Fast Polar Fourier Transform

Michael Elad*

Scientific Computing and Computational Mathematics

Stanford University

FoCM Conference, August 2002

Image and Signal Processing Workshop

IMA - Minneapolis

* Joint work with Dave Donoho (Stanford-Statistics), Amir Averbuch (TAU-CS-Israel), and Ronald Coifman (Yale-Math)

Collaborators

Dave Donoho

Statistics Department
Stanford

Amir Averbuch

CS Department
Tel-Aviv University

Ronald Coifman

Math. Department
Yale

Fast Polar Fourier Transform

- FFT is one of top 10 algorithms of 20th century.
- We'll extend utility of FFT algorithms to new class of settings in image processing.
- Create a tool which is:
  - Free of emotional involvement, &
  - Freely available over the internet.
- Current Stage:
  - We have the tool, and its analysis,
  - Have not demonstrated applications yet.

Agenda

1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Conclusions

Items 1–3: Background & Motivation

Items 4–7: New Approach and its Results

1. Thinking Polar - Continuum

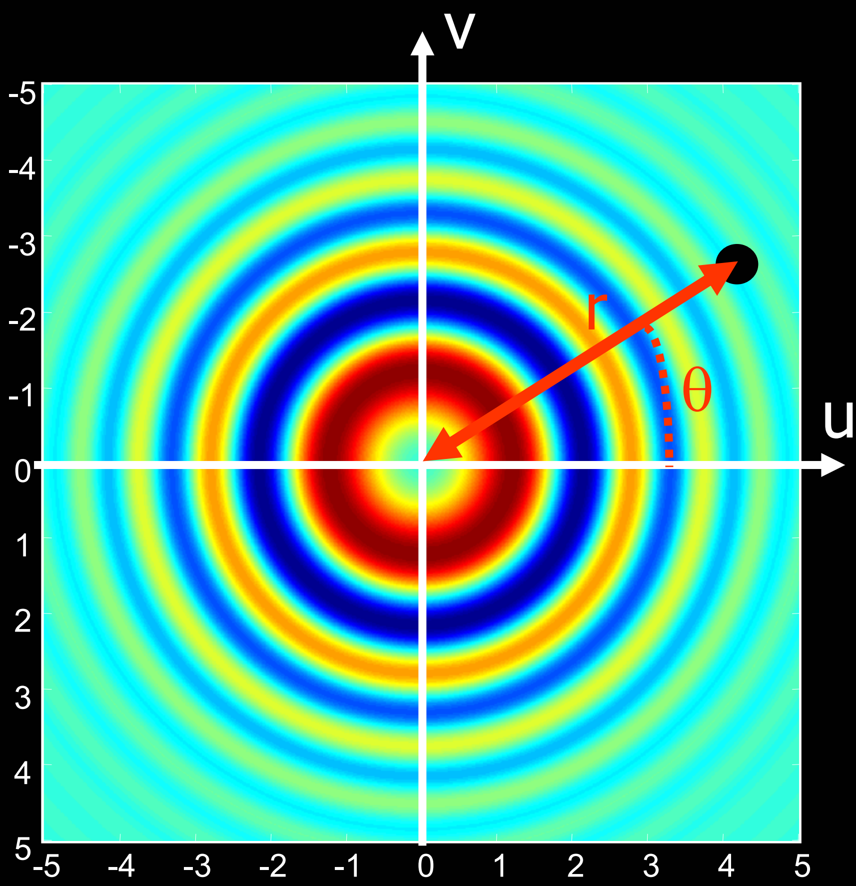
- For today f(x,y) function of $(x,y)\in\Re^2$
- Continuous Fourier Transform

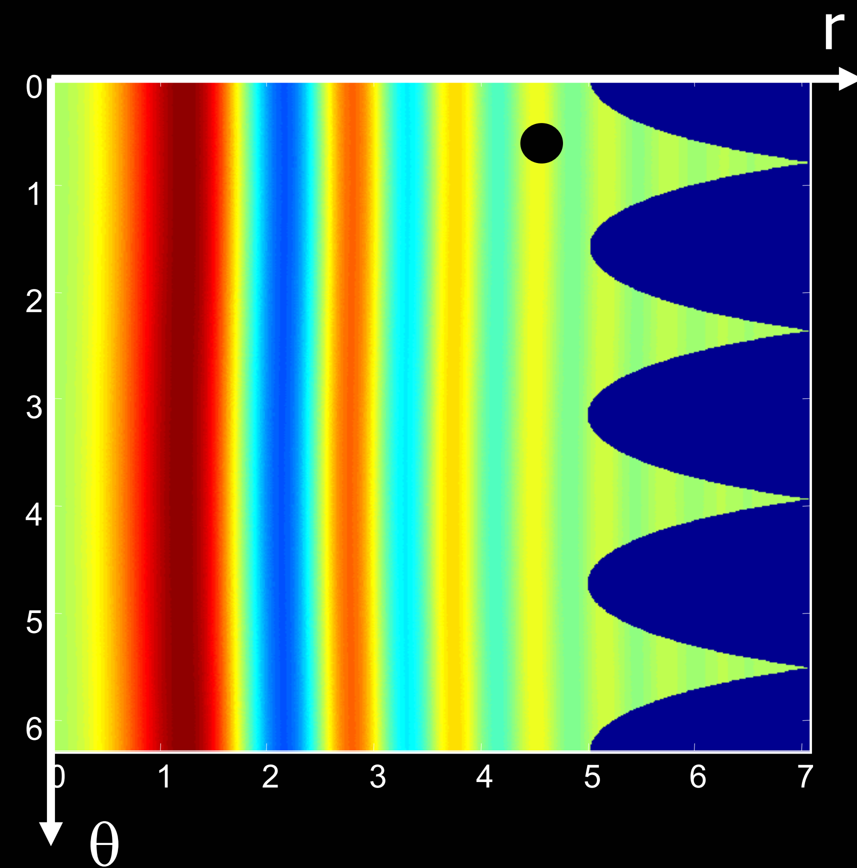
$$\hat{f}(u,v)=(\Im f)(x,y)=\int\int f(x,y)\exp\{-ixu-iyv\}dxdy$$

- Polar coordinates: $u=r\cdot\cos(\theta)$, $v=r\cdot\sin(\theta)$

$$\tilde{f}(r,\theta)=\hat{f}(r\cdot\cos(\theta),r\cdot\sin(\theta))=$$
$$=\int\int f(x,y)\exp\{-ixr\cdot\cos(\theta)-iy\cdot\sin(\theta)\}dxdy$$

- Important Processes easy to continuum polar domain.


v
u
r
θ
-5
-4
-3
-2
-1
0
1
2
3
4
5
-5 -4 -3 -2 -1 0 1 2 3 4 5


r
θ
0
1
2
3
4
5
6
0 1 2 3 4 5 6 7

Natural Operations: 1. Rotation

Using the polar coordinates, rotation is simply a shift in the angular variable.

- Q_{θ_0} – planar rotation by θ_0 degrees
- Rotation $f_{\theta_0}(x, y) = f\left(Q_{\theta_0}\{x, y\}\right)$
- In polar coordinates – shift in angular variable

$$\tilde{f}_{\theta_0}(r, \theta) = \tilde{f}(r, \theta - \theta_0)$$

Natural Operations: 2. Scaling

Using the polar coordinates, 2D scaling is simply a 1D scaling in the radial variable.

- S_α – planar scaling by a factor α
- Scaling $f_\alpha(x,y) = f(S_\alpha\{x,y\})$
- In polar coordinates – 1D scale in radial variable

$$\tilde{f}_\alpha(r,\theta) = \text{Const} \cdot \tilde{f}(\alpha r, \theta)$$

- Log-Polar – shift in the radial variable.

Natural Operations: 3. Registration

Using the polar coordinates, rotation+shift registration simply amounts to correlations.

- f(x,y) and g(x,y): $f(x,y) = g\left(Q_{\theta_0}\{x,y\} + \{x_0,y_0\}\right)$
- Goal: recover $\{x_0, y_0, \theta_0\}$.
- Angular cross-correlation between $\left|\tilde{f}(r,\theta)\right|$ and $\left|\tilde{g}(r,\theta)\right|$
  – Estimate θ_0.
- Rotation & cross-correlation on regular Fourier transform gives the shift.

Natural Operations: 4. Tomography

Using the polar coordinates, we obtain a method to obtain the Inverse Radon Transform.

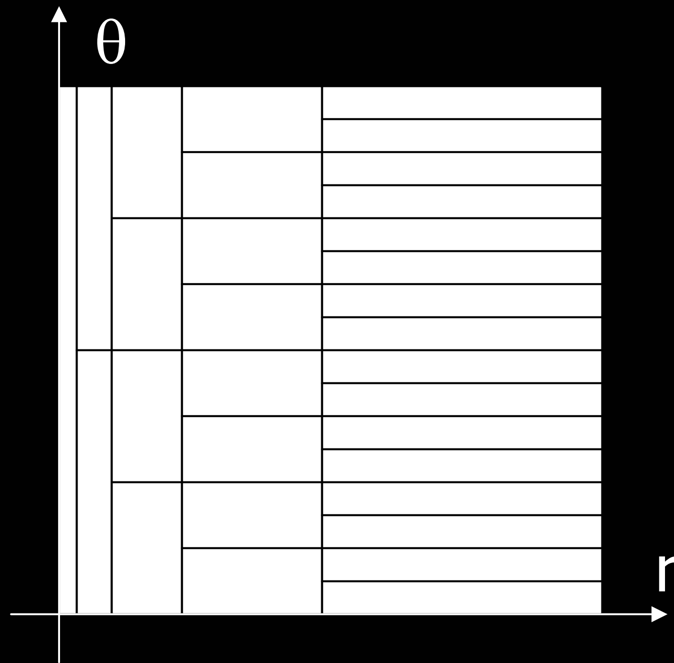
- Radon Transform:

$$Rf(t,\theta) = \iint f(x,y)\delta(x\cos(\theta) + y\sin(\theta) - t)\,dxdy$$

- Goal: Given $Rf(t,\theta)$, recover f.
- Projection-Slice-Theorem: $(\mathfrak{I}_1 Rf)(t,\theta) = \tilde{f}(r,\theta)$.
- Reconstruction: $Rf \mapsto \tilde{f} \mapsto \hat{f} \mapsto f$.

More Natural Operations

- New orthonormal bases:
  - Ridgelets,
  - Curvelets,
  - Fourier Integral operations,
  - Ridgelet packets.
- Analysis of textures.
- Analysis of singularities.
- More …



Agenda

1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Conclusions

2. Thinking Polar - Discrete

- Certain procedures very important to digitize
  - Rotation,
  - Scaling,
  - Registration,
  - Tomography, and
  - …
- These look so easy in continuous theory – Can't we use it in the discrete domain?
- Why not Polar-FFT?

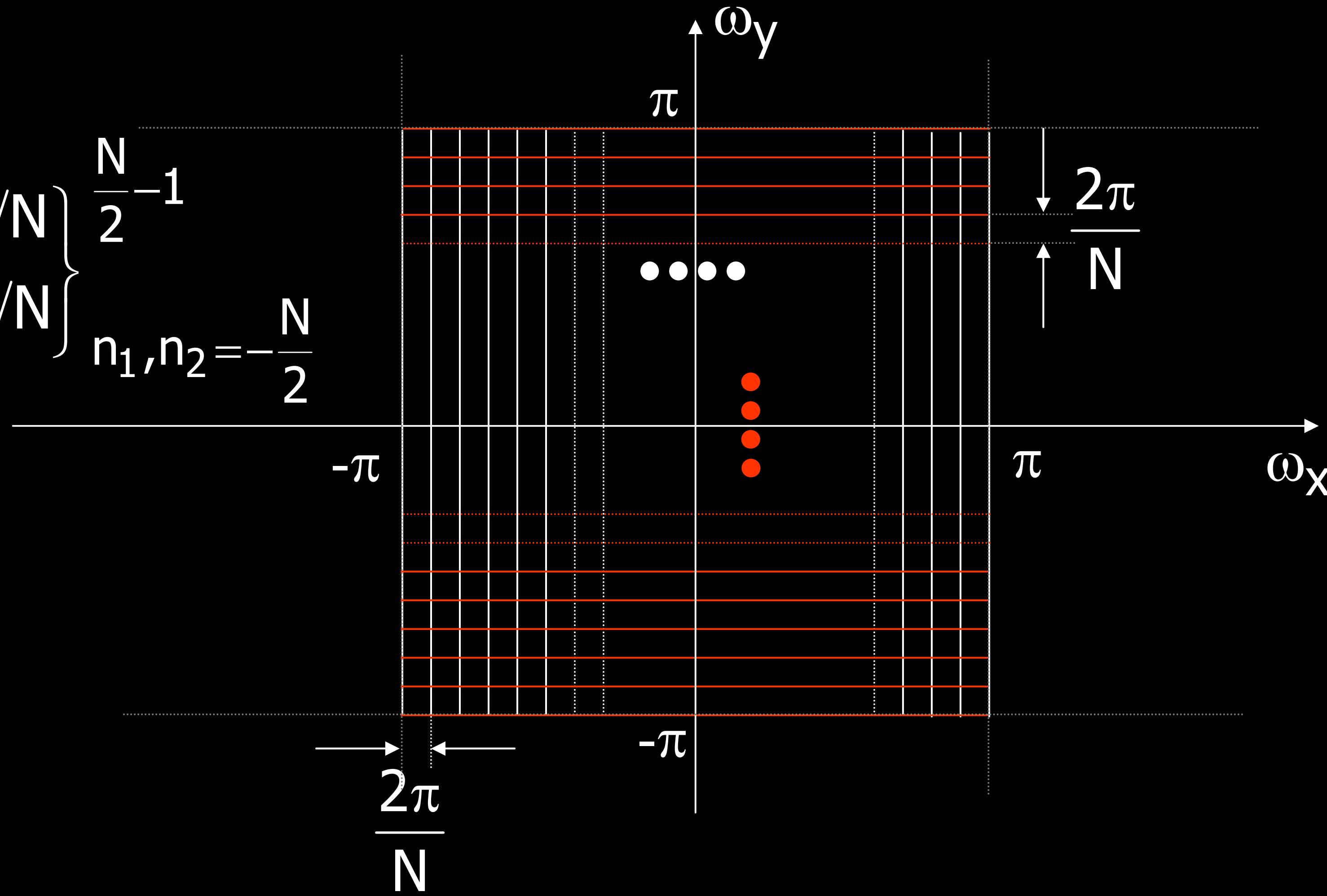
The FFT Miracles

- 1D Discrete Fourier Transform
  - Uniformly sampled in time and frequency – FFT.
  - Complexity – $O(5N\log_2 N)$ instead of $O(N^2)$.

- 2D Discrete Fourier Transform
  - Cartesian grid in space and frequency – Separability
  - Only 1D-FFT operations.
  - Smart memory management.

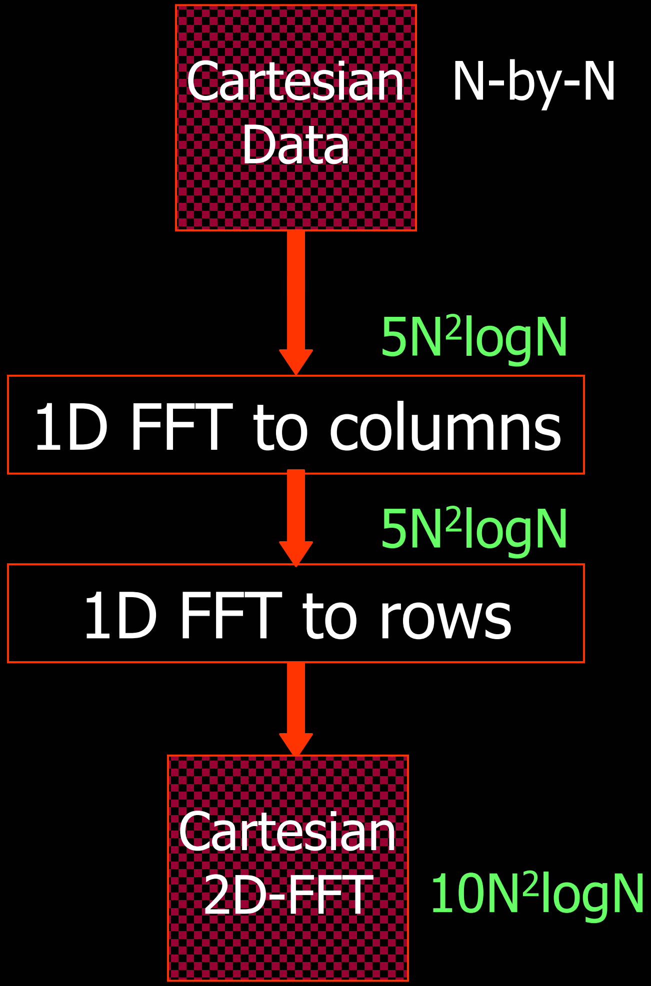
2D DFT – Cartesian Grid

$$\left\{ \begin{aligned} \omega_x &= 2\pi n_1 / N \\ \omega_y &= 2\pi n_2 / N \end{aligned} \right\}_{n_1, n_2 = -\frac{N}{2}}^{\frac{N}{2}-1}$$



2D FFT Complexity

- Complexity: $O(10N^2\log_2 N)$ instead of $O(N^4)$.
- Important Feature: All operations are 1D – leading to **efficient cache usage**



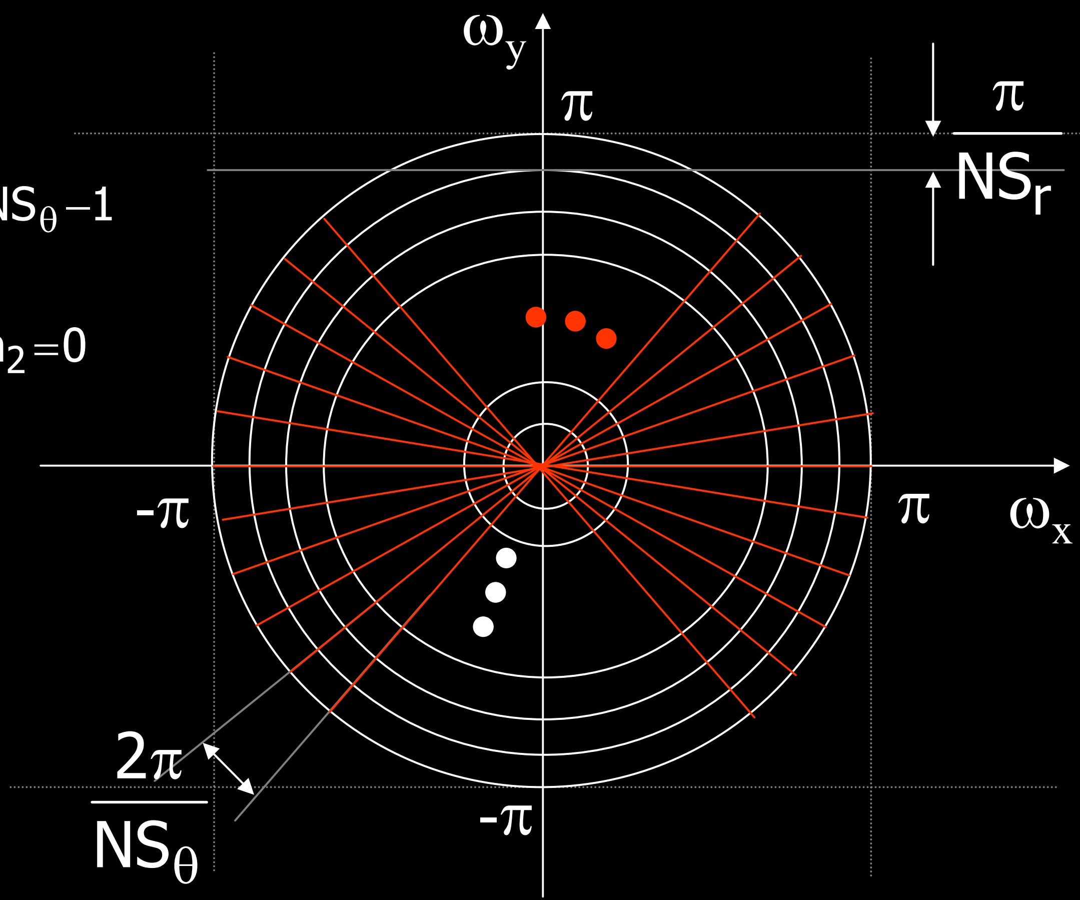
Discrete Polar Coordinates?

Choice of grid?

$$\left\{ r = \frac{\pi n_1}{NS_r} \right\}_{n_1=0}^{NS_r-1}, \left\{ \theta = \frac{2\pi n_2}{NS_\theta} \right\}_{n_2=0}^{NS_\theta-1}$$

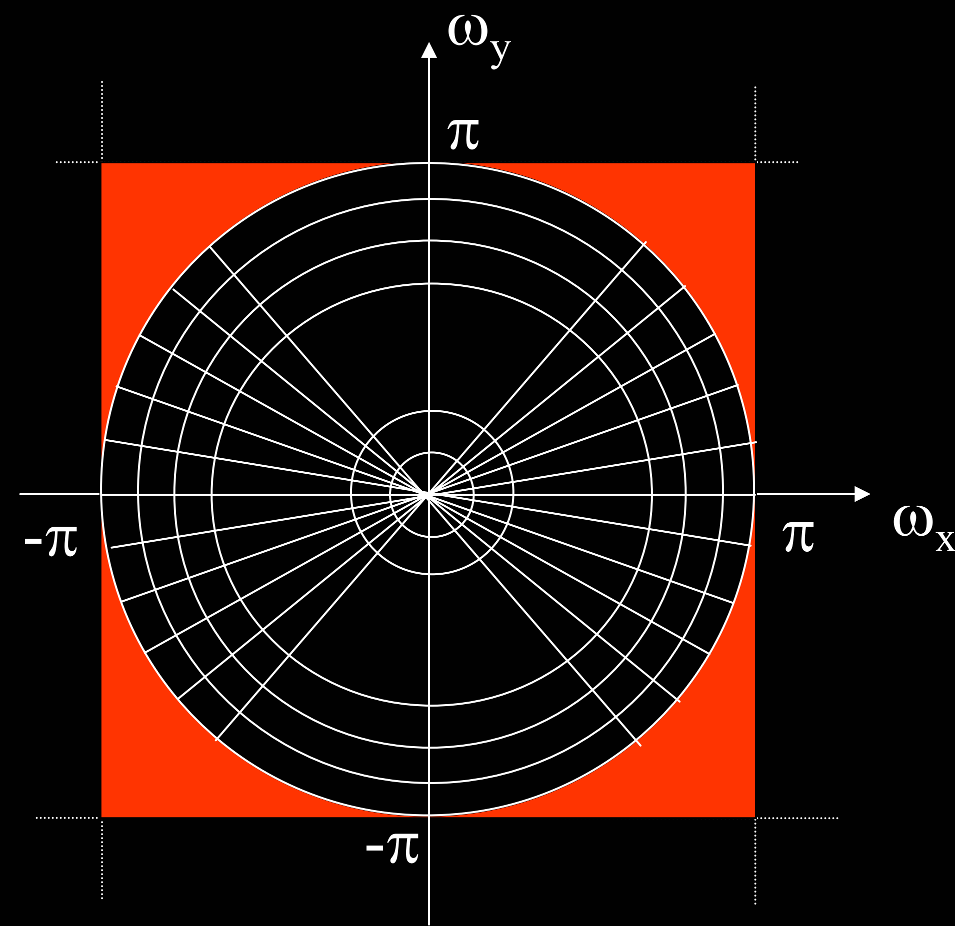
Resulting with NS_θ rays with NS_r elements on each:

For $S_\theta = S_r = 1$, we have N^2 grid points.



Grid Problematics

- ❑ Grid spacing?
- ❑ Fate of corners?
- ❑ No X-Y separability !!



Polar FFT - Current Status

- Current widespread belief - There cannot be a fast method for DFT on the polar grid. See e.g. **The DFT: an owner's manual**, Briggs and Henson, SIAM, 1995, p. 284.
- Consequence of Non-existence:
  - Continuous Fourier – vague inspiration only.
  - Fourier in implementations widely deprecated.
  - Fragmentation: each field special algorithm.

Agenda

1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Conclusions

3. Current State-Of-The-Art

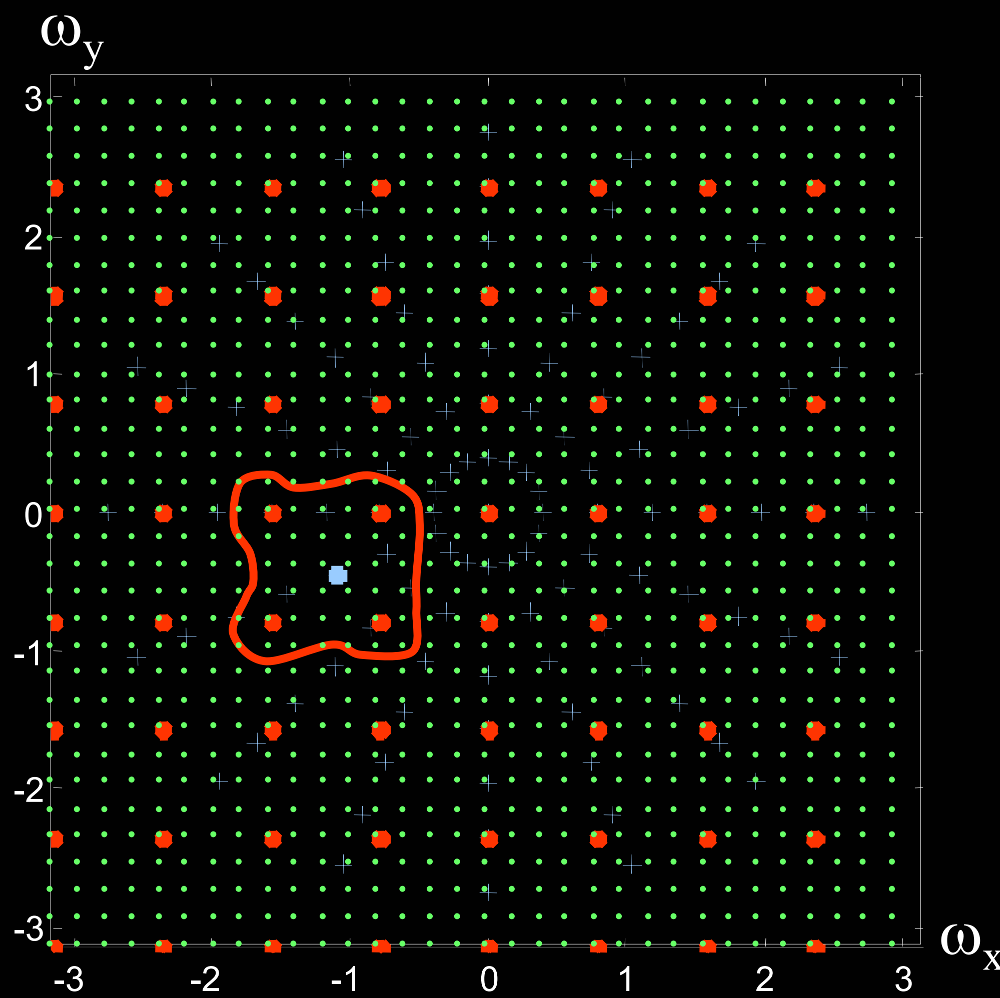
- Assessing T: Unequally-spaced FFT (USFFT)
  - Data in Cartesian set.
  - Approximate transform in non-Cartesian set.
  - Oriented to 1D – not 2D and definitely not Polar.
- Assessing $T^{\dagger}$: For Tomography
  - Data in Polar coordinates in frequency.
  - Approximate inverse transform to Cartesian grid.
  - Inspired by the projection-slice-theorem.

USFFT - Rational

+ Destination Polar grid

● Critically sampled Cartesian grid

o Over-sampled Cartesian grid



USFFT - Detailed

- Over-sample Cartesian grid.
- Rapidly evaluate FT:
  - Values F.
  - Possibly - partial derivatives.
- Associate Cartesian neighbors to each polar grid point.
- Approximate interpolation.

Our Reading of Literature

- Boyd (1992) – Over-sampling and interpolation by Euler sum or Langrangian interpolation.
- Dutt-Rokhlin (1993,1995) - Over-sampling and interpolation by the Fast-Multipole method.
- Anderson-Dahleh (1996) – Over-sampling and obtaining the partial derivatives, and then interpolating by Taylor series.
- Ware (1998) – Survey on USFFT methods.

USFFT Problematics

- Several involved parameters:
  - Over-sampling factor,
  - Method of interpolation, and
  - Order of interpolation.
- Good accuracy calls for extensive over-sampling.
- Correspondence overhead: spoils vectorizability of algorithm - causes high cache misses.
- Emotionally involved.

Agenda

1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Conclusions

4. Our Approach - General

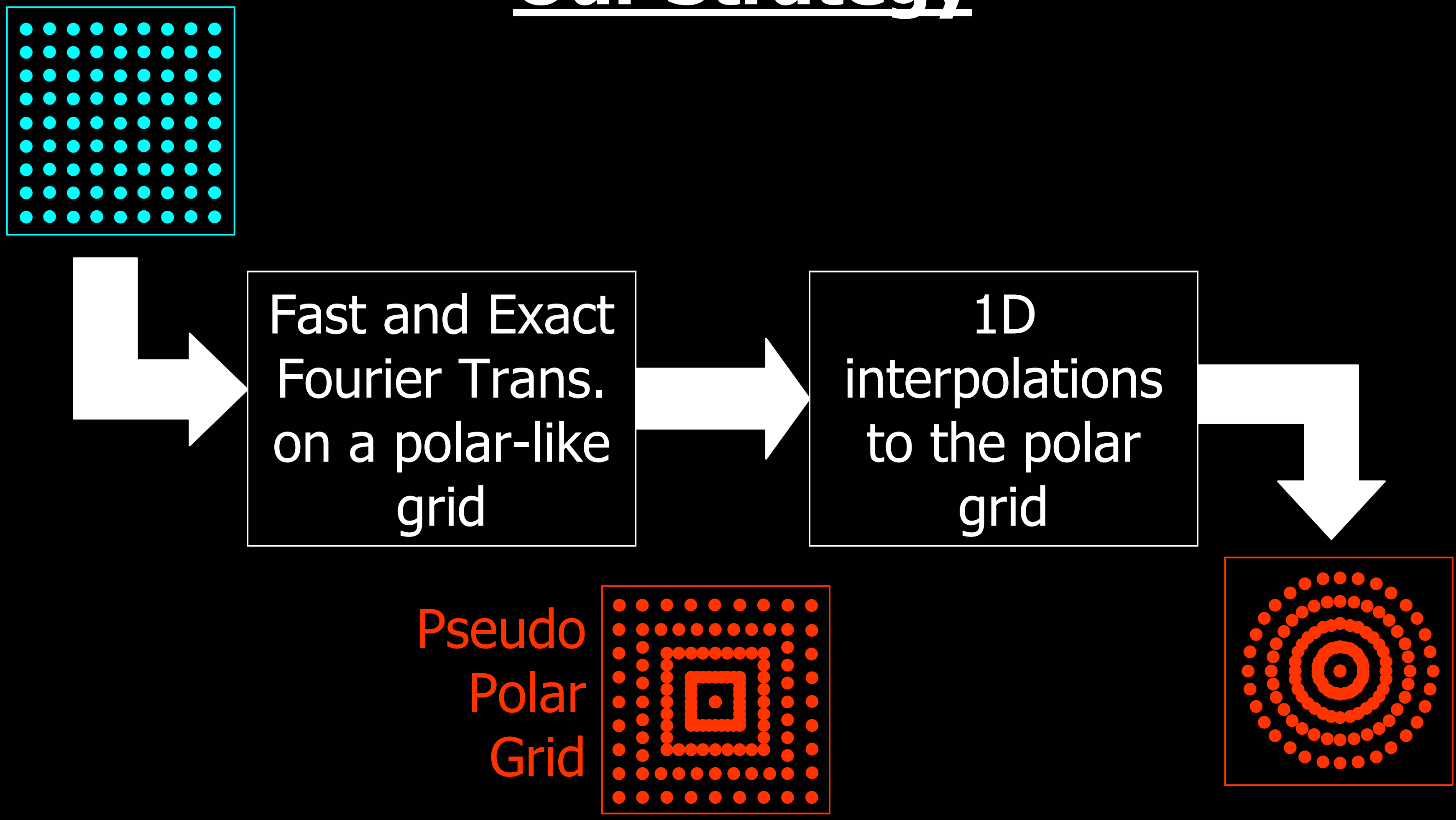
We propose a

Fast Polar Fourier Transform

with the following features:

- Low complexity – $O(Const \cdot N^2 \log_2 N)$
- Vectorizability – 1D operations only
- Non-Expansiveness – Factor 2 (or 4) only
- Stability – via Regularization
- Accuracy – 2 orders of magnitude over USFFT methods

Our Strategy



Agenda

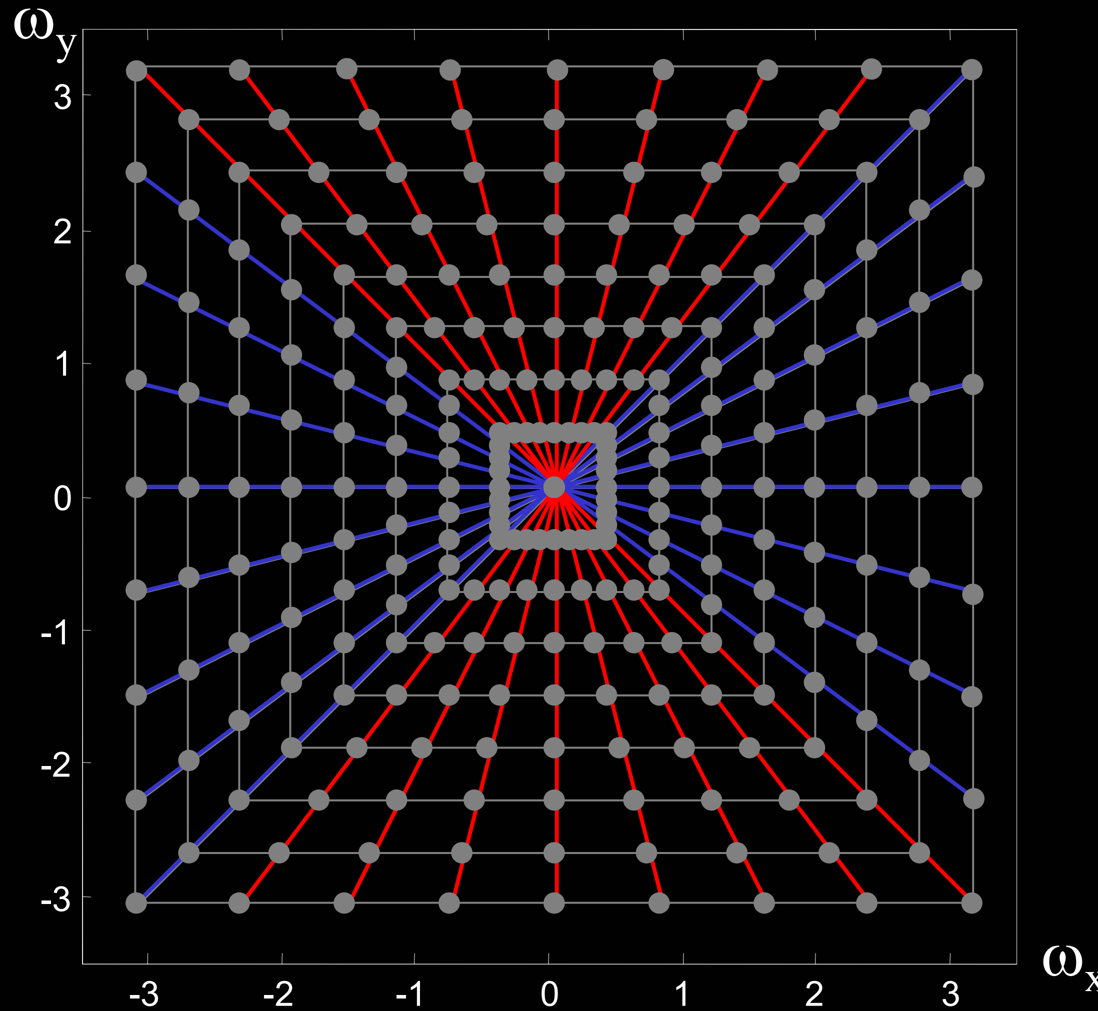
1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Conclusions

5. The Pseudo-Polar FFT

- Developed by Averbuch, Coifman, Donoho, Israeli, and Waldén (1998).
- Basic idea: A "Polar-Like" grid that enables
  - EXACT Fourier transform,
  - FAST computation,
  - 1D operations only.
- Applications: Tomography, image processing, Ridgelets, and more.

The Pseudo-Polar Skeleton

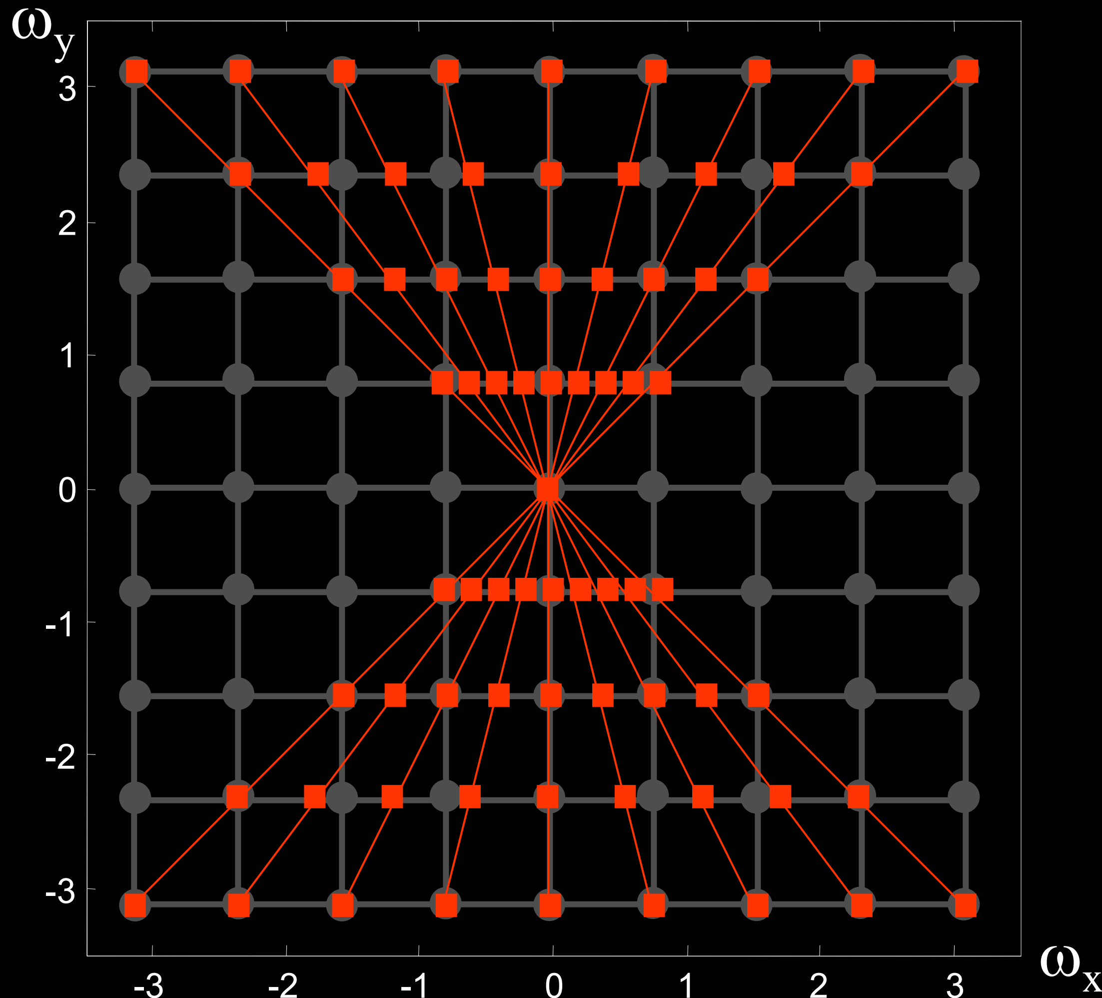
- NS_r equi-spaced concentric squares,
- NS_t 'equi-spaced' (not in angle)
- We separate our treatment to basically vertical and basically horizontal lines.

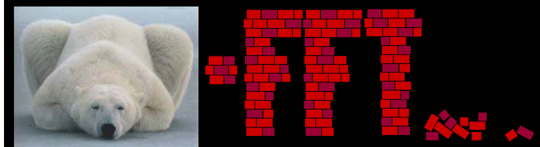


Fast Fourier Transform

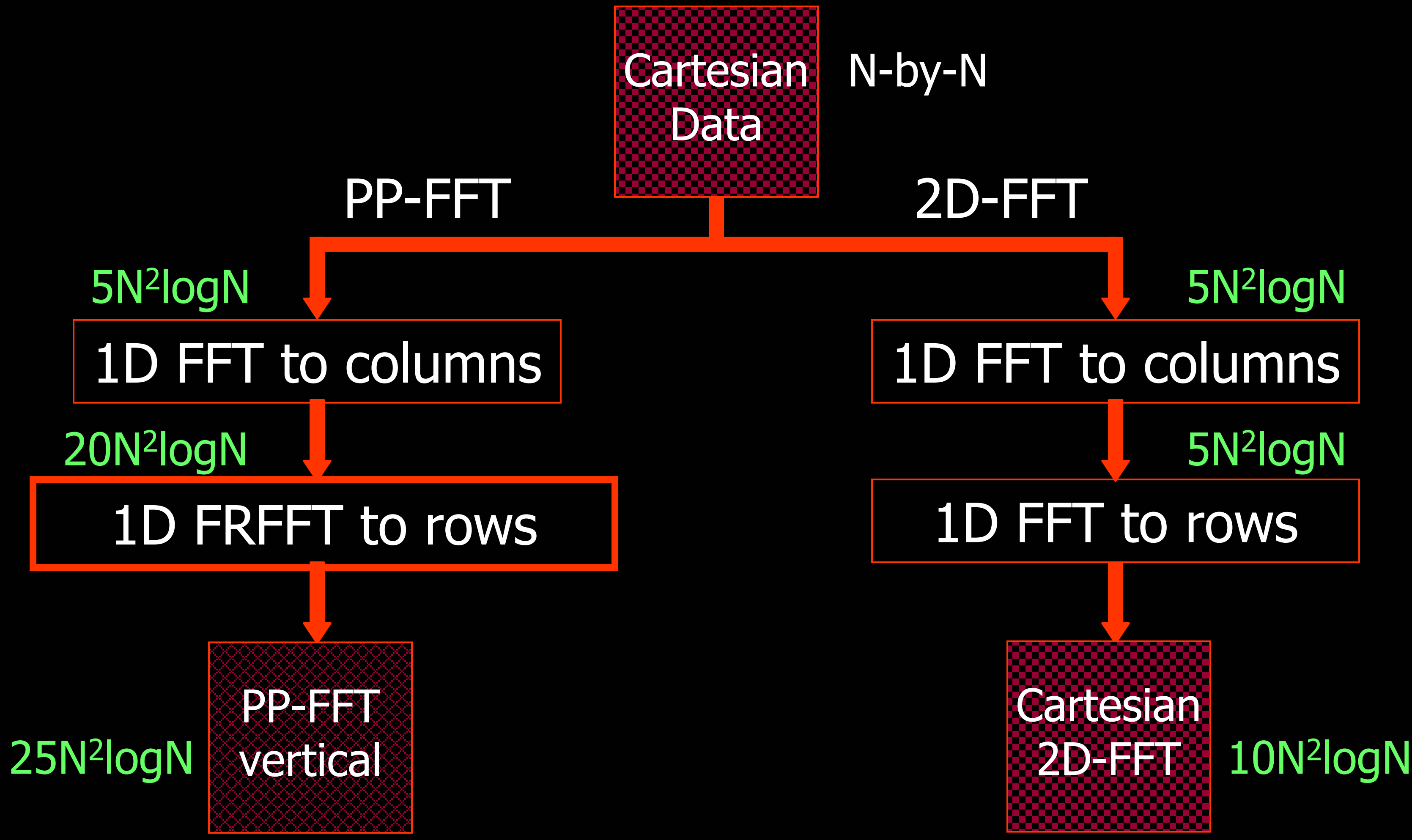
- The destination samples are uniformly sampled vertically,
- Per each row, destination samples are uniformly sampled horizontally,
- Fractional Fourier Transform is the answer (Chirp-Z), with complexity: $O(20N\log_2 N)$.

[Why?]





PP-FFT versus 2D-FFT



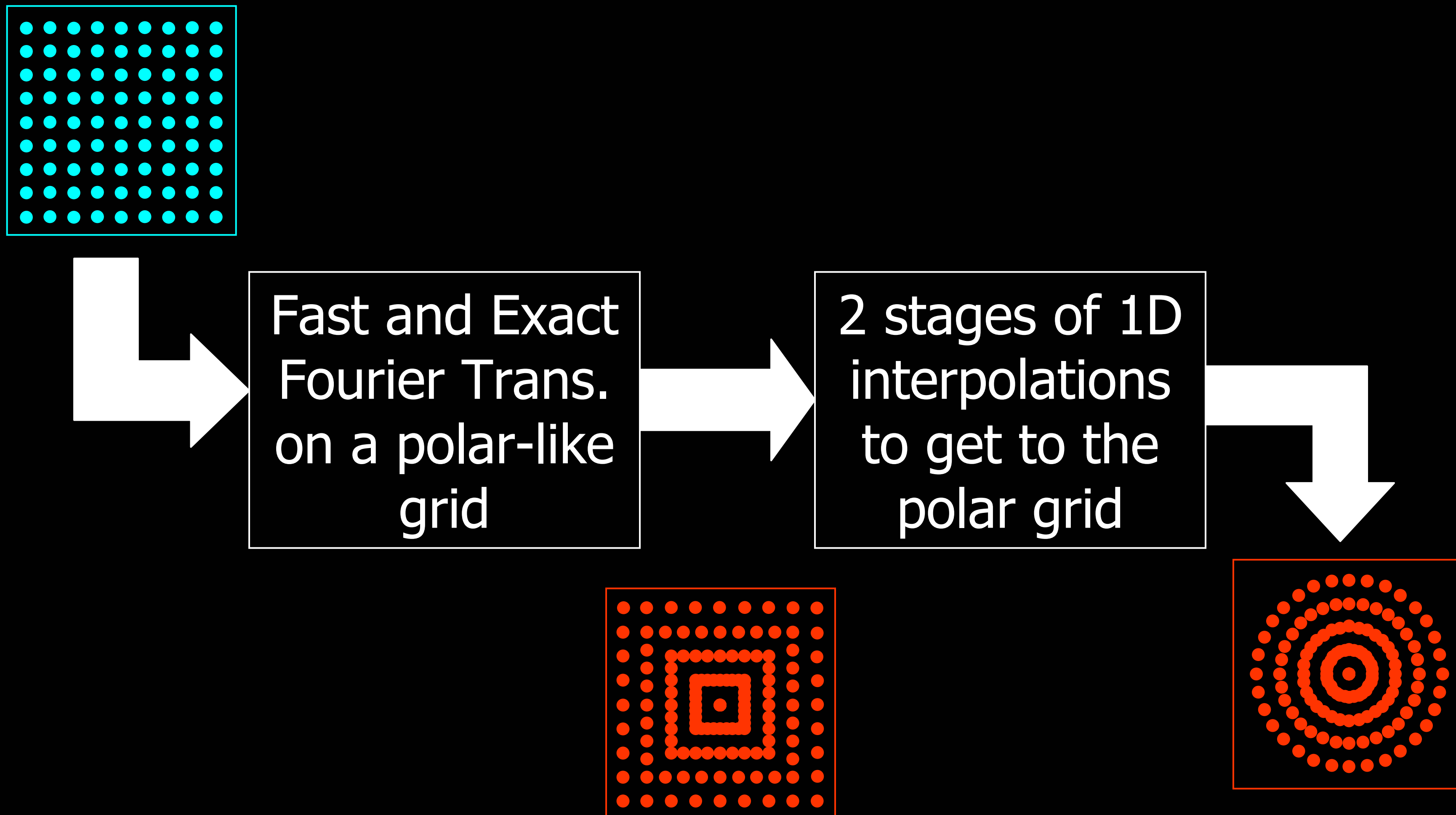
The PP-FFT - Properties

- Exact in exact arithmetic.
- No parameters involved !!
- Complexity - $O(50 \cdot N^2 \log_2 N)$ versus $O(N^4)$.
- 1D operations only.
- For the chosen grid ($S_r = S_t = 2$) - $\kappa \approx 5$.

Agenda

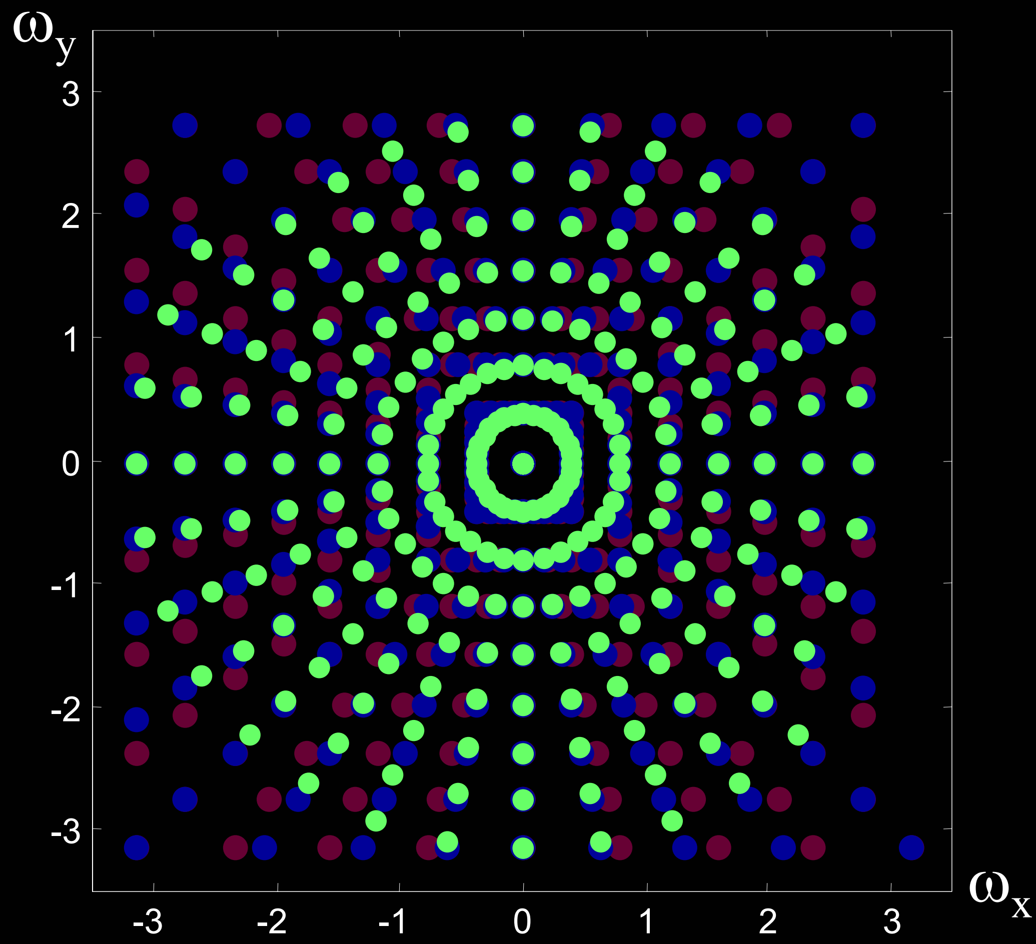
1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Conclusions

6. From Pseudo-Polar to Polar



The Interpolation Stages

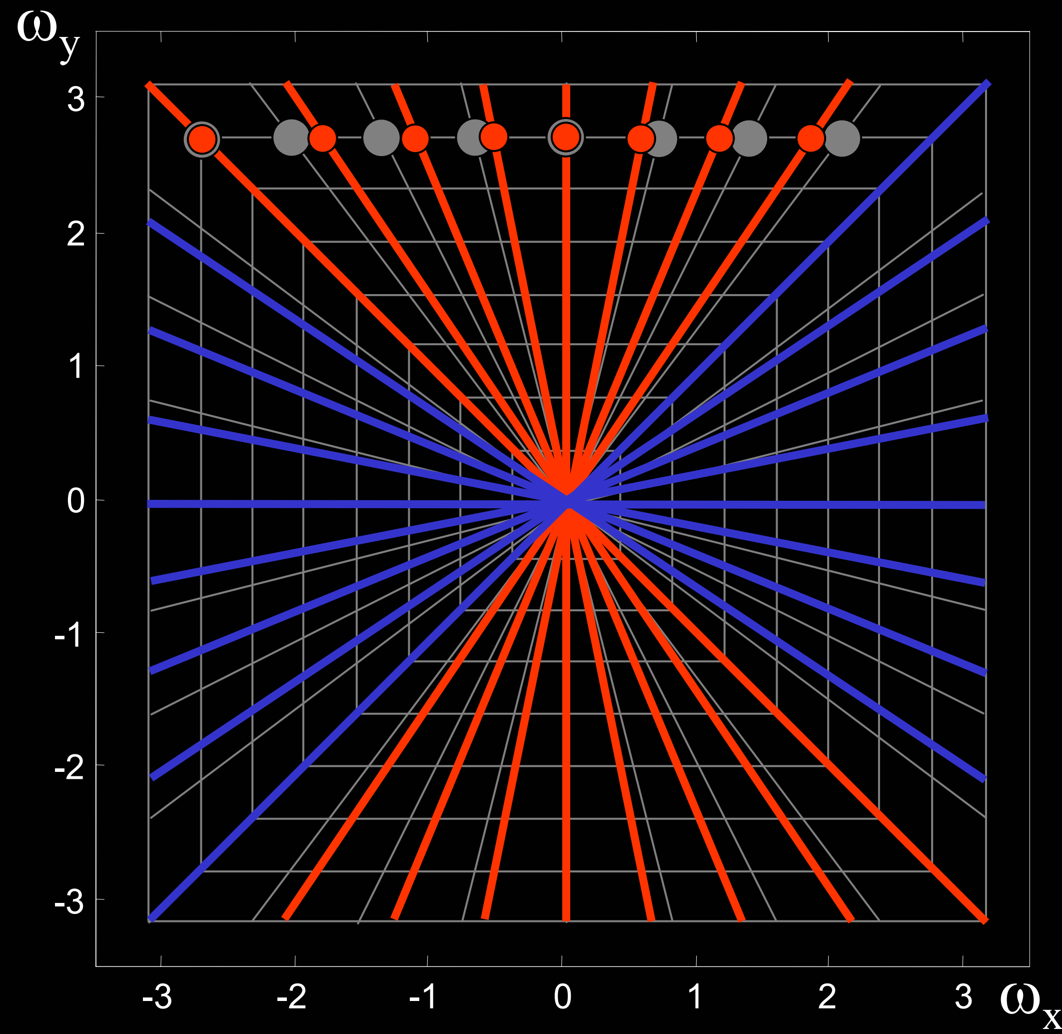
- The original Pseudo-Polar Grid
- Warping to equi-spaced angles
- Warping each ray to have the same step

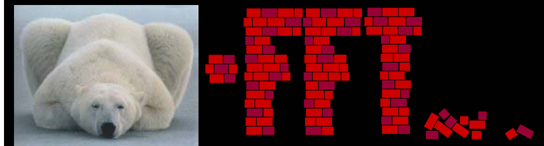


First Interpolation Stage

Rotation of the rays to have equi-spaced angles (S-Pseudo-Polar grid):

- Every row is a trigonometric polynomial of order N,
- FRFT on over-sampled array and 1D interpolation,
- Very accurate.



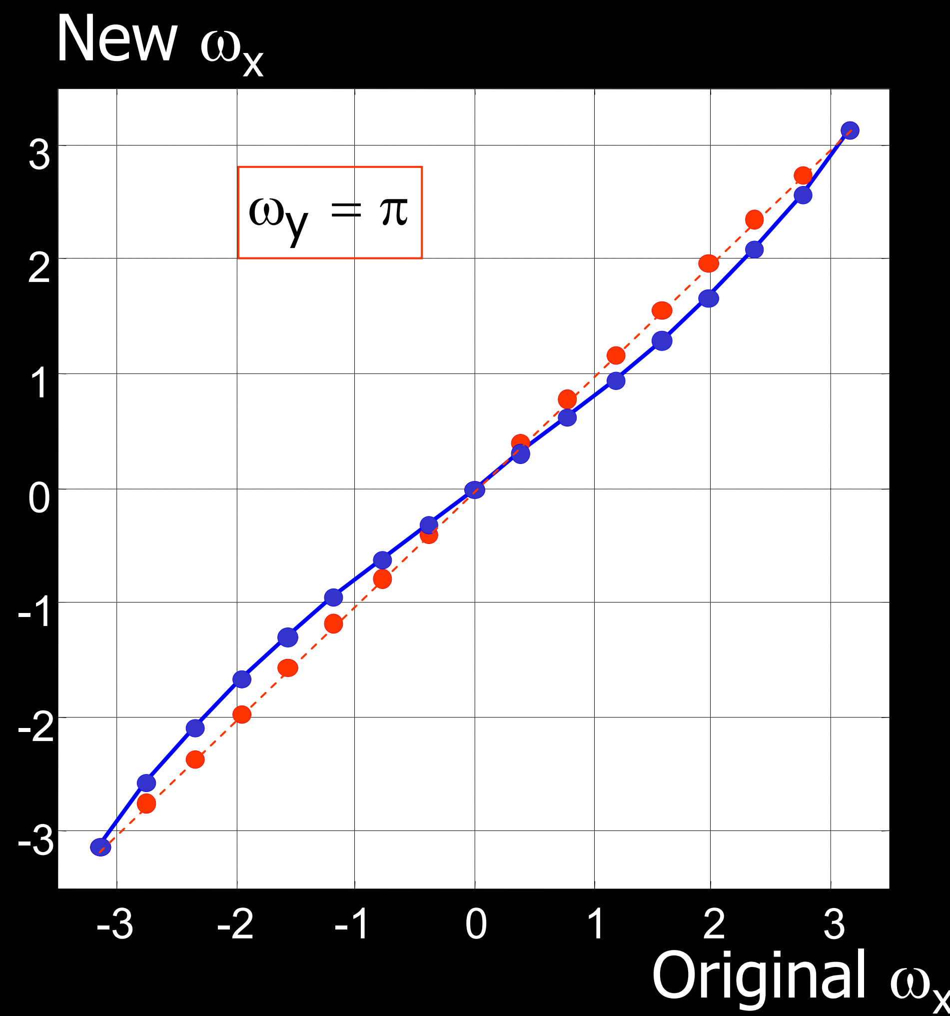


The Required Warping

Basically vertical lines:

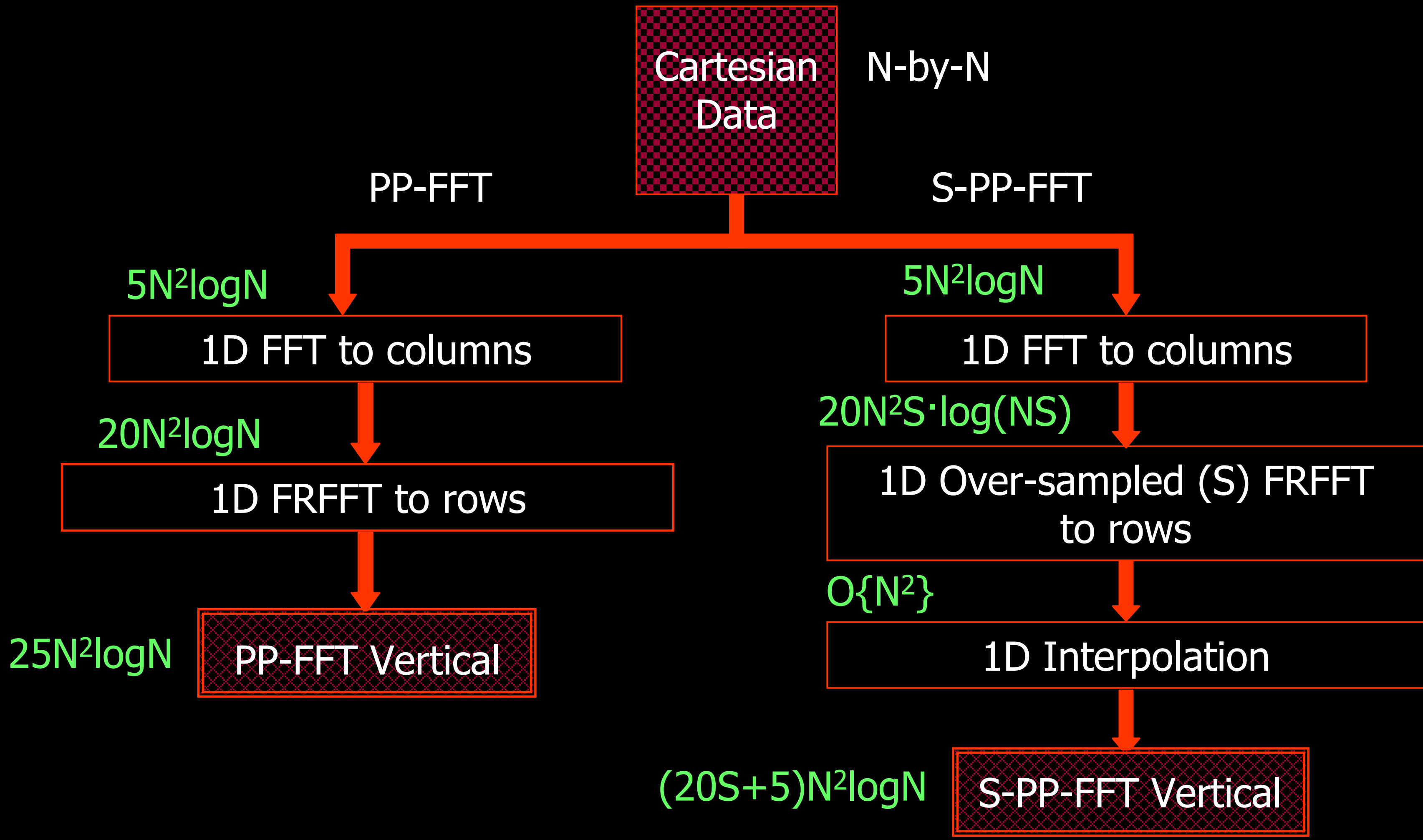
$$\left\{\omega_y = \frac{2\pi\ell}{NS_r}, \omega_x = \frac{2m}{NS_t}\omega_y\right\}_{\ell,m=-NS_t/2}^{NS_t/2-1}$$

$$\left\{\omega_x = \omega_y \cdot \tan\left(\frac{m\pi}{2NS_t}\right)\right\}_{m=-NS_t/2}^{NS_t/2-1}$$



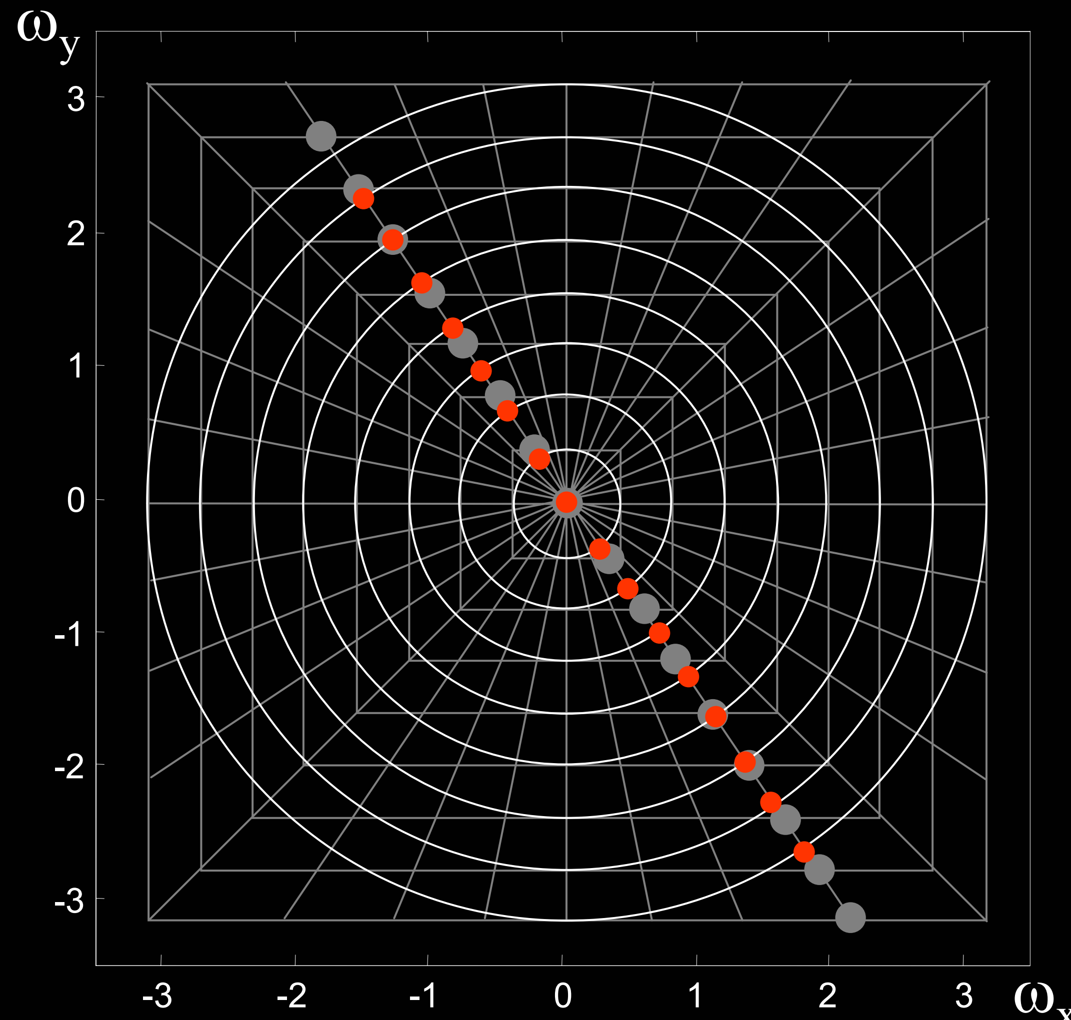
[Why?]

The Actual Interpolation



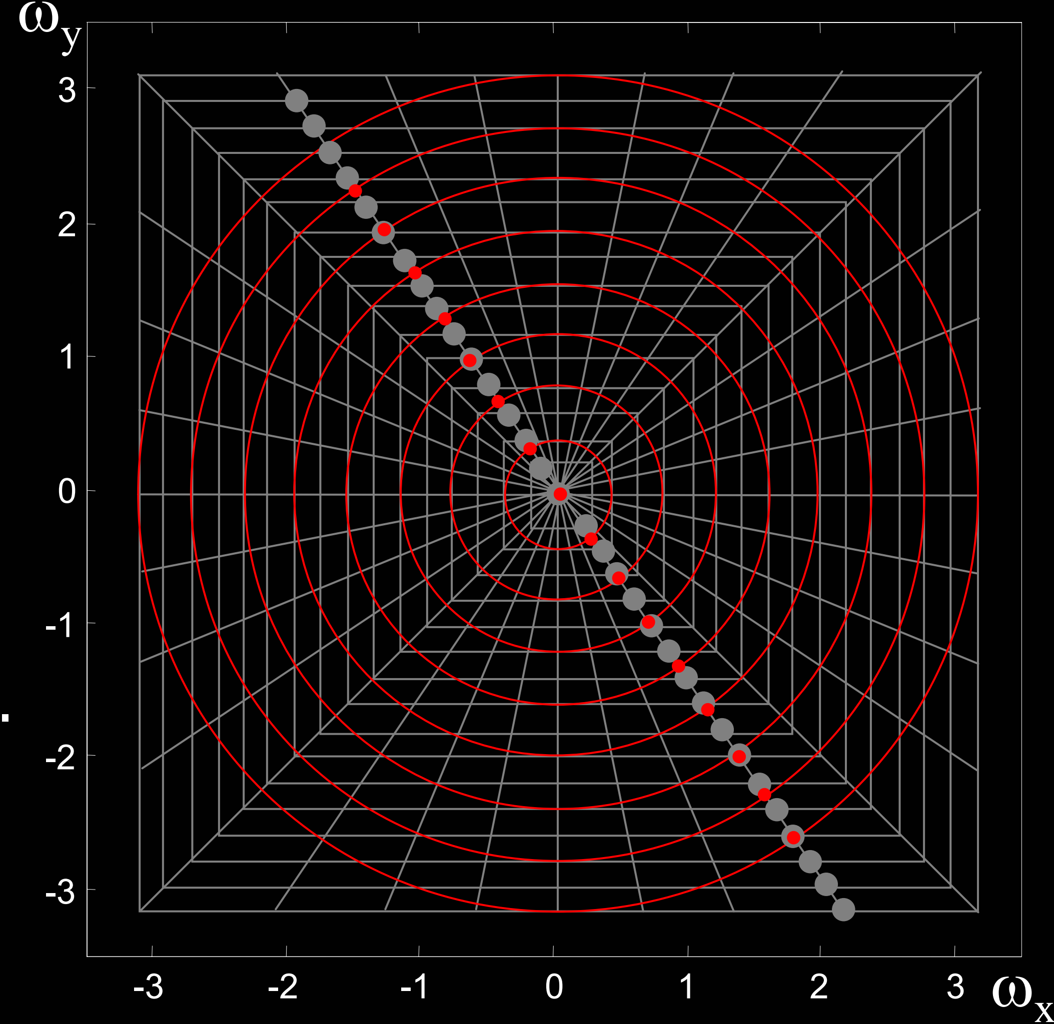
Second Interpolation Stage

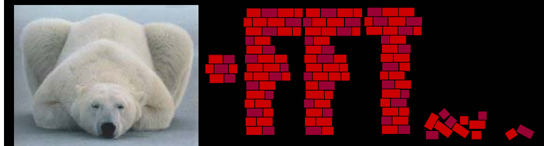
- As opposed to the previous step, the rays are not trigonometric polynomials of order N,
- We proved that the rays are band-limited (smooth) functions,
- Over-sampling and interpolation is expected to perform well.



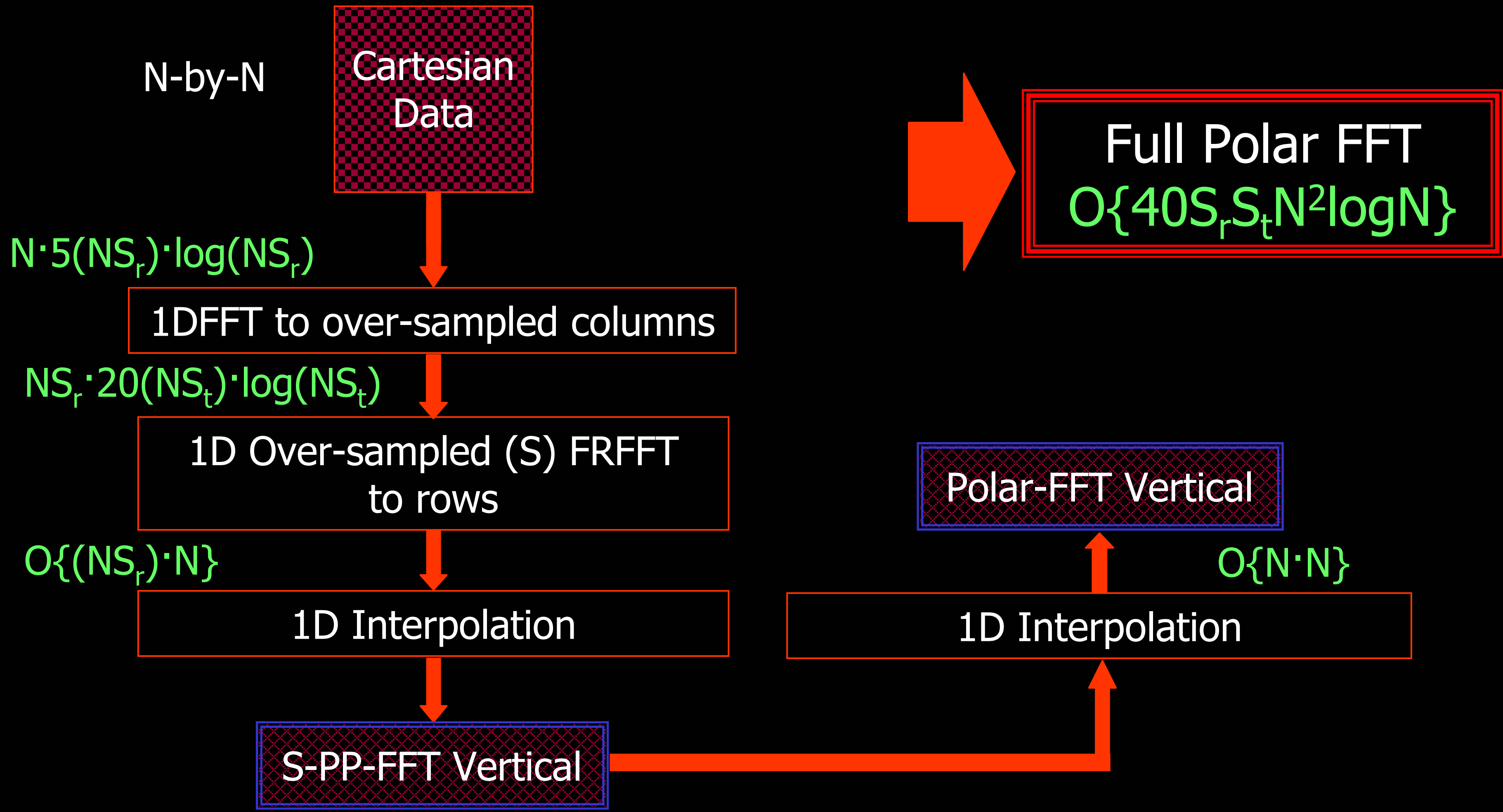
Over-Sampling Along Rays

- Over-sampling along rays cannot be done by taking the 1D ray and over-sampling it.
- S_r>1:
  - More concentric squares.
  - S_r longer 1D-FFT's at the beginning of the algorithm.
  - S_r times FRFFT operations.





The Actual Interpolation



To Summarize

We propose a

Fast Polar Fourier Transform

with the following features:

- Low complexity – $O(Const \cdot N^2 \log_2 N)$
- Vectorizability – 1D operations only
- Non-Expansiveness – Factor 2 (or 4) only
- Stability – via Regularization
- Accuracy – 2 orders of magnitude over USFFT methods

Agenda

1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Conclusions

7. Algorithm Analysis

We have a code performing the Polar-FFT in Matlab:

```
Y=Polar_FFT(X);
```

OR

$$Y=Polar\_FFT(X,S_t,S_r);$$

Where: X – Input array of N-by-N samples

S_t,S_r – Over-sampling factors in the approximations

Y – Polar-FFT result as an 2N-by-2N array with rows being the rays and columns being the concentric circles.

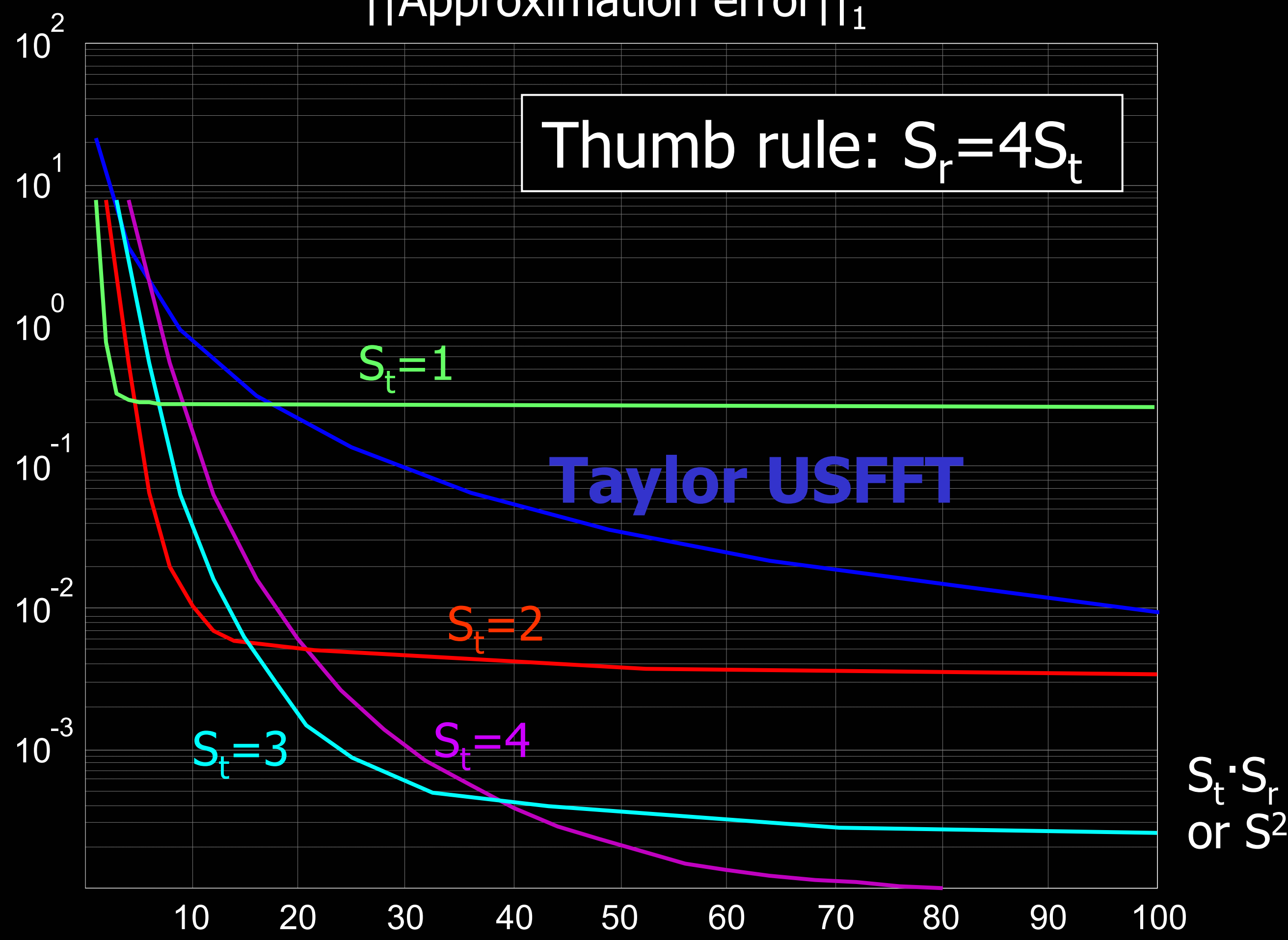
The Implementation

- The current Polar-FFT code implements Taylor method for the first interpolation stage and spline ONLY (no-derivatives) for the second stage.

- For comparison, we demonstrate the performance of the USFFT method with over-sampling S and interpolation based on the Taylor interpolation (found to be the best).

Error for Specific Signal

- Input is random 32-by-32 array,
- USFFT method uses one parameter whereas there are two for the up-sampling in the Polar-FFT.
- Thumb rule: $S_r \cdot S_t = S^2$.



Error For Specific Signals

- Similar curves obtained of $||*||_\infty$ and $||*||_2$ norms.
- Similar behavior is found for other signals.
- Conclusion: For the proper choice of S_t and S_r, we get 2-orders-of-magnitude improvement in the accuracy comparing to the best USFFT method.
- Further improvement should be achieved for Taylor interpolation in the second stage.
- US-FFT takes longer due the 2D operations.

The Transform as a Matrix

All the involved transformations (accurate and approximate) are linear - they can be represented as a matrix of size $4N^2$-by-N^2.

Approximate

$$\underline{Y}_a = \mathbf{A}\underline{x}$$

Or

$$\underline{Y}_t = \mathbf{T}\underline{x}$$

True

Regularization of T/A

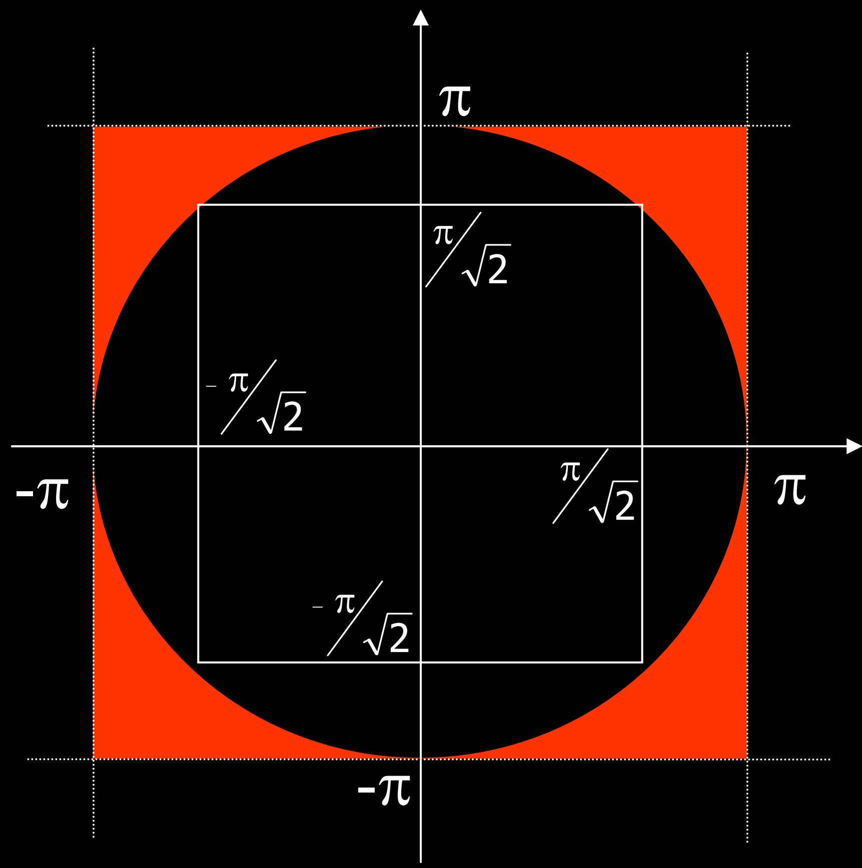
- An input signal of N-by-N is transformed to an array or 2N-by-2N.
- For N=16, **T** size is 1024-by-256, and $\kappa \approx 60{,}000$ (bad for inversion).
- Adding the assumption that the Frequency corners should be zeroed, we obtain

$$\underline{y} = \mathbf{T}_{Polar}\underline{x} \quad \Rightarrow \quad \begin{bmatrix} \mathbf{T}_{Polar} \\ \mathbf{T}_{Corner} \end{bmatrix} \underline{x} = \begin{bmatrix} \underline{y} \\ \underline{0} \end{bmatrix}$$

and the condition number becomes $\kappa \approx 5$!!!

Discarding the Corners?

- If the given signal was sampled at 1.4142 the Nyquist Rate, the corners can be removed.
- If it is not, over-sampling it can be done by FFT.



Error Analysis – Worst Signal

$$\text{Approximation error is : } \left(\mathbf{A}_{Polar-FFT} - \mathbf{T}\right)\underline{x} = \underline{e}_{Polar-FFT}$$

$$\text{Worst error : } \left\{\underline{x}_{worst}, e^2_{worst}\right\} = \underset{\underline{x}}{\text{Arg/Max}} \frac{\left\|\left(\mathbf{A}_{Polar-FFT} - \mathbf{T}_{Polar}\right)\underline{x}\right\|_2^2}{\left\|\underline{x}\right\|_2^2}$$

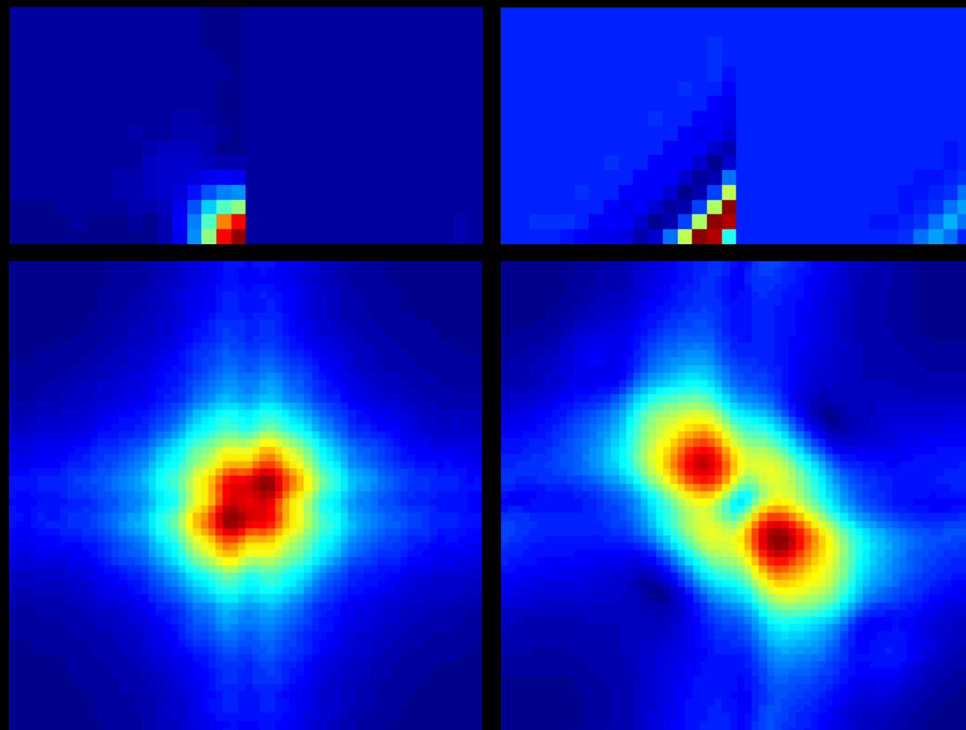
$$\text{Worst relative error : } \left\{\underline{x}_{rworst}, e^2_{rworst}\right\} = \underset{\underline{x}}{\text{Arg/Max}} \frac{\left\|\left(\mathbf{A}_{Polar-FFT} - \mathbf{T}_{Polar}\right)\underline{x}\right\|_2^2}{\left\|\mathbf{T}_{Polar}\underline{x}\right\|_2^2}$$

Worst Signal - Results

$N=16 \rightarrow \mathbf{T} \in \mathcal{C}^{1024 \times 256}$, $S=S_r=S_t=4$

USFFT
worst signal (abs. Value) $\lambda=3.469$

The worst case signal in the freq. Domain (abs. and shifted)



Polar-FFT
worst signal (abs. Value) $\lambda=0.0319$

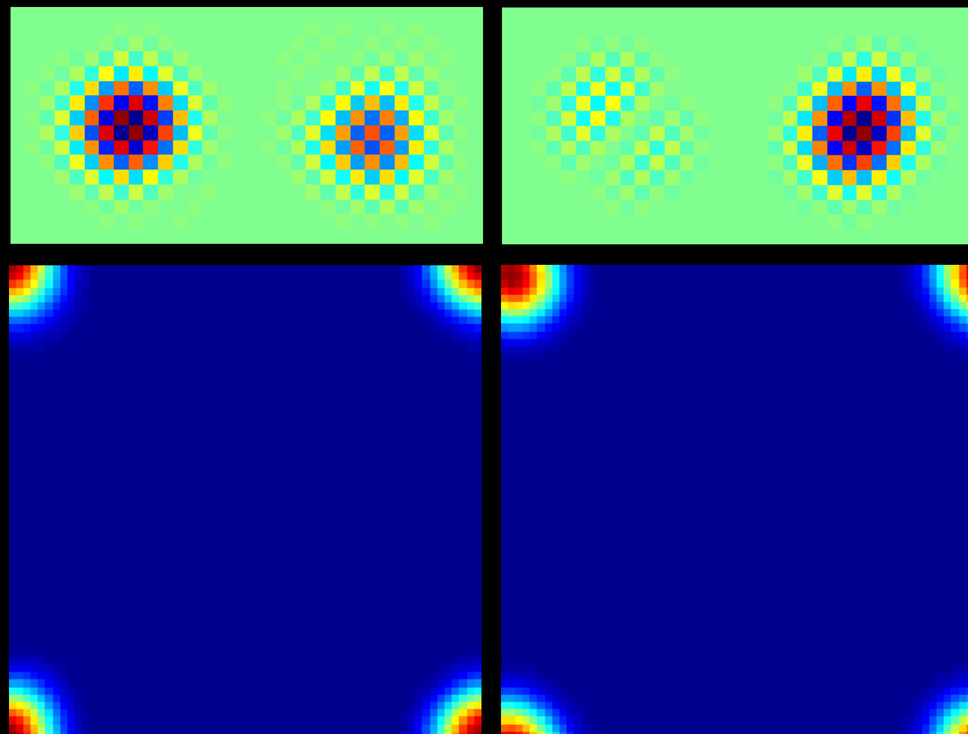
The worst case signal in the freq. Domain (abs. and shifted)

Relative Worst Signal - Results

Same parameters: $N=16 \rightarrow \mathbf{T} \in \mathcal{C}^{1024 \times 256}$, $S=S_r=S_t=4$

USFFT worst signal (abs. Value) $\lambda=0.0613$

Polar-FFT worst signal (abs. Value) $\lambda=0.0023$



The worst case signal in the freq. Domain (abs. and shifted)

The worst case signal in the freq. Domain (abs. and shifted)

Comparing Approximations

- Solve for the eigenvalue/vectors of the matrix

$$\left(\mathbf{A}_{\text{Polar}-\text{FFT}} - \mathbf{T}_{\text{Polar}}\right)^{H}\left(\mathbf{A}_{\text{Polar}-\text{FFT}} - \mathbf{T}_{\text{Polar}}\right)$$

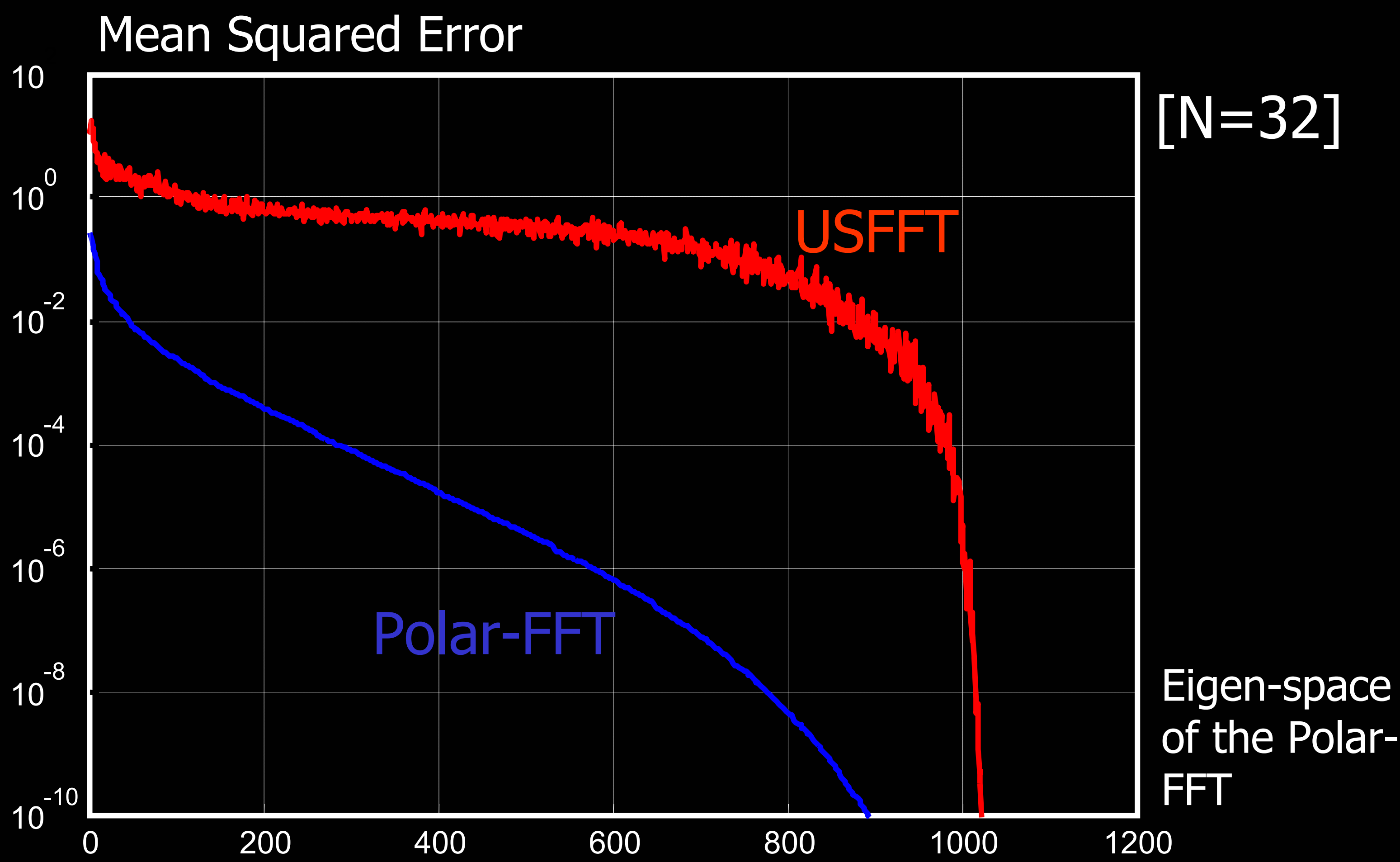
  and obtained $\{\lambda_k, \underline{x}_k\}_{k=1}^{N^2}$ (λ_k in ascending order).

- Compare to $\mathbf{A}_{\text{USFFT}}$ by computing

$$\alpha_k = \left\| \left(\mathbf{A}_{\text{USFFT}} - \mathbf{T}_{\text{Polar}}\right)\underline{x}_k \right\|_2^2$$

  using the above eigenvectors and compare to λ_k.

Comparing Approximations - Results



Agenda

1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Conclusions

8. Conclusions

- We have proposed a fast, accurate, stable, and reliable Polar Discrete-Fourier-Transform.
- By this we extend utility of FFT algorithms to new class of settings in image processing.
- Future plans:
  - Extend the analysis and improve further,
  - Demonstrate applications,
  - Publish the code for the procedure and some applications over the internet.

Beyond this slides are the appendix or redundant slides

USFFT for $T^{\dagger}$

- Over-sample Polar grid (and possibly partial derivatives).
- Associate polar neighbors to each Cartesian grid point.
- Approximate interpolation to get the Cartesian grid values.
- Perform the Cartesian 2D Inverse-FFT.

Our Reading of Literature

Direct Fourier method with over-sampling and interpolation (also called gridding) – see

- Natterer (1985).
- Jackson, Meyer, Nishimura and Macovski (1991).
- Schomberg and Timmer (1995).
- Choi and Munson (1998).

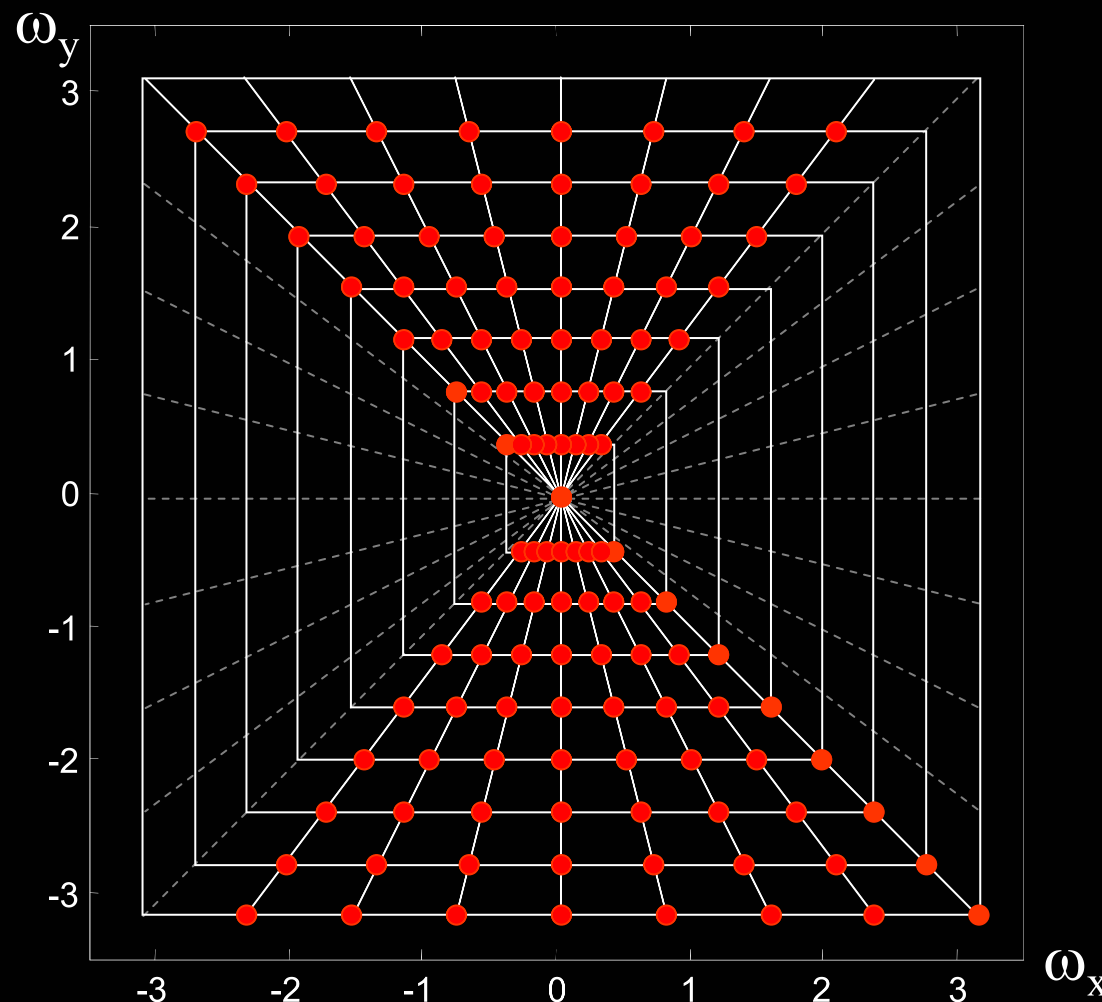
The Pseudo-Polar Sampling

Basically vertical lines:

$$\left\{\omega_y = \frac{2\pi\ell}{NS_r}\right\}_{\ell=-NS_r/2}^{NS_r/2-1}$$

$$\left\{\omega_x = \frac{2m}{NS_t}\omega_y\right\}_{m=-NS_t/2}^{NS_t/2-1}$$

For $S_t=S_r=1$, we have N^2 grid points



The Pseudo-Polar FT – Stage 1

$$F(\omega_x,\omega_y)=\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} f[k_1,k_2]\exp\{-ik_1\omega_x-ik_2\omega_y\}=$$

$$=\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} f[k_1,k_2]\exp\left\{-ik_1\frac{2m}{NS_t}\omega_y-ik_2\omega_y\right\}=$$

$$=\sum_{k_1=0}^{N-1}\exp\left\{-ik_1\frac{2m}{NS_t}\omega_y\right\}\underbrace{\sum_{k_2=0}^{N-1} f[k_1,k_2]\exp\{-ik_2\omega_y\}}_{=\hat{f}[k_1,\ell]}$$

This part is obtained by 1D-FFT along the rows !!

The Pseudo-Polar FT – Stage 2

$$F(\omega_x, \omega_y) = F[m, \ell] = \sum_{k_1=0}^{N-1} \hat{f}[k_1, \ell] \exp\left\{-ik_1 m \frac{2\omega_y}{NS_t}\right\}$$

- This summation takes columns of $\hat{f}[k_1, \ell]$ (being equi-spaced 1D signals) and computes Fourier transform of it.
- The destination grid points are also 1D equi-spaced in the frequency domain, BUT THEY ARE NOT IN THE RANGE $[-\pi,\pi]$, but rather $[-\omega_y,\omega_y]$.
- This task is called Fractional Fourier/Chirp-Z Transform.

Fractional Fourier Transform

$$F[m]=\sum_{k=0}^{N-1} f[k]\exp\left\{-i\frac{2\pi km}{N}\cdot\alpha\right\}$$

- For α=1 we get the ordinary 1D-FFT,
- For α=-1 we get the ordinary 1D-IFFT,
- There exists a Fast Fractional Fourier Transform with the complexity of $O(20\cdot N\log_2 N)$, based on 1D-FFT operations.

See: Fast fractional Fourier transforms and applications, by D. H. Bailey and P. N. Swarztrauber, *SIAM Review*, 1991, and also Bluestein, Rabiner, and Rader (1960's).

FR-FFT Detailed

$$F[m] = \sum_{k=0}^{N-1} f[k]\exp\left\{-i\frac{2\pi km}{N}\cdot\alpha\right\} =$$

$$= \sum_{k=0}^{N-1} f[k]\exp\left\{-i\frac{\pi\left[(k-m)^2-k^2-m^2\right]}{N}\cdot\alpha\right\} =$$

$$= e^{i\frac{\pi m^2}{N}\alpha}\cdot\sum_{k=0}^{N-1} f[k]\cdot e^{i\frac{\pi k^2}{N}\alpha}\cdot\exp\left\{-i\frac{\pi(k-m)^2}{N}\alpha\right\}$$

Post Multiplication

Pre-Multiplication

Convolution

[Back]

Interpolation As 1D Operation

$$F(\omega_x,\omega_y)=\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} f[k_1,k_2]\exp\{-ik_1\omega_x-ik_2\omega_y\}=$$

$$=\sum_{k_1=0}^{N-1}\exp\left\{-ik_1\tan\left(\frac{m\pi}{2NS_t}\right)\omega_y\right\}\sum_{k_2=0}^{N-1} f[k_1,k_2]\exp\{-ik_2\omega_y\}=$$

$$=\sum_{k_1=0}^{N-1}\exp\left\{-ik_1\tan\left(\frac{m\pi}{2NS_t}\right)\omega_y\right\}\hat{f}[k_1,\ell]$$

- ❑ It is a 1D operation – But it is not the Fractional-FFT.
- ❑ Can be computed by over-sampled FRFFT and interpolation.