

Searching for the optimal algorithmic and implementation configuration in 3D robot vision

Luigi Nardi, PhD

Software Performance Optimisation group
@HARSH workshop in conjunction with HPCA/CGO/PPoPP
March 12th 2016

In collaboration with:

B. Bodin, M Z. Zia, J. Mawer, E. Vespa, M. K. Emani, A. Nisbet, G. S. Shenoy, M. F. P. O'Boyle, P. H. J. Kelly, B. Franke, C. Kotselidis, M. Luján, A. J. Davison, G. Riley, N. Topham and S. Furber



The University of Manchester

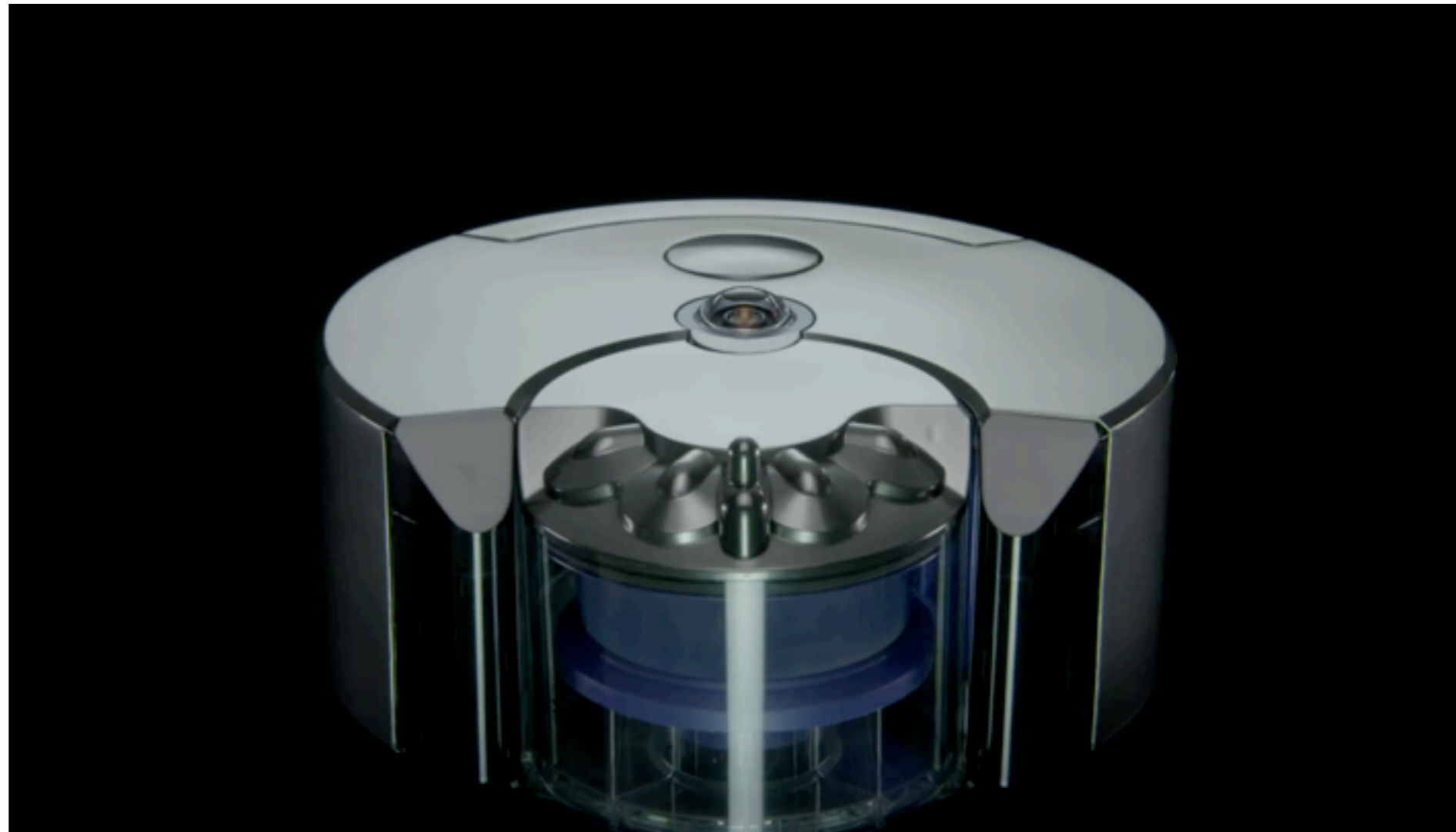
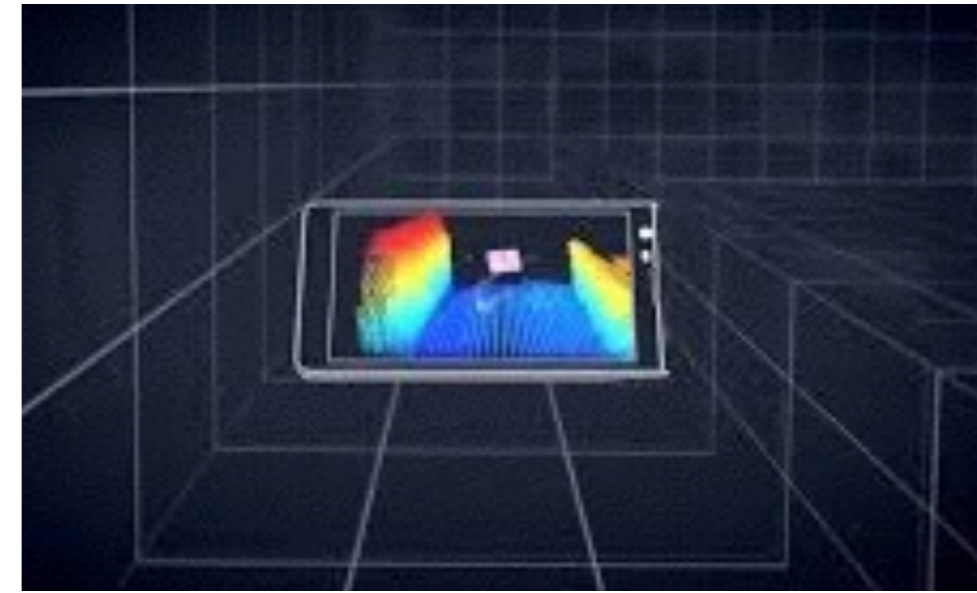


Imperial College
London



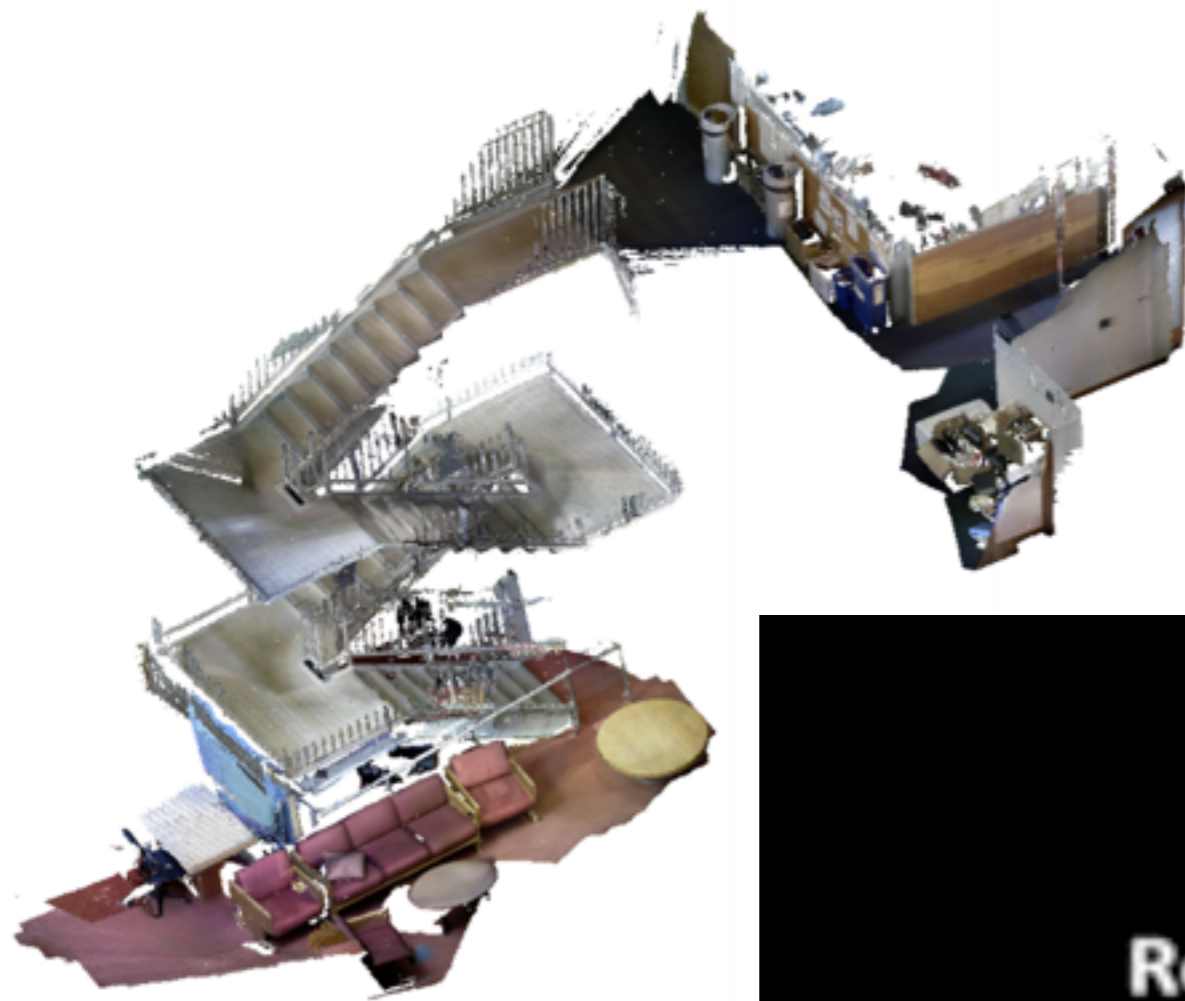
Simultaneous localisation and mapping (SLAM)

Build a coherent world representation and localise the camera in real-time



Video:
[Dyson 360 Eye](#)

Simultaneous localisation and mapping (SLAM)



[Whelan et al. 2012]



SIGGRAPH Talks 2011 KinectFusion: Real-Time Dynamic 3D Surface Reconstruction and Interaction

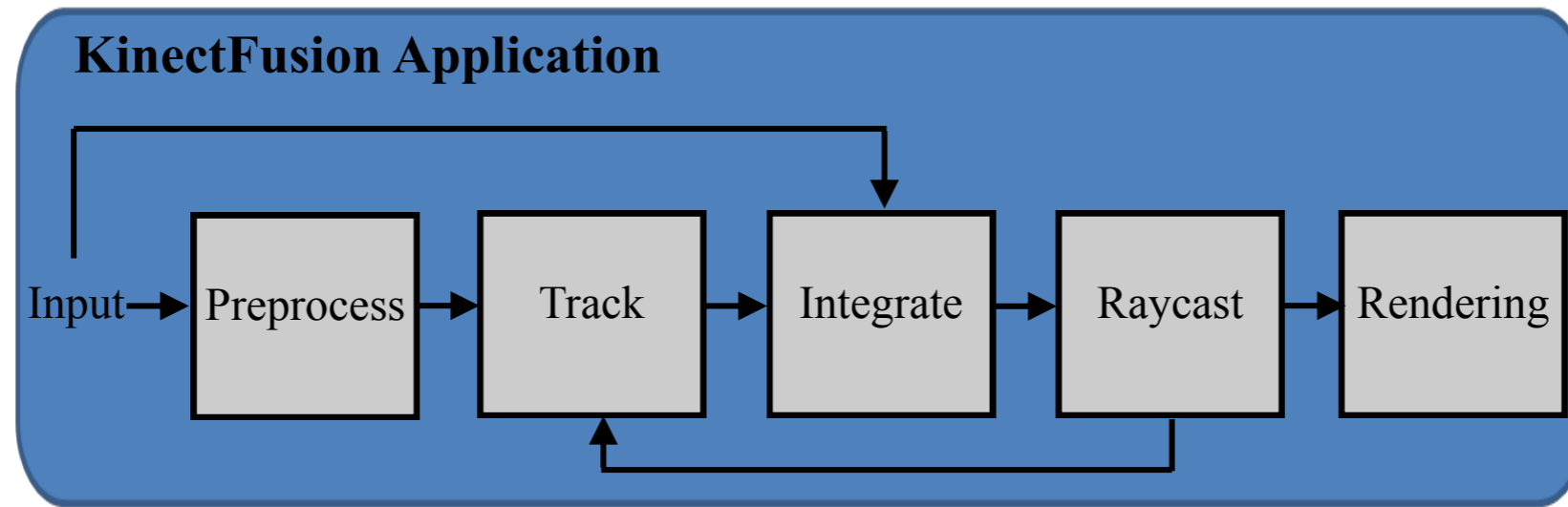
Shahram Izadi ¹, Richard Newcombe ², David Kim ^{1,3}, Otmar Hilliges ¹,
David Molyneaux ^{1,4}, Pushmeet Kohli ¹, Jamie Shotton ¹,
Steve Hodges ¹, Dustin Freeman ⁵, Andrew Davison ², Andrew Fitzgibbon ¹

¹ Microsoft Research Cambridge ² Imperial College London
³ Newcastle University ⁴ Lancaster University
⁵ University of Toronto

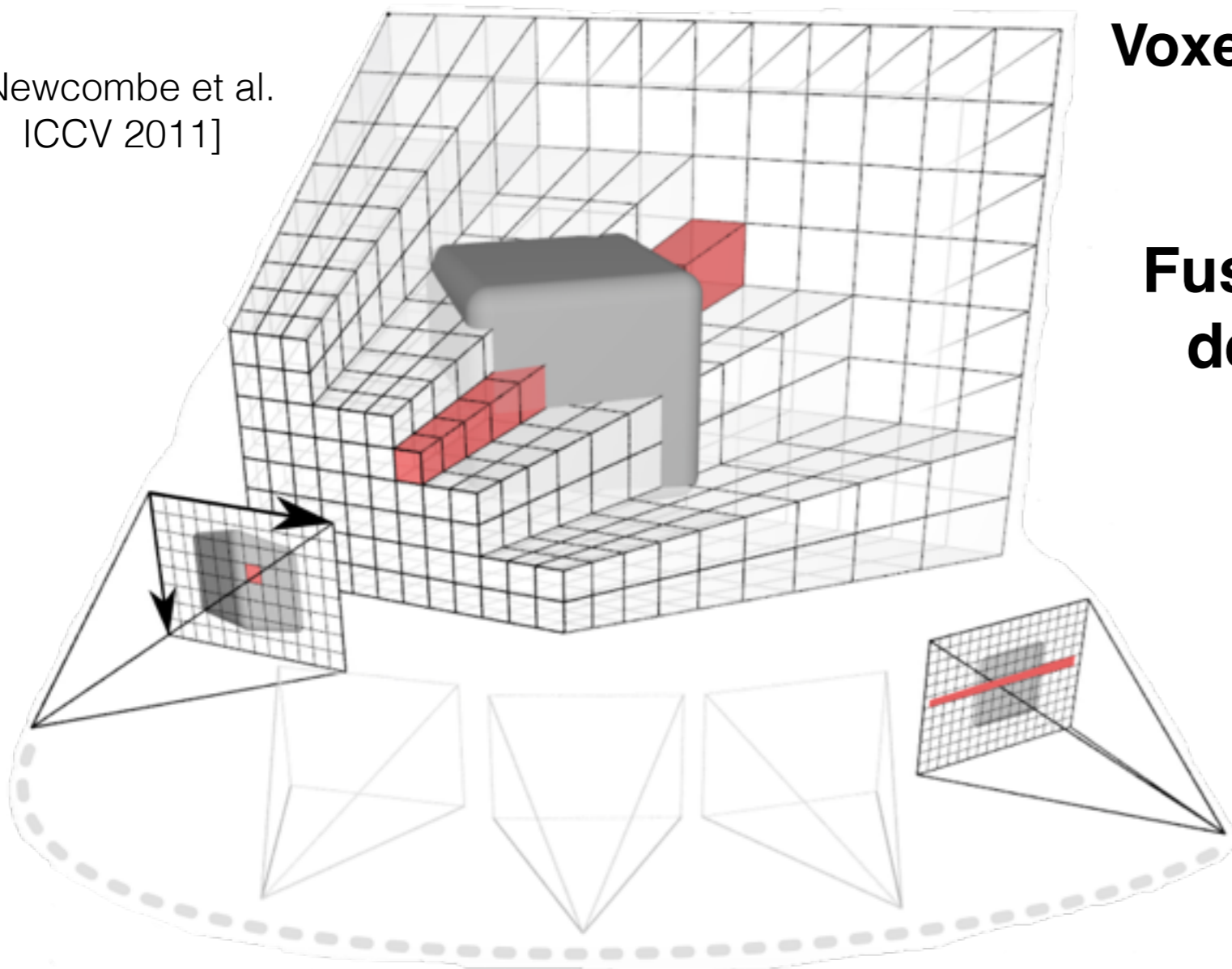
Video:

[[Newcombe et al. ISMAR 2011](#)]

KinectFusion SLAM implementation



[Newcombe et al. ICCV 2011]



Voxel grid represents **3D surfaces**
[Curless and Levoy 1996]

Fuses (or integrate) the stream **depth frames** into a **3D map**

3D surfaces recovered by **raycasting**

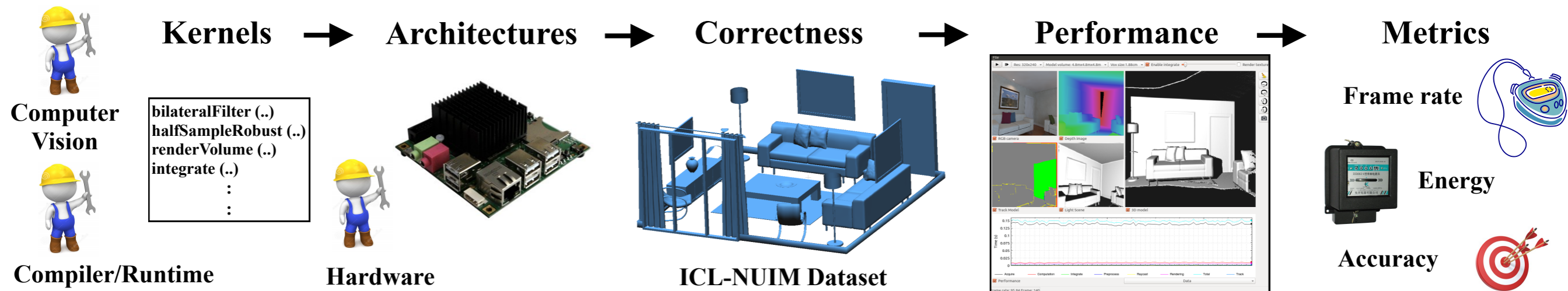
Localisation (or tracking): estimates the camera pose



Three “Performance” metrics

Holistic approach to SLAM “performance”:

SLAMBench



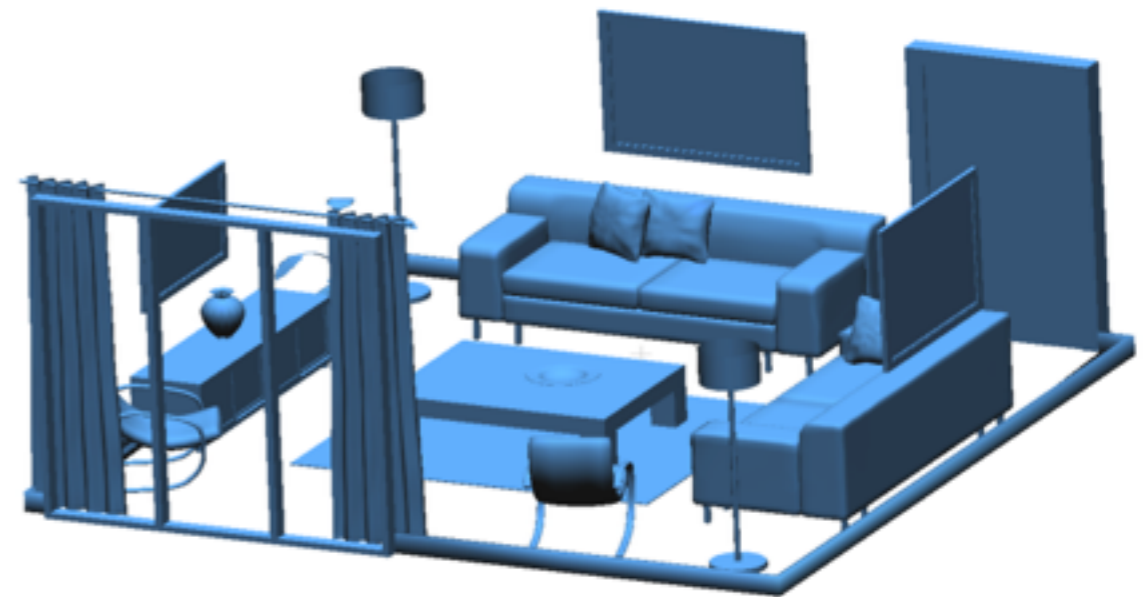
A publicly-available benchmarking framework for quantitative, comparable and validatable experimental research to investigate trade-offs in performance, accuracy and energy consumption of a SLAM system



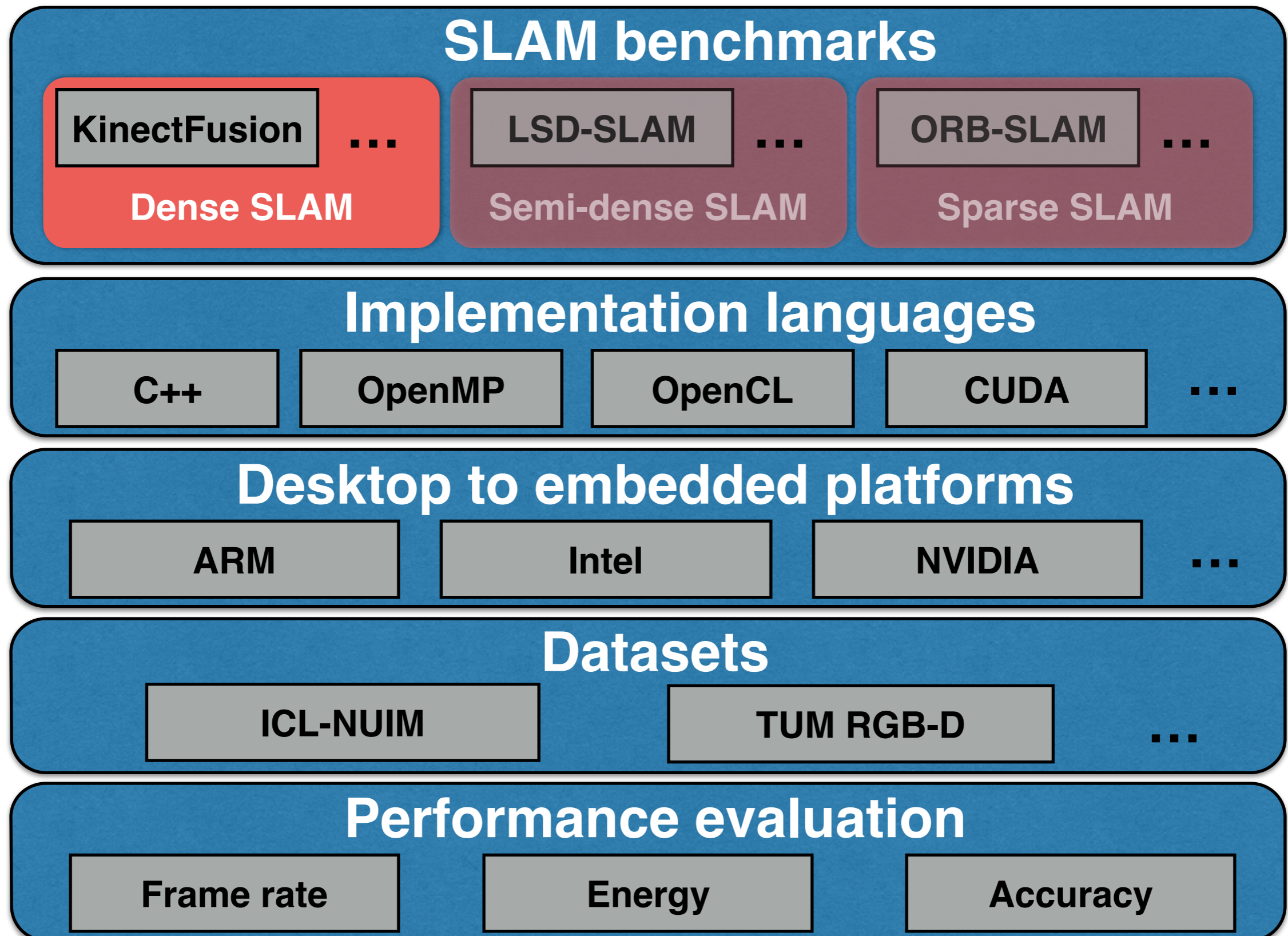
ICL-NUIM dataset



- ICL-NUIM synthetic dataset [Handa et al. 2014]
- 880 RGB-D frames at 30 FPS
- Absolute trajectory error (ATE) based on ground truth



SLAMBench framework



“Performance” on SLAMBench

- Runtime/energy/accuracy measurements
- Accuracy provided via absolute trajectory error (ATE)



Machine	CPU	CPU name	CPU GFLOPS	CPU cores	GPU	GPU name	GPU GFLOPS	TDP Watts
Hardkernel ODROID-XU3	ARM A15 + A7	Exynos 5422	80	4 + 4	ARM	Mali-T628	60 + 30	10

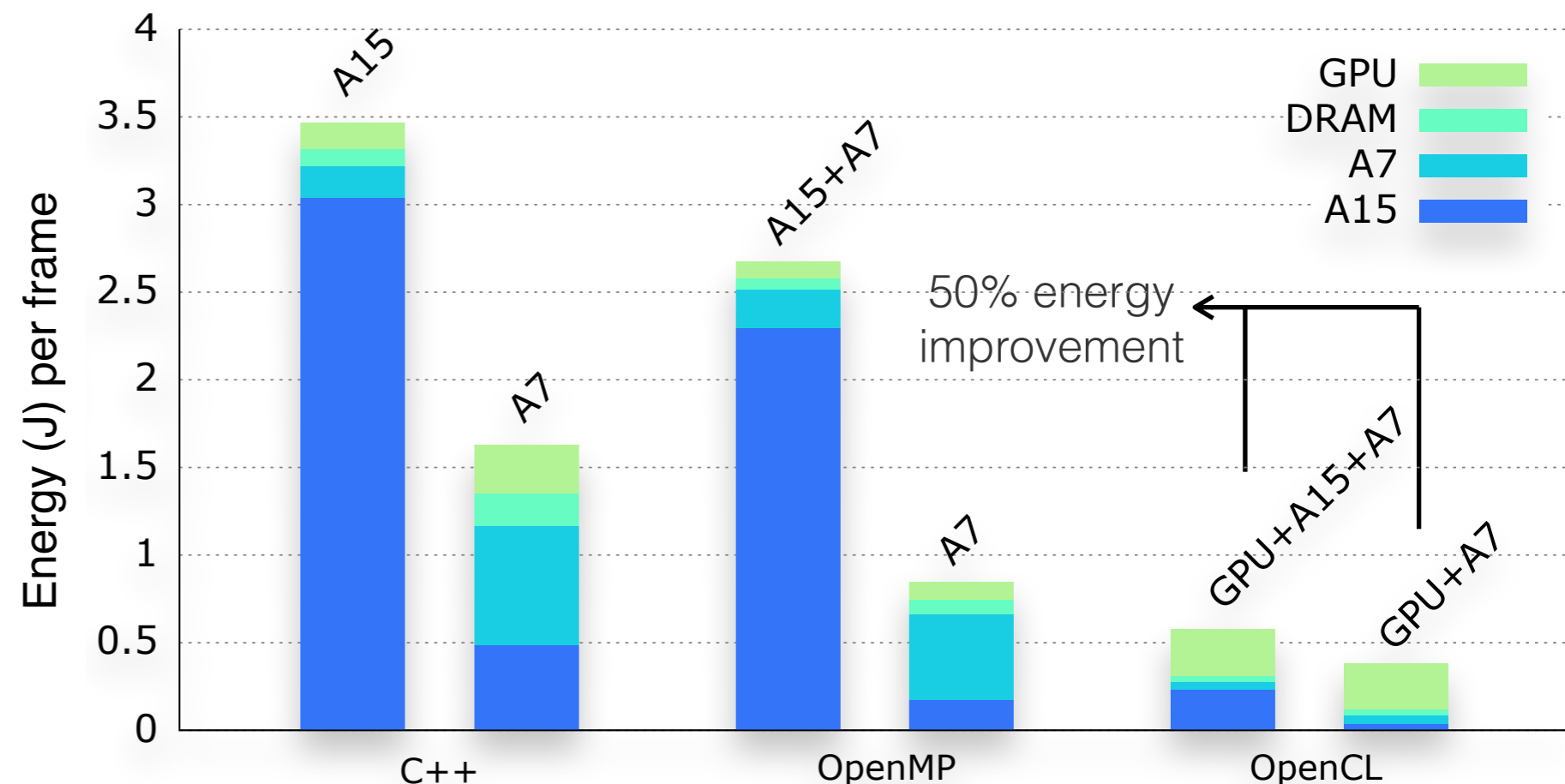
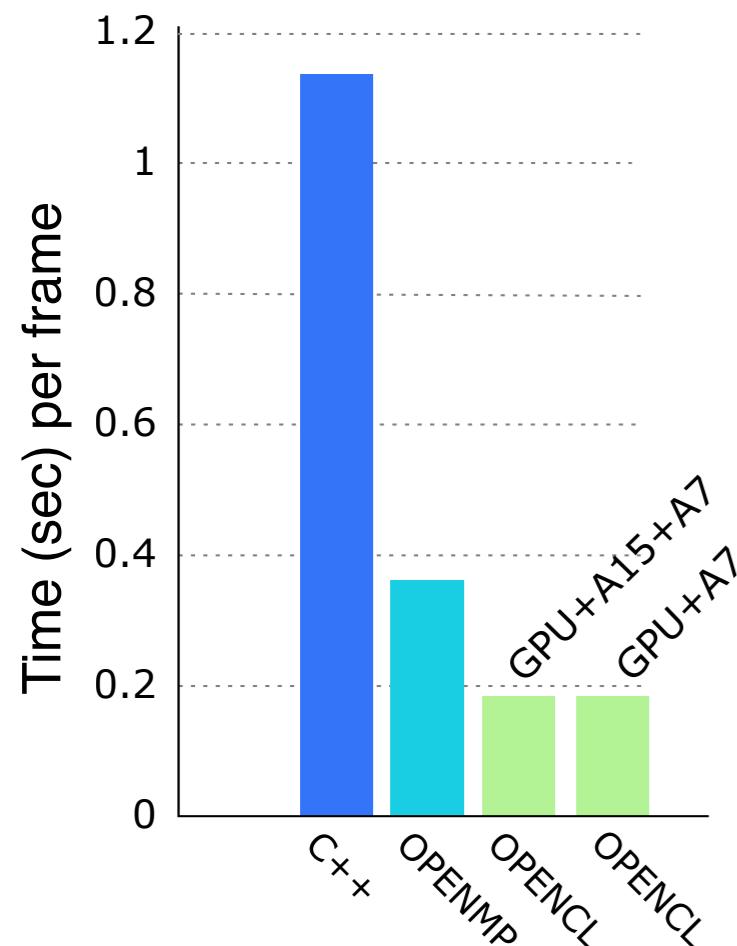
“Performance” on SLAMBench

- Runtime/energy/accuracy measurements
- Accuracy provided via absolute trajectory error (ATE)



ATE in cm	
C++	2.06
OpenMP	2.06
OpenCL	2.01

Machine	CPU	CPU name	CPU GFLOPS	CPU cores	GPU	GPU name	GPU GFLOPS	TDP Watts
Hardkernel ODROID-XU3	ARM A15 + A7	Exynos 5422	80	4 + 4	ARM	Mali-T628	60 + 30	10



What is the optimisation space?

Configuration parameters:

Space 1

1. Algorithmic:
 - Application-specific parameters
 - Minimisation methods
 - Early exit condition values

What is the optimisation space?

Configuration parameters:

Space 1	<ol style="list-style-type: none">1. Algorithmic:<ul style="list-style-type: none">• Application-specific parameters• Minimisation methods• Early exit condition values
Space 2	<ol style="list-style-type: none">2. Compilation:<ul style="list-style-type: none">• opencl-params: -cl-mad-enable, -cl-fast-relaxed-math, etc.• LLVM flags: O1, O2, O3, vectorize-slp-aggressive, etc.• Local work group size: 16/32/64/96/112/128/256• Vectorisation: width (1/2/4/8), direction (x/y)• Thread coarsening: factor (1/2/4/8/16/32), stride (1/2/4/8/16/32), dimension (x/y)

What is the optimisation space?

Configuration parameters:

Space 1	<p>1. Algorithmic:</p> <ul style="list-style-type: none"> • Application-specific parameters • Minimisation methods • Early exit condition values
Space 2	<p>2. Compilation:</p> <ul style="list-style-type: none"> • opencl-params: -cl-mad-enable, -cl-fast-relaxed-math, etc. • LLVM flags: O1, O2, O3, vectorize-slp-aggressive, etc. • Local work group size: 16/32/64/96/112/128/256 • Vectorisation: width (1/2/4/8), direction (x/y) • Thread coarsening: factor (1/2/4/8/16/32), stride (1/2/4/8/16/32), dimension (x/y)
Space 3	<p>3. Architecture:</p> <ul style="list-style-type: none"> • GPU frequency: 177/266/350/420/480/543/600/DVFS • # of active big cores: 0/1/2/3/4 • # of active LITTLE cores: 1/2/3/4

What is the optimisation space?

Configuration parameters:

Co-design space	Space 1	<ol style="list-style-type: none"> Algorithmic: <ul style="list-style-type: none"> Application-specific parameters Minimisation methods Early exit condition values
	Space 2	<ol style="list-style-type: none"> Compilation: <ul style="list-style-type: none"> opencl-params: -cl-mad-enable, -cl-fast-relaxed-math, etc. LLVM flags: O1, O2, O3, vectorize-slp-aggressive, etc. Local work group size: 16/32/64/96/112/128/256 Vectorisation: width (1/2/4/8), direction (x/y) Thread coarsening: factor (1/2/4/8/16/32), stride (1/2/4/8/16/32), dimension (x/y)
	Space 3	<ol style="list-style-type: none"> Architecture: <ul style="list-style-type: none"> GPU frequency: 177/266/350/420/480/543/600/DVFS # of active big cores: 0/1/2/3/4 # of active LITTLE cores: 1/2/3/4

Warning: huge spaces, impossible to run exhaustively

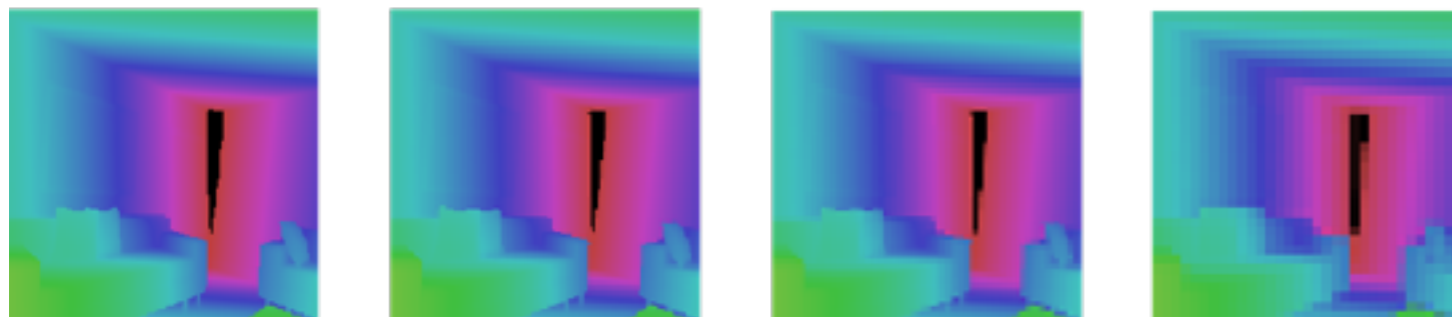
KinectFusion algorithmic features

Features	Ranges
Volume resolution	64x64x64, 128x128x128, 256x256x256, 512x512x512
μ distance	0 .. 0.5
Pyramid level iterations (3 levels)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Image resolution (image ratio)	1, 2, 4, 8
Tracking rate	1, 2, 3, 4, 5
ICP threshold	10^{-6} .. 10^2
Integration rate	1 .. 30

KinectFusion algorithmic features

Features	Ranges
Volume resolution	64x64x64, 128x128x128, 256x256x256, 512x512x512
μ distance	0 .. 0.5
Pyramid level iterations (3 levels)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Image resolution (image ratio)	1, 2, 4, 8
Tracking rate	1, 2, 3, 4, 5
ICP threshold	10^{-6} .. 10^2
Integration rate	1 .. 30

Image resolution (image ratio)



640x480

320x240

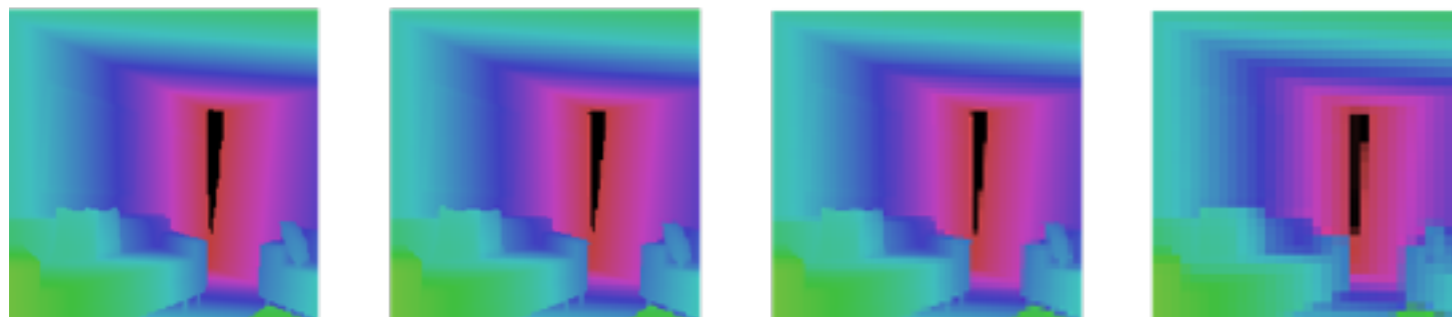
160x120

80x60

KinectFusion algorithmic features

Features	Ranges
Volume resolution	64x64x64, 128x128x128, 256x256x256, 512x512x512
μ distance	0 .. 0.5
Pyramid level iterations (3 levels)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Image resolution (image ratio)	1, 2, 4, 8
Tracking rate	1, 2, 3, 4, 5
ICP threshold	10^{-6} .. 10^2
Integration rate	1 .. 30

Image resolution (image ratio)



640x480

320x240

160x120

80x60

Volume resolution

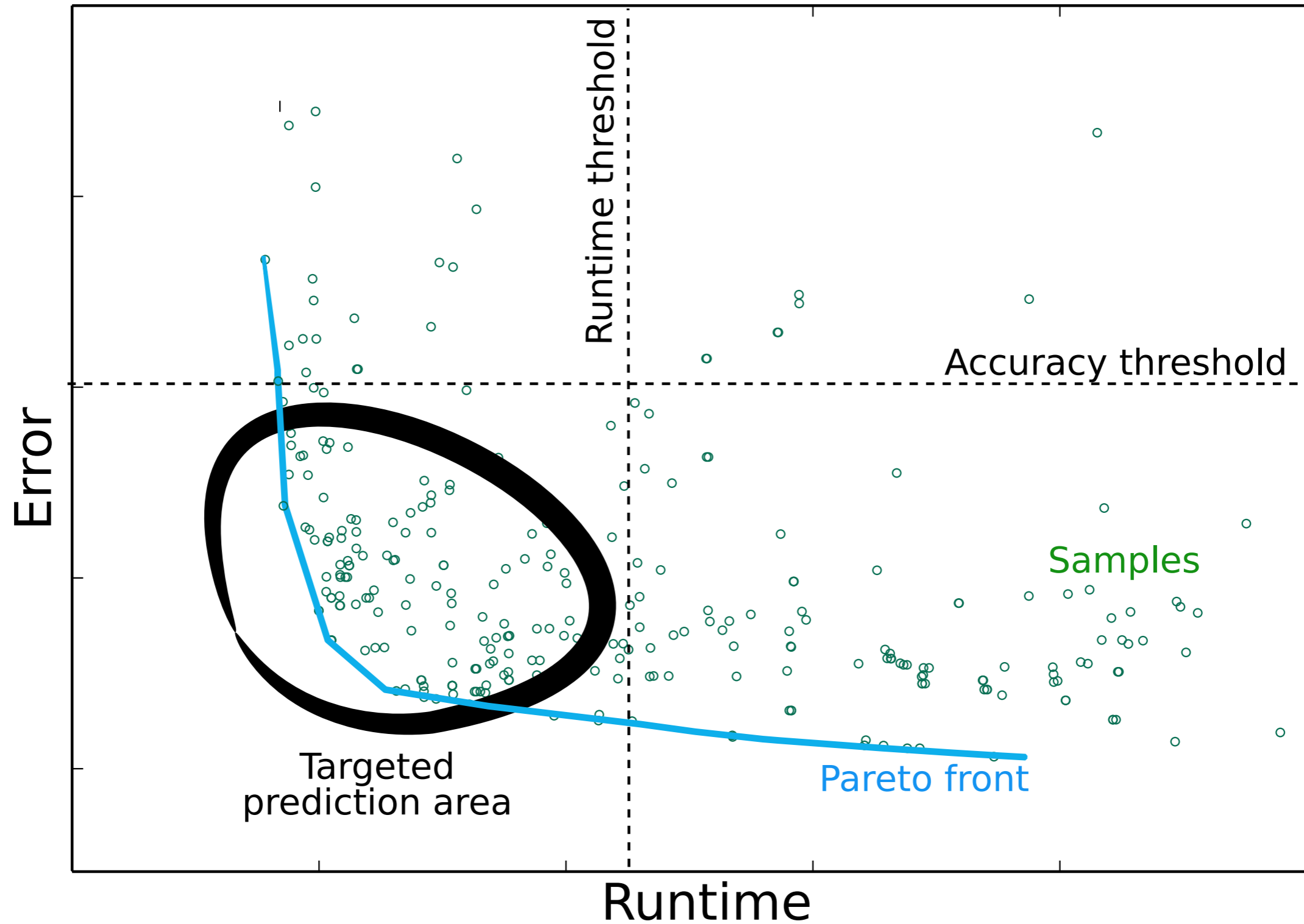


2x2x2

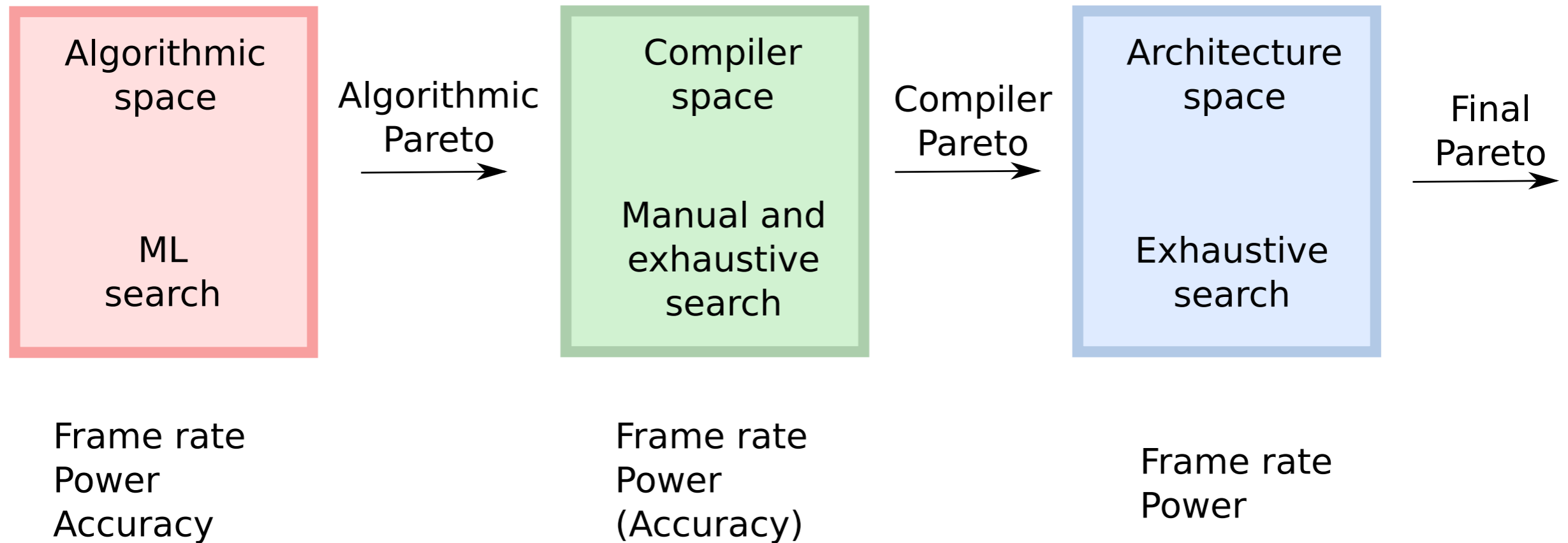
3x3x3

6x6x6

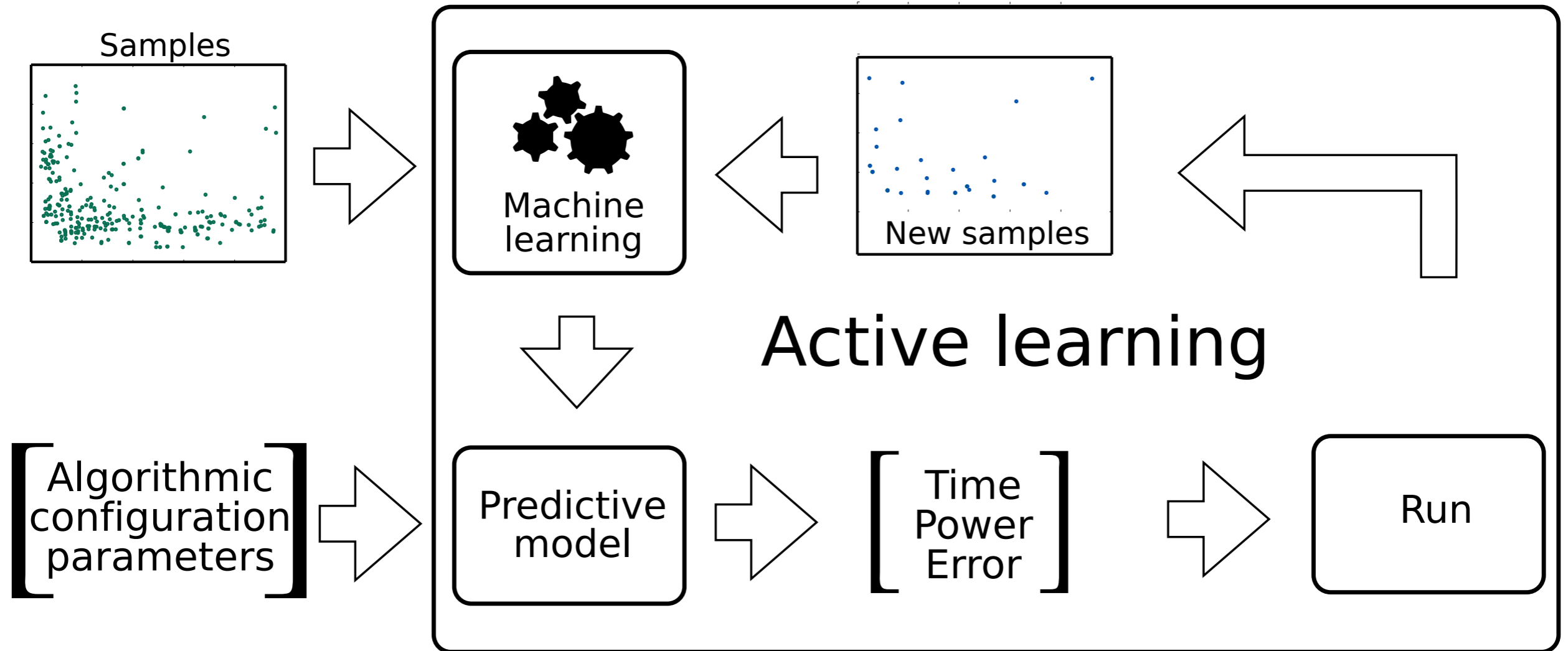
Exploration goal



Incremental exploration approach

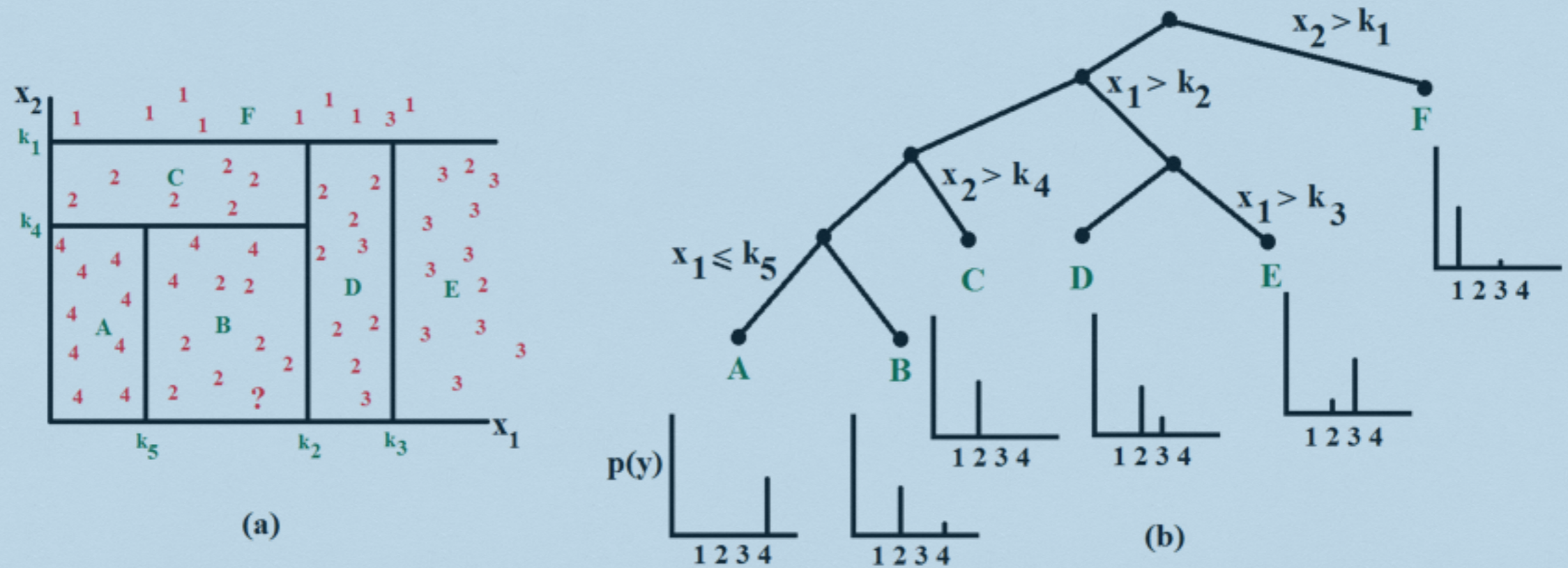


Algo design-space exploration (DSE)

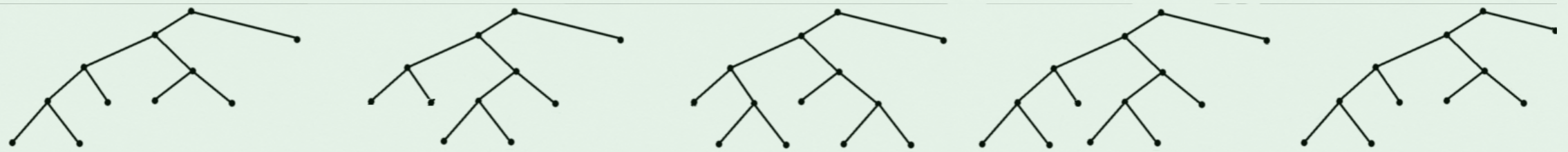


Machine learning methods used

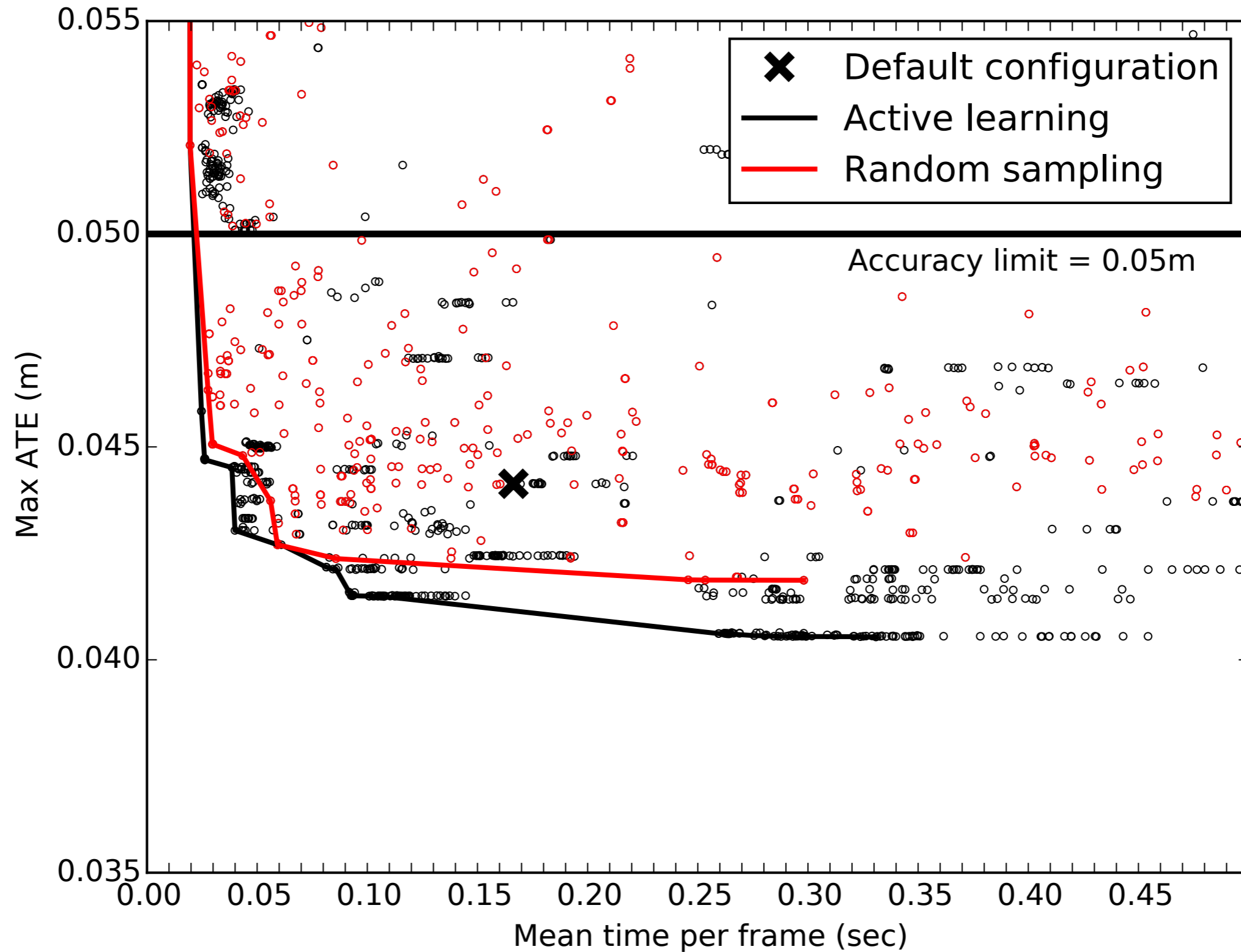
Decision tree



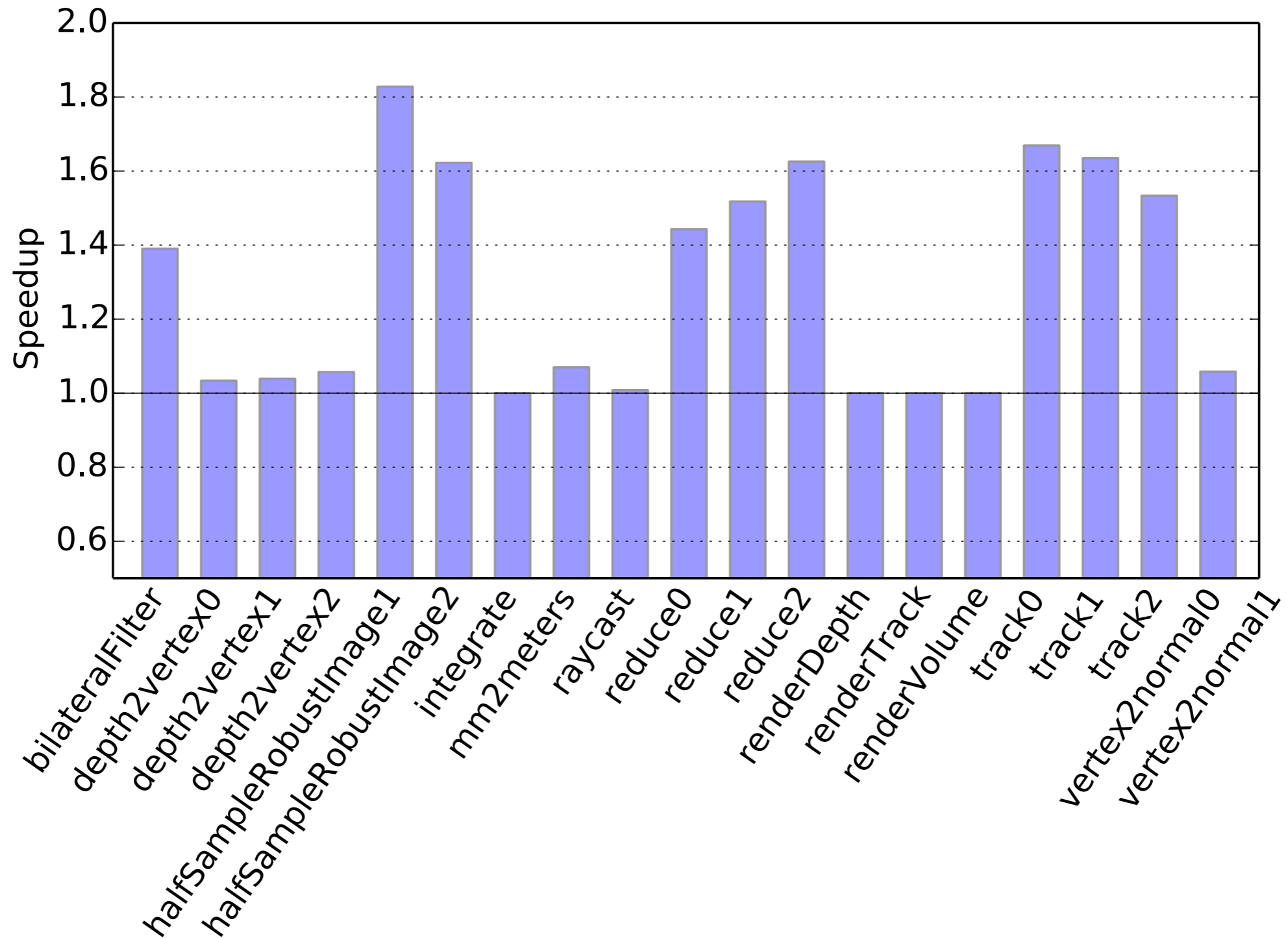
Random forest



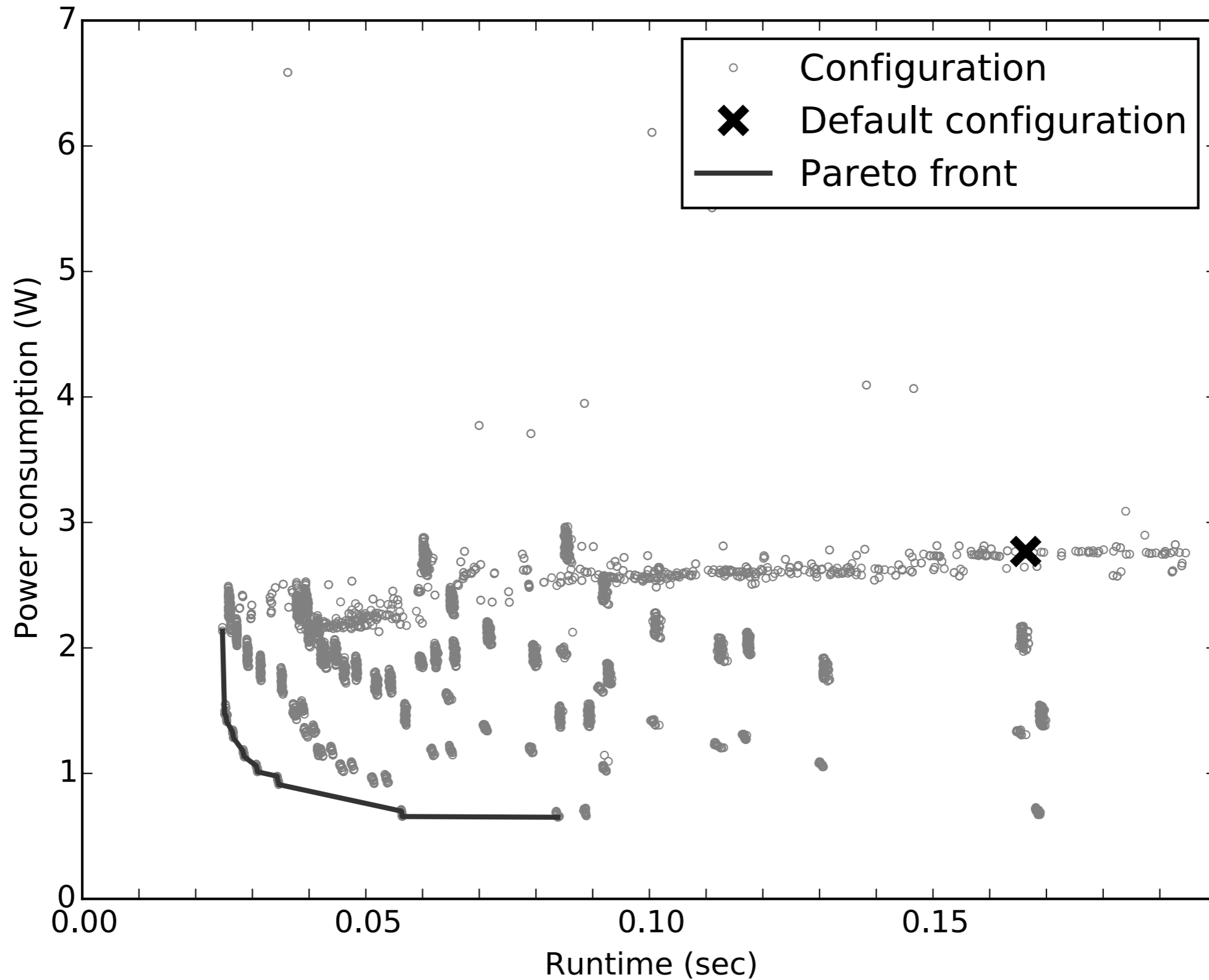
DSE on algorithmic parameters error/runtime



DSE compiler parameters speedup



DSE architecture parameters power/runtime

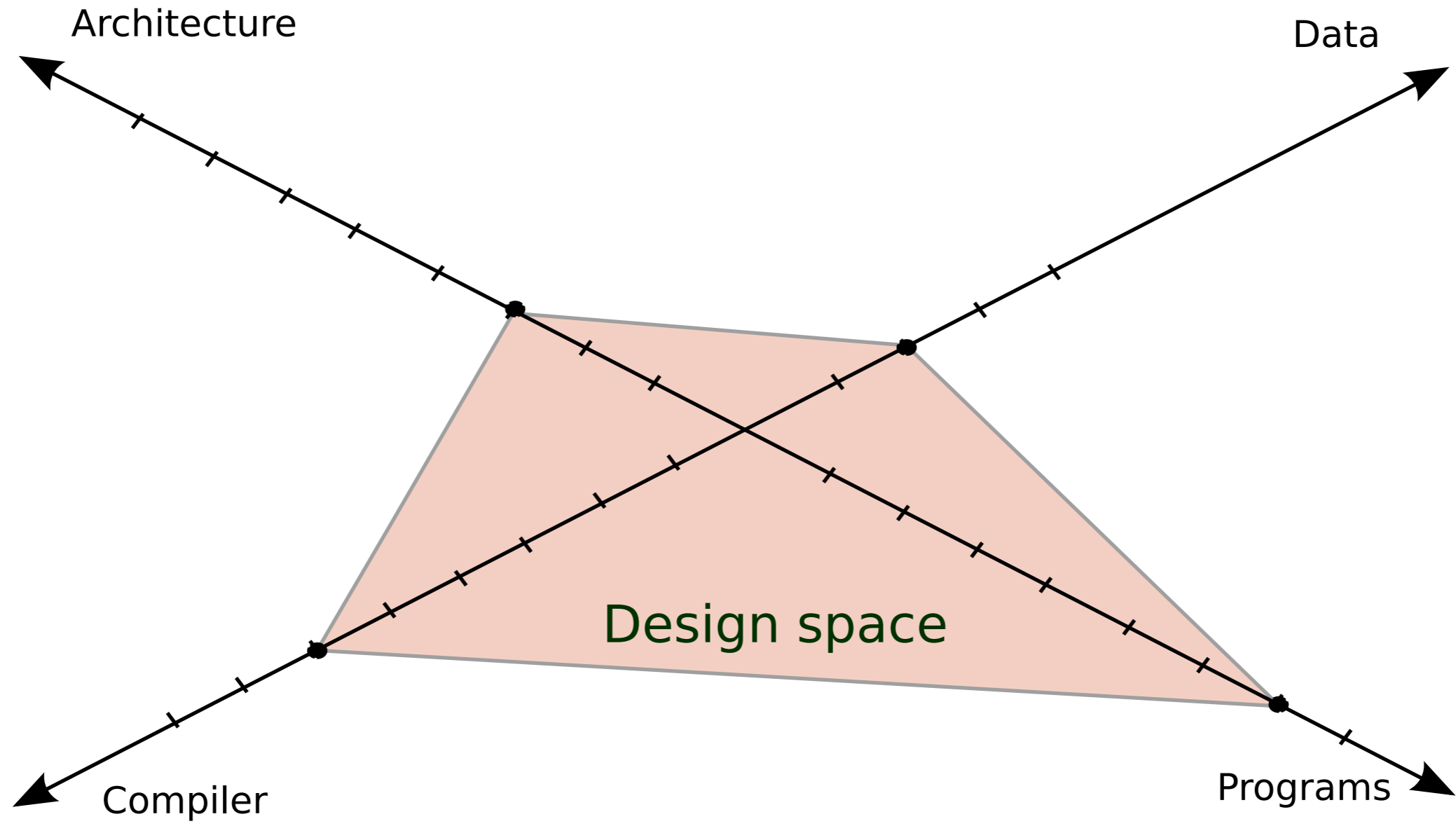


DSE final results and conclusion

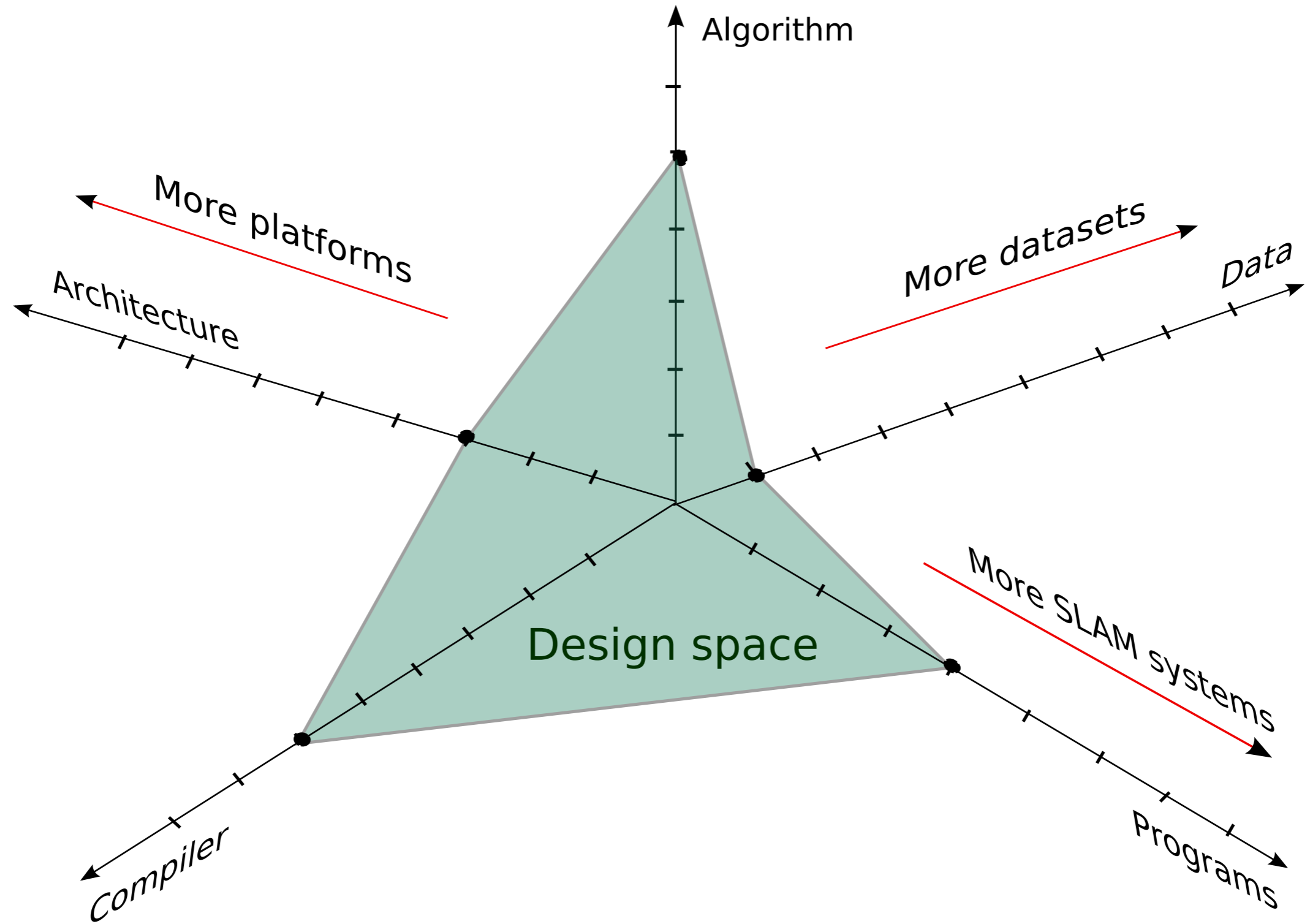
Constraint	Runtime (FPS)	Max ATE (cm)	Power (Watts)
Default	6.03	4.41	2.77
Best runtime	39.85	4.47	1.47
Best accuracy	1.51	3.30	2.38
Best power	11.92	4.45	0.65
Power < 1W	29.09	4.47	0.98
Power < 2W	39.85	4.47	1.47
FPS > 10	11.92	4.45	0.65
FPS > 20	28.87	4.47	0.91
FPS > 30	32.38	4.47	1.01

- Multi-objective optimisation: frame rate/power/accuracy
- Auto-tuning tool to pick interesting points
- Most of the improvement comes from the algorithmic space
- KinectFusion real-time on a popular embedded device
- Enabling auto-tuning at the domain-specific language (DSL) level

DSE the big picture I



DSE the big picture II



SLAMBench today

- Publicly released
13/11/2014
(1100+ downloads)

- Early adopters:
 - Computer Vision
 - Compiler/runtime
 - Architecture



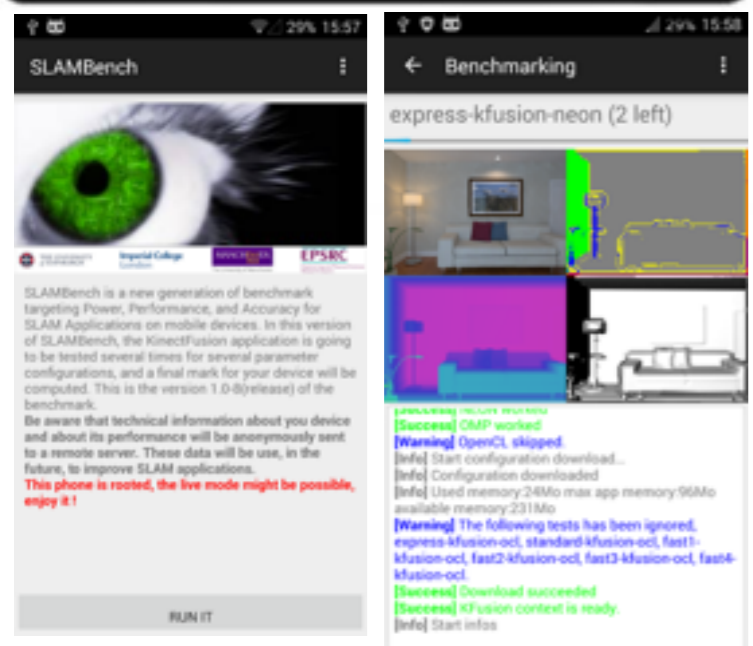
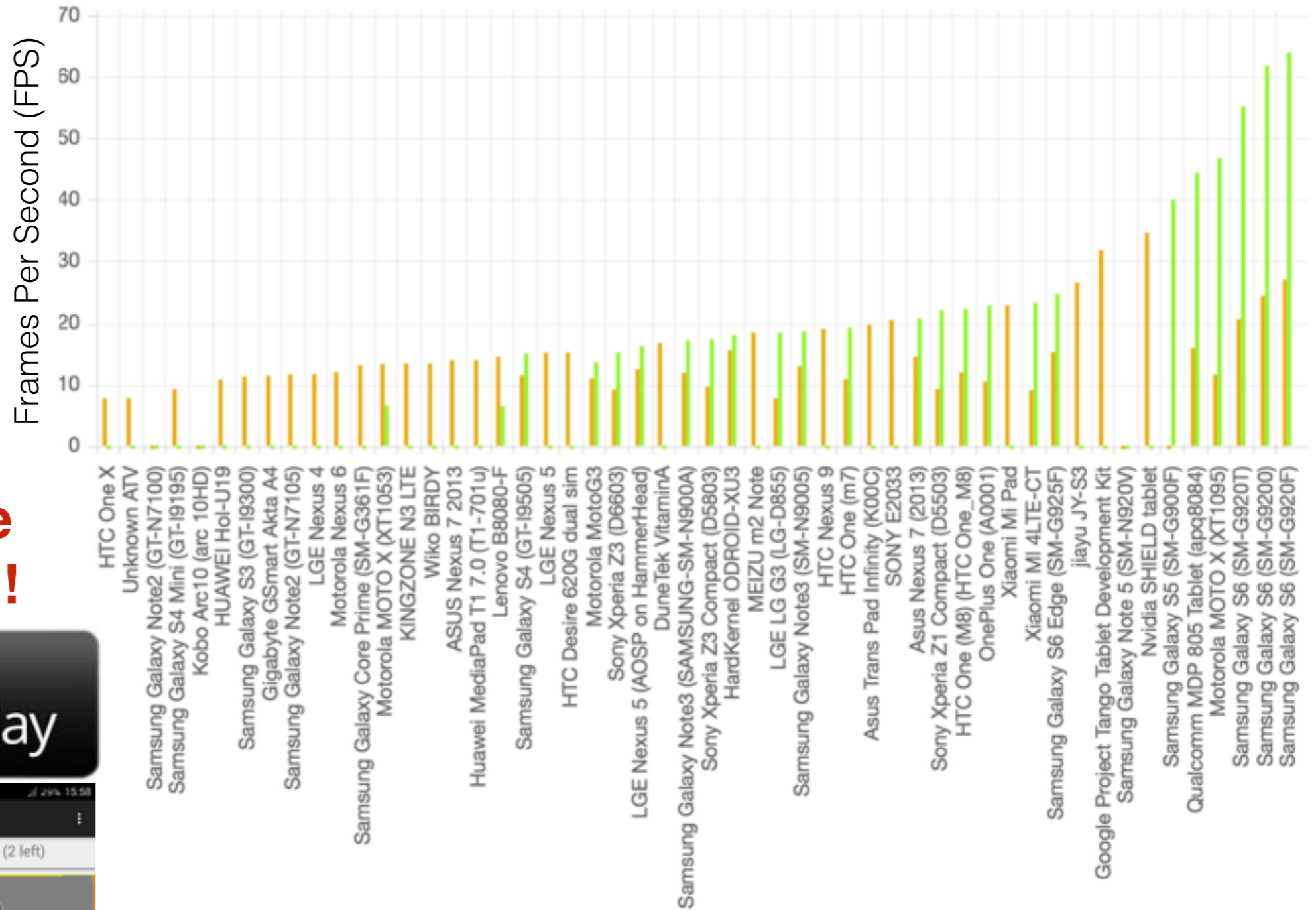
Web: apt.cs.manchester.ac.uk/projects/PAMELA/tools/SLAMBench/

- Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM (ICRA 2015)*
 - Comparative Design Space Exploration of Dense and Semi-Dense SLAM (ICRA 2016)*
- Papers for further reads**

Crowdsourcing mobile Android SLAMBench

- SLAMBench OpenMP
- SLAMBench OpenCL

**Get it now,
And see where
your device is!!**



- It runs a set of configurations on the available languages on your device
- Then shows the best achieved result



References I

1. [Nardi et al. 2015] L. Nardi, B. Bodin, M. Z. Zia, J. Mawer, A. Nisbet, P. H. J. Kelly, A. J. Davison, M. Luján, M. F. P. O'Boyle, G. Riley, N. Topham, and S. Furber. "Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM." Submitted, arXiv:1410.2167, 2015.
2. [Newcombe et al. ICCV 2011] R. A. Newcombe, S. J. Lovegrove and A. J. Davison. "DTAM: Dense tracking and mapping in real-time." Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.
3. [Rusinkiewicz and Levoy 2001] S. Rusinkiewicz, and M. Levoy. "Efficient variants of the ICP algorithm." 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on. IEEE, 2001.
4. [Chen et al. 2013] J. Chen, D. Bautembach, and S. Izadi, Scalable real-time volumetric surface reconstruction, in ACM Trans. Graph., 2013.
5. [Newcombe et al. ISMAR 2011] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. "KinectFusion: Real-time dense surface mapping and tracking." 10th IEEE Int. Symp. on Mixed and augmented reality (ISMAR), 2011.
6. [Handa et al. 2014] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM. IEEE Int. Conf. on Robotics and Automation, ICRA 2014.
7. [Reitmayr] G. Reitmayr. KFusion github 2011. <https://github.com/GerhardR/kfusion>
8. [Curless and Levoy 1996] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In Proc. Computer graphics and interactive technique. ACM, 1996.
9. [Whelan et al. 2012] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald. Kintinuous: Spatially extended kinectfusion. 2012.



References II

- [Ogilvie 2014] Ogilvie, William, et al. "Fast automatic heuristic construction using active learning." Proceedings of the Workshop on Languages and Compilers for Parallel Computing (LCPC'14). 2014.
- [Siegmund 2015] Siegmund Norbert et al. "Performance-influence models for highly configurable systems", submitted FSE 2015.
- [Guo 2013] Guo, Jianmei, et al. "Variability-aware performance prediction: A statistical learning approach." Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on. IEEE, 2013.
- [Grewe 2011] Grewe, Dominik et al. "A static task partitioning approach for heterogeneous systems using OpenCL." Compiler Construction. Springer Berlin Heidelberg, 2011.
- [Kurek 2013] Kurek, Maciej, Tianchi Liu, and Wayne Luk. "MULTI-OBJECTIVE SELF-OPTIMIZATION OF RECONFIGURABLE DESIGNS WITH MACHINE LEARNING." 2nd Workshop on Self-Awareness in Reconfigurable Computing Systems (SRCS'13). 2013.
- [Balaprakash 2013] Balaprakash, Prasanna, Robert B. Gramacy, and Stefan M. Wild. "Active-learning-based surrogate models for empirical performance tuning." Cluster Computing (CLUSTER), 2013 IEEE International Conference on. IEEE, 2013.
- [Vespa 2015] Vespa Emanuele. "Sparse voxelization of dense volumetric reconstruction with automated analysis of scene reconstruction quality." M.Res. thesis, Imperial College London, 2015.

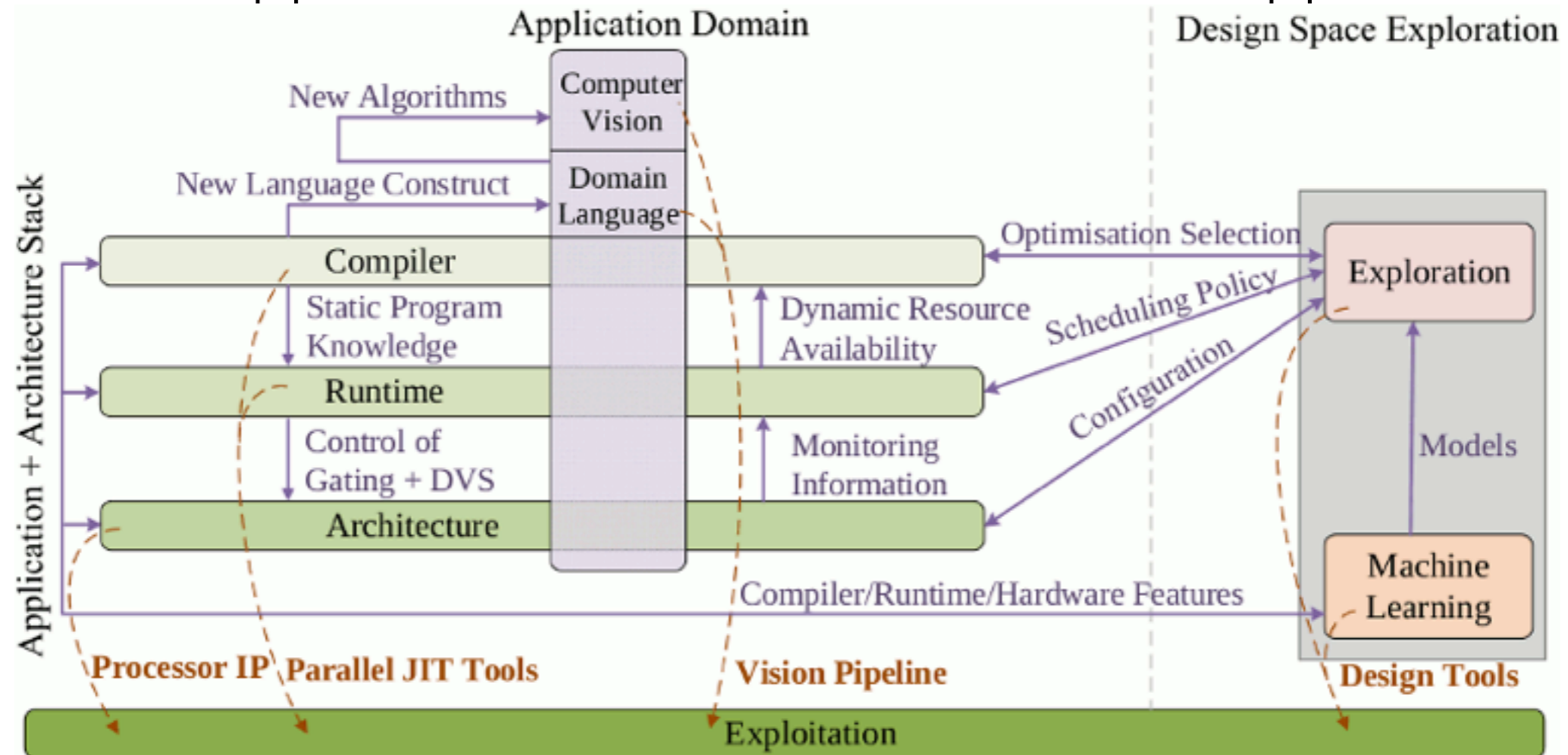


Backup slides



PAMELA project

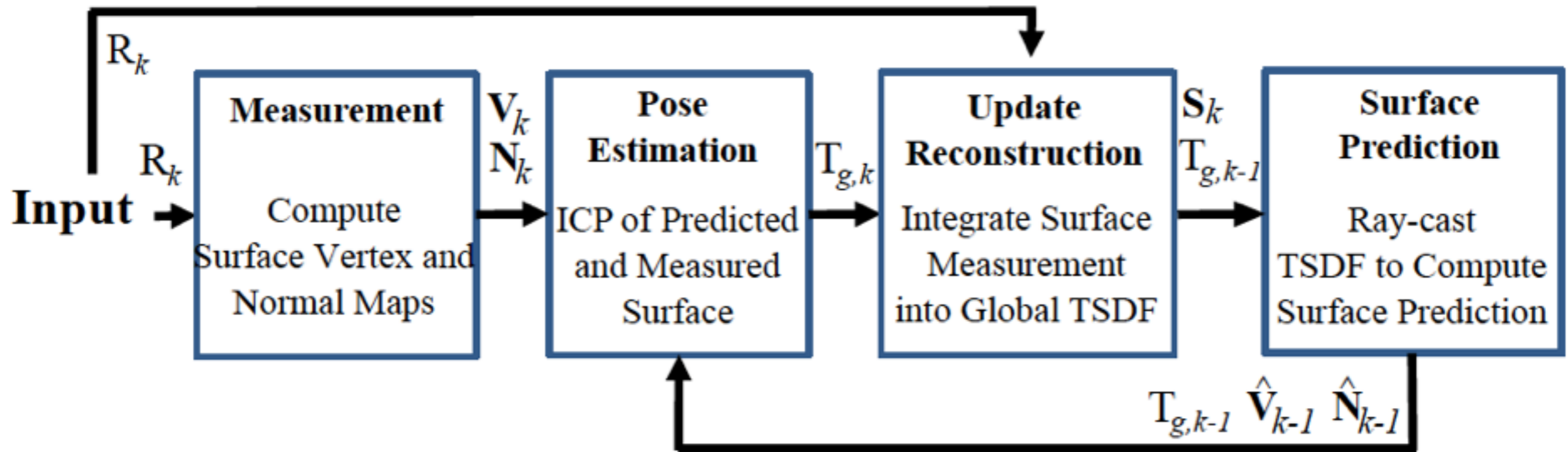
Panoramic Approach to the Many-core LANDscape -
from application to end-device: a holistic approach



In collaboration with:

B. Bodin, M Z. Zia, J. Mawer, E. Vespa, M. K. Emani, A. Nisbet, G. S. Shenoy, M. F. P. O'Boyle, P. H. J. Kelly, B. Franke, C. Kotselidis, M. Luján, A. J. Davison, G. Riley, N. Topham and S. Furber



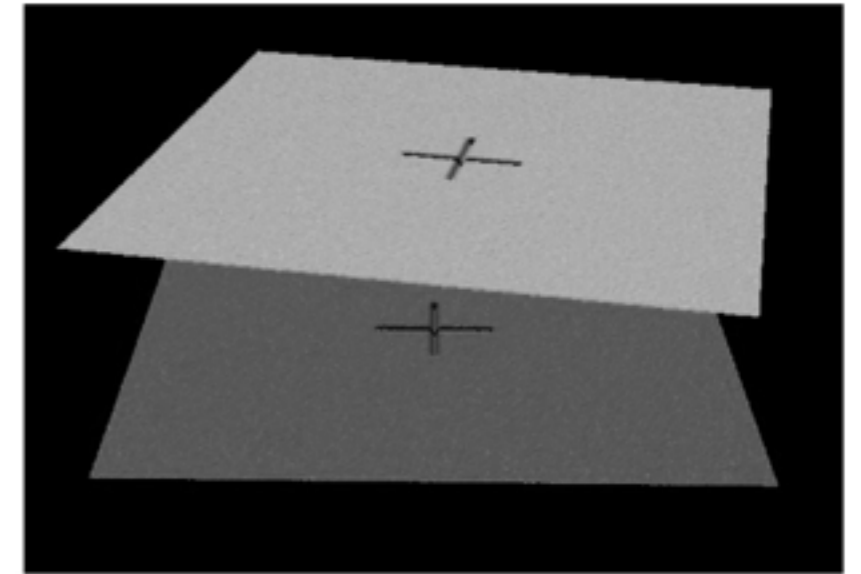
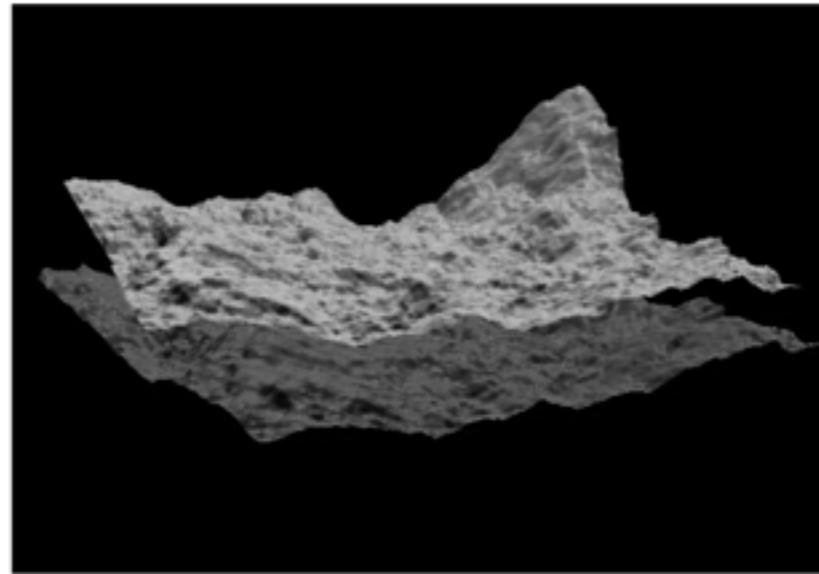
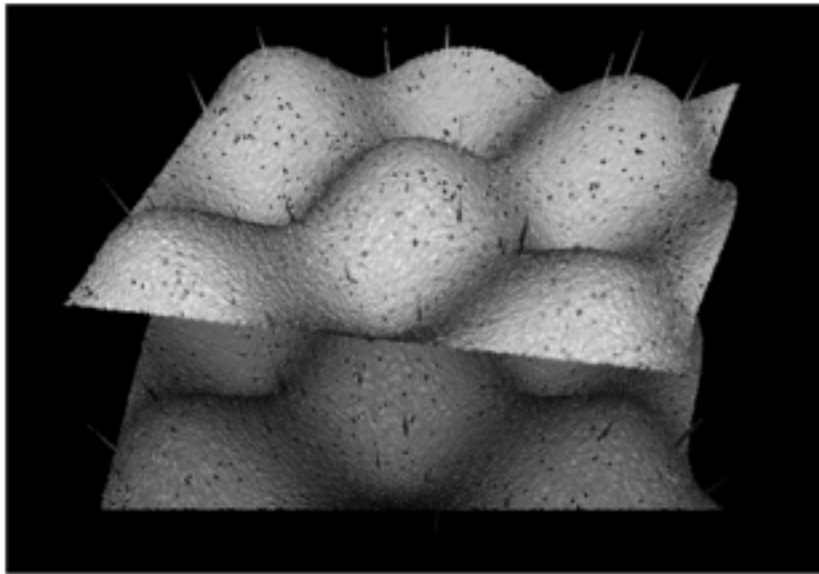


KinectFusion pipeline

- First dense monocular SLAM algorithm [Newcombe et al. ISMAR 2011]
- Adopted as a major building block in more recent SLAM systems
- Implementation based on [Reitmayr 2011] CUDA implementation



ICP registration

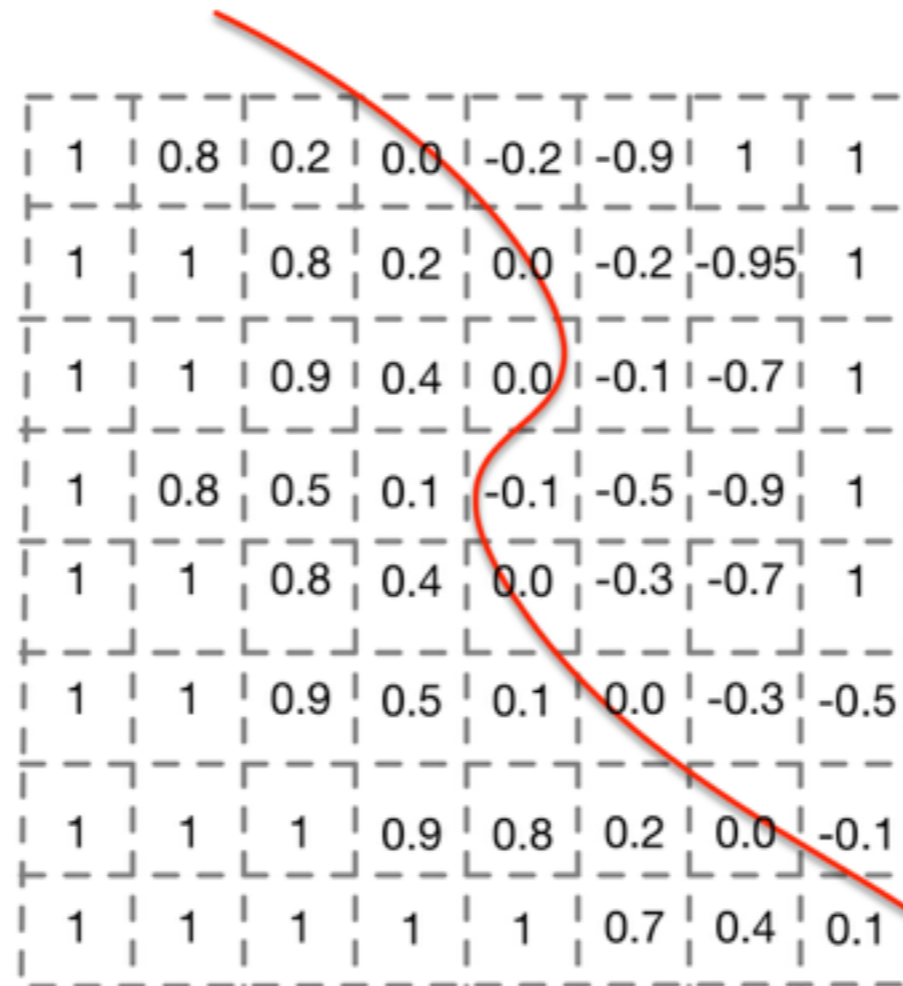


[Rusinkiewicz and Levoy 2001]

- Iterative Closest Point (ICP):
6 DoF rigid body transform from frame $k-1$ to frame k
- Iterative algorithm computing an energy function minimisation



Truncated Signed Distance Function



[Vespa 2015]

- Example of a 2D Truncated Signed Distance Function (TSDF)
- The red line shows the zero isosurface representing the best estimate of the observed surface



How to measure SLAM “Performance”?

SLAM computation depends on:

- Images acquired
- Way the camera is moved
- Numerical approximations
- Processing frame rate
(depends on hardware capability)

**Need for reproducibility
and accuracy check**

Pre-recorded scenes

Process-every-frame mode

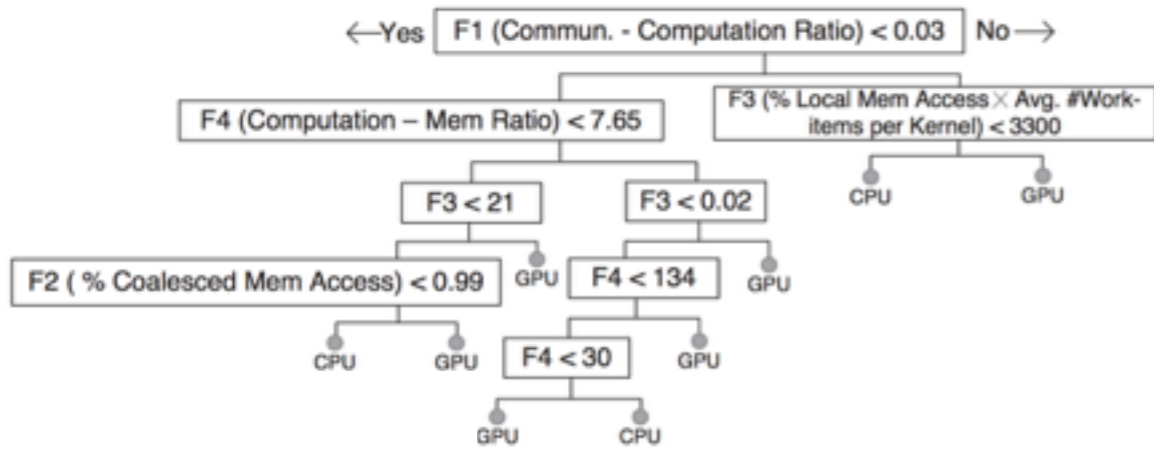
*ICL-NUIM dataset
(frames and ground truth)*



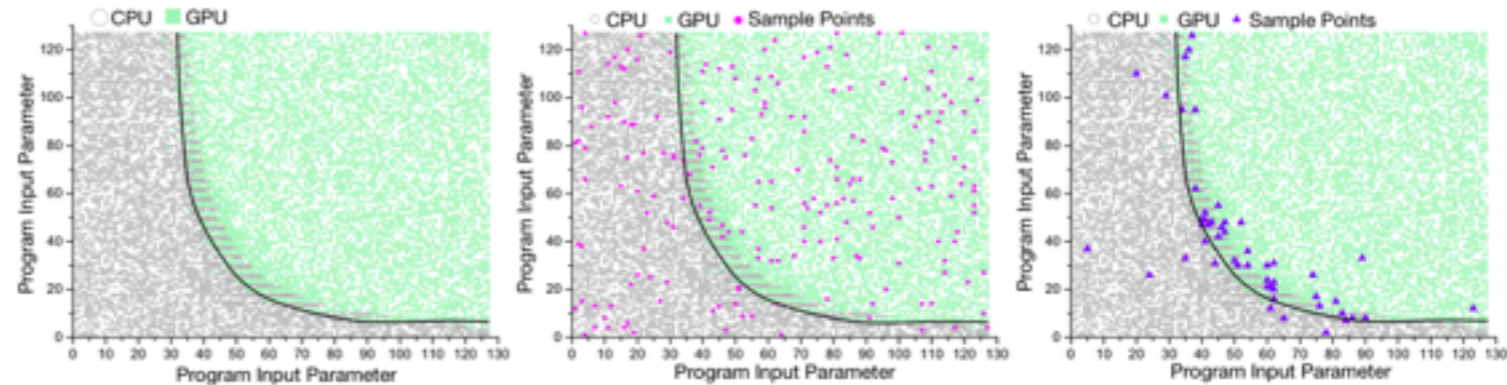
Why the DSE work is unique?

Luigi Nardi - Imperial College London

[Grewe 2011]: CPU/GPU mapping using a classification decision tree on code features

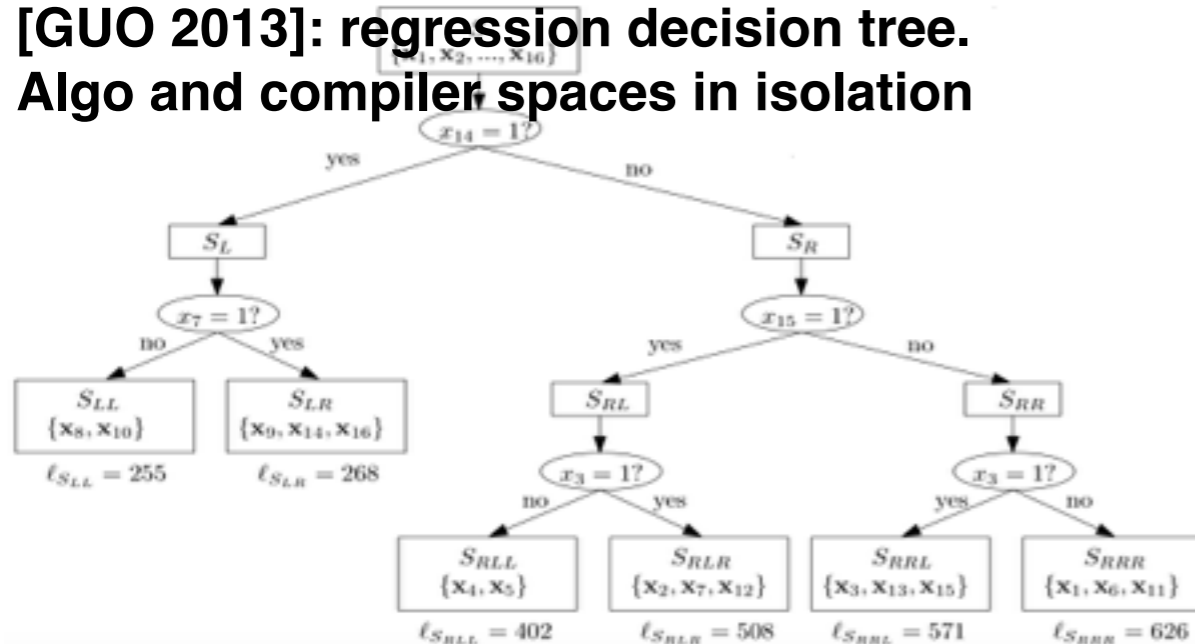


[Ogilvie 2014]: CPU/GPU static mapping using active learning and query by committee classifiers



(a) The problem space (b) Random sample points (c) Intelligent sample points

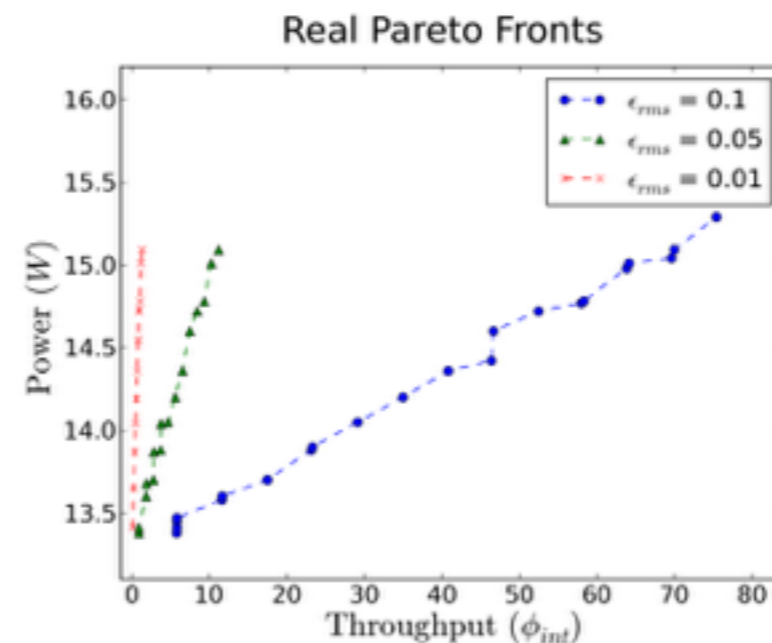
[GUO 2013]: regression decision tree. Algo and compiler spaces in isolation



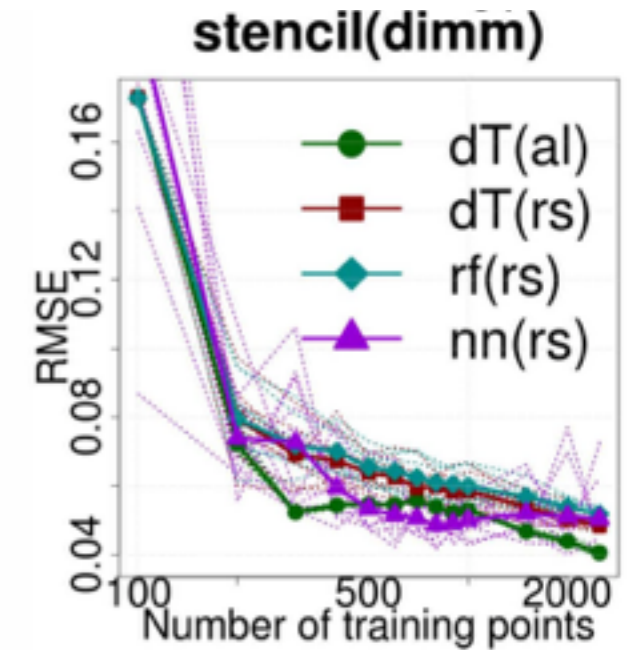
[Siegmond 2015]: performance-influence models using linear regression. Algo and compiler spaces in isolation

$$\Pi(c) = 50 + \underbrace{20 \cdot c(E)}_{\Phi_{E,C}} + \underbrace{15 \cdot c(C)}_{\Phi_{E,P}} + \underbrace{5 \cdot c(S)}_{\Phi_{E,C,D}} - \underbrace{0.5 \cdot c(P)}_{\Phi_{E,C,D}} + \underbrace{1.5 \cdot c(D)}_{\Phi_{E,C,D}}^2 - \underbrace{10 \cdot c(E) \cdot c(C)}_{\Phi_{E,C}} + \underbrace{0.3 \cdot c(E) \cdot c(P)}_{\Phi_{E,P}} + \underbrace{2.5 \cdot c(E) \cdot c(C) \cdot c(D)}_{\Phi_{E,C,D}}$$

[Kurek 2013]: MOMLO GPs, SVM, PSO. Joint algo and HW spaces



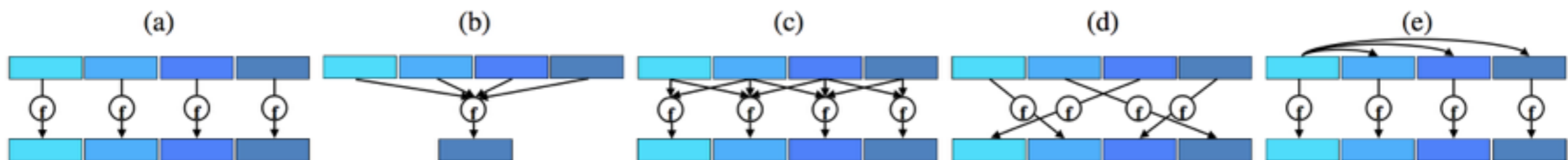
[Balaprakash 2013]: new parallel dynamic trees AL method for HPC. Compiler space MOMLO on runtime and power



SLAMBench DSE: joint space exploration, multi-objective function, actual application, improvement in "performance"

SLAMBench kernels

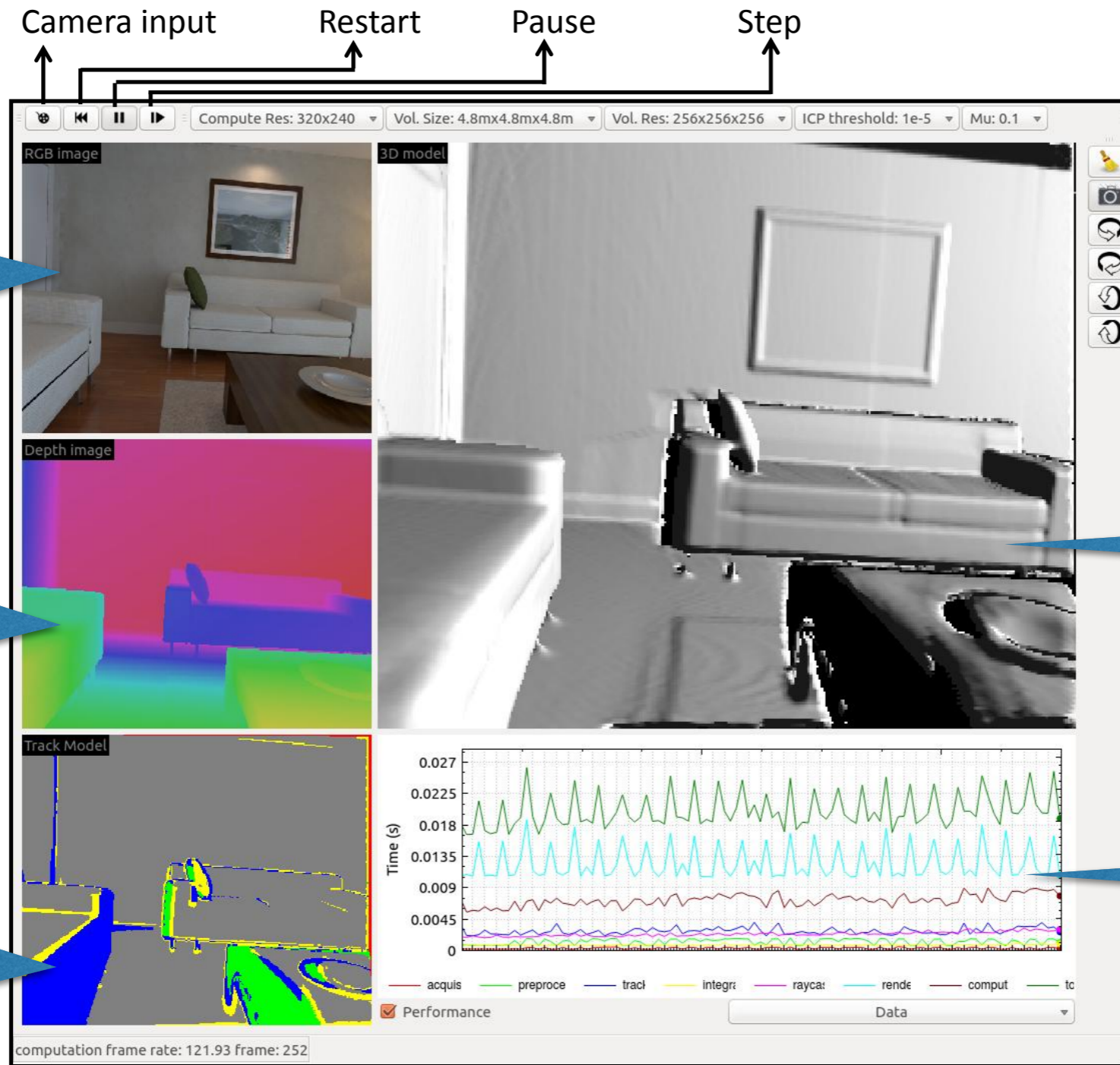
Kernels	Pipeline	Pattern	In	Out	%
acquire	Acquire	n/a	pointer	2D	0.03
mm2meters	Preprocess	Gather	2D	2D	0.06
bilateralFilter	Preprocess	Stencil	2D	2D	33.68
halfSample	Track	Stencil	2D	2D	0.05
depth2vertex	Track	Map	2D	2D	0.11
vertex2normal	Track	Stencil	2D	2D	0.27
track	Track	Map/Gather	2D	2D	4.72
reduce	Track	Reduction	2D	6x6	2.99
solve	Track	Sequential	6x6	6x1	0.02
integrate	Integrate	Map/Gather	2D/3D	3D	12.85
raycast	Raycast	Search/Stencil	2D/3D	2D	35.87
renderDepth	Rendering	Map	2D	2D	0.12
renderTrack	Rendering	Map	2D	2D	0.06
renderVolume	Rendering	Search/Stencil	3D	2D	9.18



Parallel patterns: (a) Map, (b) Reduction, (c) Stencil, (d) Gather and (e) Search.



SLAMBench GUI



RGB camera
(not used)

Depth camera

Tracked
points

Reset
volume

Toggle
viewpoint
of 3D model

3D model

Performance



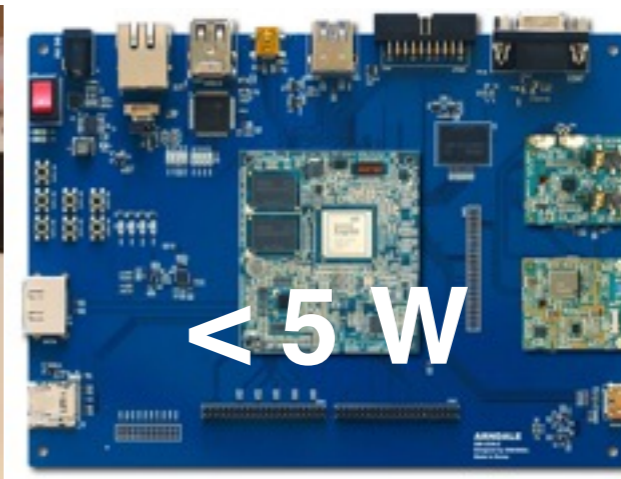
Platforms

Machine names	TITAN	GTX870M	TK1	ODROID (XU3)	Arndale
Machine type	Desktop	Laptop	Embedded	Embedded	Embedded
CPU	i7 Haswell	i7 Haswell	NVIDIA 4-Plus-1	Exynos 5422	Exynos 5250
CPU cores	4	4	4 (Cortex-A15) + 1	4 (Cortex-A15) + 4 (Cortex-A7)	2 (Cortex-A15)
CPU GHz	3.5	2.4	2.3	1.8	1.7
GPU	NVIDIA TITAN	NVIDIA GTX 870M	NVIDIA Tegra K1	ARM Mali-T628-MP6	ARM Mali-T604-MP4
GPU architecture	Kepler	Kepler	Kepler	Midgard 2nd gen.	Midgard 1st gen.
GPU FPU32s	2688	1344	192	60	40
GPU MHz	837	941	852	600	533
GPU GFLOPS (SP)	4500	2520	330	60+30 (72+36)	60 (71)
Language	CUDA/OpenCL/C++	CUDA/OpenCL/C++	CUDA/C++	OpenCL/C++	OpenCL/C++
OpenCL version	1.1	1.1	n/a	1.1	1.1
Toolkit version	CUDA 5.5	CUDA 5.5	CUDA 6.0	Mali SDK1.1.	Mali SDK1.1
Ubuntu OS (kernel)	13.04 (3.8.0)	14.04 (3.13.0)	14.04 (3.10.24)	14.04 (3.10.53)	12.04 (3.11.0)



Platforms

Machine names	TITAN	GTX870M	TK1	ODROID (XU3)	Arndale
Machine type	Desktop	Laptop	Embedded	Embedded	Embedded
CPU	i7 Haswell	i7 Haswell	NVIDIA 4-Plus-1	Exynos 5422	Exynos 5250
CPU cores	4	4	4 (Cortex-A15) + 1	4 (Cortex-A15) + 4 (Cortex-A7)	2 (Cortex-A15)
CPU GHz	3.5	2.4	2.3	1.8	1.7
GPU	NVIDIA TITAN	NVIDIA GTX 870M	NVIDIA Tegra K1	ARM Mali-T628-MP6	ARM Mali-T604-MP4
GPU architecture	Kepler	Kepler	Kepler	Midgard 2nd gen.	Midgard 1st gen.
GPU FPU32s	2688	1344	192	60	40
GPU MHz	837	941	852	600	533
GPU GFLOPS (SP)	4500	2520	330	60+30 (72+36)	60 (71)
Language	CUDA/OpenCL/C++	CUDA/OpenCL/C++	CUDA/C++	OpenCL/C++	OpenCL/C++
OpenCL version	1.1	1.1	n/a	1.1	1.1
Toolkit version	CUDA 5.5	CUDA 5.5	CUDA 6.0	Mali SDK1.1.	Mali SDK1.1
Ubuntu OS (kernel)	13.04 (3.8.0)	14.04 (3.13.0)	14.04 (3.10.24)	14.04 (3.10.53)	12.04 (3.11.0)



“Performance”: accuracy

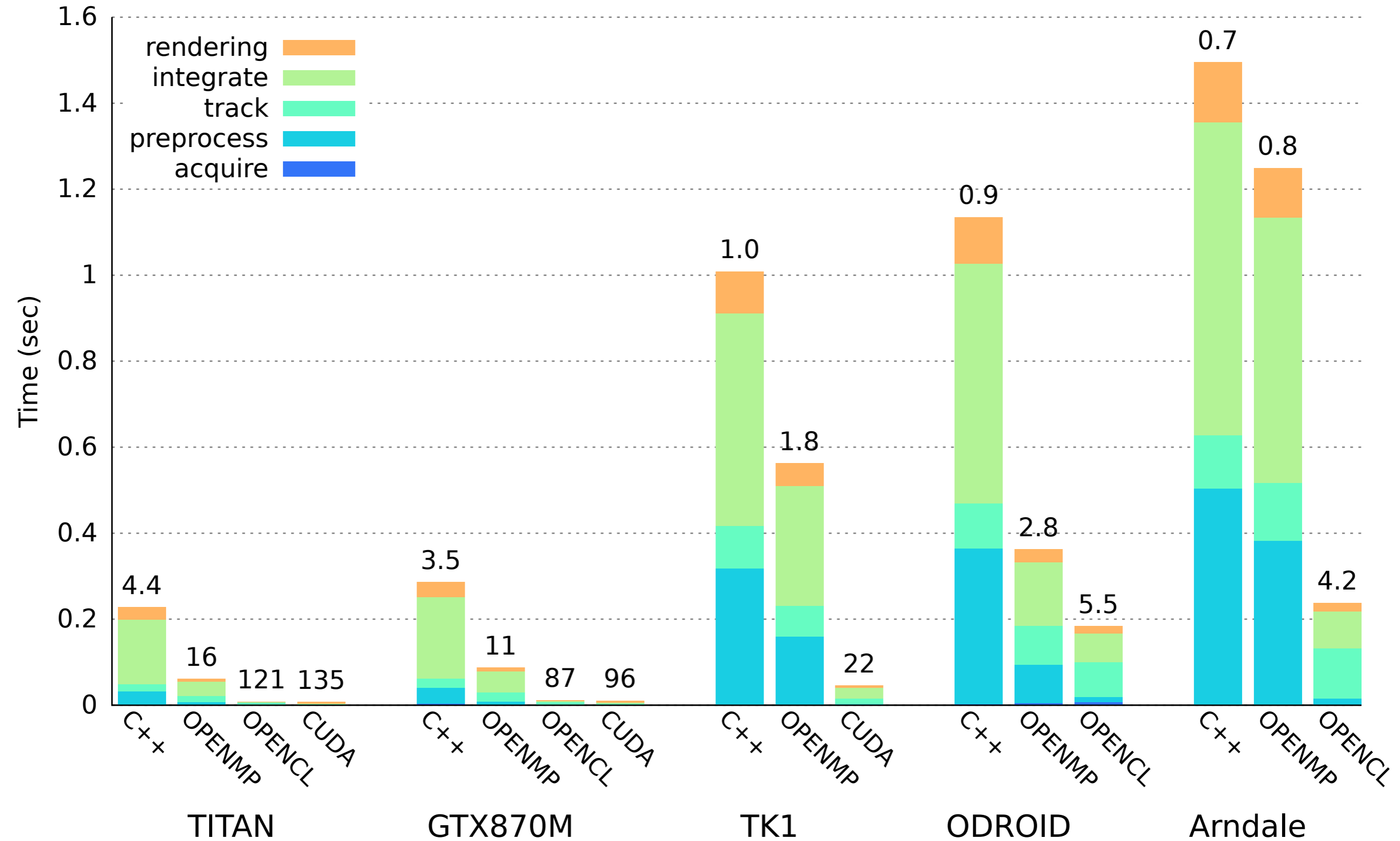
Absolute trajectory error (ATE) in cm - default algorithmic configuration

ATE in cm	TITAN	GTX870M	TK1	ODROID	Arndale
C++	2.07	2.07	2.06	2.06	2.06
OpenMP	2.07	2.07	2.06	2.06	2.06
OpenCL	2.07	2.07	n/a	2.01	2.07
CUDA	2.07	2.07	2.07	n/a	n/a

- ATE easy-to-use tool for non computer vision experts
- Semantic validation instead than bitwise accuracy

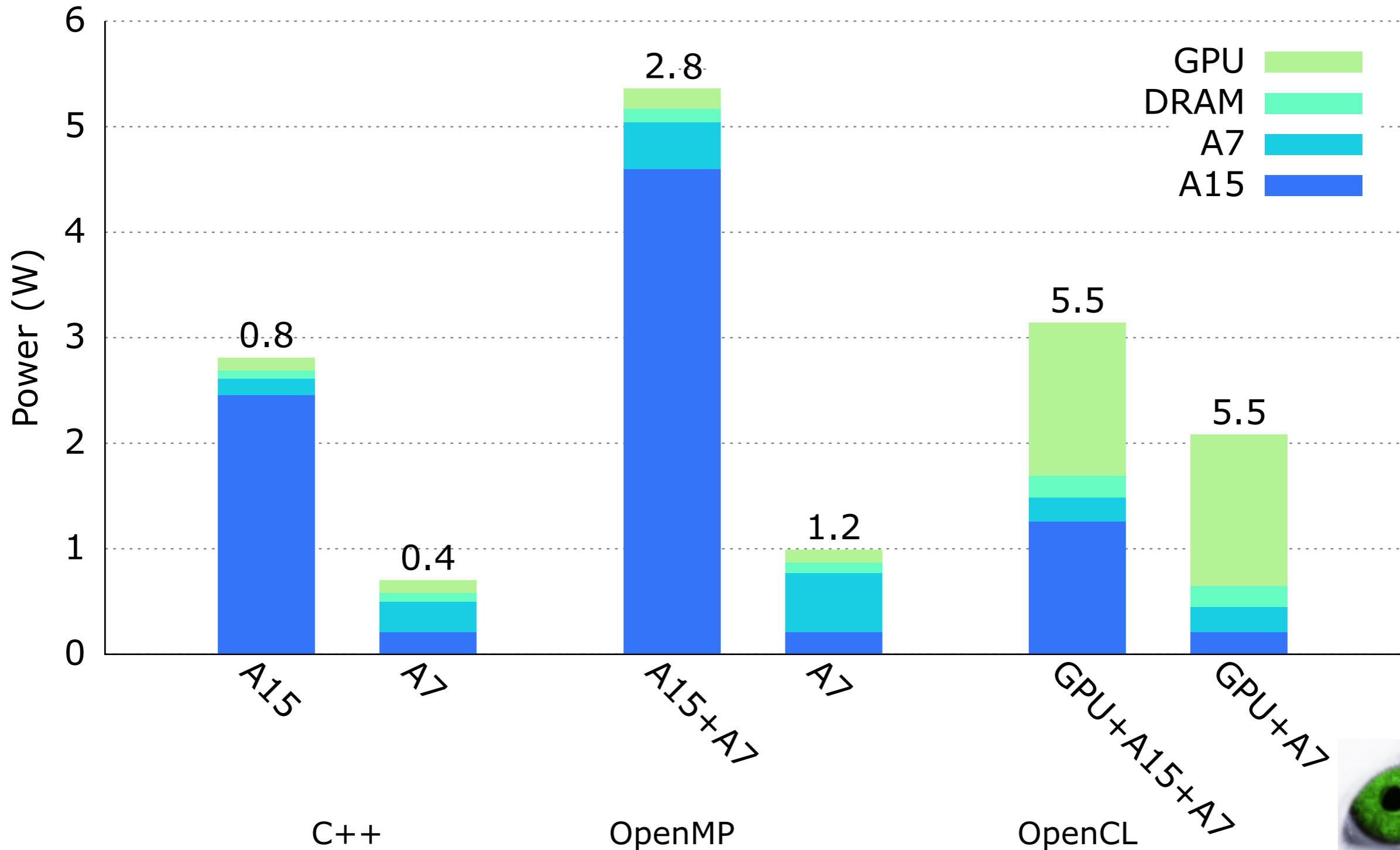
“Performance”: execution time

Mean time per frame (lower is better)

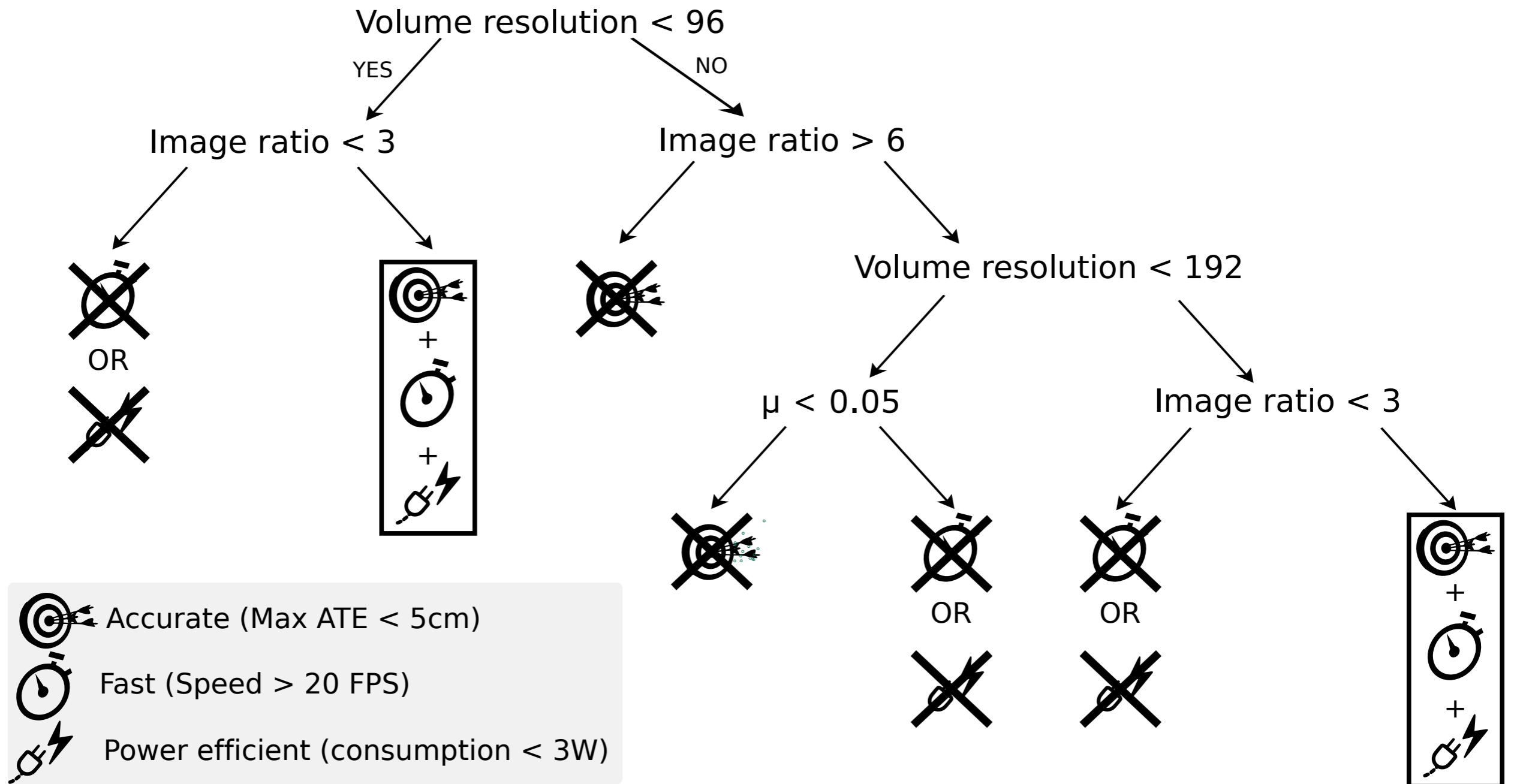


“Performance” power (ODROID-XU3)

On-board voltage/current sensors and split power rails:
power measured individually on big (A15), LITTLE (A7), GPU and DRAM



Predominant algorithmic features



Textual User Interface



```
hickory% make 2.openc1.log
LD_PRELOAD=./build/kfusion/thirdparty/liboclwrapper.so ./build/kfusion/kfusion-benchmark-openc1 -s 4.8 -p 0.34,0.5,0.24 -z 4 -c 2 -r 2 -k 481.2,480,320,240 -i living_room_traj2_
encl.log 2> oclwrapper.2.openc1.log
End of file(garbage found).
cat oclwrapper.2.openc1.log |grep -E ".+ [0-9]+ [0-9]+ [0-9]+" |cut -d" " -f1,4 > kernels.2.openc1.log
./kfusion/thirdparty/checkPos.py benchmark.2.openc1.log livingRoom2.gt.freiburg > resume.2.openc1.log
./kfusion/thirdparty/checkKernels.py kernels.2.openc1.log >> resume.2.openc1.log
hickory%
```

For benchmarking purposes,
SSH and post-processing
tools friendly



Textual User Interface

```
hickory% make 2.openc1.log
LD_PRELOAD=./build/kfusion/thirdparty/liboclwrapper.so ./build/kfusion/kfusion-benchmark-openc1 -s 4.8 -p 0.34,0.5,0.24 -z 4 -c 2 -r 2 -k 481.2,480,320,240 -i living_room_traj2_
encl.log 2> oclwrapper.2.openc1.log
End of file(garbage found).
cat oclwrapper.2.openc1.log |grep -E ".+ [0-9]+ [0-9]+ [0-9]+" |cut -d" " -f1,4 > kernels.2.openc1.log
./kfusion/thirdparty/checkPos.py benchmark.2.openc1.log livingRoom2.gt.freiburg > resume.2.openc1.log
./kfusion/thirdparty/checkKernels.py kernels.2.openc1.log >> resume.2.openc1.log
hickory%
```

Easy
run


frame grated	acquisition	preprocessing	tracking	integration	raycasting	rendering	computation	total	X	Y	Z	tracked	inte
0	0.003564	0.000451	0.001622	0.000461	0.000002	0.001124	0.002536	0.007223	0.0000	0.000000	0.000000	0	1
1	0.002724	0.000352	0.000903	0.000467	0.000001	0.000212	0.001722	0.004658	0.0000	0.000000	0.000000	0	1
2	0.002697	0.000336	0.000904	0.000459	0.000001	0.000212	0.001701	0.004611	0.0000	0.000000	0.000000	0	1
3	0.002663	0.000335	0.000905	0.000467	0.000604	0.000219	0.002311	0.005193	0.0000	0.000000	0.000000	0	1

Raw
data

For benchmarking purposes,
SSH and post-processing
tools friendly



Textual User Interface



Easy
run



Raw
data



Sum
up

For benchmarking purposes,
SSH and post-processing
tools friendly

```
hickory% make 2.openc1.log
LD_PRELOAD=./build/kfusion/thirdparty/liboclwrapper.so ./build/kfusion/kfusion-benchmark-openc1 -s 4.8 -p 0.34,0.5,0.24 -z 4 -c 2 -r 2 -k 481.2,480,320,240 -i living_room_traj2_
encl.log 2> oclwrapper.2.openc1.log
End of file(garbage found).
cat oclwrapper.2.openc1.log |grep -E ".+ [0-9]+ [0-9]+ [0-9]+" |cut -d" " -f1,4 > kernels.2.openc1.log
./kfusion/thirdparty/checkPos.py benchmark.2.openc1.log livingRoom2.gt.freiburg > resume.2.openc1.log
./kfusion/thirdparty/checkKernels.py kernels.2.openc1.log >> resume.2.openc1.log
hickory%
```

frame	acquisition	preprocessing	tracking	integration	raycasting	rendering	computation	total	X	Y	Z	tracked	inte
0	0.003564	0.000451	0.001622	0.000461	0.000002	0.001124	0.002536	0.007223	0.0000	0.000000	0.000000	0	1
1	0.002724	0.000352	0.000903	0.000467	0.000001	0.000212	0.001722	0.004658	0.0000	0.000000	0.000000	0	1
2	0.002697	0.000336	0.000904	0.000459	0.000001	0.000212	0.001701	0.004611	0.0000	0.000000	0.000000	0	1
3	0.002663	0.000335	0.000905	0.000467	0.000604	0.000219	0.002311	0.005193	0.0000	0.000000	0.000000	0	1

```
1 Get kfusion output data.
2 Skip kfusion line :
3 Skip num line :
4 kfusion file: Valid frames 882 dropped frames: 0
5 kfusion result      : 882 positions.
6 NUIM result        : 880 positions.
7 Working position is : 880
8 Untracked frames: 0
9 Shift kfusion trajectory...
10
11 A detailed statistical analysis is provided.
12 All durations are in seconds and the absolute trajectory error (ATE) is in centimeters.
13 ATE Min : 0.000000 Max : 0.049309 Mean : 0.020662 Total : 18.18239335
14 acquisition Min : 0.000056 Max : 0.009033 Mean : 0.002044 Total : 1.80289800
15 computation Min : 0.001701 Max : 0.009234 Mean : 0.005894 Total : 5.19872500
16 integration Min : 0.000001 Max : 0.000821 Mean : 0.000258 Total : 0.22773500
17 preprocessing Min : 0.000284 Max : 0.001884 Mean : 0.000441 Total : 0.38904600
18 raycasting Min : 0.000001 Max : 0.003313 Mean : 0.001345 Total : 1.18644400
19 rendering Min : 0.000201 Max : 0.003452 Mean : 0.000572 Total : 0.50420900
20 total Min : 0.003882 Max : 0.020054 Mean : 0.008510 Total : 7.50584700
21 tracking Min : 0.000903 Max : 0.006743 Mean : 0.003850 Total : 3.39543600
22 Get SlamBench data.
23 ResetVolume Count : 1 Min : 401056 Max : 401056 Mean : 401056.000000 Total : 401056
24 bilateral_filter Count : 882 Min : 139520 Max : 205632 Mean : 159880.199546 Total : 141014336
25 depth2vertex Count : 2646 Min : 5440 Max : 48384 Mean : 10585.167045 Total : 28008352
26 halfSampleRobust Count : 1764 Min : 6624 Max : 328480 Mean : 9546.557823 Total : 16840128
27 integrate Count : 443 Min : 270880 Max : 793664 Mean : 485648.469526 Total : 215142272
28 mm2meters Count : 882 Min : 6912 Max : 13632 Mean : 8977.306122 Total : 7917984
29 raycast Count : 879 Min : 435584 Max : 3284768 Mean : 1323028.423208 Total : 1162941984
30 reduce Count : 12117 Min : 31488 Max : 523488 Mean : 138305.687546 Total : 1675850016
31 renderDepth Count : 882 Min : 9568 Max : 39904 Mean : 12002.975057 Total : 10586624
32 renderTrack Count : 882 Min : 16608 Max : 33088 Mean : 18716.081633 Total : 16507584
33 renderVolume Count : 221 Min : 479840 Max : 3143200 Mean : 1338055.819005 Total : 295710336
34 track Count : 12117 Min : 13472 Max : 129344 Mean : 51505.231988 Total : 624088896
35 vertex2normal Count : 2646 Min : 8544 Max : 73504 Mean : 22275.362056 Total : 58940608
```



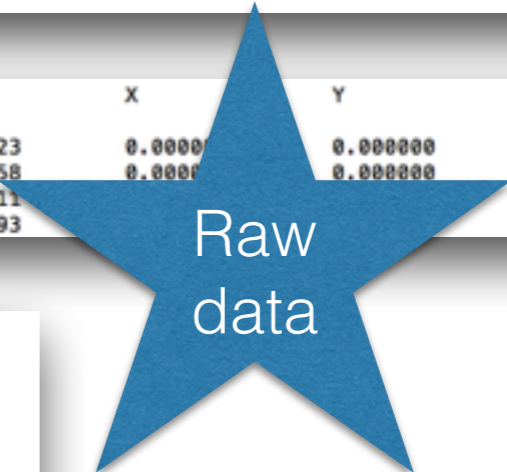
Textual User Interface



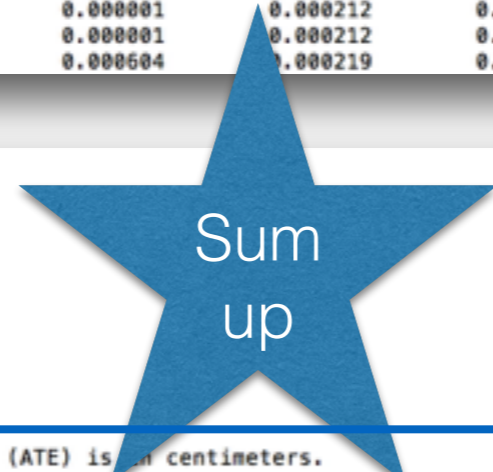
Easy
run

```
hickory% make 2.openc1.log
LD_PRELOAD=./build/kfusion/thirdparty/liboclwrapper.so ./build/kfusion/kfusion-benchmark-openc1 -s 4.8 -p 0.34,0.5,0.24 -z 4 -c 2 -r 2 -k 481.2,480,320,240 -i living_room_traj2_
encl.log 2> oclwrapper.2.openc1.log
End of file(garbage found).
cat oclwrapper.2.openc1.log |grep -E ".+ [0-9]+ [0-9]+ [0-9]+" |cut -d" " -f1,4 > kernels.2.openc1.log
./kfusion/thirdparty/checkPos.py benchmark.2.openc1.log livingRoom2.gt.freiburg > resume.2.openc1.log
./kfusion/thirdparty/checkKernels.py kernels.2.openc1.log >> resume.2.openc1.log
hickory%
```

frame	acquisition	preprocessing	tracking	integration	raycasting	rendering	computation	total	X	Y	Z	tracked	inte
0	0.003564	0.000451	0.001622	0.000461	0.000002	0.001124	0.002536	0.007223	0.0000	0.000000	0.000000	0	1
1	0.002724	0.000352	0.000903	0.000467	0.000001	0.000212	0.001722	0.004658	0.0000	0.000000	0.000000	0	1
2	0.002697	0.000336	0.000904	0.000459	0.000001	0.000212	0.001701	0.004611	0.0000	0.000000	0.000000	0	1
3	0.002663	0.000335	0.000905	0.000467	0.000604	0.000219	0.002311	0.005193	0.0000	0.000000	0.000000	0	1

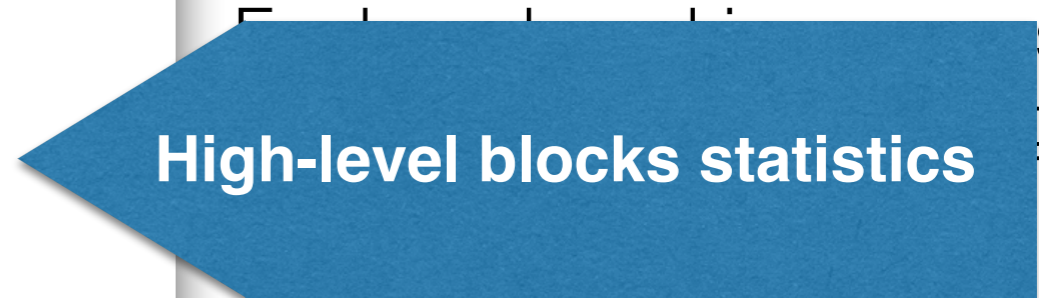


Raw
data



Sum
up

```
1 Get kfusion output data.
2 Skip kfusion line :
3 Skip num line :
4 kfusion file: Valid frames 882 dropped frames: 0
5 kfusion result      : 882 positions.
6 NUIM result        : 880 positions.
7 Working position is : 880
8 Untracked frames: 0
9 Shift kfusion trajectory...
10
11 A detailed statistical analysis is provided.
12 All durations are in seconds and the absolute trajectory error (ATE) is in centimeters.
13
14 ATE      Min : 0.000000 Max : 0.049309 Mean : 0.020662 Total : 18.18239335
15 acquisition Min : 0.000056 Max : 0.009033 Mean : 0.002044 Total : 1.80289800
16 computation Min : 0.001701 Max : 0.009234 Mean : 0.005894 Total : 5.19872500
17 integration Min : 0.000001 Max : 0.000821 Mean : 0.000258 Total : 0.22773500
18 preprocessing Min : 0.000284 Max : 0.001884 Mean : 0.000441 Total : 0.38904600
19 raycasting Min : 0.000001 Max : 0.003313 Mean : 0.001345 Total : 1.18644400
20 rendering Min : 0.000201 Max : 0.003452 Mean : 0.000572 Total : 0.50420900
21 total Min : 0.003882 Max : 0.020054 Mean : 0.008510 Total : 7.50584700
22 tracking Min : 0.000903 Max : 0.006743 Mean : 0.003850 Total : 3.39543600
23 Get SlamBench data.
24 ResetVolume Count : 1 Min : 401056 Max : 401056 Mean : 401056.000000 Total : 401056
25 bilateral_filter Count : 882 Min : 139520 Max : 205632 Mean : 159880.199546 Total : 141014336
26 depth2vertex Count : 2646 Min : 5440 Max : 48384 Mean : 10585.167045 Total : 28008352
27 halfSampleRobust Count : 1764 Min : 6624 Max : 328480 Mean : 9546.557823 Total : 16840128
28 integrate Count : 443 Min : 270880 Max : 793664 Mean : 485648.469526 Total : 215142272
29 mm2meters Count : 882 Min : 6912 Max : 13632 Mean : 8977.306122 Total : 7917984
30 raycast Count : 879 Min : 435584 Max : 3284768 Mean : 1323028.423208 Total : 1162941984
31 reduce Count : 12117 Min : 31488 Max : 523488 Mean : 138305.687546 Total : 1675850016
32 renderDepth Count : 882 Min : 9568 Max : 39904 Mean : 12002.975057 Total : 10586624
33 renderTrack Count : 882 Min : 16608 Max : 33088 Mean : 18716.081633 Total : 16507584
34 renderVolume Count : 221 Min : 479840 Max : 3143200 Mean : 1338055.819005 Total : 295710336
35 track Count : 12117 Min : 13472 Max : 129344 Mean : 51505.231988 Total : 624088896
36 vertex2normal Count : 2646 Min : 8544 Max : 73504 Mean : 22275.362056 Total : 58940608
```



High-level blocks statistics

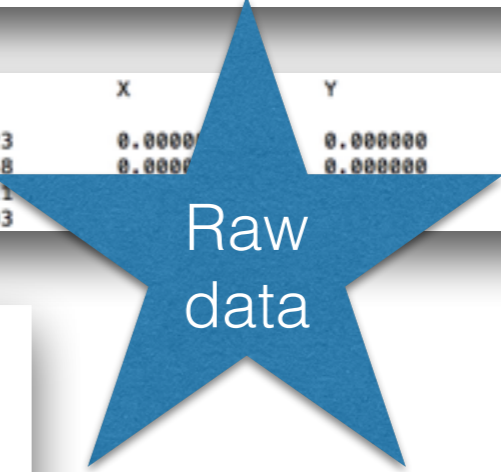


Textual User Interface

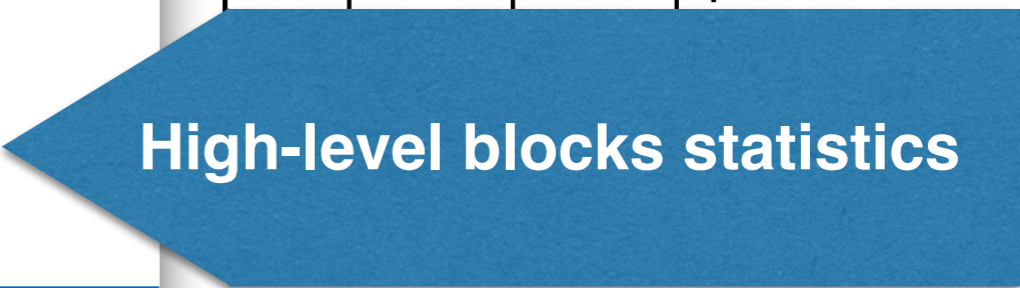
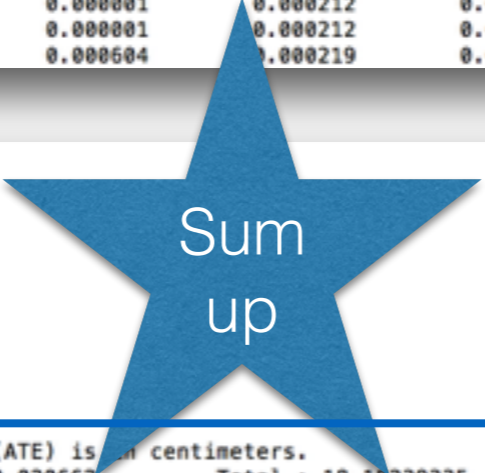


```
hickory% make 2.openc1.log
LD_PRELOAD=./build/kfusion/thirdparty/liboclwrapper.so ./build/kfusion/kfusion-benchmark-openc1 -s 4.8 -p 0.34,0.5,0.24 -z 4 -c 2 -r 2 -k 481.2,480,320,240 -i living_room_traj2_
encl.log 2> oclwrapper.2.openc1.log
End of file(garbage found).
cat oclwrapper.2.openc1.log |grep -E ".+ [0-9]+ [0-9]+ [0-9]+" |cut -d" " -f1,4 > kernels.2.openc1.log
./kfusion/thirdparty/checkPos.py benchmark.2.openc1.log livingRoom2.gt.freiburg > resume.2.openc1.log
./kfusion/thirdparty/checkKernels.py kernels.2.openc1.log >> resume.2.openc1.log
hickory%
```

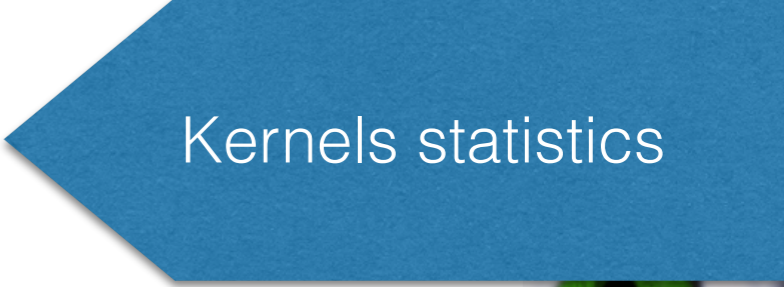
frame	acquisition	preprocessing	tracking	integration	raycasting	rendering	computation	total	X	Y	Z	tracked	inte
0	0.003564	0.000451	0.001622	0.000461	0.000002	0.001124	0.002536	0.007223	0.0000	0.000000	0.000000	0	1
1	0.002724	0.000352	0.000903	0.000467	0.000001	0.000212	0.001722	0.004658	0.0000	0.000000	0.000000	0	1
2	0.002697	0.000336	0.000904	0.000459	0.000001	0.000212	0.001701	0.004611	0.0000	0.000000	0.000000	0	1
3	0.002663	0.000335	0.000905	0.000467	0.000604	0.000219	0.002311	0.005193	0.0000	0.000000	0.000000	0	1



```
1 Get kfusion output data.
2 Skip kfusion line :
3 Skip num line :
4 kfusion file: Valid frames 882 dropped frames: 0
5 kfusion result : 882 positions.
6 NUIM result : 880 positions.
7 Working position is : 880
8 Untracked frames: 0
9 Shift kfusion trajectory...
10
11 A detailed statistical analysis is provided.
12 All durations are in seconds and the absolute trajectory error (ATE) is in centimeters.
13 ATE Min : 0.000000 Max : 0.049309 Mean : 0.020662 Total : 18.18239335
14 acquisition Min : 0.000056 Max : 0.009033 Mean : 0.002044 Total : 1.80289800
15 computation Min : 0.001701 Max : 0.009234 Mean : 0.005894 Total : 5.19872500
16 integration Min : 0.000001 Max : 0.000821 Mean : 0.000258 Total : 0.22773500
17 preprocessing Min : 0.000284 Max : 0.001884 Mean : 0.000441 Total : 0.38904600
18 raycasting Min : 0.000001 Max : 0.003313 Mean : 0.001345 Total : 1.18644400
19 rendering Min : 0.000201 Max : 0.003452 Mean : 0.000572 Total : 0.50420900
20 total Min : 0.003882 Max : 0.020054 Mean : 0.008510 Total : 7.50584700
21 tracking Min : 0.000903 Max : 0.006743 Mean : 0.003850 Total : 3.39543600
22 Get SlamBench data.
```



23	ResetVolume	Count : 1	Min : 401056	Max : 401056	Mean : 401056.000000	Total : 401056
24	bilateral_filter	Count : 882	Min : 139520	Max : 205632	Mean : 159880.199546	Total : 141014336
25	depth2vertex	Count : 2646	Min : 5440	Max : 48384	Mean : 10585.167045	Total : 28008352
26	halfSampleRobust	Count : 1764	Min : 6624	Max : 328480	Mean : 9546.557823	Total : 16840128
27	integrate	Count : 443	Min : 270880	Max : 793664	Mean : 485648.469526	Total : 215142272
28	mm2meters	Count : 882	Min : 6912	Max : 13632	Mean : 8977.306122	Total : 7917984
29	raycast	Count : 879	Min : 435584	Max : 3284768	Mean : 1323028.423208	Total : 1162941984
30	reduce	Count : 12117	Min : 31488	Max : 523488	Mean : 138305.687546	Total : 1675850016
31	renderDepth	Count : 882	Min : 9568	Max : 39904	Mean : 12002.975057	Total : 10586624
32	renderTrack	Count : 882	Min : 16608	Max : 33088	Mean : 18716.081633	Total : 16507584
33	renderVolume	Count : 221	Min : 479840	Max : 3143200	Mean : 1338055.819005	Total : 295710336
34	track	Count : 12117	Min : 13472	Max : 129344	Mean : 51505.231988	Total : 624088896
35	vertex2normal	Count : 2646	Min : 8544	Max : 73504	Mean : 22275.362056	Total : 58940608



Copyrights

- Author: unknown. Microsoft Kinect camera. [Image]. Retrieved from <http://channel9.msdn.com/Series/KinectSDKQuickstarts/Understanding-Kinect-Hardware>
- Author: Dyson Ltd. Dyson 360 Eye. [Video]. Retrieved from <https://www.youtube.com/watch?v=OadhulCDAjk>
- Author: Google Inc. Google Tango project. [Image]. Retrieved from <http://blogthinkbig.com/en/project-tango-googles-mobile-kinect/>
- Author: unknown. Audi autonomous car. [Photograph]. Retrieved from <http://www.wired.com/2010/06/audis-robotic-car-looks-hot-in-old-school-livery/>
- Author: ExtremeTech. Google Shaft robot. [Photograph]. Retrieved from <http://www.extremetech.com/extreme/173318-google-wins-darpa-robotics-challenge-wonders-if-it-was-a-good-idea-to-turn-down-future-military-contracts>
- Author: HardKernel. ODROID-XU3 board. [Photograph]. Retrieved from http://www.hardkernel.com/main/products/prdt_info.php?g_code=G135235611947
- Author: PC Specialist Ltd. Vortex series laptop. [Photograph]. Retrieved from <https://www.pcspecialist.co.uk/forums/showthread.php?23366-My-new-beast-15-6-quot-Vortex-III>
- Author: Arndale.org. Arndale board. [Photograph]. Retrieved from http://www.arndaleboard.org/wiki/index.php/Main_Page
- Author: Unknown. Chip. [Image]. Retrieved from <https://cajalesygalileos.wordpress.com/2013/06/23/un-chip-ultrasensible-identifica-15-cepas-de-gripe/>
- Author: Unknown. Eye. [Image]. Retrieved from <http://gallery.digitalculture.asu.edu/?/interactive-environments/computer-vision/>
- Author: Unknown. Compiler. [Image]. Retrieved from <http://d3q6qq2zt8nhwv.cloudfront.net/107/large-icon.png>

