# Lecture 2: Curvelets

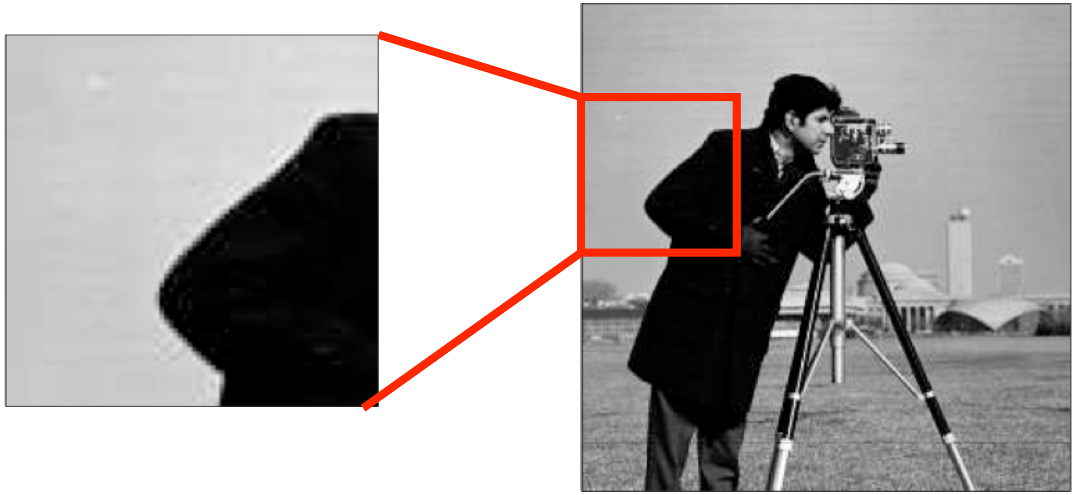
*Emmanuel Candès*, California Institute of Technology

# Sparsity and Applications

- We have seen that sparse representations are critical for
  - compression
  - estimation
  - inverse problems
- This talk: Curvelets, a sparse representation for images with geometrical structure

# Image Model

- Images of interest: smooth regions separated by smooth contours



- Geometrical fragment model:
  - smooth regions: $C^2$ functions of two variables
  - edge contour: $C^2$ functions of one variable

# Judging Image Representations

- Representation $\{\psi_i\}$

$$f(x_1, x_2) = \sum_i \alpha_i \psi_i(x_1, x_2)$$

- $f_m$ = best $m$-term approximation

$$f_m(x_1, x_2) = \sum_{i \in \Gamma(m)} \alpha_i \psi_i(x_1, x_2)$$

  where $\Gamma$ is chosen such that $|\Gamma| = m$ and $\|f - f_m\|_2^2$ is minimized

- How fast does $\|f - f_m\|_2^2 \to 0$?
- Fundamental limit:

$$\|f - f_m\|_2^2 \asymp m^{-2}$$

  – No basis can do better than this
  – No depth-search limited dictionary can do better
  – No pre-existing basis does anything near this well

# Fourier is Awful

- Discontinuities in the image lead to slow decay of the Fourier coefficients (edges have a lot of "high frequency content")
- $m$-term approximation error

$$\|f - f_m\|_2^2 \asymp m^{-1/2}$$

- Example:

original

1% of Fourier coeffs

10% of Fourier coeffs

# Wavelets are Bad

- Many wavelets are needed to represent an edge
  (number depends on the length of the edge, not the smoothness)

- $m$-term approximation error

$$\|f - f_m\|_2^2 \asymp m^{-1}$$

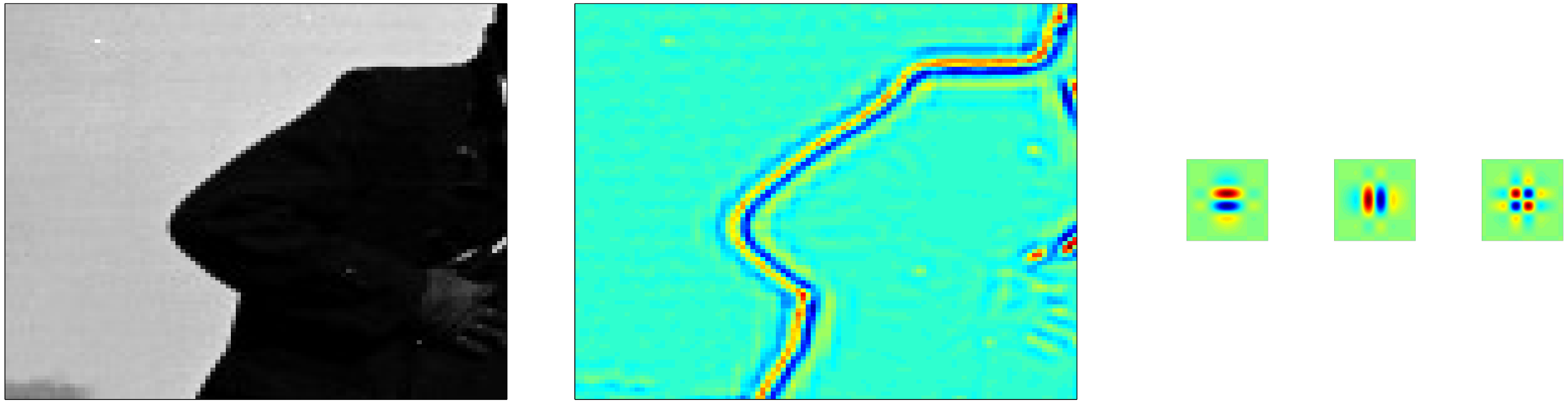
- Example:

original

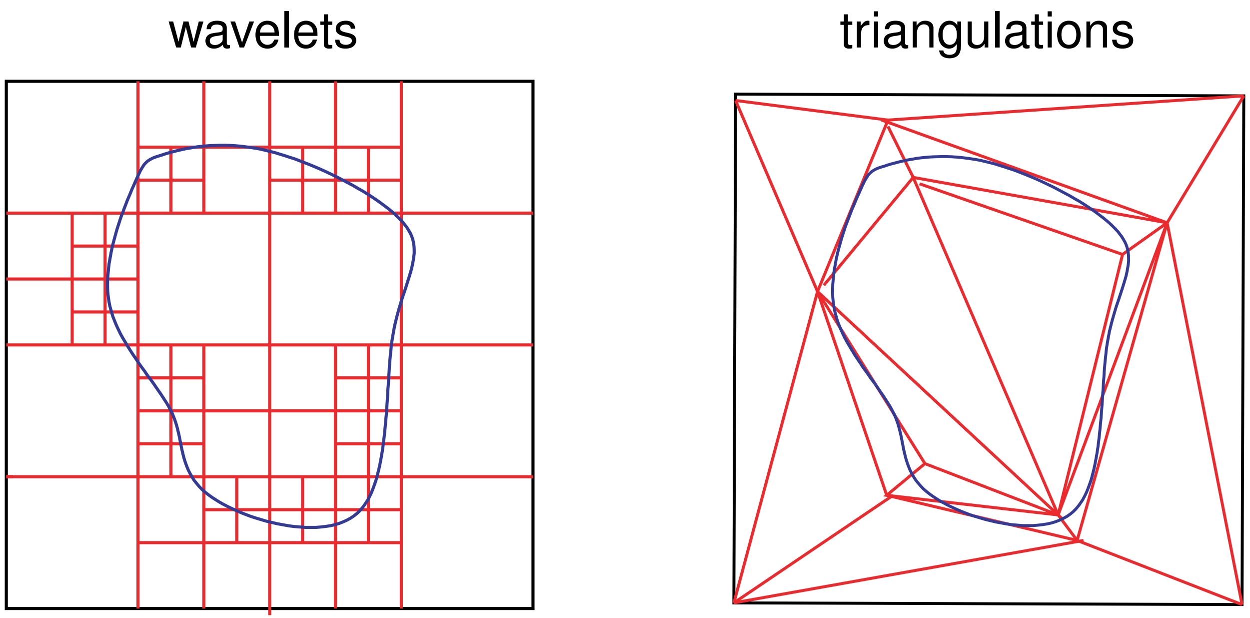
1% of wavelet coeffs

10% of wavelet coeffs

# Wavelets and Geometry



- Wavelet basis functions are isotropic
  $\Rightarrow$ they cannot "adapt" to geometrical structure




- We need a more refined scaling concept...

# Wavelet Pyramids

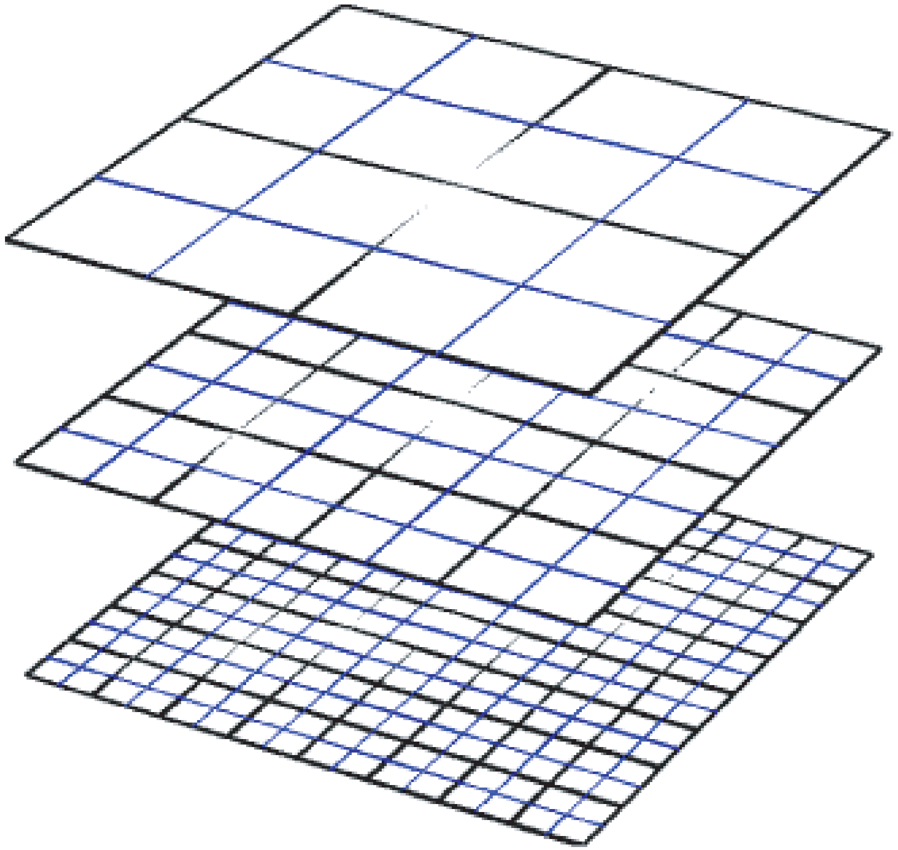
Canonical Pyramid Ideas (1980-present)

- Laplacian Pyramid (Adelson/Burt)
- Orthonormal Wavelet Pyramid (Mallat/Meyer)
- Steerable Pyramid (Adelson/Heeger/Simoncelli)
- Multiwavelets (Alpert/Beylkin/Coifman/Rokhlin)

Shared features

- Elements at dyadic scales/locations
- Fixed number of elements at each scale/location

# Wavelet Pyramid



# Limitations of Existing Scaling Concepts

Traditional Scaling

$$f_a(x_1, x_2) = f(ax_1, ax_2), \qquad a > 0.$$

Curves exhibit different kinds of scaling

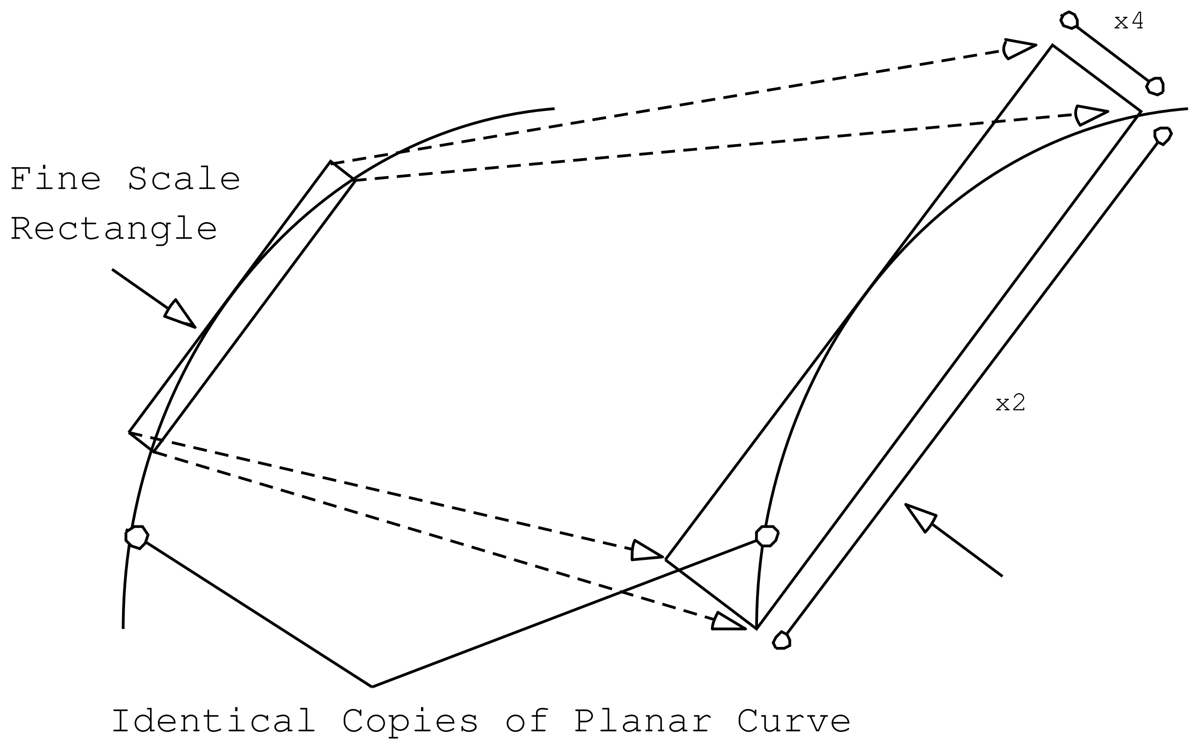
- Anisotropic
- Locally Adaptive

If $f(x_1, x_2) = 1_{\{y \geq x^2\}}$ then

$$f_a(x_1, x_2) = f(a \cdot x_1, a^2 x_2)$$

In Harmonic Analysis called Parabolic Scaling.

# Curves are invariant under anisotropic scaling



# Curvelets

C. and Donoho, 1999–2004

New multiscale pyramid:

- Multiscale
- Multi-orientations
- *Parabolic (anisotropy) scaling*

$$width \approx length^2$$

# Space-side Picture

- Start with a waveform $\varphi(x) = \varphi(x_1, x_2)$.
  - oscillatory in $x_1$
  - lowpass in $x_2$
- Parabolic rescaling

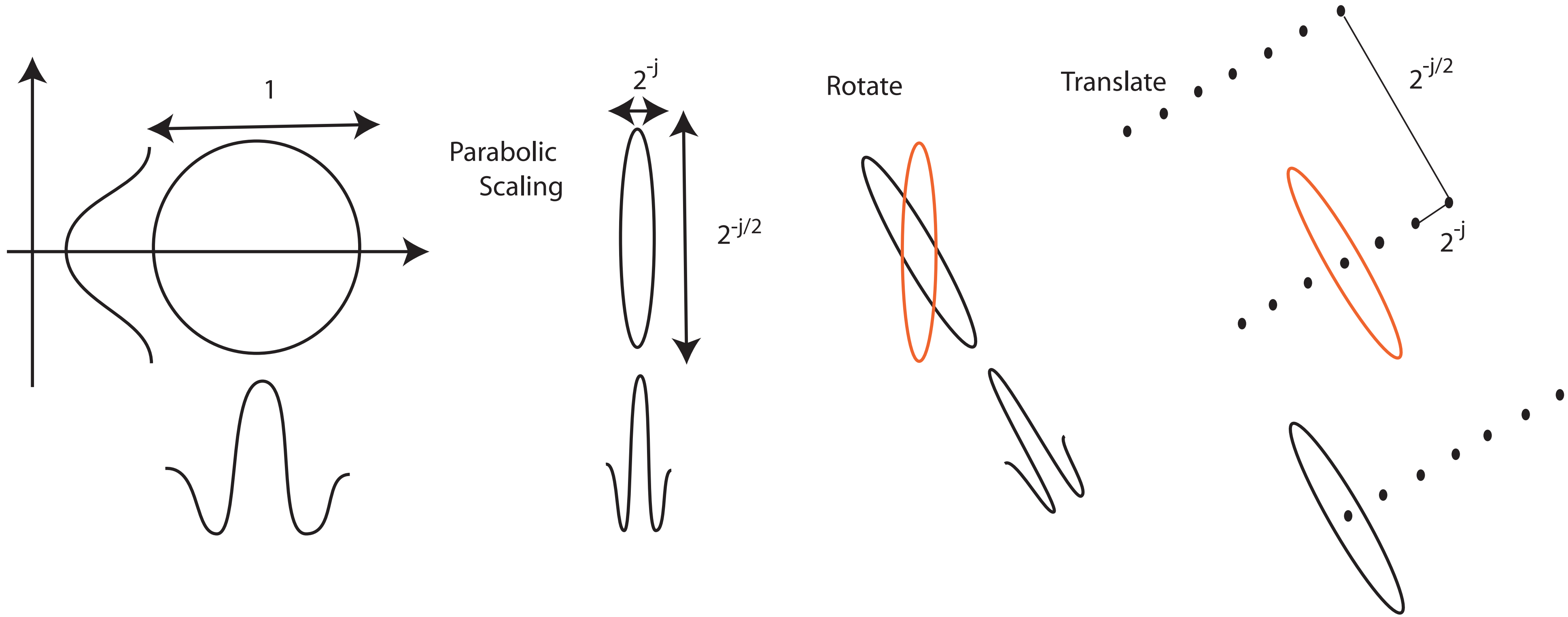
$$|D_j|\varphi(D_j x) = 2^{3j/4}\varphi(2^j x_1, 2^{j/2} x_2), \quad D_j = \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}, \; j \geq 0$$

- Rotation (scale dependent)

$$2^{3j/4}\varphi(D_j R_{\theta_{j\ell}} x), \quad \theta_{j\ell} = 2\pi \cdot \ell 2^{-\lfloor j/2 \rfloor}$$

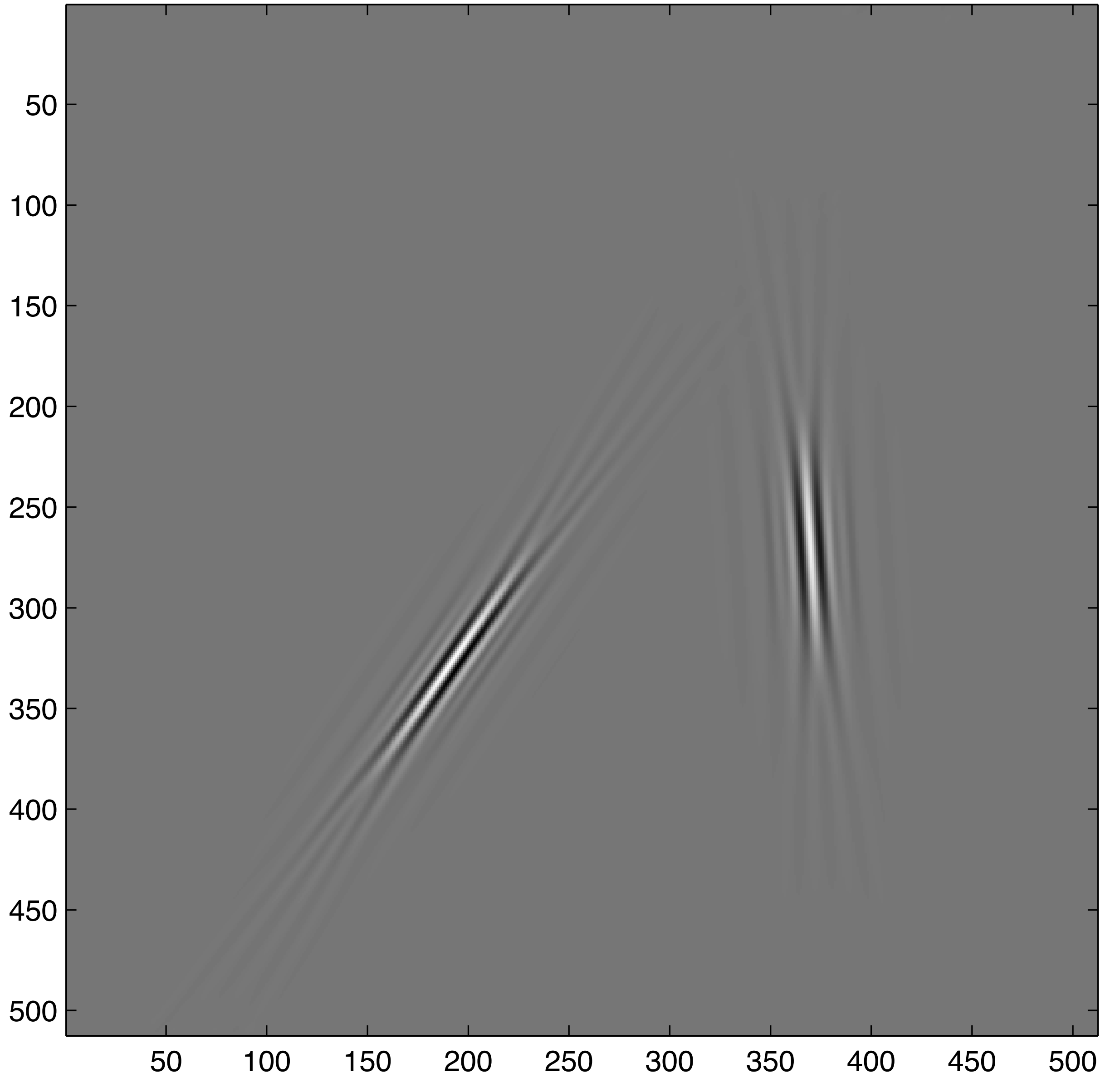
- Translation (oriented Cartesian grid with spacing $2^{-j} \times 2^{-j/2}$);

$$2^{3j/4}\varphi(D_j R_{\theta_{j\ell}} x - k), \quad k \in \mathbb{Z}^2$$

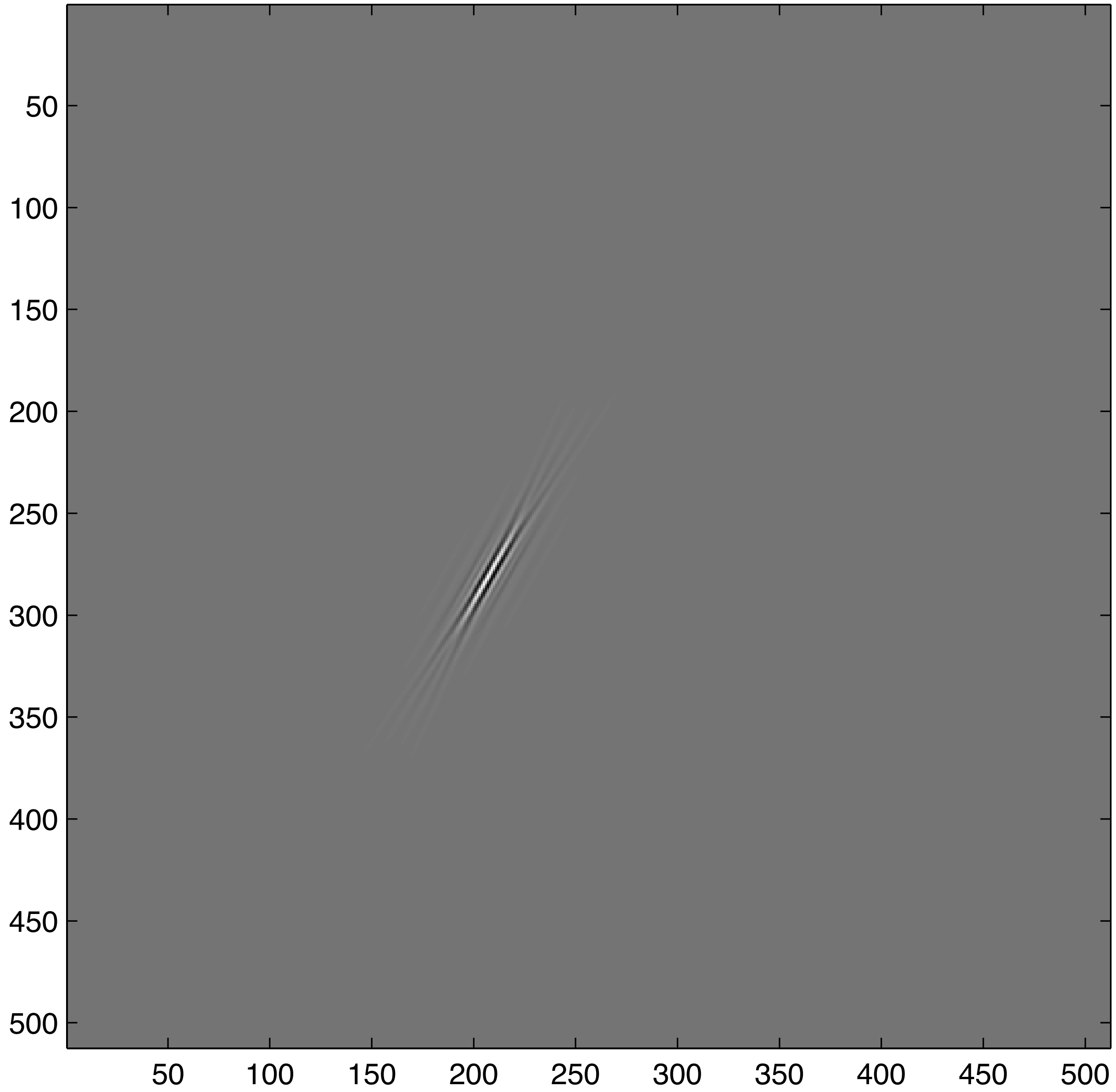



Curvelets parameterized by *scale, location,* and *orientation*

# Digital Curvelets



# Digital Curvelets



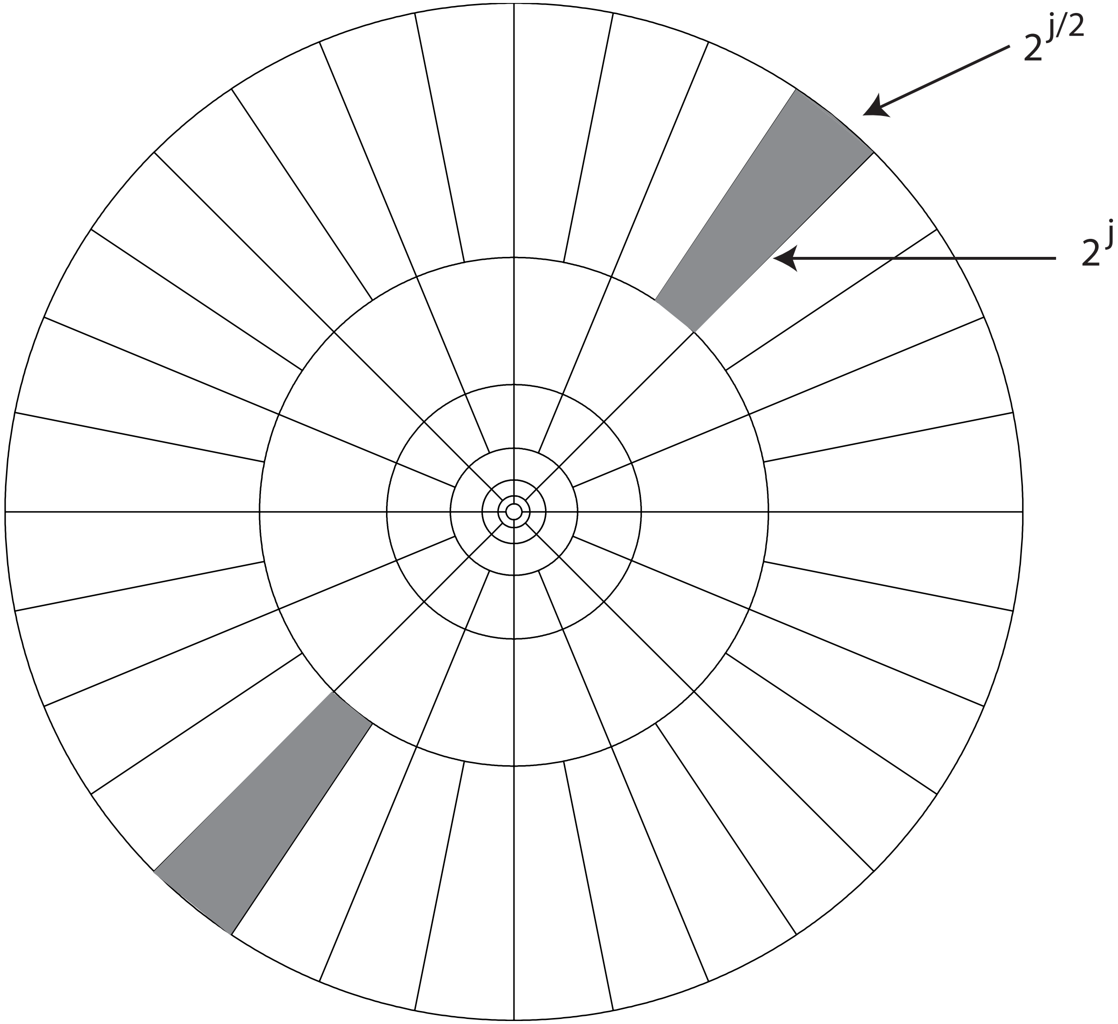
# Frequency-side Picture

Frequency-domain definition

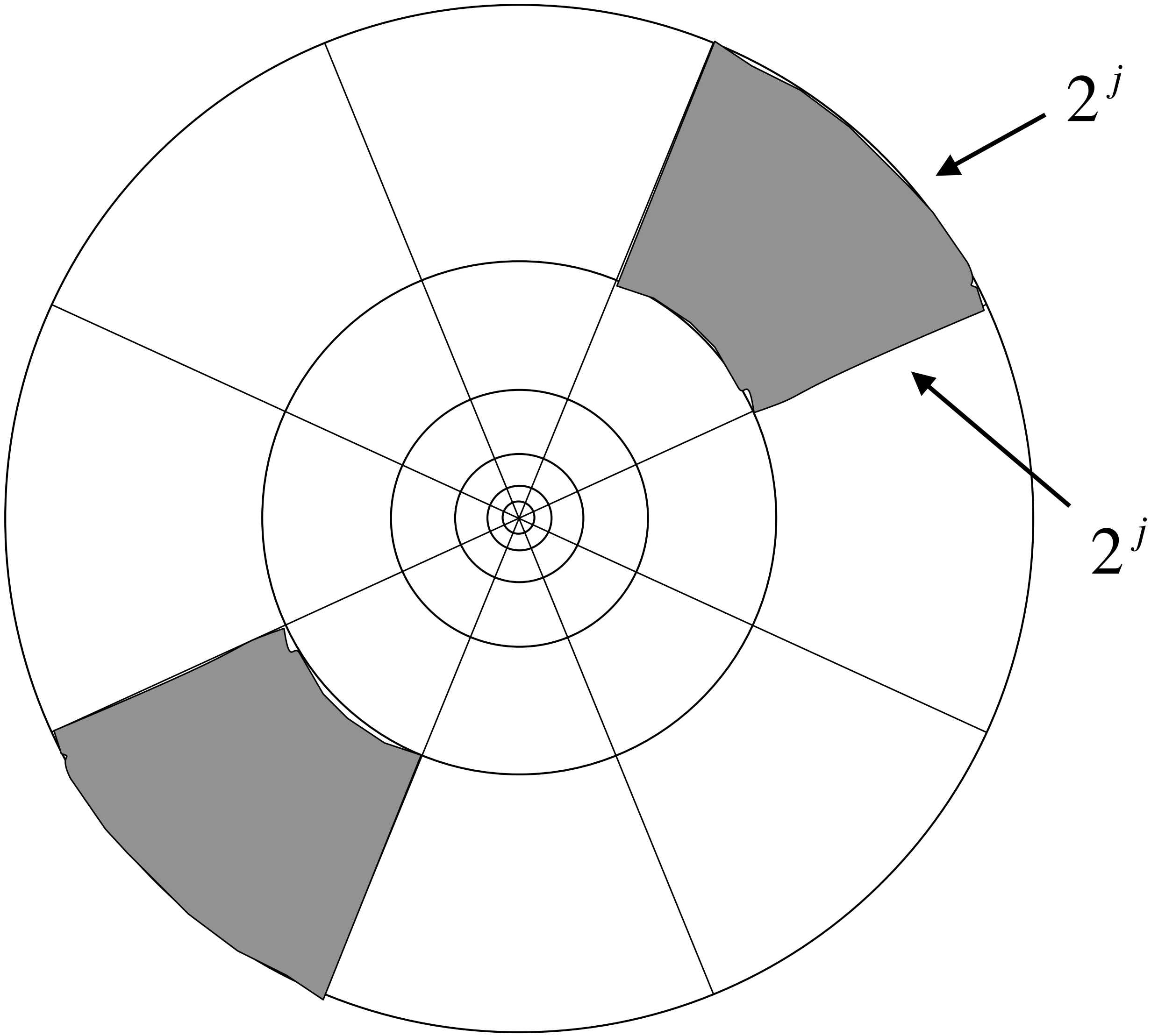
$$\hat{\varphi}_\mu(\xi) = w(2^{-j}|\xi|)\nu(2^{\lfloor j/2 \rfloor}\theta - \pi\ell)e^{i\langle k^{j,\ell}, \xi\rangle}$$

- $w(\cdot) =$ window for scale $j$
- $\nu(\cdot) =$ window for orientation $\theta$
- $e^{i\langle k^{j,\ell}, \xi\rangle}$ shifts to location $(k, \ell)$

# Curvelet Tiling



# Compare to Wavelet Tiling



# Curvelet Properties

- Tight frame, the curvelet transform obeys Parseval

$$f = \sum_{\mu} \langle f, \varphi_\mu \rangle \varphi_\mu \qquad ||f||_2^2 = \sum_{\mu} \langle f, \varphi_\mu \rangle^2$$

- Geometric pyramid structure
  - dyadic scale
  - dyadic location
  - direction (angular resolution doubles every other scale)
- "Needle shaped": width $\sim 2^{-j}$, length $\sim 2^{-j/2}$

# Curvelet Approximation

- Curvelets build up edges in images using “broad strokes”
- $m$-term approximation error

$$\|f - f_m\|_2^2 \asymp m^{-2} \log^3 m$$

within $\log$ factors of optimal rate $m^{-2}$

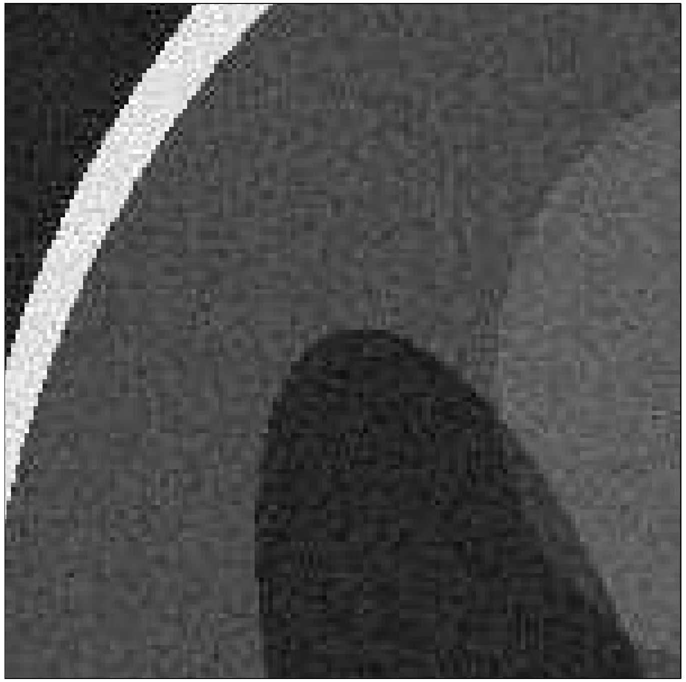
- Example:

original | 1% of curvelet coeffs | 10% of curvelet coeffs

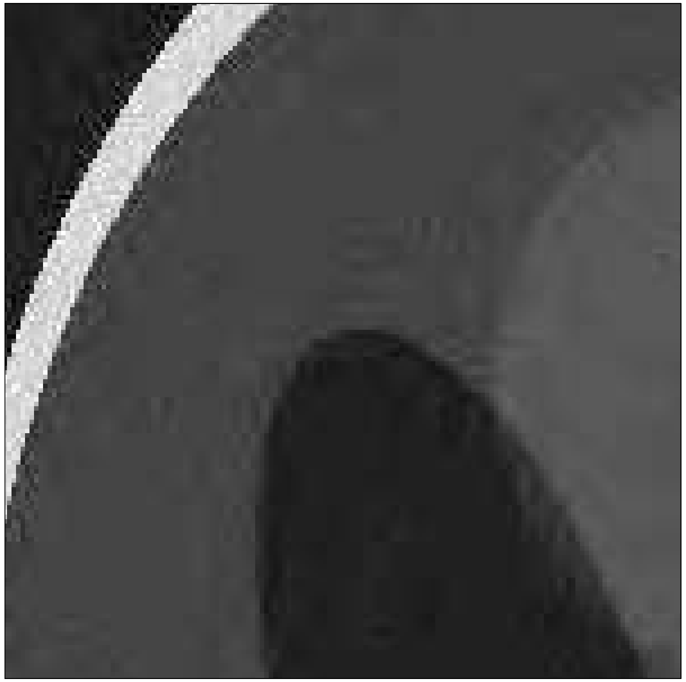
# Application: Curvelet Denoising

Zoom-in on piece of phantom

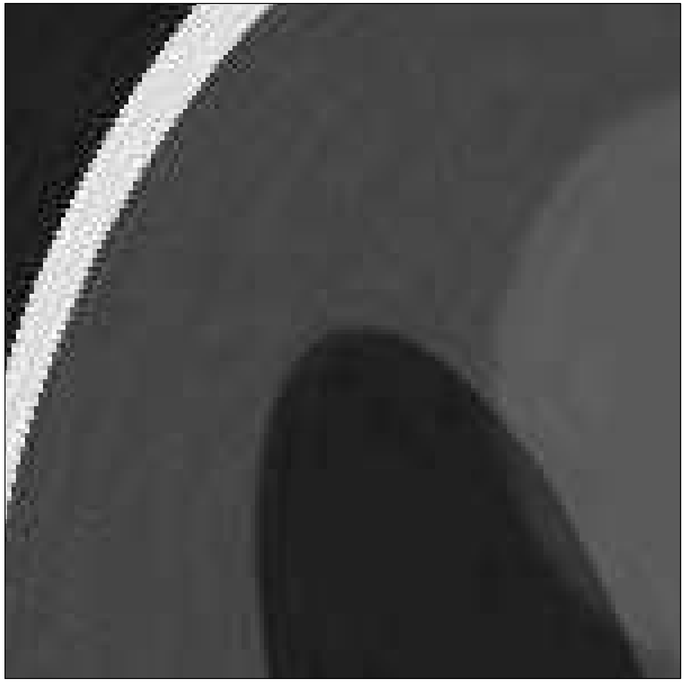
noisy



wavelet thresholding



curvelet thresholding



# Application: Curvelet Denoising

Photograph-like image:

noisy

wavelet thresholding

curvelet thresholding

wavelet thresholding

curvelet thresholding

# Curvelet Thresholding

$$y = f + \sigma z$$

- Model: $f$ is $C^2$ away from $C^2$ edges
- Curvelet shrinkage attains the risk (up to $\log$ factors)

$$\inf_m \|f - f_m\|^2 + m\sigma^2 \asymp \sigma^{4/3}$$

- No estimator can do fundamentally better!

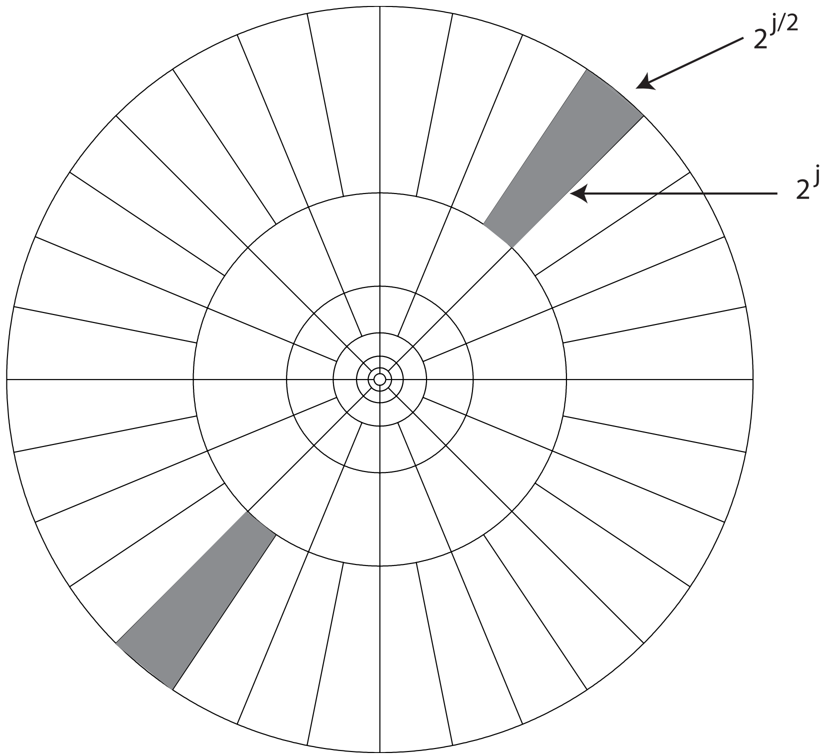
# The Fast Digital Curvelet Transform

# Digital Curvelets and Sampling

- Digital images are sampled on a Cartesian grid
- Main difficulty: rotations are not natural (grid is not closed under rotation)
- Use shearing in place of rotation
- Use pseudo-polar grid in place of polar grid

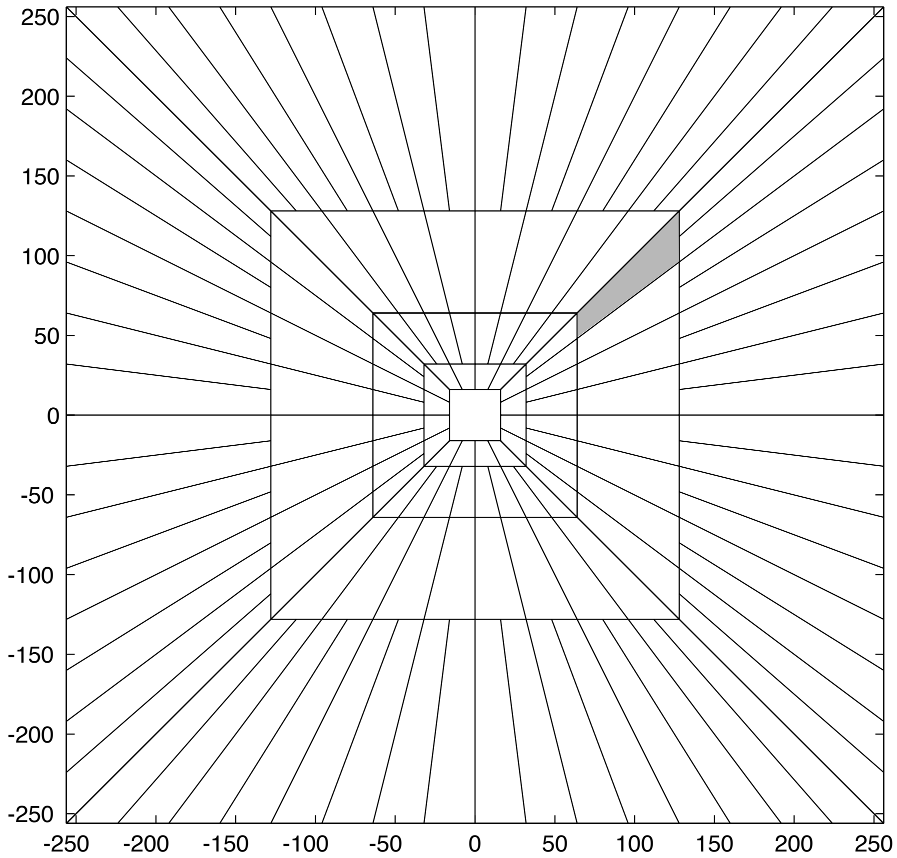
# Curvelet Tilings

continuous time




polar grid

digital

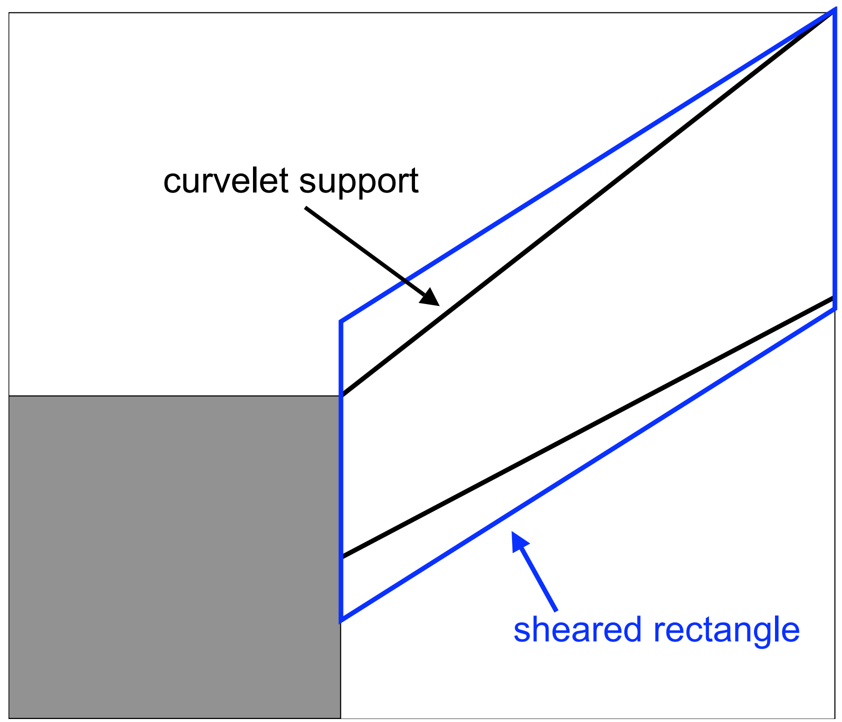



pseudo-polar grid

# Discrete Curvelet Coefficients

Assume that window $W_{j0}(n_1, n_2)$ is supported within a sheared rectangle

$$\mathcal{P}_j = \{(n_1, n_2) : 0 \le n_1 - n_0 < L_j,\ -l_j/2 \le n_2 < l_j/2\}.$$



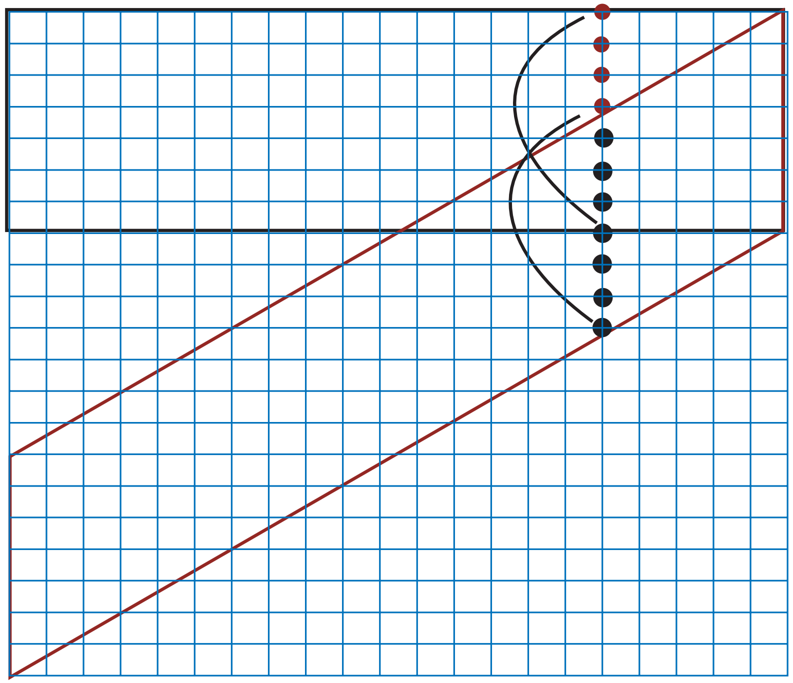

Discrete curvelet coefficient

$$\theta^D_{j,\ell,k} = \sum_{n_1, n_2 \in \mathcal{P}_j} \hat{f}(n_1, n_2 + n_1 \tan\theta_{j,\ell}) W_{j0}(n_1, n_2) e^{-i2\pi(n_1 k_1/L_j + n_2 k_2/l_j)},$$

Need to evaluate $\hat{f}$ inside the sheared rectangle

# FFTs on Parallelograms

- Samples inside each parallelogram tile by periodic wrap-around
  $\Rightarrow$ can be calculated by taking FFTs on rectangular tiles



Periodic wrap around

- This makes the whole transform an isometry (inverse=adjoint)

# DCT: Putting it Together

Initial data: Cartesian array $f(i_1, i_2)$, $0 \le i_1, i_2 \le N-1$.

1. FFT: Apply the 2D FFT and obtain Fourier samples $\hat{f}(n_1, n_2)$, $-N/2 \le n_1, n_2 < N/2$.

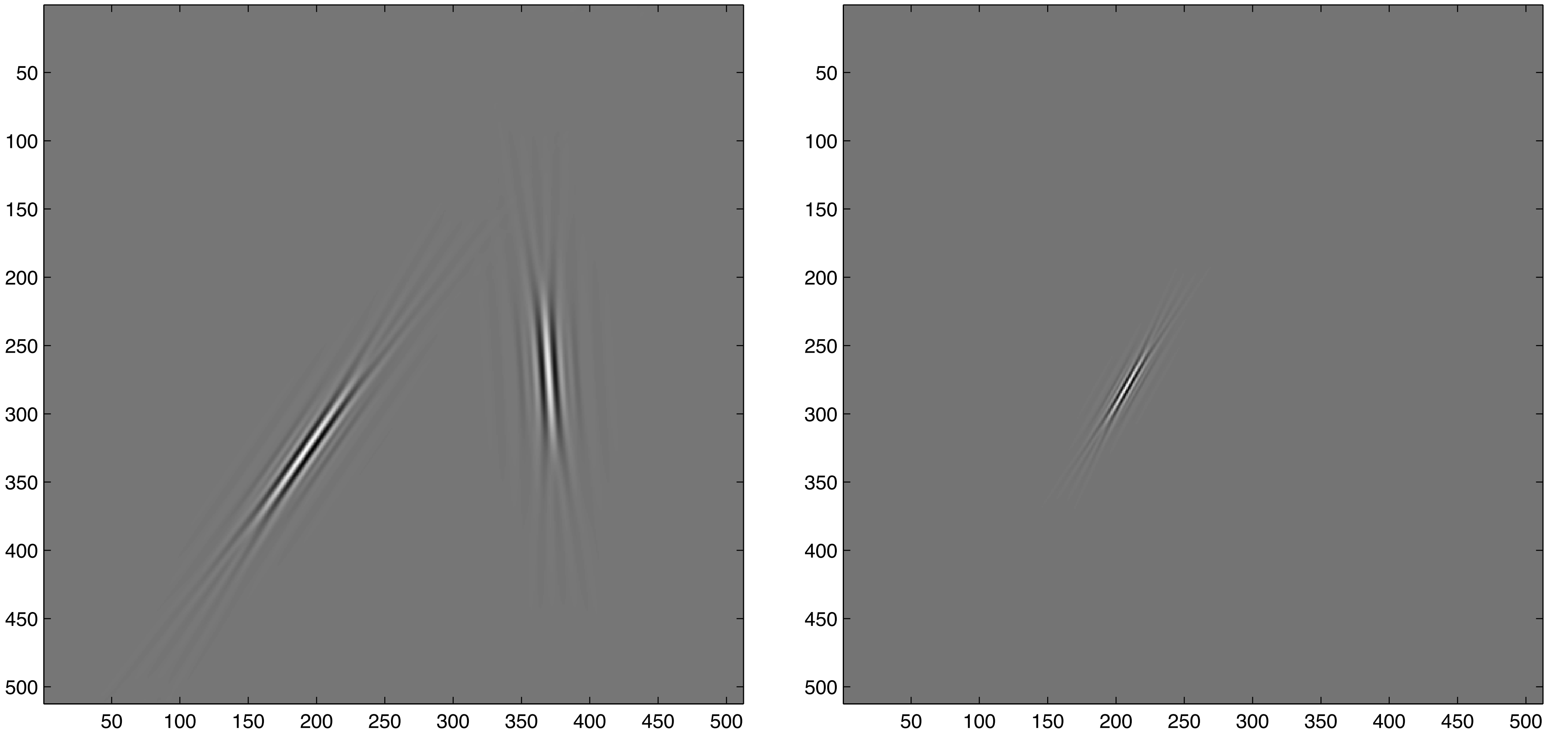
2. Resample: For each scale/angle pair $(j, \ell)$, calculate the sample values inside the parallelipiped $\mathcal{P}_{j,\ell} := \{(n_1, n_2 + n_1 \tan\theta_{j,\ell})\}$, $n_1, n_2 \in \mathcal{P}_j$.

3. Multiply the interpolated (or sheared) object $\hat{f}$ with the parabolic window $\tilde{W}_{j0}$, effectively localizing $\hat{f}$ near the parallelipiped with orientation $\theta_{j,\ell}$, and obtain

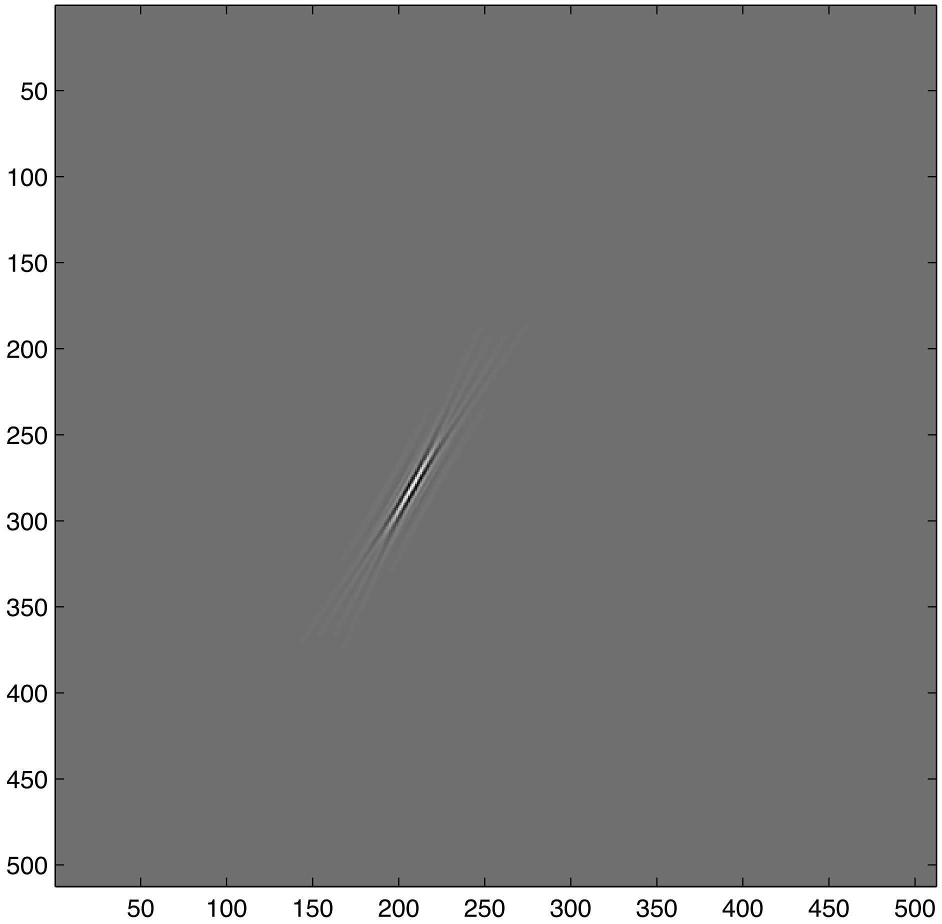
$$\tilde{f}_{j,\ell}(n_1, n_2) = \hat{f}(n_1, n_2 + n_1 \tan\theta_{j,\ell}) W_{j0}(n_1, n_2).$$

4. Inverse FFT: Apply the inverse 2D FFT to each $\tilde{f}_{j,\ell}$, hence collecting the discrete coefficients $\theta^D_{j,\ell,k}$.

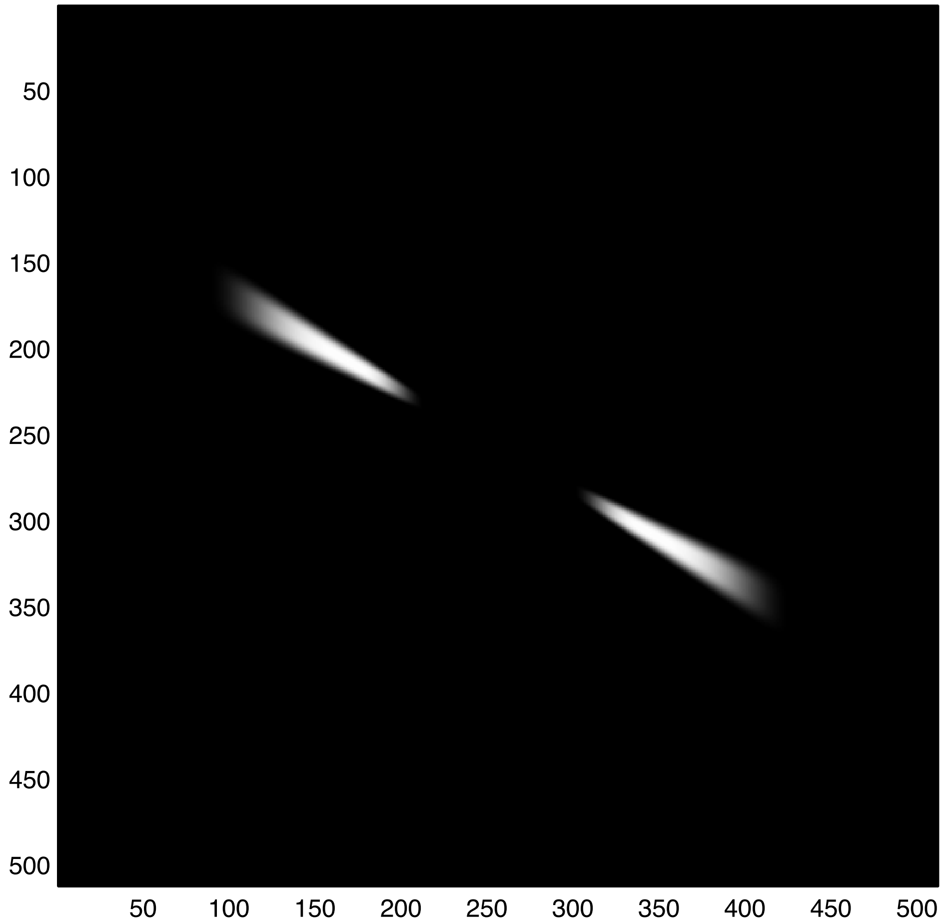
# Digital Curvelets



# Digital Curvelets: Frequency Localization

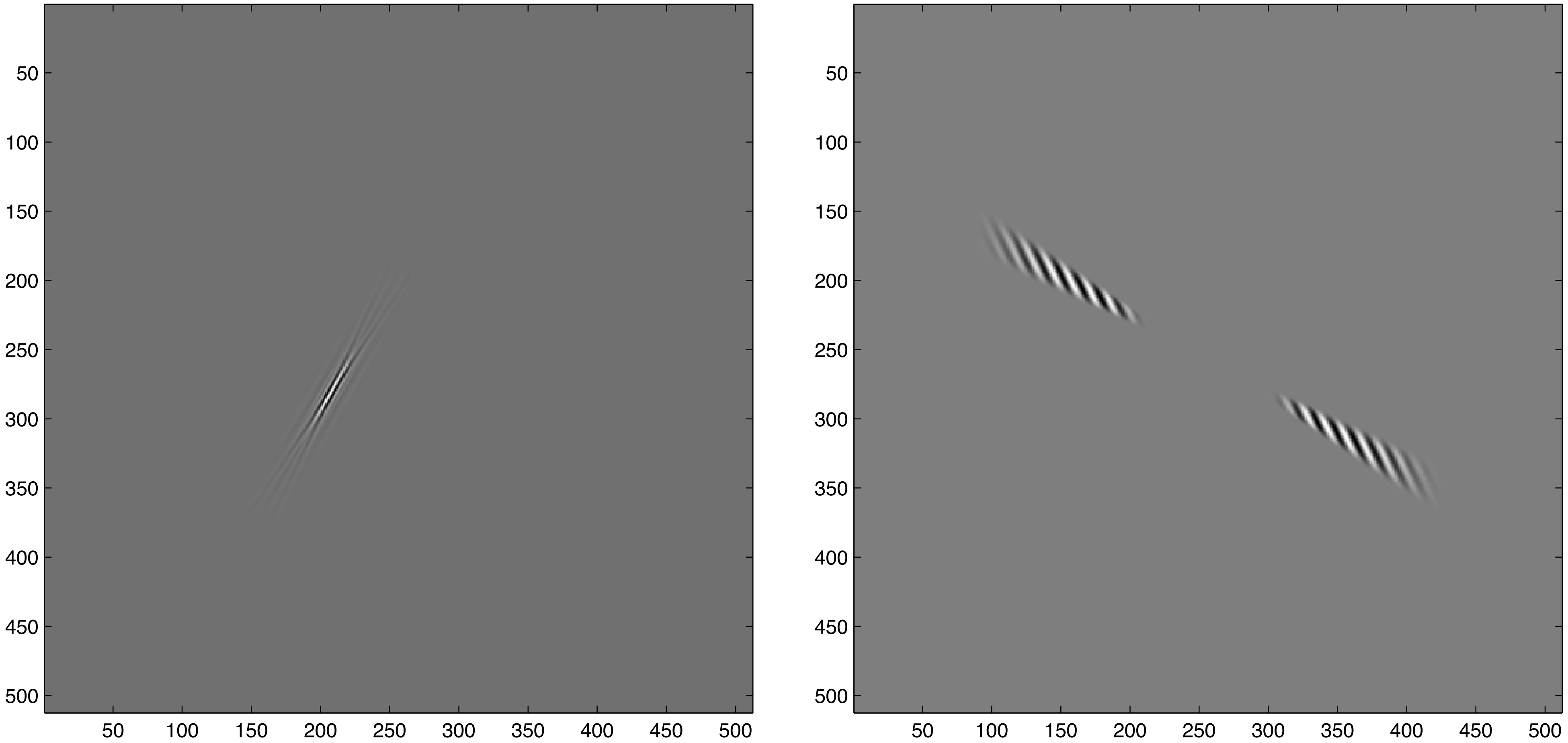



Time domain




Frequency domain (modulus)

# Digital Curvelets: : Frequency Localization



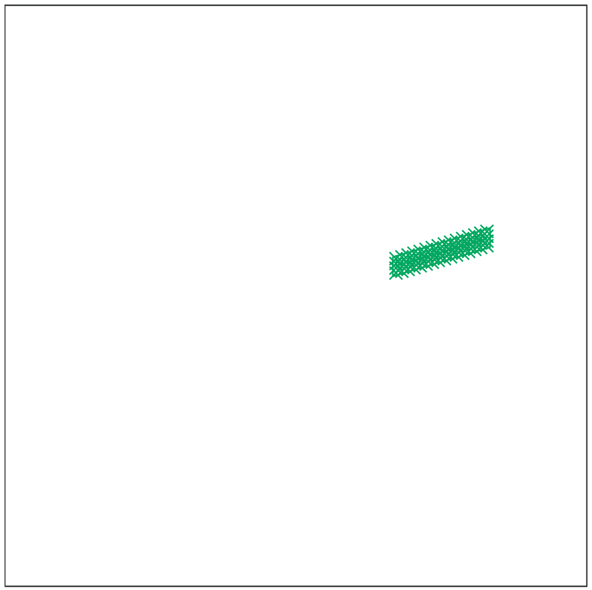

Time domain

Frequency domain (real part)

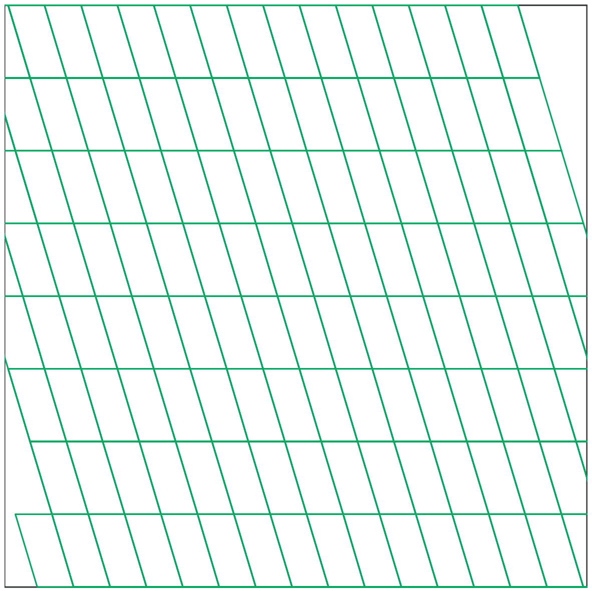
# DCT: Architecture

1. Each coefficient is defined by a direct summation over a parallelipipedal, anisotropic 'tilted' lattice in the frequency domain.
2. The regions obey the parabolic scaling relation $width \approx \sqrt{length}$.
3. The coefficients associated with a single orientation and scale 'tile' the spatial domain according to a dual 'tilted' lattice. The corresponding basis functions sharing that orientation and scale have support tiling the space according to a dual tilted lattice.

Freq-domain parallelogram



Spatial-domain tiling



4. The Riesz representers of the coefficients obey sharp frequency localization.
5. The transform is a near-isometry; all steps except one involve either orthogonal transforms or tight frames.
6. The transform is cache-aware: all component steps involve processing $n$ items in the array in sequence, e.g. there is frequent use of 1-D FFTs to compute $n$ intermediate results simultaneously.
7. Transform can be made arbitrarily tight at the cost of oversampling.

# DCT: Summary

- Cartesian data structure
- For practical purposes, algorithm runs $O(n^2 \log(n))$ flops, where $n^2$ is the number of pixels. (Runs in about 5s for a 512 by 512 image on my MAC).
- The approach is flexible, and can be used with a variety of choices of parallelipipedal tilings, for example, including based on principles besides parabolic scaling.