Finite Element-Wavelet Hybrid Algorithm for Atmospheric Tomography

Misha Yudytskiy

March 26, 2014

Edinburgh, Scotland

This research was partially supported by the Austrian Science Fund (FWF): W1214-N15, project DK8.

Joint work with

Misha Yudytskiy

Tapio Helin

Ronny Ramlau

Finite element-wavelet hybrid algorithm for atmospheric tomography,
in Journal of Optical Society of America A 31(3), 2014.

Wavelet-methods in multi-conjugate adaptive optics,
in Inverse problems 29(8), 2013.

Outline

- Current algorithms for atmospheric tomography
- FEWHA for atmospheric tomography
- FEWHA: speed and quality

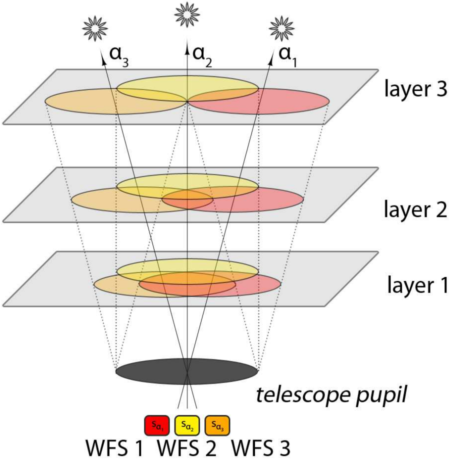
Atmospheric tomography

- Use several guidestars (LGS & NGS)
- Goal: quality in the field of view

◦ Laser Tomography AO (LTAO)

◦ Multi Object AO (MOAO)

◦ Multi Conjugate AO (MCAO)

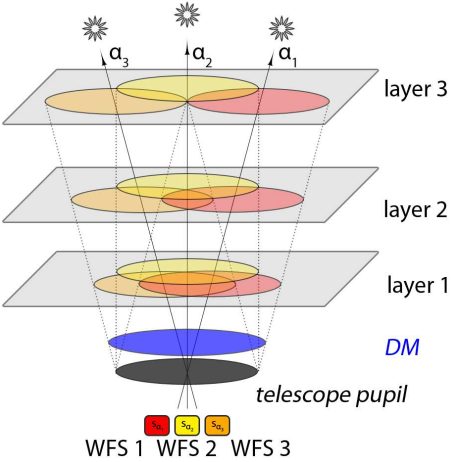


Atmospheric tomography: WFS measurements → layers

Atmospheric tomography

- Use several guidestars (LGS & NGS)
- Goal: quality in the field of view

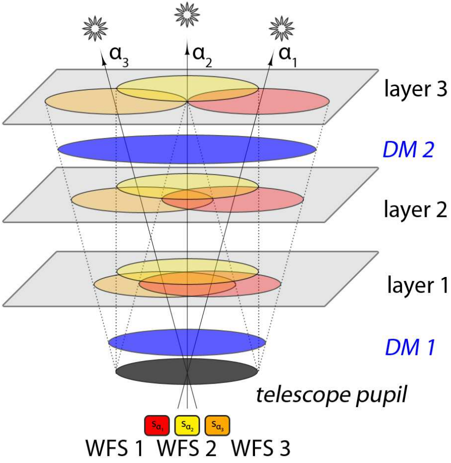
- **Laser Tomography AO (LTAO)**
- Multi Object AO (MOAO)
- Multi Conjugate AO (MCAO)



Atmospheric tomography: WFS measurements → layers

Atmospheric tomography

- Use several guidestars (LGS & NGS)
- Goal: quality in the field of view



- ○ Laser Tomography AO (LTAO)
- ○ Multi Object AO (MOAO)
- ○ **Multi Conjugate AO (MCAO)**

Atmospheric tomography: WFS measurements → layers

Atmospheric tomography

- Use several guidestars (LGS & NGS)
- Goal: quality in the field of view

○ Laser Tomography AO (LTAO)

○ Multi Object AO (MOAO)

○ **Multi Conjugate AO (MCAO)**

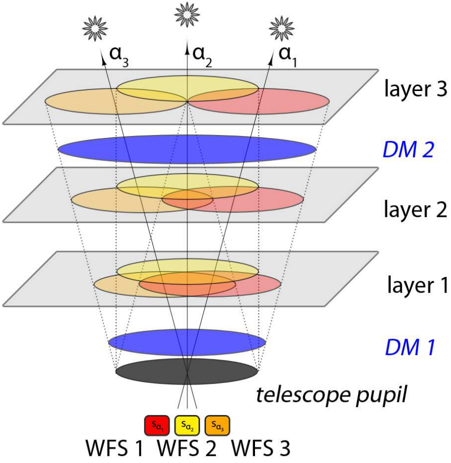
$$A\phi = s$$

↗ ↑ ↖

atmospheric tomography operator

turbulence layers

noisy measurements



Atmospheric tomography: WFS measurements → layers

Standard approach: MVM

Minimum variance turbulence profile estimate:

$$(A^* \overset{\substack{\text{inverse noise covariance}\\ \downarrow}}{C_\eta^{-1}} \underset{\substack{\nearrow\\ \text{atmospheric tomography}}}{A} + \overset{\substack{\text{inverse turbulence covariance}\\ \downarrow}}{C_\phi^{-1}})\underset{\substack{\uparrow\\ \text{turbulence layers}}}{\phi} = A^* C_\eta^{-1} \underset{\substack{\nwarrow\\ \text{noisy measurements}}}{s}$$

Standard approach: MVM

Minimum variance turbulence profile estimate:

$$\left(A^* \overset{\substack{\text{inverse noise covariance}\\ \downarrow}}{C_\eta^{-1}} \underset{\substack{\nearrow\\ \text{atmospheric tomography}}}{A} + \overset{\substack{\text{inverse turbulence covariance}\\ \downarrow}}{C_\phi^{-1}}\right) \underset{\substack{\uparrow\\ \text{turbulence layers}}}{\phi} = A^* C_\eta^{-1} \underset{\substack{\nwarrow\\ \text{noisy measurements}}}{s}$$

Standard approach - MVM:

1. Compute $\mathbf{R} = P(A^* C_\eta^{-1} A + C_\phi^{-1})^{-1} A^* C_\eta^{-1}$ *(off-line)* $\mathcal{O}(n^3)$
2. Multiply $a = \mathbf{R}s$ *(on-line)* $\mathcal{O}(n^2)$

Standard approach: MVM

Minimum variance turbulence profile estimate:

$$(A^* \overset{\substack{\text{inverse noise covariance}\\ \downarrow}}{C_\eta^{-1}} \underset{\substack{\nearrow\\ \text{atmospheric tomography}}}{A} + \overset{\substack{\text{inverse turbulence covariance}\\ \downarrow}}{C_\phi^{-1}})\underset{\substack{\uparrow\\ \text{turbulence layers}}}{\phi} = A^* C_\eta^{-1} \underset{\substack{\nwarrow\\ \text{noisy measurements}}}{s}$$

Standard approach - MVM:

1. Compute $\mathbf{R} = P(A^* C_\eta^{-1} A + C_\phi^{-1})^{-1} A^* C_\eta^{-1}$ *(off-line)* $\mathcal{O}(n^3)$
2. Multiply $a = \mathbf{R}s$ *(on-line)* $\mathcal{O}(n^2)$

Dimensions of $\mathbf{R}$ for E-ELT:

LTAO/MOAO: 99900×5402
MCAO: 66612×9296

⟶ **Very high computational costs!** (even with parallelization)

Alternative approach: iterative methods

Solve:

$$\underbrace{(A^* C_\eta^{-1} A + C_\phi^{-1})}_{M} \phi = A^* C_\eta^{-1} s$$

iteratively: conjugate gradient (CG) method. No matrix inversion!

Alternative approach: iterative methods

Solve:

$$\underbrace{\left(A^* C_\eta^{-1} A + C_\phi^{-1}\right)}_{M} \phi = A^* C_\eta^{-1} s$$

iteratively: conjugate gradient (CG) method. No matrix inversion!

$$\boxed{\text{cost of CG} \approx \text{cost of applying } M \cdot \# \text{ of iterations}}$$

cost of applying M

- discretization
- representation
- parallelization

\# of iterations

↘ warm restart

↘ preconditioning

Alternative approach: iterative methods

Solve:

$$\underbrace{(A^* C_\eta^{-1} A + C_\phi^{-1})}_{M} \phi = A^* C_\eta^{-1} s$$

iteratively: conjugate gradient (CG) method. No matrix inversion!

$$\boxed{\text{cost of CG} \approx \text{cost of applying } M \cdot \# \text{ of iterations}}$$

cost of applying M

- discretization
- representation
- parallelization

\# of iterations

↘ warm restart

↘ preconditioning

Iterative algorithms:

- ○ FD-PCG $\mathcal{O}(n \log n)$
- ○ FrIM $\mathcal{O}(n)$

Alternative: 3-step methods:

- ○ Kaczmarz $\mathcal{O}(n)$
- ○ Gradient-based $\mathcal{O}(n)$
- ○ CG $\mathcal{O}(n)$

Discretization of layers with wavelets

Concept: use **wavelets** to represent **turbulence layers**

Wavelets:

- tool to represent and analyze signals
- used in JPEG compression

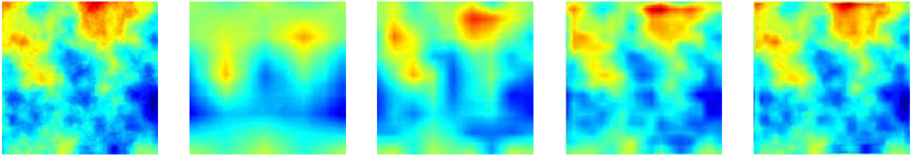
Discretization of layers with wavelets

Concept: use **wavelets** to represent **turbulence layers**

Wavelets:

- tool to represent and analyze signals
- used in JPEG compression

Wavelets decomposition of a layer:



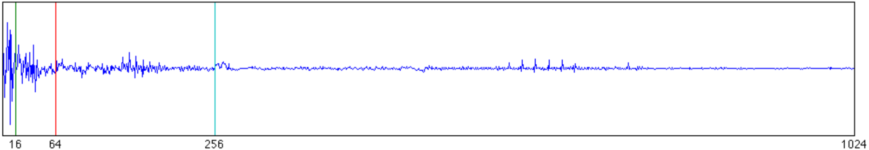
turb. layer
16,384 coeff

2 scales
16 coeff

3 scales
64 coeff

4 scales
256 coeff

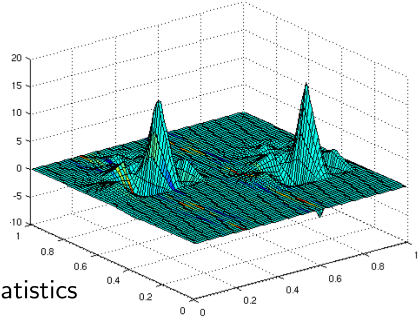
5 scales
1024 coeff



Wavelet properties

Advantages of wavelets:

- good approximative properties
- discrete wavelet transform (DWT)
  - DWT is $\mathcal{O}(n)$, parallelizable!
- useful properties in frequency domain
  - efficient representation of turbulence statistics

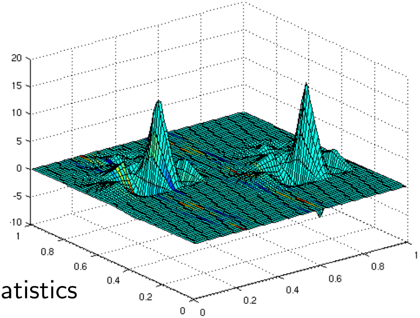


two Daubechies 3 wavelets

Wavelet properties

Advantages of wavelets:

- good approximative properties
- discrete wavelet transform (DWT)
  - DWT is $\mathcal{O}(n)$, parallelizable!
- useful properties in frequency domain
  - efficient representation of turbulence statistics



two Daubechies 3 wavelets

Von Karman:

$$C_\phi = c\mathcal{F}^{-1}M\mathcal{F} \qquad C_\phi \simeq cW^{-1}DW$$

$$(Mf)(\xi) = |\xi|^{-11/3}f(\xi) \qquad D = \mathrm{diag}(\ldots, 2^{-\frac{11}{3}j}, \ldots)$$

j... wav scale

Finite Element-Wavelet Hybrid Algorithm (FEWHA)

Dual domain discretization:

$$(W A^* C_\eta^{-1} A W^{-1} + \alpha D^{-1}) c = W A^* C_\eta^{-1} s$$

discrete wavelet transform $\mathcal{O}(n)$

atmospheric tomography bilinear basis (sparse)

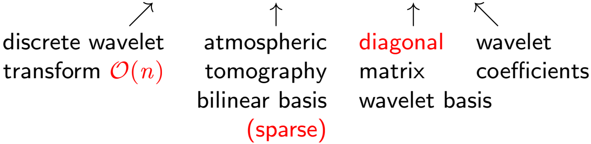
diagonal matrix wavelet basis

wavelet coefficients

Finite Element-Wavelet Hybrid Algorithm (FEWHA)

Dual domain discretization:

$$(W A^* C_\eta^{-1} A W^{-1} + \alpha D^{-1}) c = W A^* C_\eta^{-1} s$$



discrete wavelet transform $\mathcal{O}(n)$

atmospheric tomography bilinear basis (sparse)

diagonal matrix wavelet basis

wavelet coefficients

operator M

- sparse in bilinear / wavelet domains
- matrix-free representation
- parallelizable (shared memory!)

Finite Element-Wavelet Hybrid Algorithm (FEWHA)

Dual domain discretization:

$$(W A^* C_\eta^{-1} A W^{-1} + \alpha D^{-1}) c = W A^* C_\eta^{-1} s$$

↗ discrete wavelet transform $\mathcal{O}(n)$

↑ atmospheric tomography bilinear basis (sparse)

↑ diagonal matrix wavelet basis

↖ wavelet coefficients

operator M

- sparse in bilinear / wavelet domains
- matrix-free representation
- parallelizable (shared memory!)

\# of iterations

↘ warm restart

↘ frequency-dependent preconditioner

↘ ground layer multi-scale (GLMS)

Finite Element-Wavelet Hybrid Algorithm (FEWHA)

Dual domain discretization:

$$(W A^* C_\eta^{-1} A W^{-1} + \alpha D^{-1}) c = W A^* C_\eta^{-1} s$$

discrete wavelet transform $\mathcal{O}(n)$

atmospheric tomography bilinear basis (sparse)

diagonal matrix wavelet basis

wavelet coefficients

operator M

- sparse in bilinear / wavelet domains
- matrix-free representation
- parallelizable (shared memory!)

# of iterations

↘ warm restart

↘ frequency-dependent preconditioner

↘ ground layer multi-scale (GLMS)

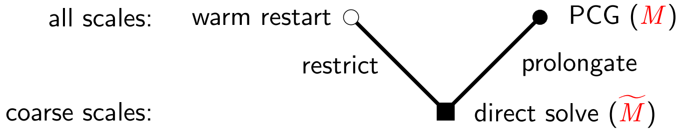
$\longrightarrow$ **FEWHA:** $\mathcal{O}(n)$ **complexity**

✓ **fast convergence**

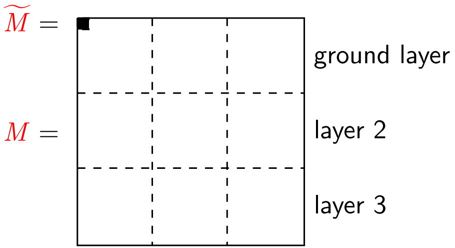
✓ **parallelizable**

Ground layer multi-scale (GLMS) method

Multi-scale method:


all scales:
warm restart
PCG (M)
restrict
prolongate
coarse scales:
direct solve ($\widetilde{M}$)

Coarse scale sub-problem:


$\widetilde{M} =$
$M =$
ground layer
layer 2
layer 3

- $\dim M \approx 150,000 \times 150,000$ $\sim$ 7 scales at 9 layers
- $\dim \widetilde{M} = 256 \times 256$ $\sim$ 4 scales of ground layer

Properties of FEWHA

Modularity:

- Systems: LTAO / MOAO / MCAO / (SCAO too)
- Sensors: Shack-Hartmann - LGS / NGS / tip-tilt (low order NGS)
- Control: closed loop (pseudo-open loop control) / open loop

Properties of FEWHA

Modularity:

- Systems: LTAO / MOAO / MCAO / (SCAO too)
- Sensors: Shack-Hartmann - LGS / NGS / tip-tilt (low order NGS)
- Control: closed loop (pseudo-open loop control) / open loop

Easy to update (at runtime):

- star position
- photon flux, C_n^2
- DM / WFS alignment

- ○ no matrix inversion
- ○ swap a few numbers in memory

Properties of FEWHA

Modularity:

- Systems: LTAO / MOAO / MCAO / (SCAO too)
- Sensors: Shack-Hartmann - LGS / NGS / tip-tilt (low order NGS)
- Control: closed loop (pseudo-open loop control) / open loop

Easy to update (at runtime):

- star position
- photon flux, C_n^2
- DM / WFS alignment

- no matrix inversion
- swap a few numbers in memory

Parameters:

- not parameter-free

- but few and easy to tune

Quality: Summary

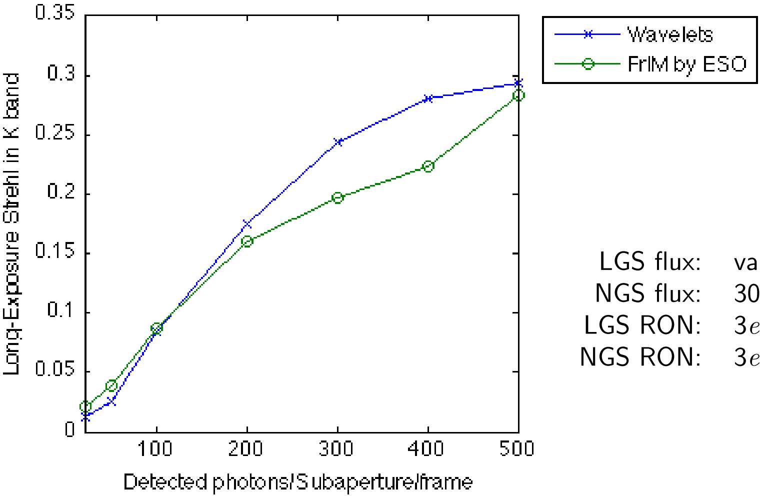
Numerical simulation:

- ESO's OCTOPUS
- 1 second (500 time-steps)
- 9 layer "ESO standard" atmosphere
- 42 m E-ELT configuration

	LTAO	MCAO
Star asterism	6 LGS @ 7.5' diam 3 NGS @ 10' diam	6 LGS @ 2' diam 3 TTS @ 2.67' diam
SH-WFS	84×84 LGS 84×84 NGS	84×84 LGS 2×2, 1×1 TTS
DM(s)	1	3
Probe-stars	1 (zenith)	25 over FoV
Benchmark	FrIM by ESO	MVM by ESO
FEWHA	9 layers, projection	9 layers, fitting
		3 layers on DMs

- LGS: cone effect / tilt-tilt indetermination / spot elongation

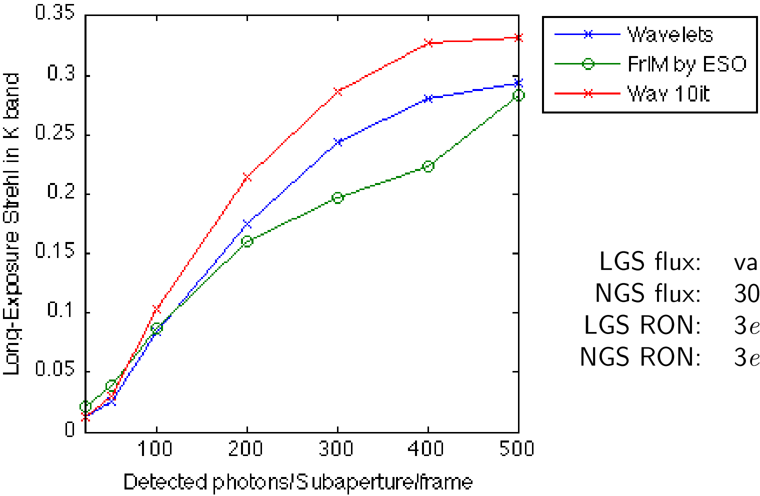
Quality: LTAO spot elongated



LGS flux:	variable
NGS flux:	300
LGS RON:	$3e^-$
NGS RON:	$3e^-$

- Algorithm: 4 it.

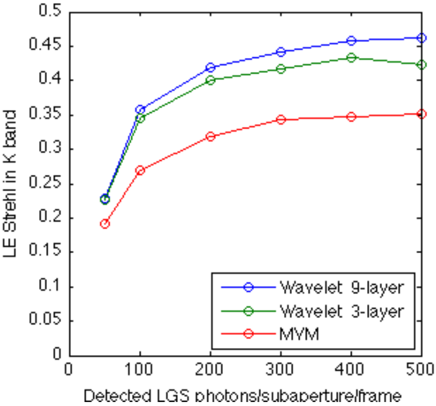
Quality: LTAO spot elongated



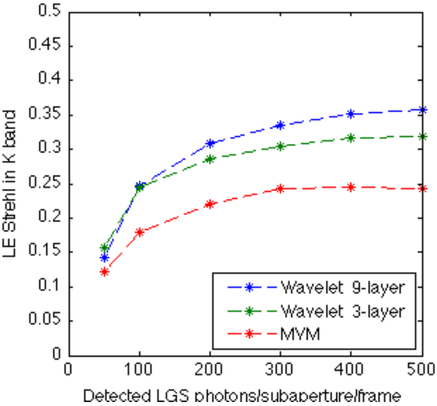
LGS flux:	variable
NGS flux:	300
LGS RON:	$3e^-$
NGS RON:	$3e^-$

- Algorithm: 4 it. / 10 it.

Quality: MCAO spot elongated



On-axis



Field average

- Algorithm: 4 it.

LGS flux:	variable	LGS RON:	$3e^-$
TTS flux:	500	NGS RON:	$5e^-$

Computational complexity

	MVM	FEWHA
Setup	$\mathcal{O}(n^3)^*$ (matrix inversion)	$\mathcal{O}(n)$
Runtime	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Memory	$\mathcal{O}(n^2)$ (matrix storage)	$\mathcal{O}(n)$ (matrix-free)
Parallelization	very high	high (shared memory)
Pipelining	very high	moderate

---

* an iterative method can be used for MVM setup

RTC: Summary

Computing system configurations:

1. Intel Xeon X5650 @ 2.66GHz
   - 12 cores (dual hexacore)
   - Q1 2010, €5000
2. Intel Xeon E5-1650 @ 3.20GHz
   - 6 cores
   - Q1 2012, €1800

System	LTAO[1]	MCAO[2]
Allotted computing time	1 ms	1 ms
Computing time MVM	104 ms	72 ms
Computing time FEWHA	6.2 ms	1.5 ms
Speed-up factor	17	48

Memory MVM	2.2 Gb	2.3 Gb
Memory FEWHA	10.3 Mb	3.2 Mb
	(4 it)	(4 it)

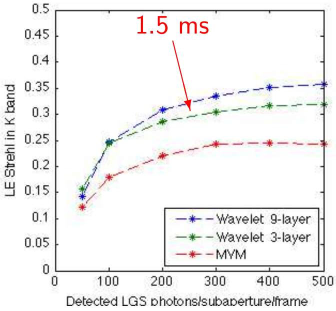
RTC: Summary

Computing system configurations:

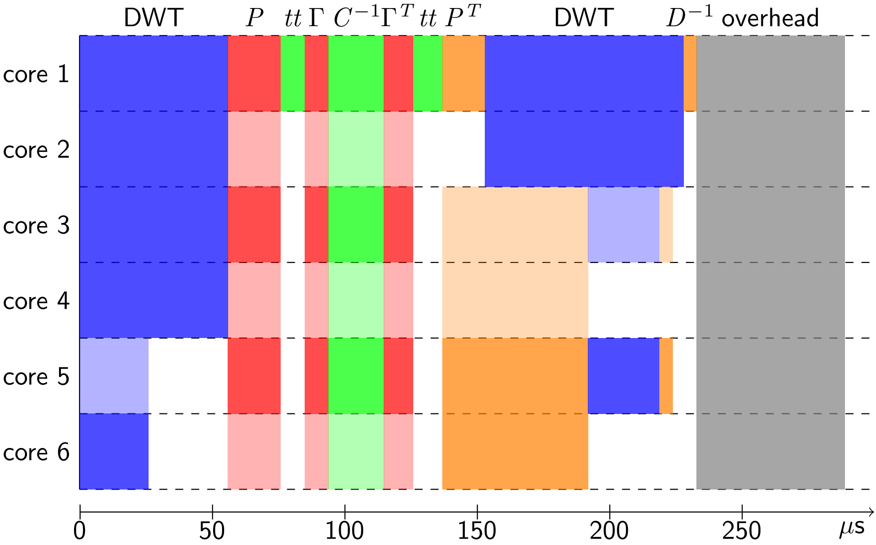
1. Intel Xeon X5650 @ 2.66GHz
   - 12 cores (dual hexacore)
   - Q1 2010, €5000
2. Intel Xeon E5-1650 @ 3.20GHz
   - 6 cores
   - Q1 2012, €1800

System	LTAO[1]	MCAO[2]
Allotted computing time	1 ms	1 ms
time MVM	104 ms	72 ms
time FEWHA	6.2 ms	1.5 ms
ctor	17	48

/M	2.2 Gb	2.3 Gb
WHA	10.3 Mb (4 it)	3.2 Mb (4 it)


1.5 ms

Parallelization of MCAO operator M - tetris



Summary and Outlook

Summary

- FEWHA: light-weight versatile method for MCAO/LTAO/MOAO
- Superior quality
- Fast and compact:
  - MCAO: 1.5 ms on off-the-shelf hardware

Outlook

- Study of algorithm behavior: more simulations and optical bench tests
- RTC on GPUs / CPU with more cores
- Algorithm development → multiscale methods, predictive schemes

Summary and Outlook

Summary

- FEWHA: light-weight versatile method for MCAO/LTAO/MOAO
- Superior quality
- Fast and compact:
  - MCAO: 1.5 ms on off-the-shelf hardware

Outlook

- Study of algorithm behavior: more simulations and optical bench tests
- RTC on GPUs / CPU with more cores
- Algorithm development → multiscale methods, predictive schemes

Thank you for your attention!