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-Math-Israel), and Ronald Coifman (Yale-Math)

Collaborators

Dave Donoho

Statistics Department
Stanford

Amir Averbuch

Math. Department
Tel-Aviv University

Ronald Coifman

Math. Department
Yale

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. Open questions & Future work

Background & Motivation

New Approach and its Results

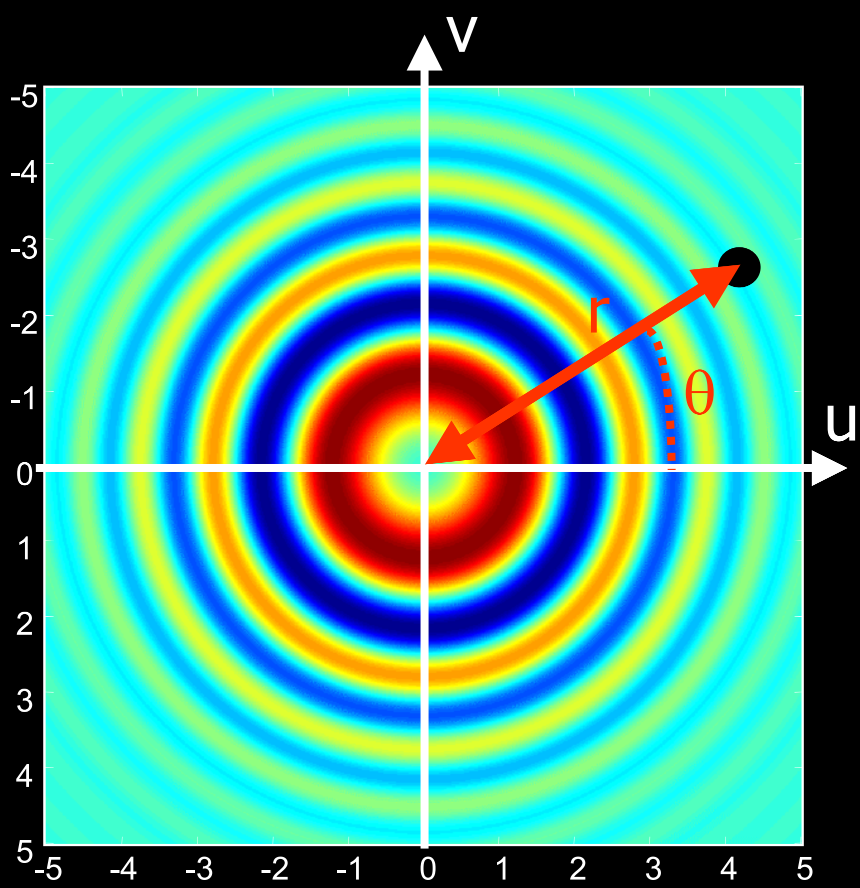
1. Thinking Polar - Continuum

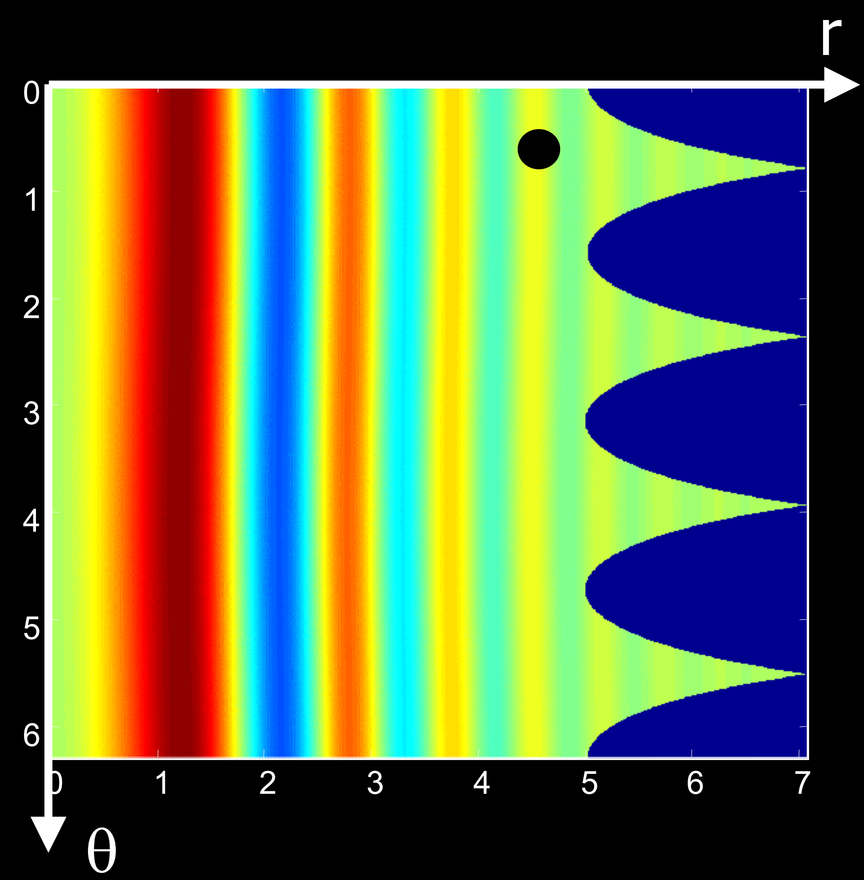
- For today f(x,y) function of $(x,y)\in\Re^2$
- Continuous Fourier Transform

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

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

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


v
u
r
θ
-5
-4
-3
-2
-1
0
1
2
3
4
5


r
θ
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.

- Let Q_{θ_0} be the operator of planar rotation by θ_0 degrees
- Rotation of a function $f_{\theta_0}(x, y) = f\left(Q_{\theta_0}\{x, y\}\right)$
- 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.

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

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

- Using log-Polar coordinates - 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) satisfy: $f(x,y) = g\left(Q_{\theta_0}\{x,y\} + \{x_0,y_0\}\right)$
- Goal: given f and g, recover $\{x_0, y_0, \theta_0\}$.
- Obtaining $\tilde{f}(r,\theta)$ and $\tilde{g}(r,\theta)$, angular cross-correlation on the absolute values gives the angle θ_0.
- After compensating for this angle, 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:

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

- Goal: Given $Rf(t,\theta)$, recover f.
- Projection-Slice-Theorem: $\left(\Im_1 Rf\right)(t,\theta) = \tilde{f}(r,\theta)$
- Reconstruction: $Rf \mapsto \tilde{f} \mapsto \hat{f} \mapsto f$

Natural Operations: 5. Singularities

- f is smooth (C^∞) apart from discontinuity along line

$$x\cos(\theta_0) + y\sin(\theta_0) = t$$

- Radial asymptotics for $\theta \neq \theta_0$:

$$\tilde{f}(r,\theta) = o\{r^{-m}\}, \ r \to \infty, m = 1,2,3,\ldots$$

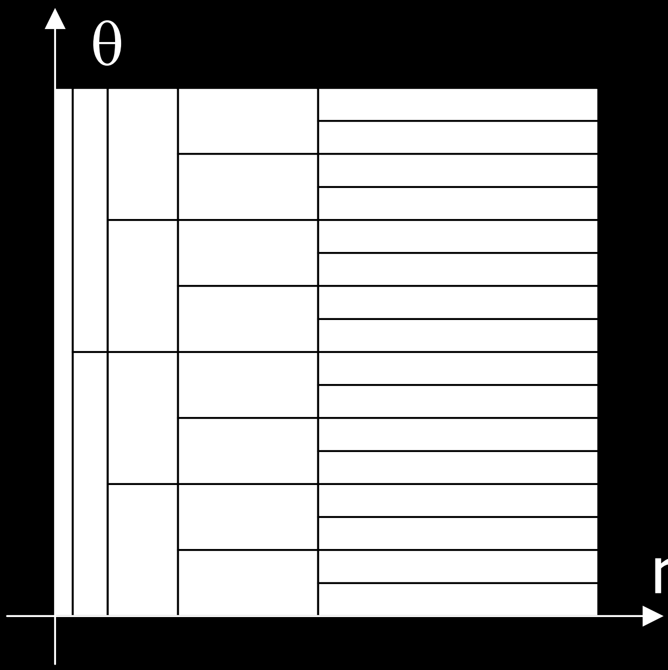
- Radial asymptotics for $\theta = \theta_0$: $\tilde{f}(r,\theta) = c/|r|, \ r \to \infty$

- For curves (– vertical to their direction):

$$\tilde{f}(r,\theta) = c/|r|^{3/2}, \ r \to \infty$$

Natural Operations: More

- New orthonormal bases
  - Ridgelets
  - Curvelets
  - Fourier Integral operations
  - Ridgelet packets
- Analysis of textures
- 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. Open questions & Future work

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 Miracle

- 1D Discrete Fourier Transform
  - The miracle of 1D fast procedure is built on the fact that we have equi-spaced points in time and frequency.
  - The result is 1D-FFT. Its complexity is $O(5N\log_2 N)$ operations instead of $O(N^2)$ by direct approach.
- 2D Discrete Fourier Transform
  - Space and frequency grids are both Cartesian.
  - Can be broken (separable) into 1D-FFT operations.

2D DFT - Definition

- The 2D-Fourier Transform on $\{f[k_1,k_2],\ 0 \le k_1,k_2 < N\}$

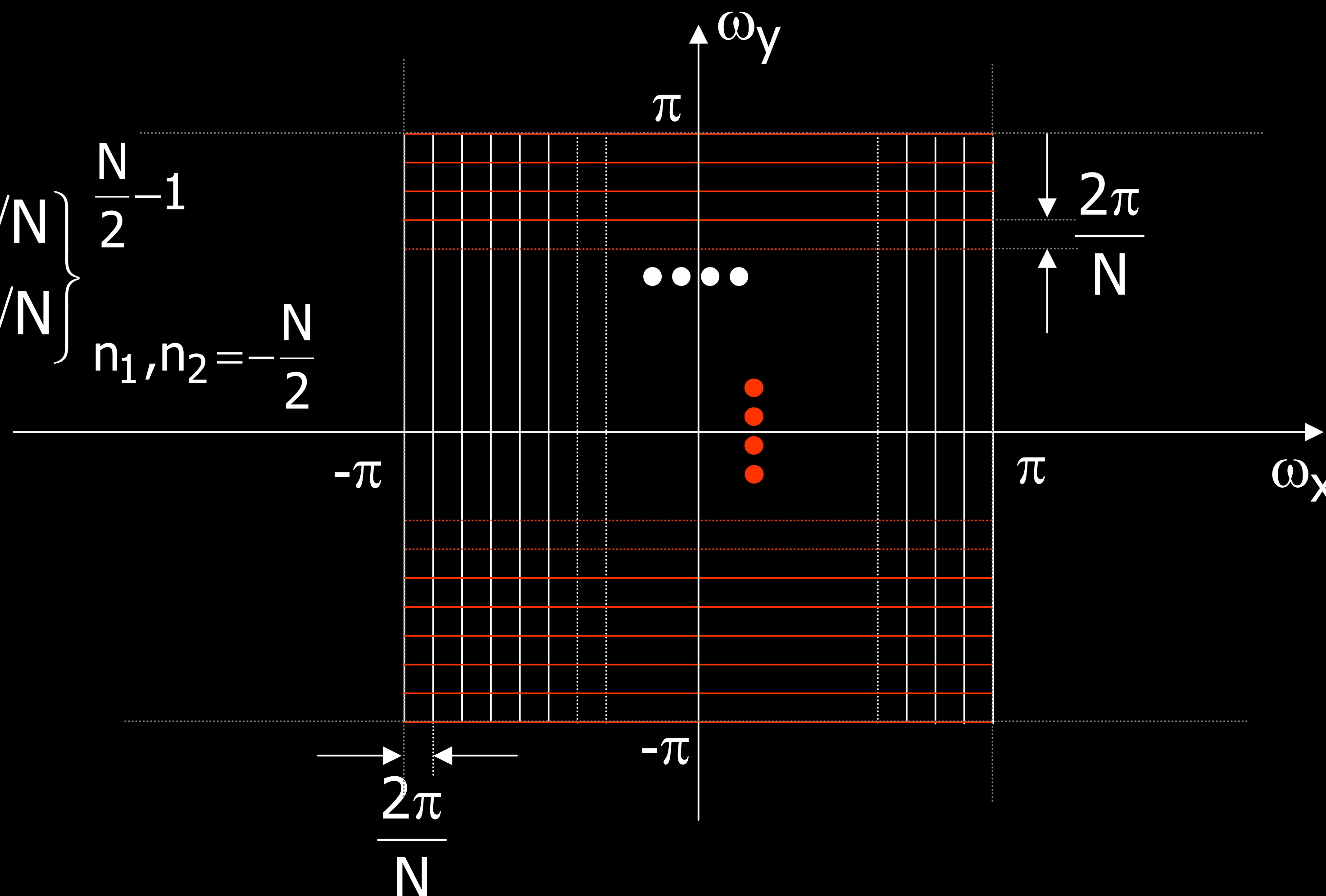
$$F(\omega_x,\omega_y) = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} f[k_1,k_2] \exp\{-ik_1\omega_x - ik_2\omega_y\}$$

- 2D-DFT for $\{\omega_x = 2\pi n_1/N,\ \omega_y = 2\pi n_2/N\}_{n_1,n_2=0}^{N-1}$

$$F[n_1,n_2] = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} f[k_1,k_2] \exp\left\{-\frac{i2\pi}{N}(k_1n_1 + k_2n_2)\right\}$$

2D DFT – Frequency Grid

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



2D FFT

$$F[n_1,n_2] = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} f[k_1,k_2] \exp\left\{-\frac{i2\pi}{N}(k_1 n_1 + k_2 n_2)\right\}$$

$$= \sum_{k_1=0}^{N-1} \exp\left\{-\frac{i2\pi k_1 n_1}{N}\right\} \sum_{k_2=0}^{N-1} f[k_1,k_2] \exp\left\{-\frac{i2\pi k_2 n_2}{N}\right\}$$

- $\hat{f}[k_1,n_2]$ is obtained by 1D-FFT over the rows of $f[k_1,k_2]$ and fill into the original array.
- The remaining stage is to do the same on the columns of $\hat{f}[k_1,n_2]$ obtaining $F[n_1,n_2]$.

2D FFT Complexity

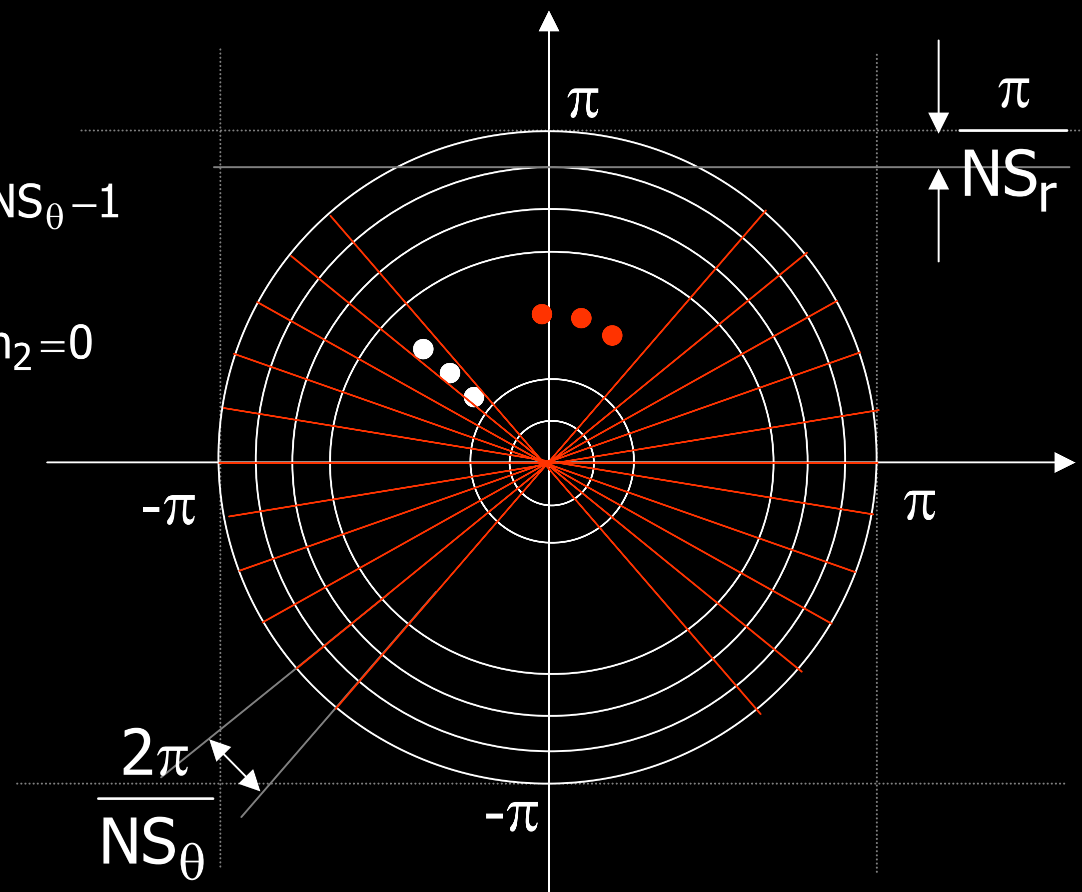
- Applying 1D-FFT on a sequence of N samples requires $O(5N\log_2 N)$ operations.
- In the 2D FFT, we have N such 1D FFT-s for the rows and N such 1D FFT-s for the columns.
- Thus: 2D-FFT requires $O(10N^2\log_2 N)$ operations, instead of $O(N^4)$ by the direct approach.
- Important Feature: All operations are 1D operations, processing one columns/row at a time – leading to **efficient cache usage**.

Discrete Polar Coordinates?

Choice of grid?

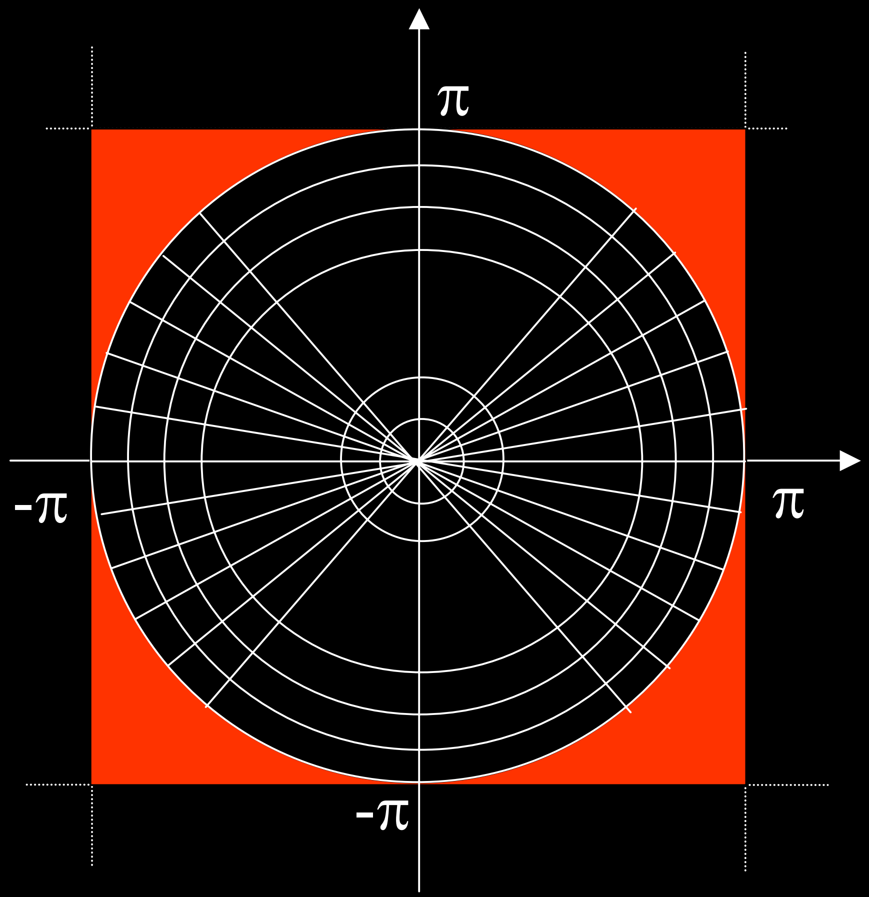
$$\left\{ r = \frac{\pi n_1}{NS_r} \right\}_{n_1=0}^{NS_r-1}, \left\{ \theta = \frac{2\pi n_2}{NS_\theta} \right\}_{n_2=0}^{NS_\theta-1}$$

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 Belief

- Current widespread belief is that there cannot be a fast method for implementing the Discrete Fourier Transform (DFT) on the polar grid.
- See for example
  - "The DFT: an owner's manual" by Briggs and Henson, SIAM, 1995.
  - A comprehensive and authoritative book on the Discrete Fourier Transform.
  - Index item: ***Polar Domain.***
  - Index entry: no FFT for polar domain (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. Open questions & Future work

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.

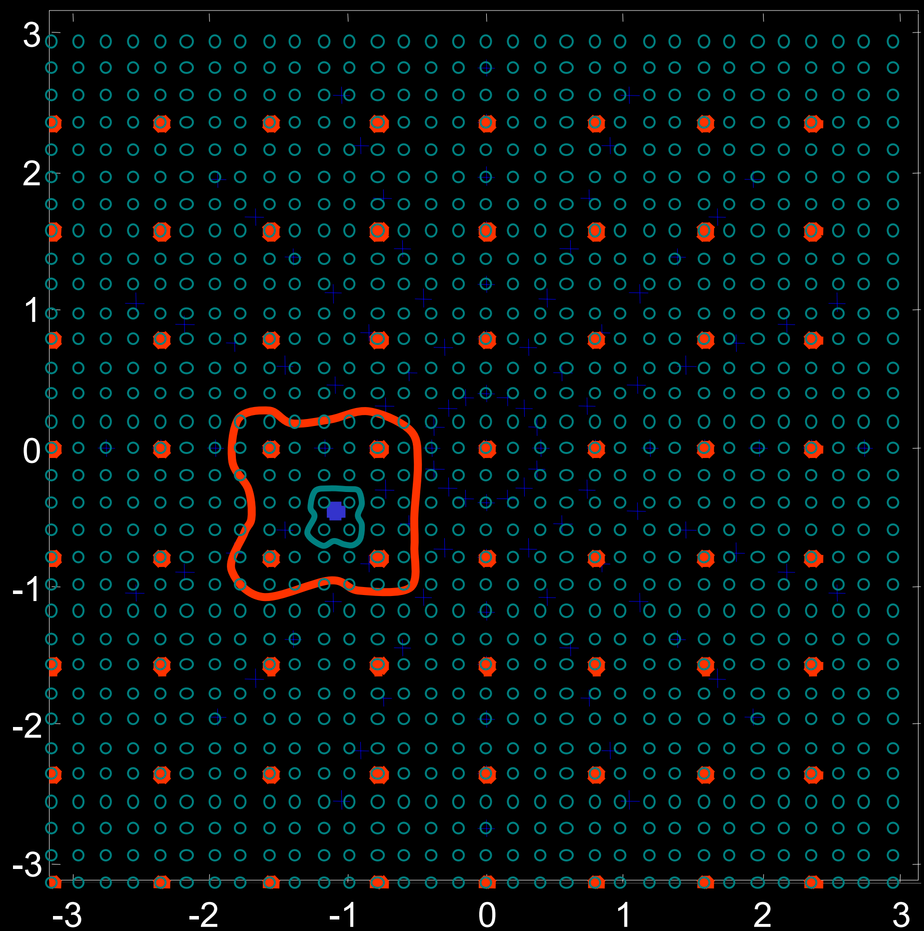
- Assessing T: Unequally-spaced FFT (USFFT)
  - Data in Cartesian set.
  - Approximate transform in non-Cartesian set.
  - Oriented to 1D – not 2D and definitely not Polar.

USFFT - Rational

- Define over-sampled Cartesian grid.
- Rapidly evaluate FT at over-sampled grid:
  - Function values F.
  - (possibly) partial derivatives through order L.
- Associate several neighboring Cartesian points to each polar grid point.
- Approximate interpolation using the Cartesian values (and derivatives) to get the polar grid values.

USFFT - Interpolation

- \+ Destination Polar grid
- ● Critically sampled Cartesian grid
- o Over-sampled Cartesian grid



Our Reading of Literature

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

USFFT for $T^{\dagger}$

- Define over-sampled Polar grid.
- Rapidly evaluate FT at over-sampled grid:
  - Function values $\tilde{F}$.
  - (possibly) partial derivatives through order L.
- Associate several neighboring Polar points to each Cartesian grid point.
- Approximate interpolation using the Polar values (and derivatives) 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).

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, and 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. Open questions & Future work

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

Strategy to obtain Polar FFT

- Define an intermediate "Polar-Like" grid that
  - Is close to the polar grid,
  - It enables fast and exact FFT evaluation with 1D FFT-s only, and
  - It enables going to the exact polar grid by 1D interpolations.

  This will be the Pseudo-Polar grid.
- Close the gap between Pseudo-Polar and exact Polar by 1D high-accuracy interpolation.

Agenda

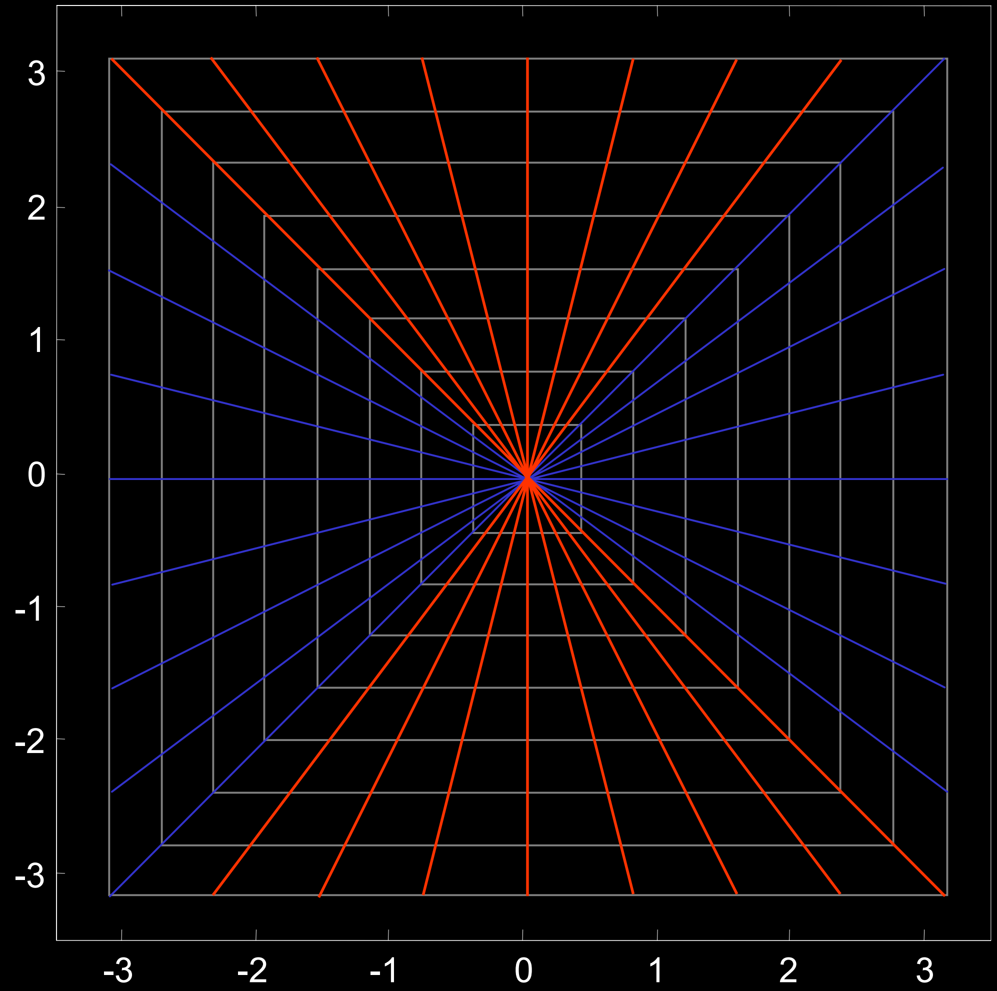
1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Open questions & Future work

5. The Pseudo-Polar FFT

- Developed by Averbuch, Coifman, Donoho, Israeli, and Waldén (1998).
- Basic idea: Define a "Polar-Like" grid that replaces concentric circles by squares, and equispaced angles by equispaced distances.
- The resulting grid enables fast Fourier Transform evaluation – ACCURATELY!!!
- Applications: Tomography, image processing, Ridgelets, and more.

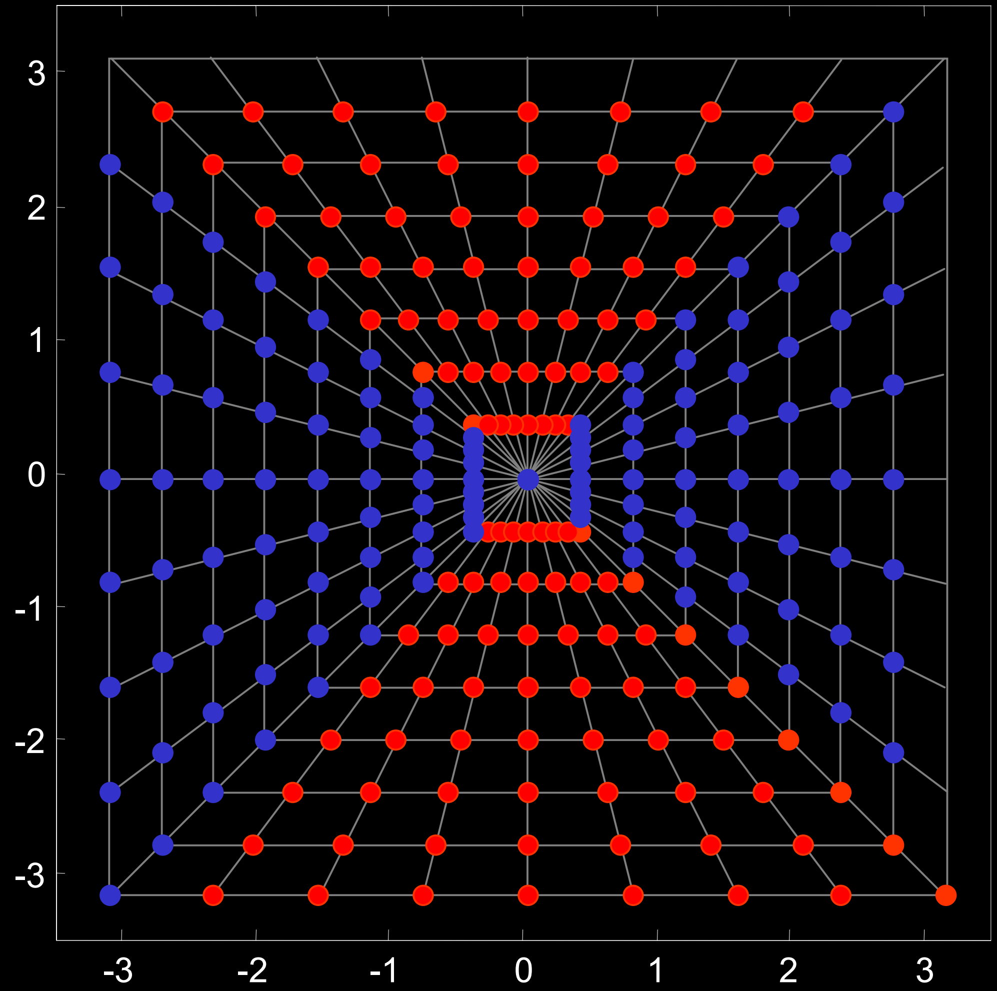
The Pseudo-Polar Skeleton

- NS_r equi-spaced concentric squares,
- NS_t equi-spaced points along each line (end-to-end),
- We separate our treatment to basically vertical and basically horizontal lines.



The Pseudo-Polar Grid

- In this example, we have 16 lines with 16 points along each one.
- Note that the outer square is not fully covered.
- The angles are not equally-spaced!!



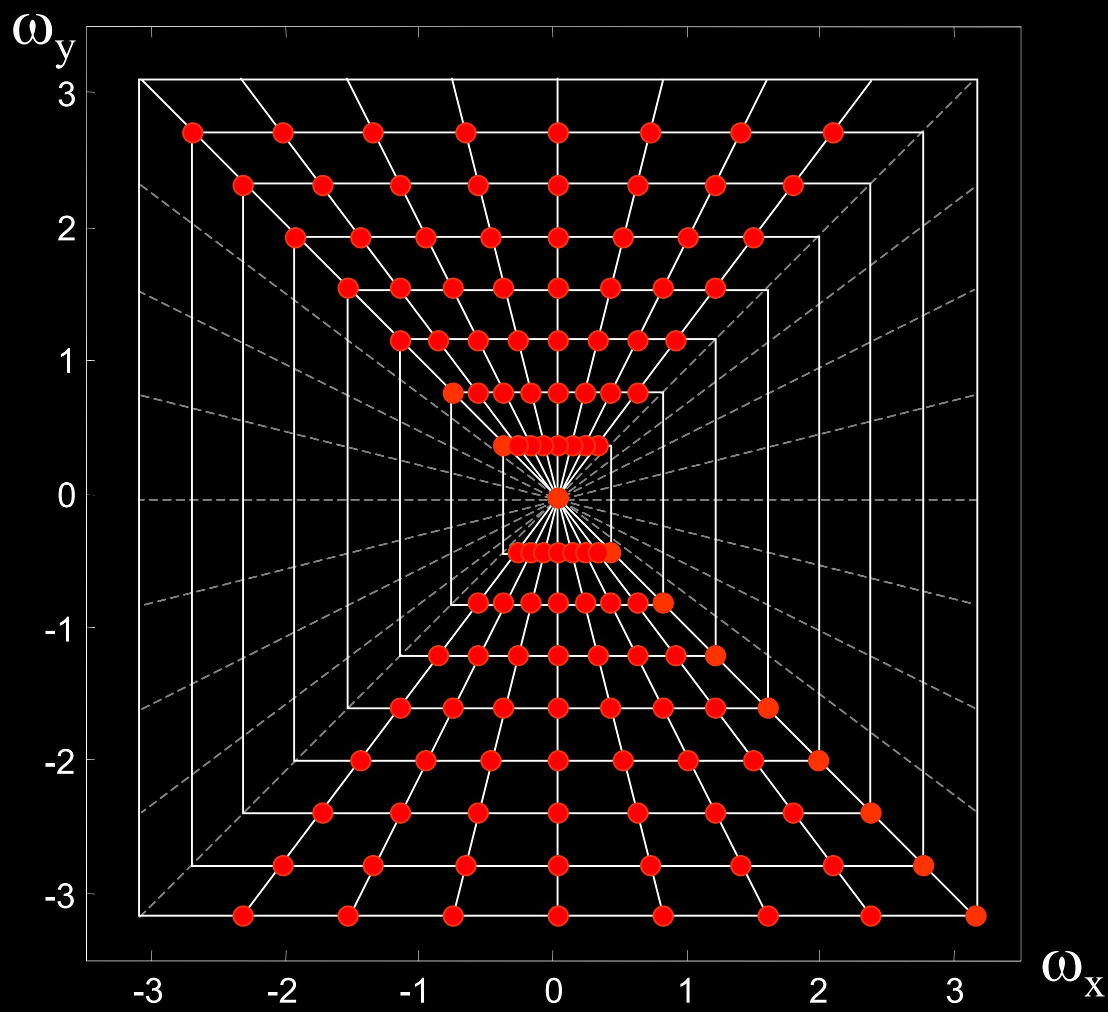
The Pseudo-Polar Sampling

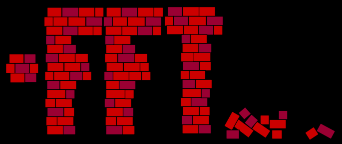
Basically vertical lines:

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

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

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





The Pseudo-Polar FT – Stage 1

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

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

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

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.

Fast Track

Fractional Fourier Transform

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

- 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).

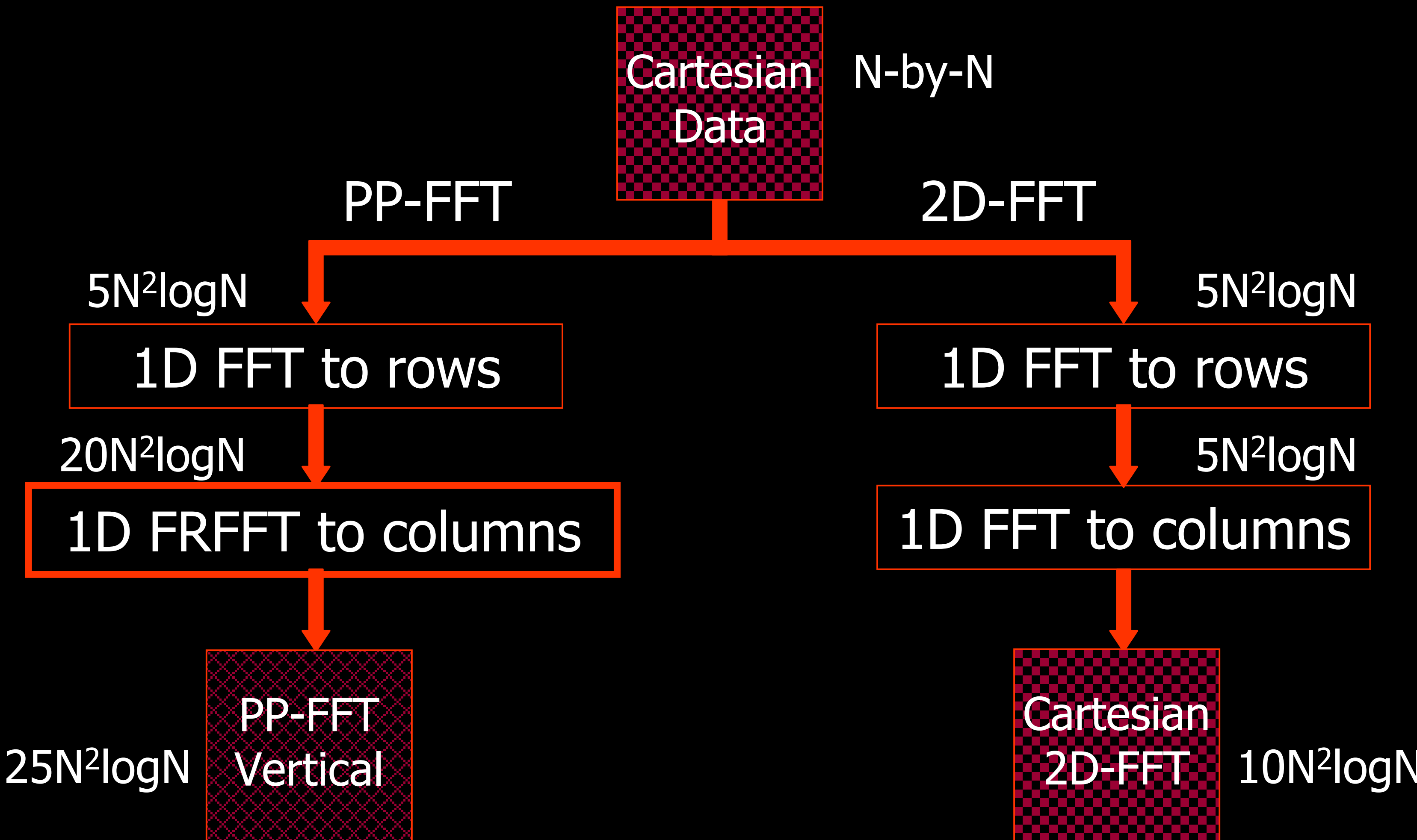
FR-FFT Detailed

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

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

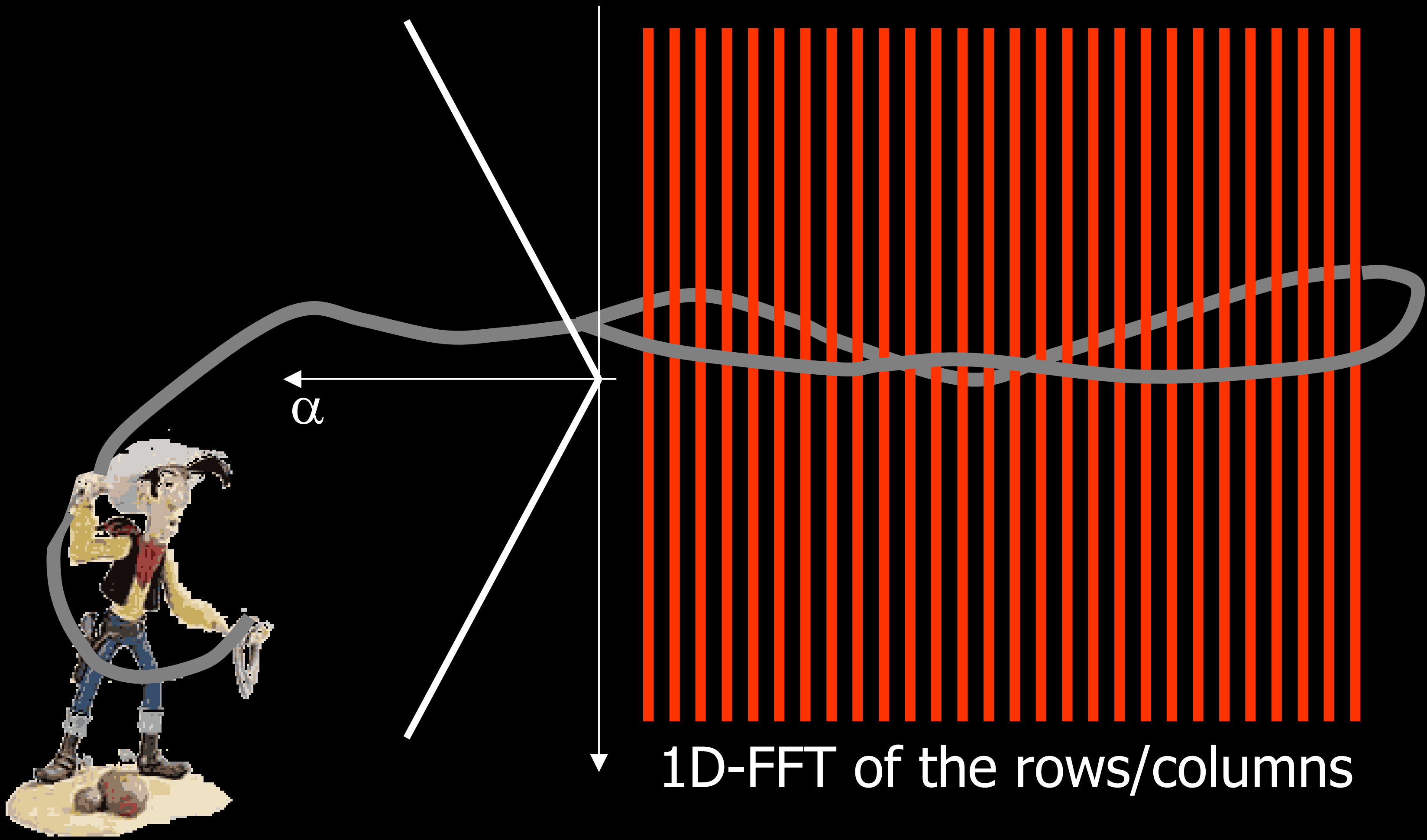
$$=\underbrace{e^{i\frac{\pi m^2}{N}\alpha}}_{\text{Post Multiplication}}\cdot\underbrace{\sum_{k=0}^{N-1}\underbrace{f[k]\cdot e^{i\frac{\pi k^2}{N}\alpha}}_{\text{Pre-Multiplication}}\cdot\exp\left\{-i\frac{\pi(k-m)^2}{N}\alpha\right\}}_{\text{Convolution}}$$

PP-FFT versus 2D-FFT



Fast Track

PP-FFT Intuition



The PP-FFT - Properties

- The transform is exact in exact arithmetic – no approximations involved.
- There are no parameters involved !!
- Order of complexity - $O(50 \cdot N^2 \log_2 N)$, compared to $O(N^4)$ in the direct approach.
- Only 1D operations are required.
- The chosen grid yields a stable transform with low condition number (~5).

Agenda

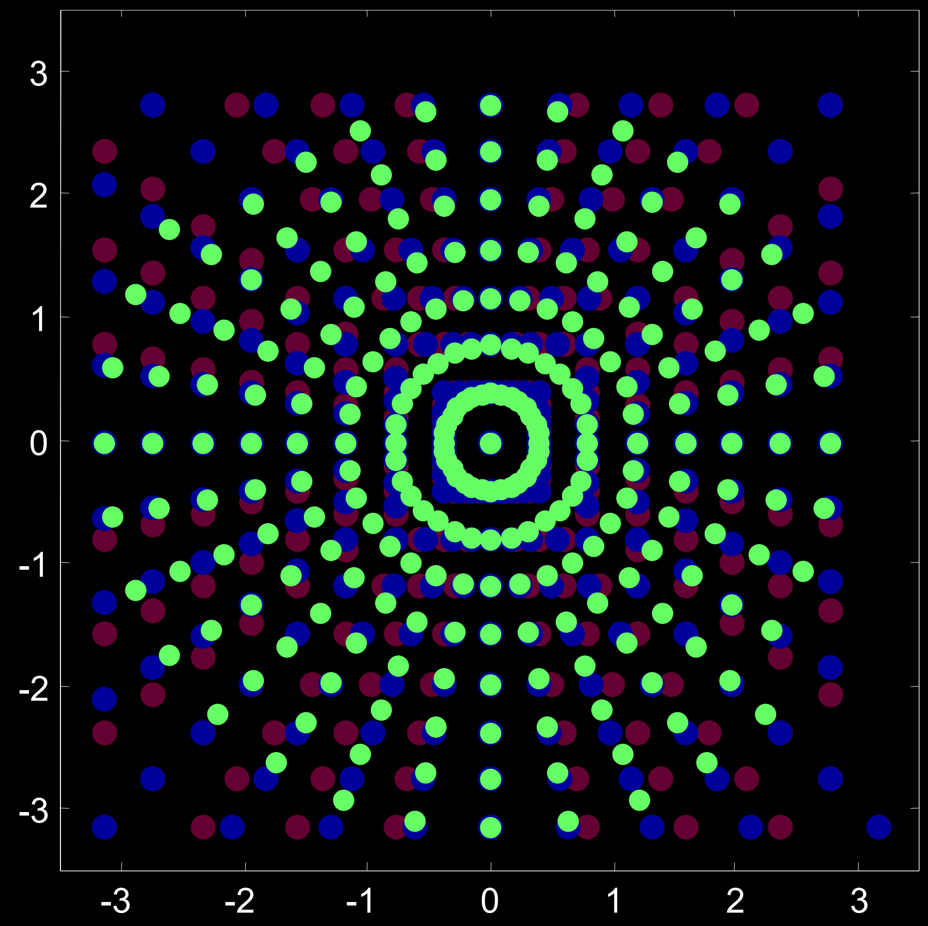
1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Open questions & Future work

6. From Pseudo-Polar to Polar

- Goal: Given the Fourier Transform on the Pseudo-Polar grid, approximate (accurately) the values on the Polar grid.
- Method: Two stages of **1D** interpolations
  - Replacing every row/column on the concentric squares with new angularly equi-spaced points.
  - Replacing every line (ray) with equi-spaced points with a new set of values on this very line.

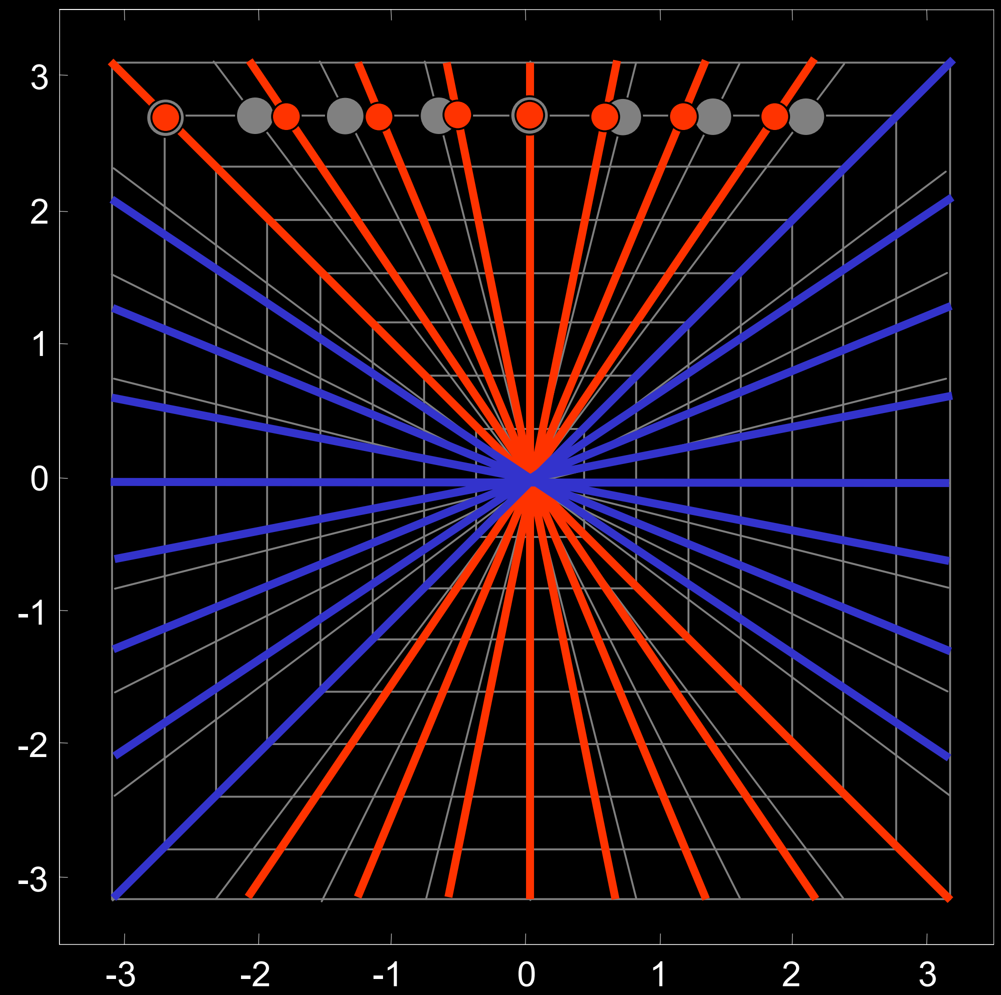
The Interpolation Stages

- 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):



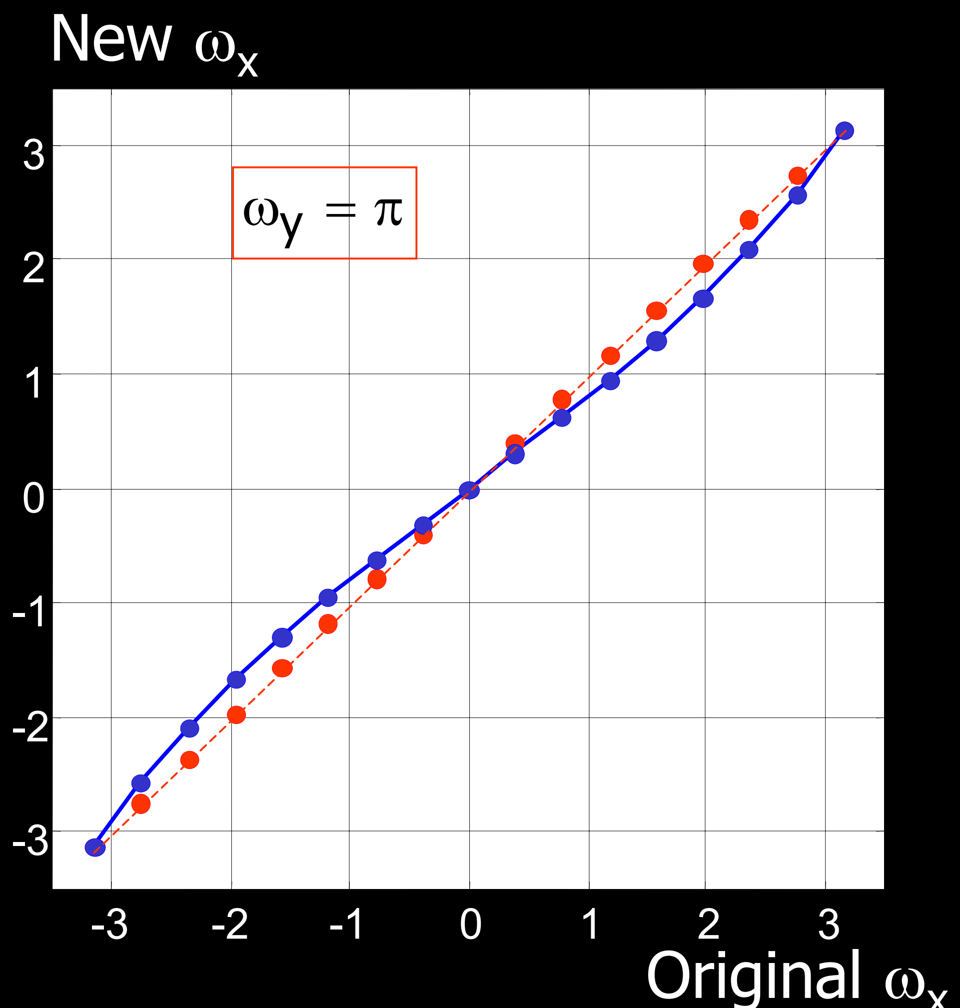
- Warping each set of points on a side of a concentric square.
- Done separately for the basically horizontal and vertical lines.

The Required Warping

Basically vertical lines:

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

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



Interpolation As 1D Operation

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

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

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

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

The Actual Interpolation

Cartesian Data — N-by-N

PP-FFT:

- $5N^2\log N$ → 1D FFT to rows
- $20N^2\log N$ → 1D FRFFT to columns
- $25N^2\log N$ → PP-FFT Vertical

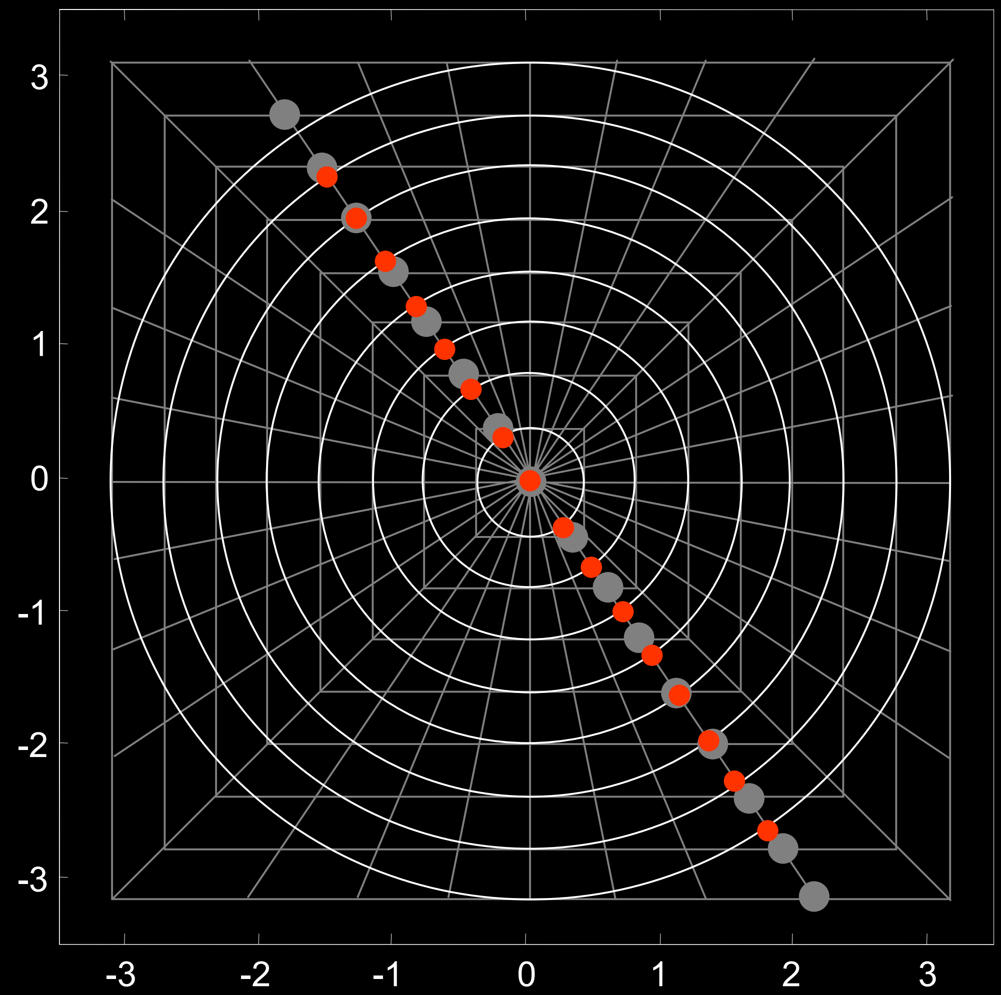
S-PP-FFT:

- $5N^2\log N$ → 1D FFT to rows
- $20N^2S\cdot\log(NS)$ → 1D Over-sampled (S) FRFFT to columns
- $O\{N^2\}$ → 1D Interpolation
- $(20S+5)N^2\log N$ → S-PP-FFT Vertical

Second Interpolation Stage

Squeezing the rays to have the same equi-spaced sampling (Creating the Polar grid):

- Warping each set of points along a ray.
- Both the source and destination seq. are uniformly sampled.



Interpolation as 1D Operation

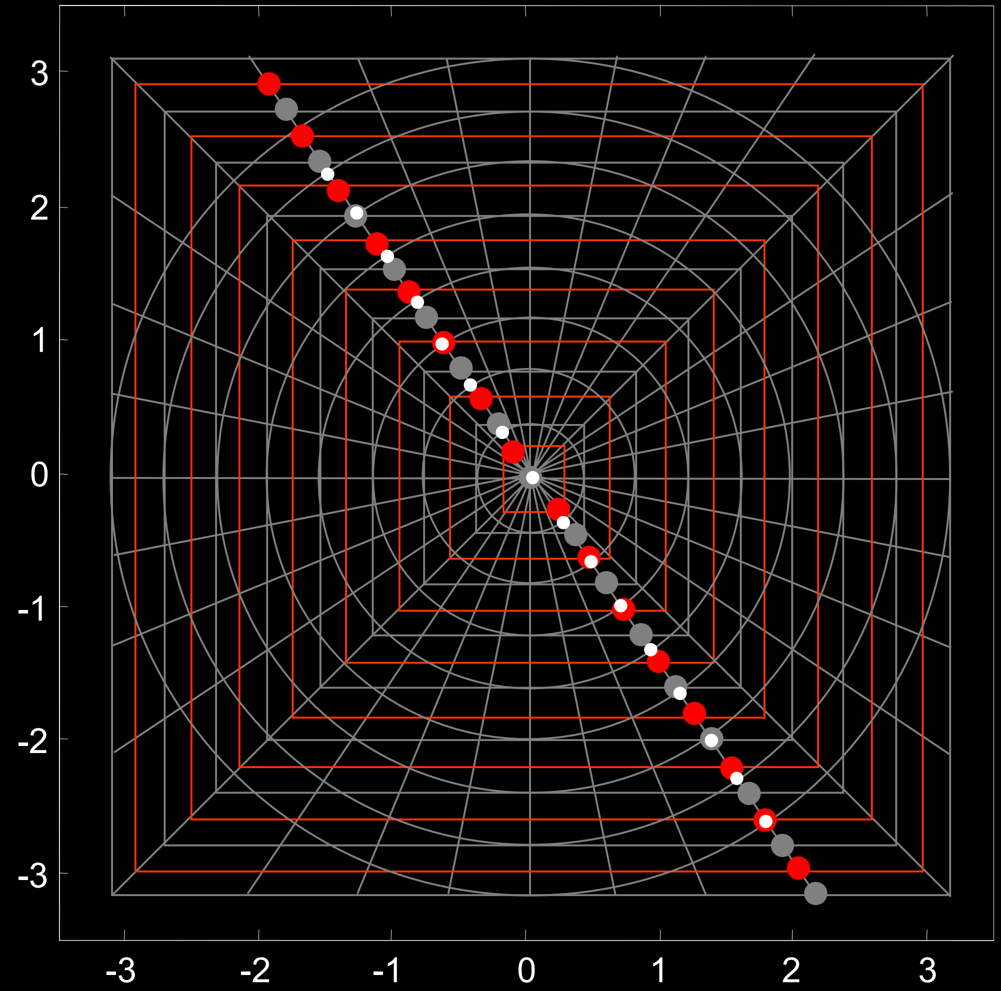
- As opposed to the previous interpolation stage, this stage requires the 2D data to be perfectly exact. Thus, 1D interpolation is an approximation!!
- BUT: we have proven that the function

$$F(r,\theta)=\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}\hat{f}[k_1,k_2]e^{-ir\cdot(k_1\cos(\theta)+k_2\sin(\theta))}$$

is band limited (i.e. smooth) and thus low S_r values (around 2-4) lead to good accuracy.

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

N-by-N

Cartesian Data

$N\cdot 5(NS_r)\cdot\log(NS_r)$

1D FFT to over-sampled rows

$NS_r\cdot 20(NS_t)\cdot\log(NS_t)$

1D Over-sampled (S) FRFFT to columns

$O\{(NS_r)\cdot N\}$

1D Interpolation

S-PP-FFT Vertical

1D Interpolation

$O\{N\cdot N\}$

Polar-FFT Vertical

Full Polar FFT
$O\{40S_rS_tN^2\log N\}$

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. Open questions & Future work

7. Algorithm Analysis

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

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

OR

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

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

S_t,S_r – Over-sampling factors in the approximations

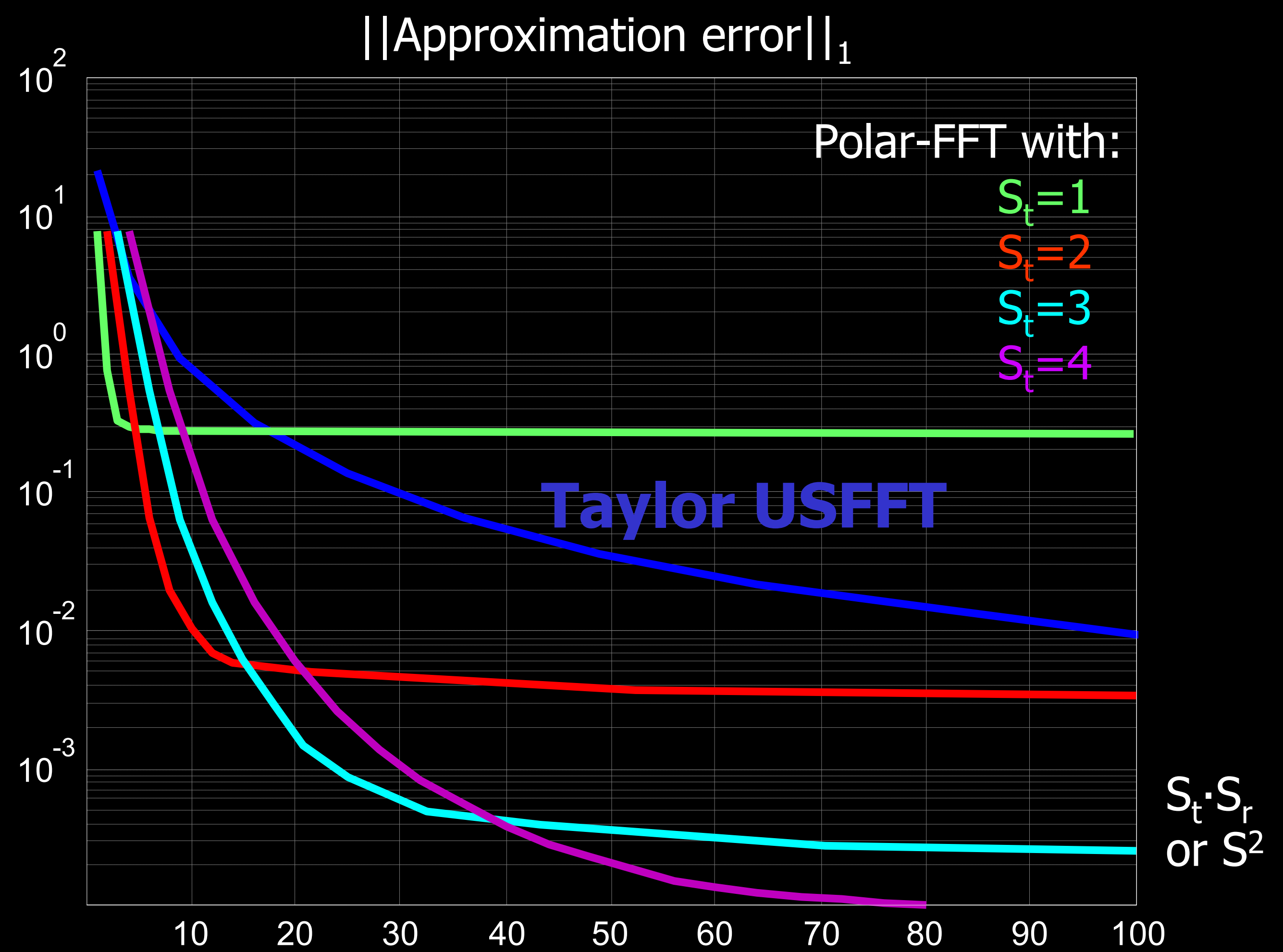
Y – Polar-FFT result as an 2N-by-2N array with rows being the rays and columns being the concentric circles.

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 are 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 when comparing the best USFFT method and the Polar-FFT.
- Further improvement should be achieved for Taylor interpolation in the second stage.

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

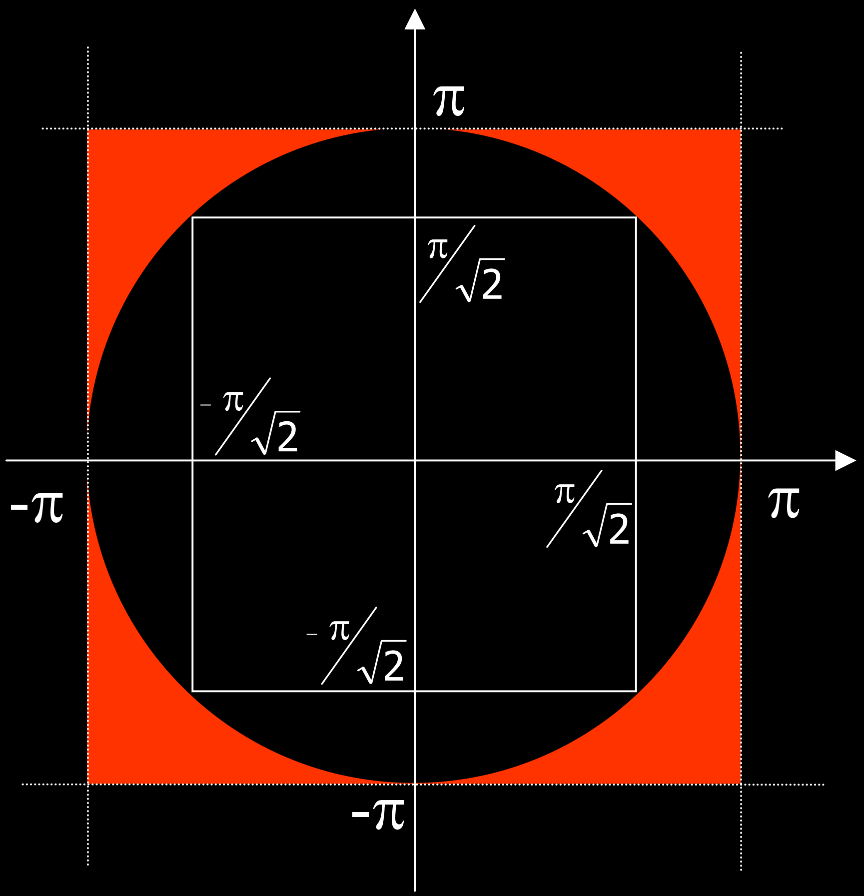
- An input signal of N-by-N is transformed to an array or 2N-by-2N.
- For N=16, **T** size is 1024-by-256, and $\kappa \approx 60{,}000$ (bad for inversion).
- Adding the assumption that the Frequency corners should be zeroed, we obtain

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

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

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

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

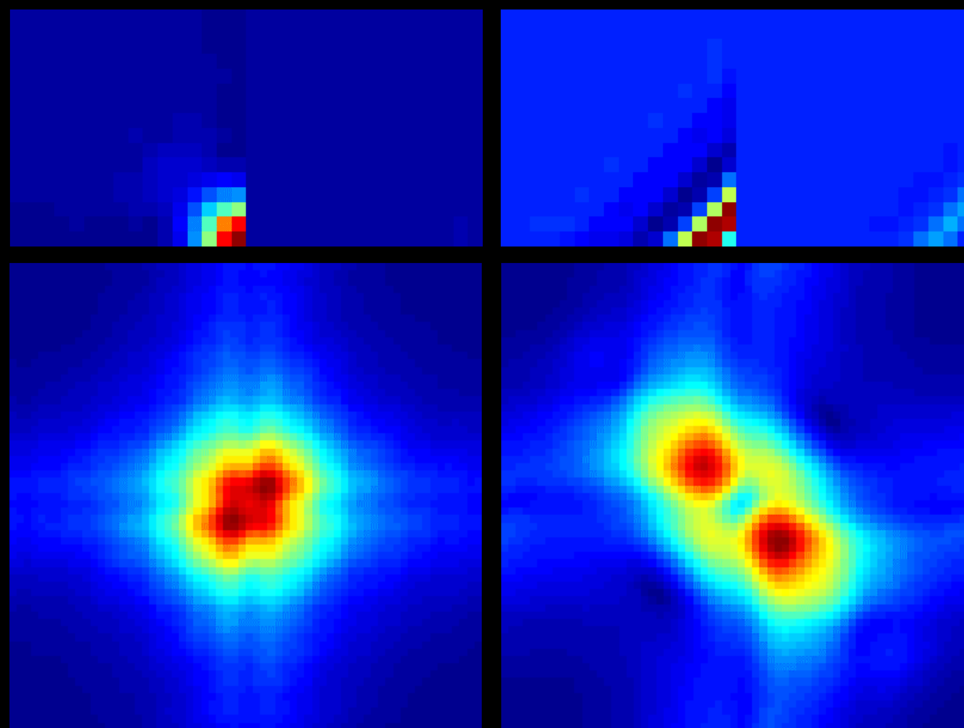
$$\text{Worst relative error : } \left\{\underline{x}_{\text{rworst}}, e^{2}_{\text{rworst}}\right\} = \underset{\underline{x}}{\mathrm{Arg/Min}} \frac{\left\|\left(\mathbf{A}_{\text{Polar-FFT}} - \mathbf{T}_{\text{Polar}}\right)\underline{x}\right\|_2^2}{\left\|\mathbf{T}_{\text{Polar}}\underline{x}\right\|_2^2}$$

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)



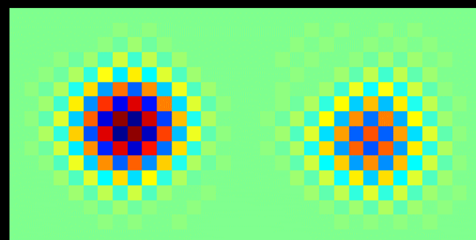
Polar-FFT
worst signal (abs. Value) $\lambda=0.0319$

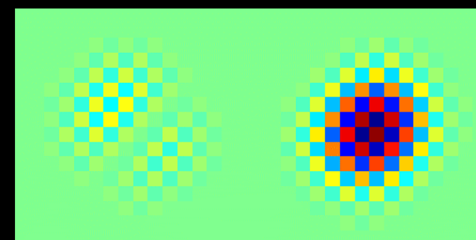
The worst case signal in the freq. Domain (abs. and shifted)

Relative Worst Signal - Results

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

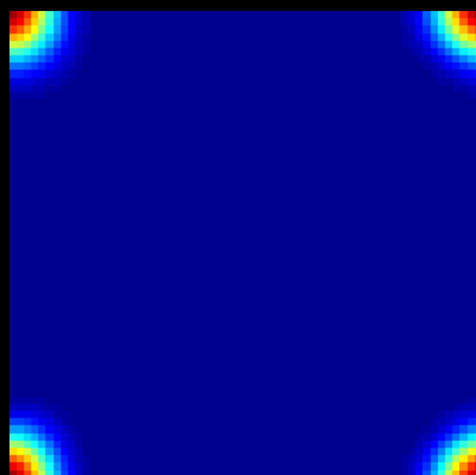
USFFT worst signal (abs. Value) $\lambda=0.0613$

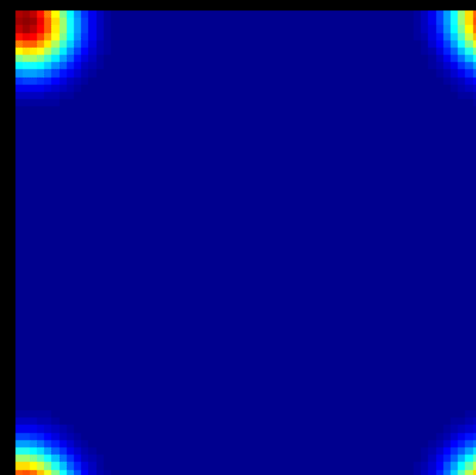




Polar-FFT worst signal (abs. Value) $\lambda=0.0023$

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

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

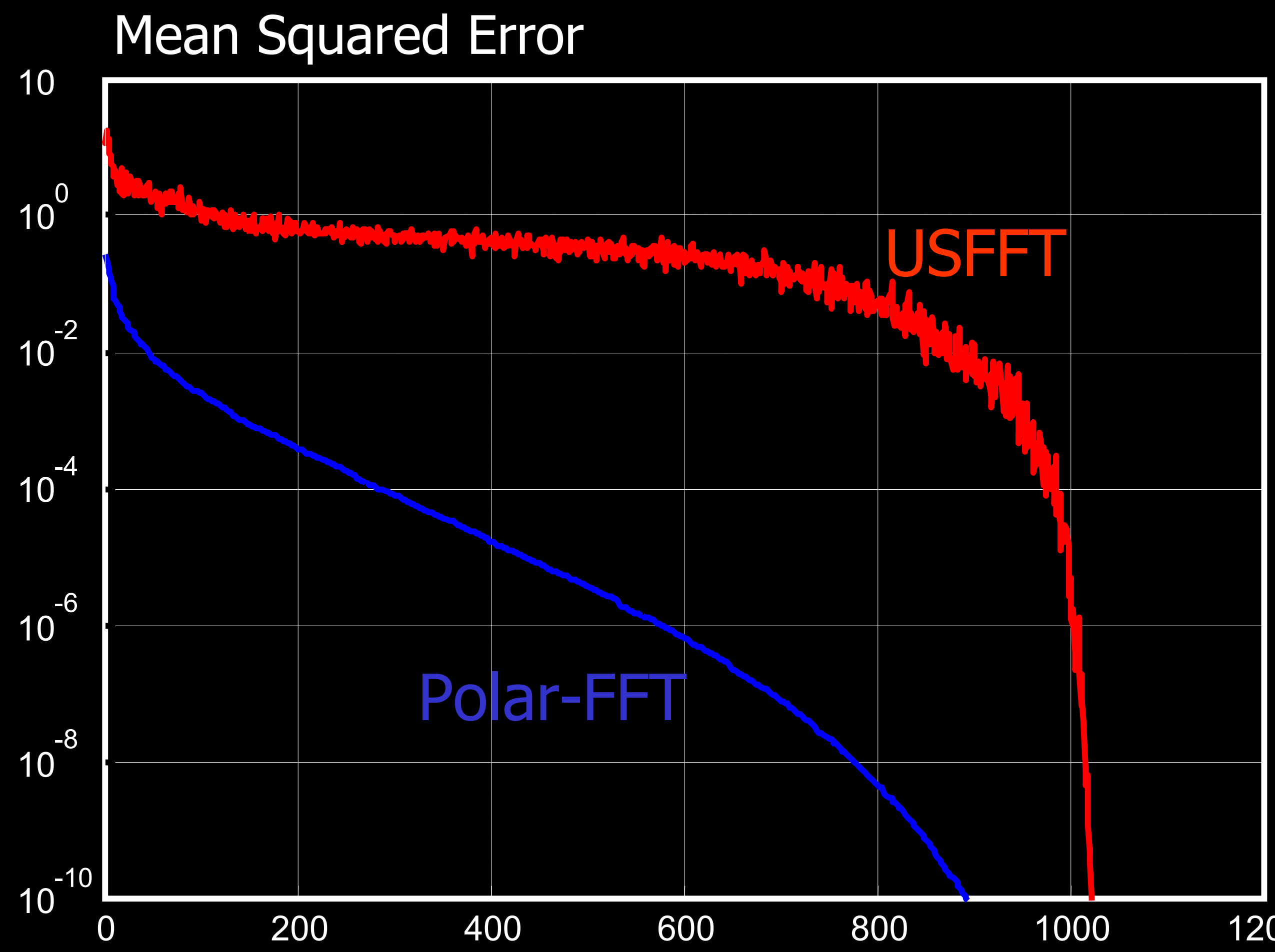
  and obtained $\left\{\lambda_k, \underline{x}_k\right\}_{k=1}^{N^2}$ (λ_k in ascending order).

- 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



[N=32]

Eigen-space of the Polar-FFT

Agenda

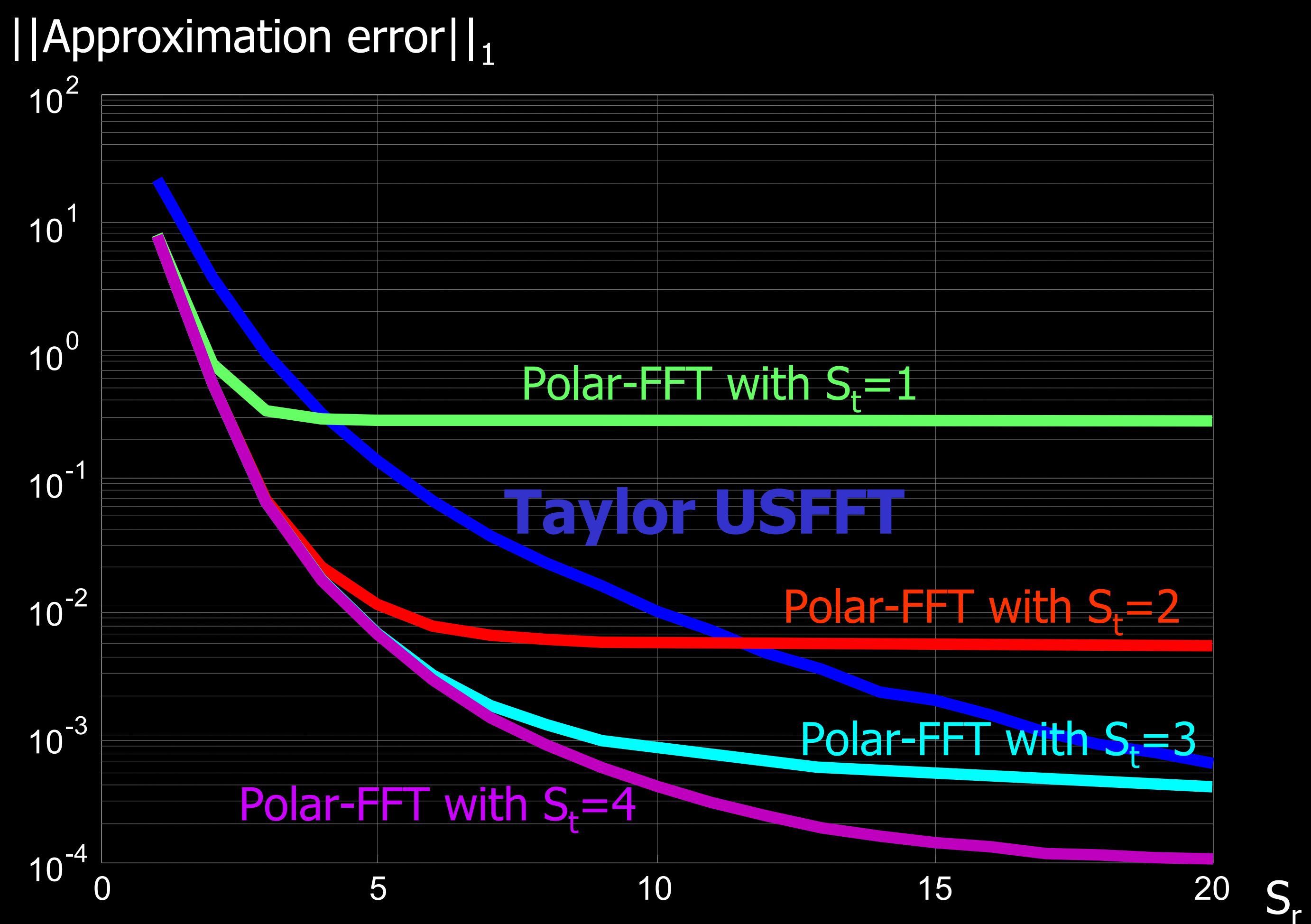
1. Thinking Polar – Continuum
2. Thinking Polar – Discrete
3. Current State-Of-The-Art
4. Our Approach - General
5. The Pseudo-Polar Fast Transform
6. From Pseudo-Polar to Polar
7. Algorithm Analysis
8. Open questions & Future work

8. Open questions & Future work

- ❑ Polar Inverse FFT - Choice of pre-conditioning.
- ❑ Efficient implementation of the inverse.
- ❑ Bounding the approximation error theoretically.
- ❑ Further improving the interpolation stages.
- ❑ Solve analytically for the worst signals for $N\to\infty$.
- ❑ Eeigenvalue-analysis for large N using the QR/QZ.
- ❑ Error analysis for families of signals (e.g. smooth).
- ❑ Applications (rotation, registration, Tomography, Transforms, and more) – Theory and Experiments.

Beyond this slides are the appendix or redundant slides

Error for Specific Signal

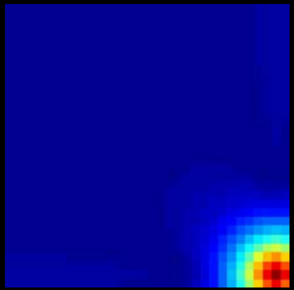


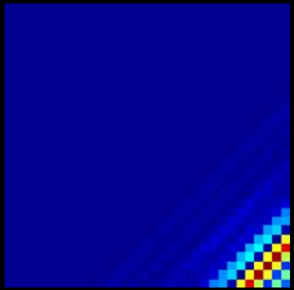
- Input is random 32-by-32 array,
- USFFT method uses one parameter whereas there are two for the up-sampling in the Polar-FFT.
- Thumb rule: $S_r \cdot S_t = S^2$.

Worst Signal - Results

$$N=32 \rightarrow \mathbf{T} \in C^{4096 \times 1024},\ S=S_r=S_t=4$$

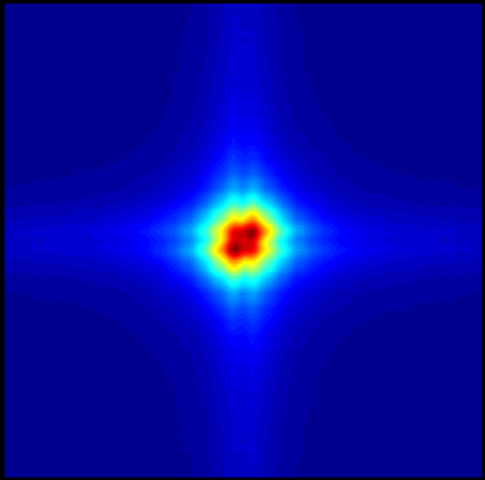
USFFT
worst signal (abs. Value) λ=30.049

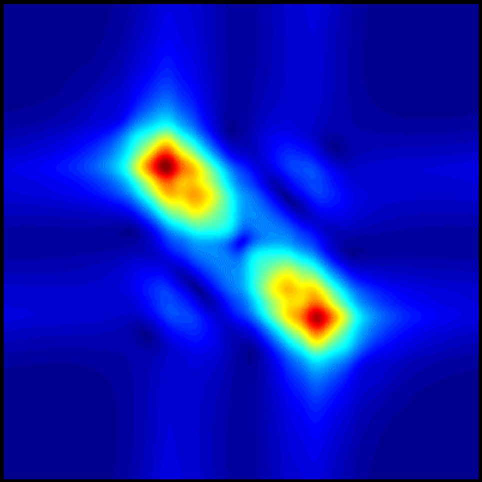




Polar-FFT
worst signal (abs. Value) λ=0.28

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



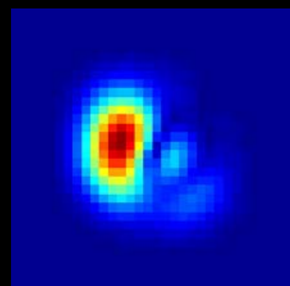


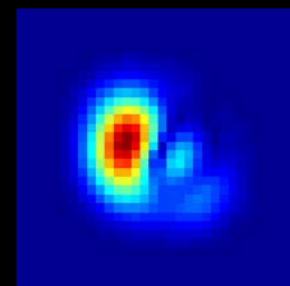
The worst case signal in the freq. Domain (abs. and shifted)

Relative Worst Signal - Results

Same parameters: $N=32 \rightarrow \mathbf{T} \in \mathcal{C}^{4096 \times 1024}$, $S=S_r=S_t=4$

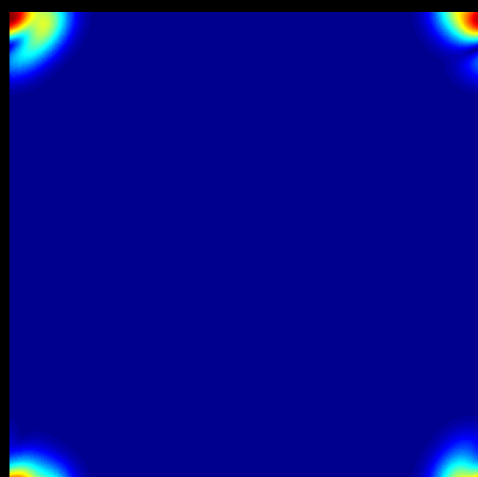
USFFT
worst signal (abs. Value) λ=97.34

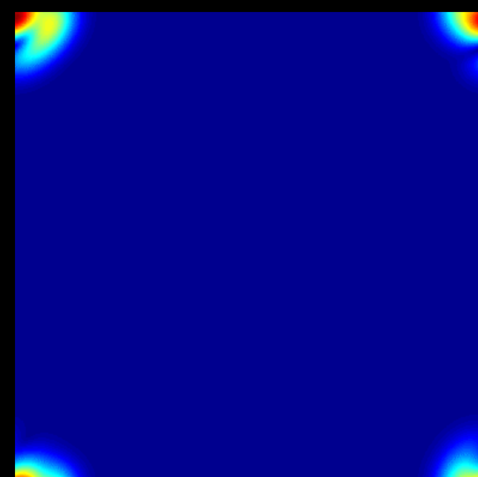




Polar-FFT
worst signal (abs. Value) λ=3.19

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





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