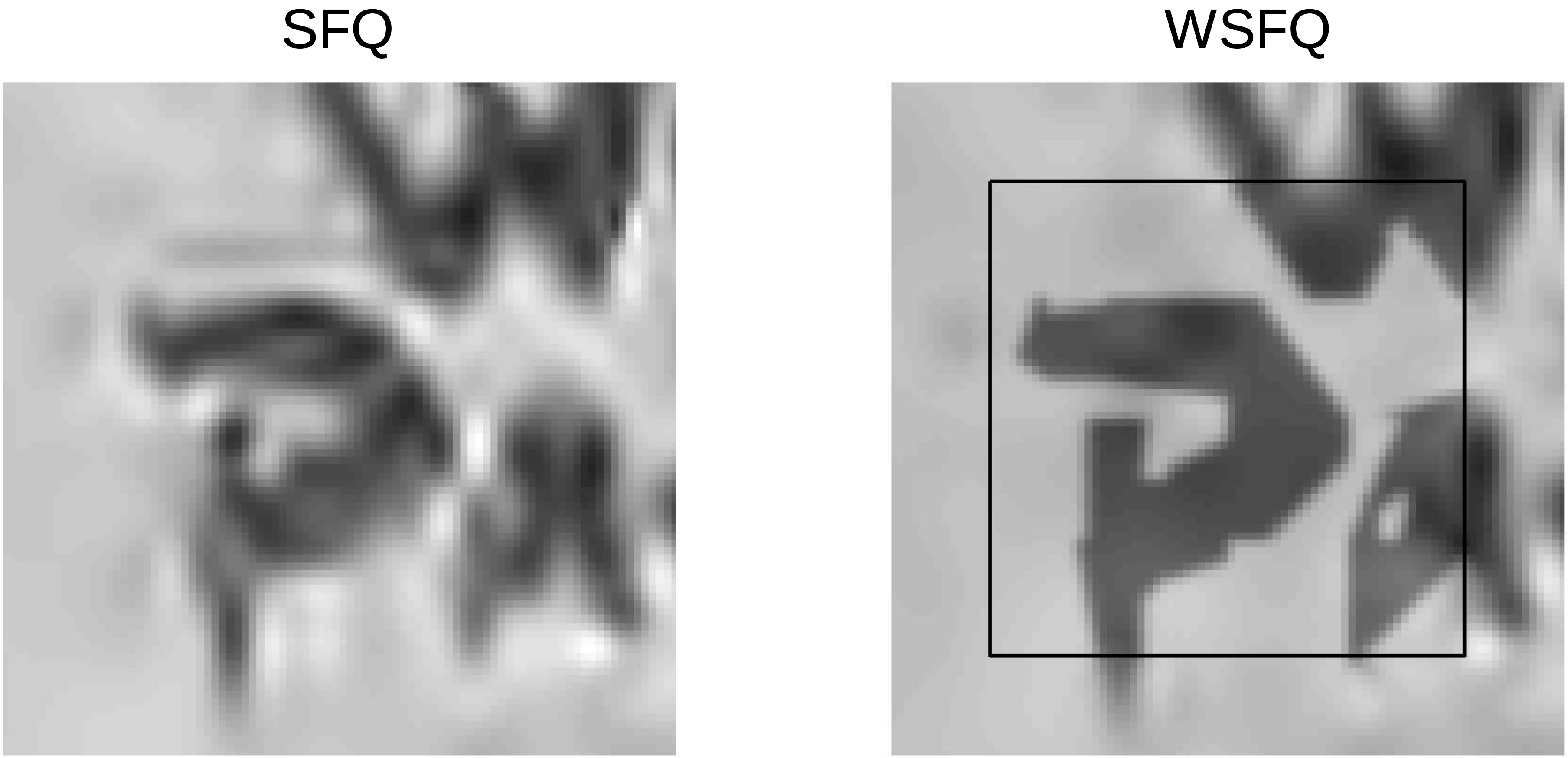
WSFQ Contour Trees



WSFQ Zoom

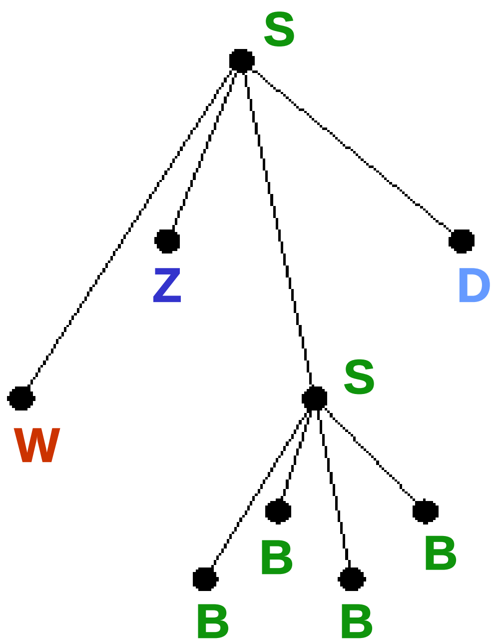


SFQ vs. WSFQ

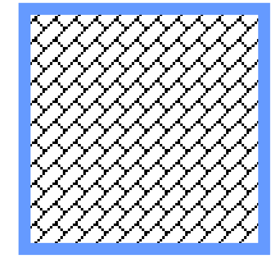


Extensions

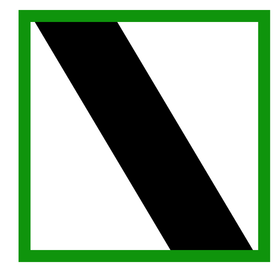
zerotree



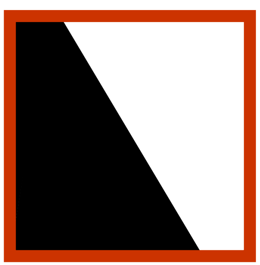
DCTprint

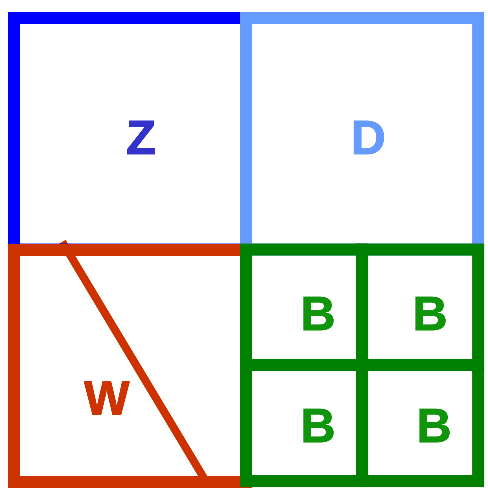


barprint



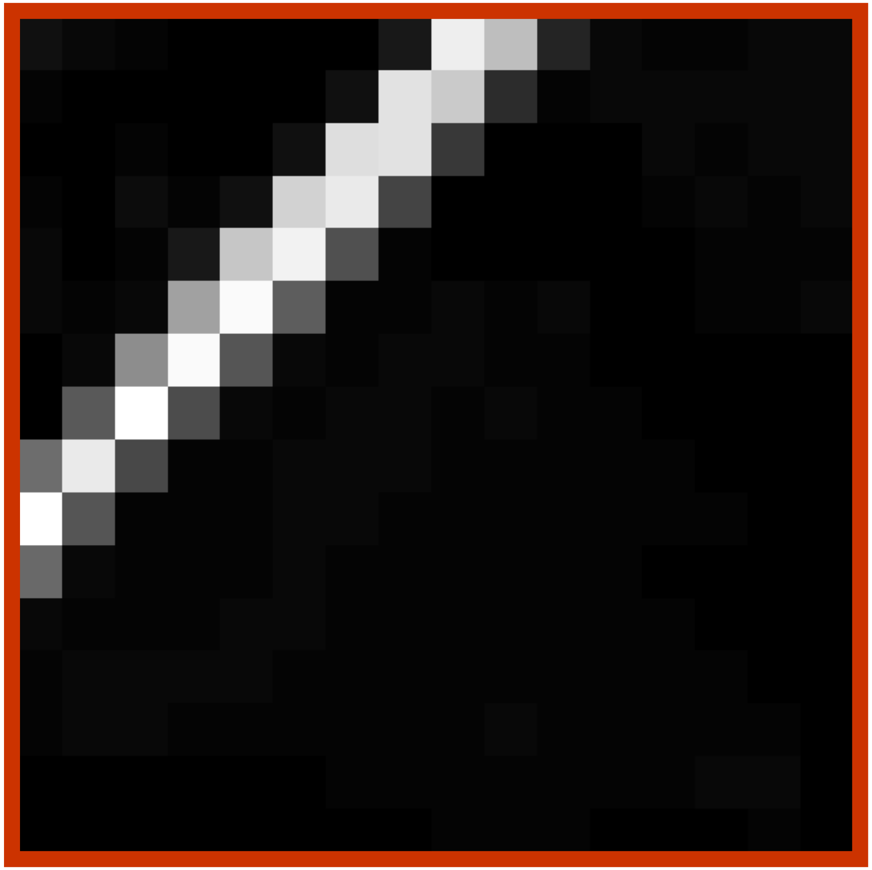
wedgeprint





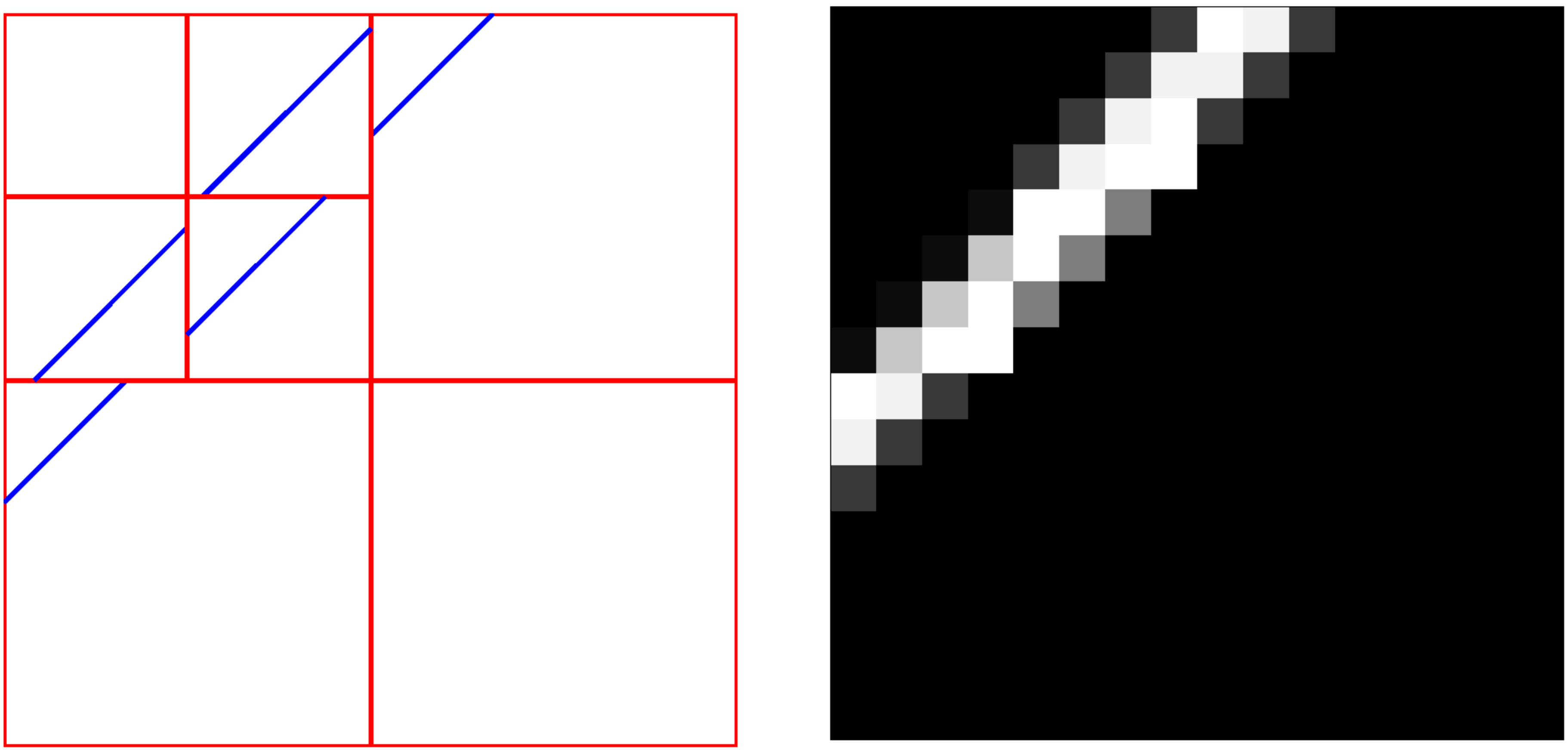
"Coifman's Dream"

Bars / Ridges



16x16 image block

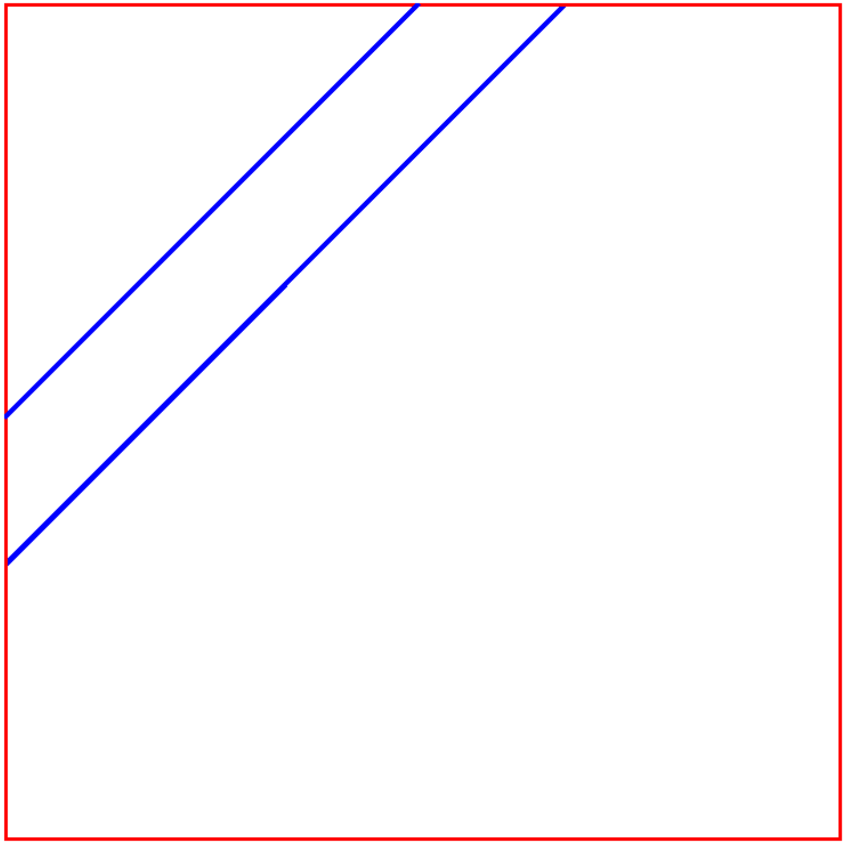
Multiple Wedgelet Coding

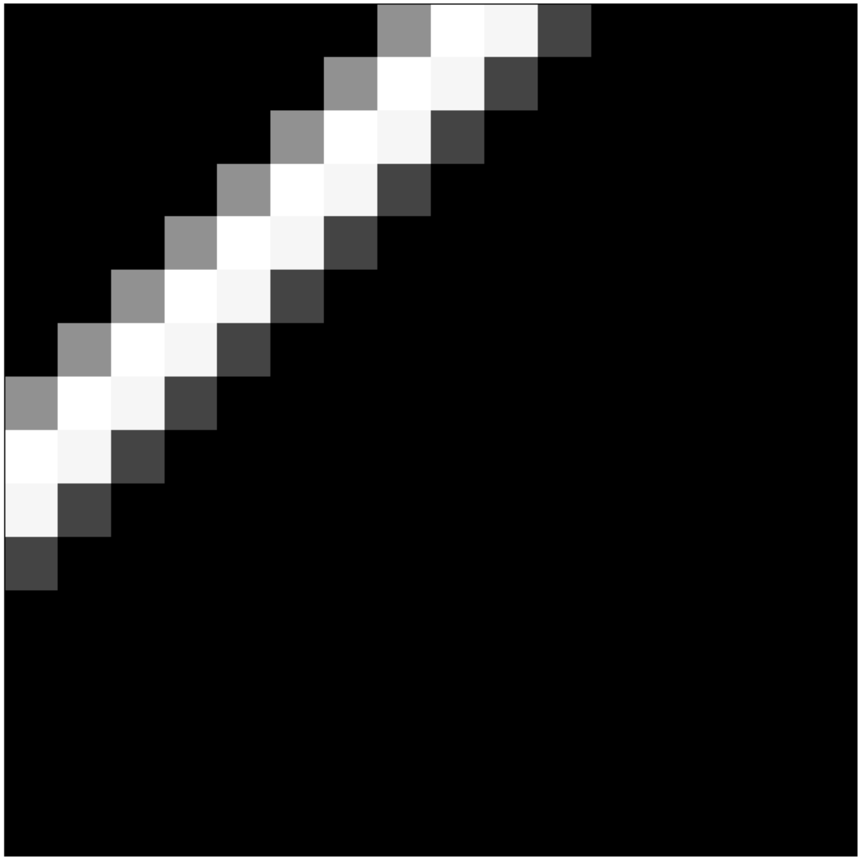


80 bits to jointly encode 5 wedgelets

“Barlet” Coding

(fat edgelet/beamlet)

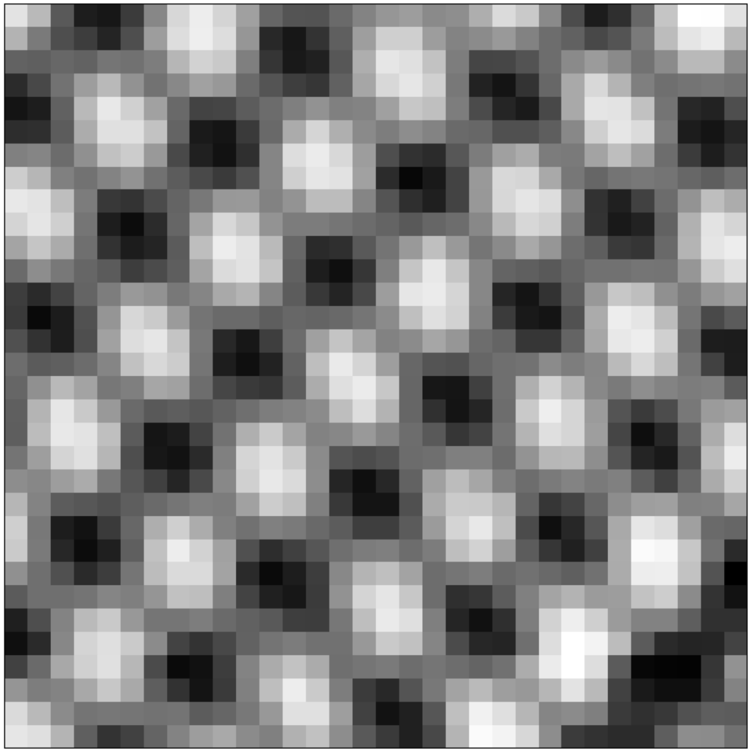




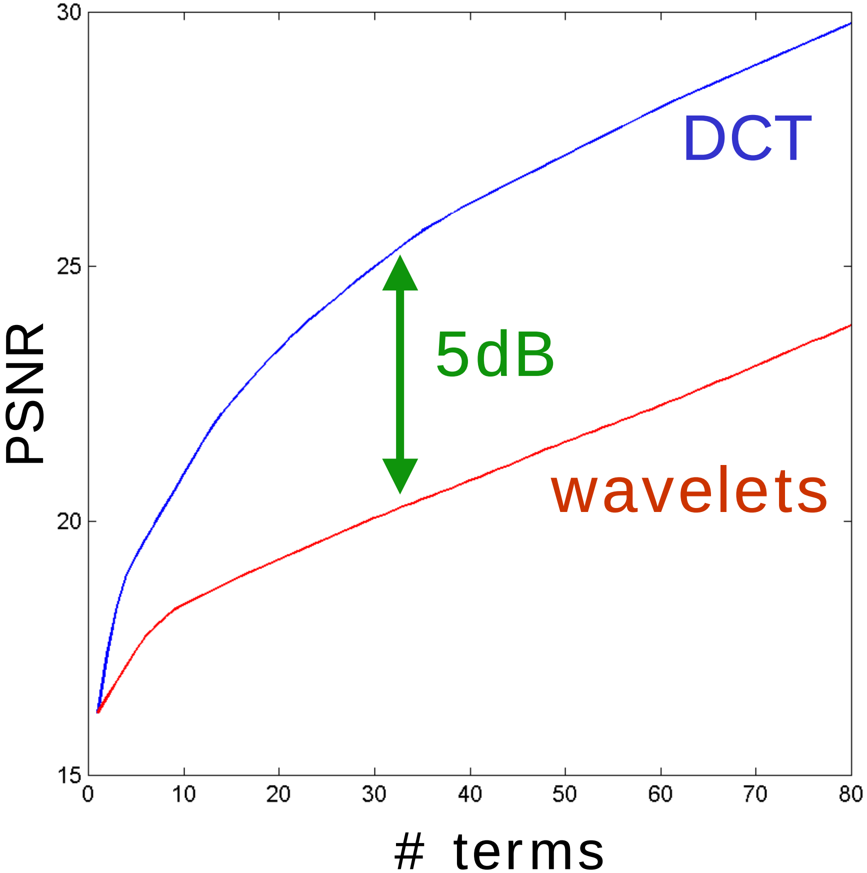
22 bits to encode 1 barlet

Periodic Textures

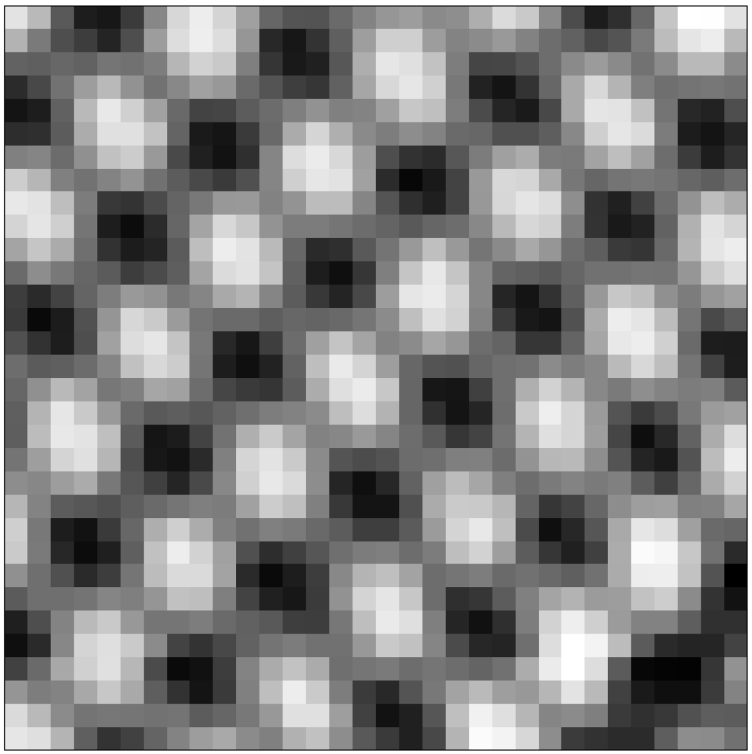
DCTprint



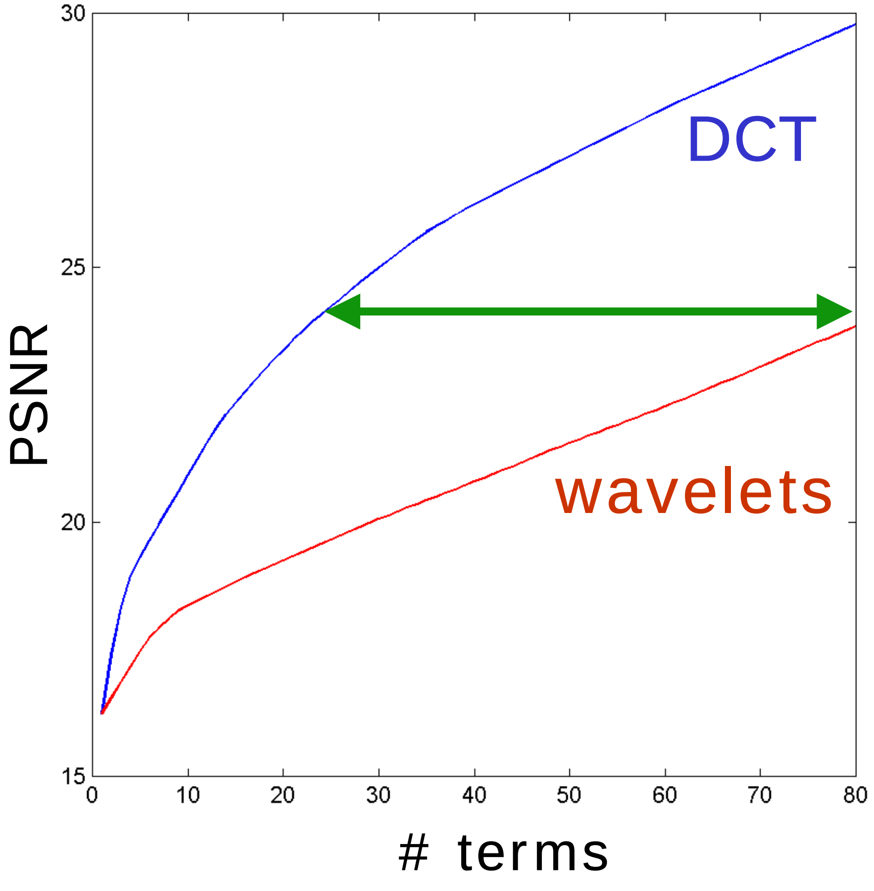
32x32 block = 1024 pixels



DCTprint

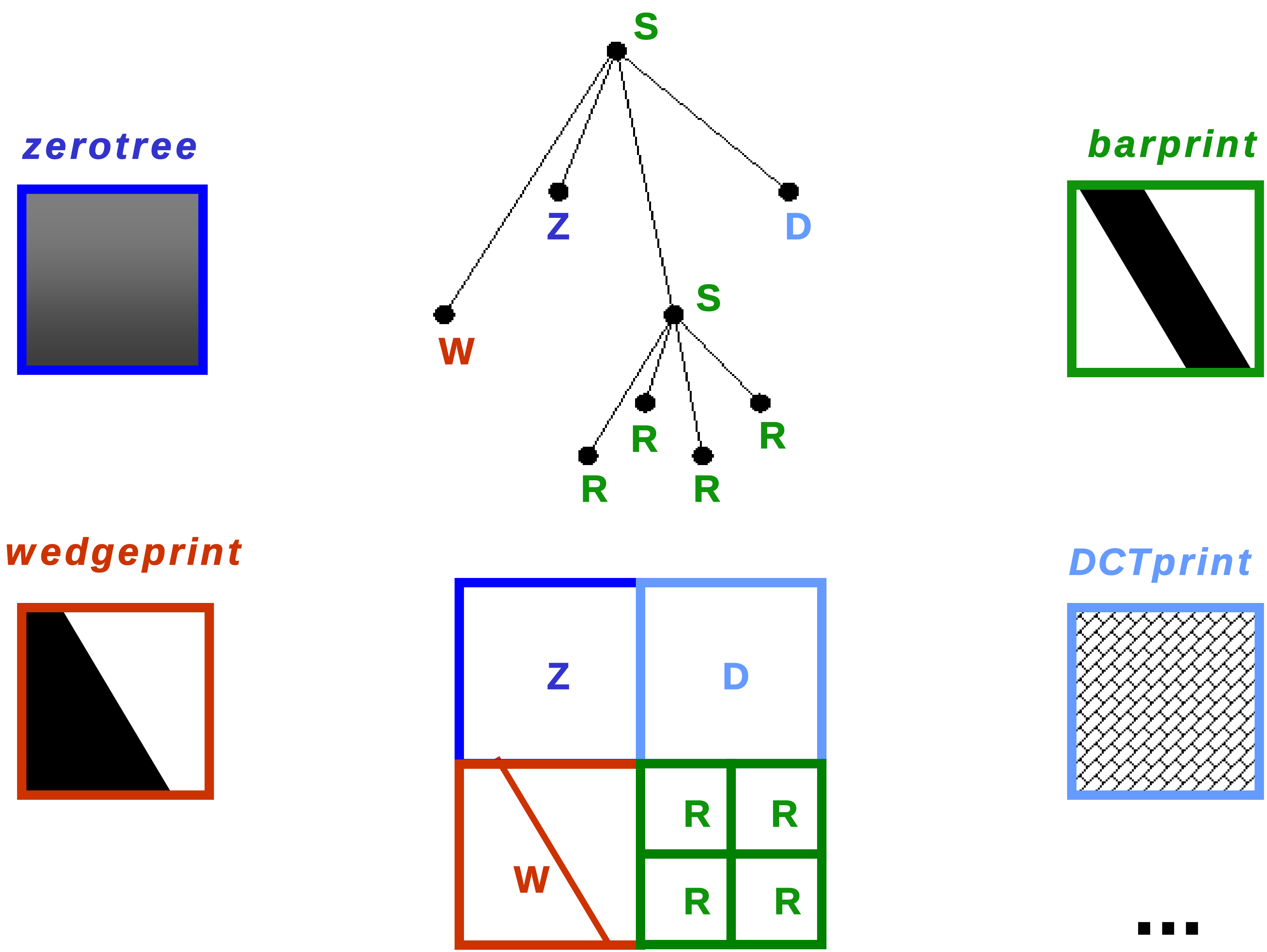


32x32 block = 1024 pixels



4x fewer coefficients

Extensions

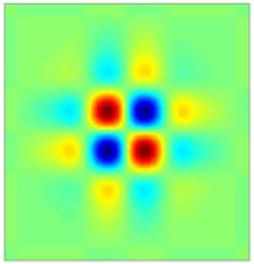


zerotree
barprint
S
Z
D
W
S
R
R
R
R
wedgeprint
DCTprint
Z
D
W
R
R
R
R
...

Conclusions

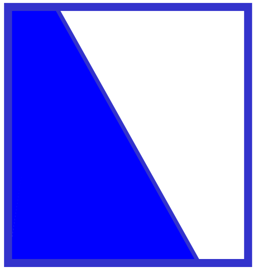
- Capturing *geometrical information* in images requires new tools

wavelets

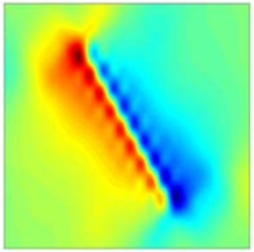


basis

wedgelets



model

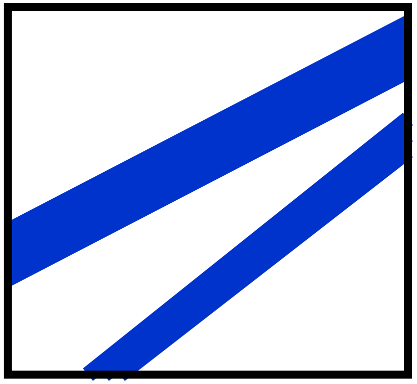


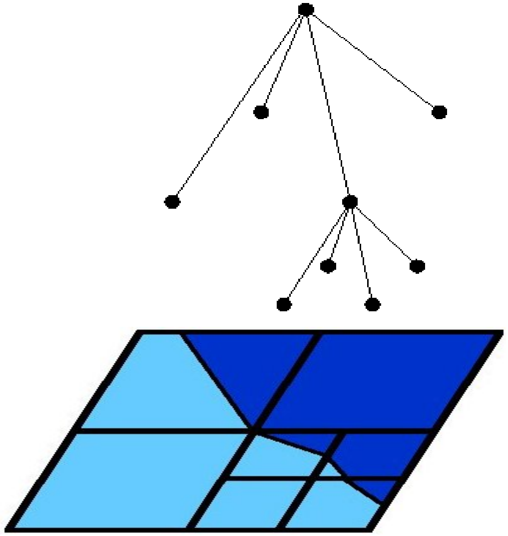
wedgeprints

- *Optimal approximation* of C^2/C^2 function class

Isotropic Anisotropy?

- Wedgeprints+MGM achieve optimal performance of *anisotropic* curvelets
- But wavelet/wedgelet tiles are *isotropic* (based on dyadic squares, quadtree)
- Ultimately could prove a curse for closely spaced geometrical features





Related Work

- Multiresolution Fourier transform
  [Calway, Pearson, Wilson]

- “Prune and join” quadtree approximation
  [Shukla, Dragotti, Do, Vetterli]

- Platelets
  [Willett and Nowak]

- Wavelet footprints
  [Dragotti, Vetterli]

- Bandelets
  [Le Pennec, Mallat]

dsp.rice.edu