Holistic benchmarking in 3D Robot Vision: the SLAMBench open source framework

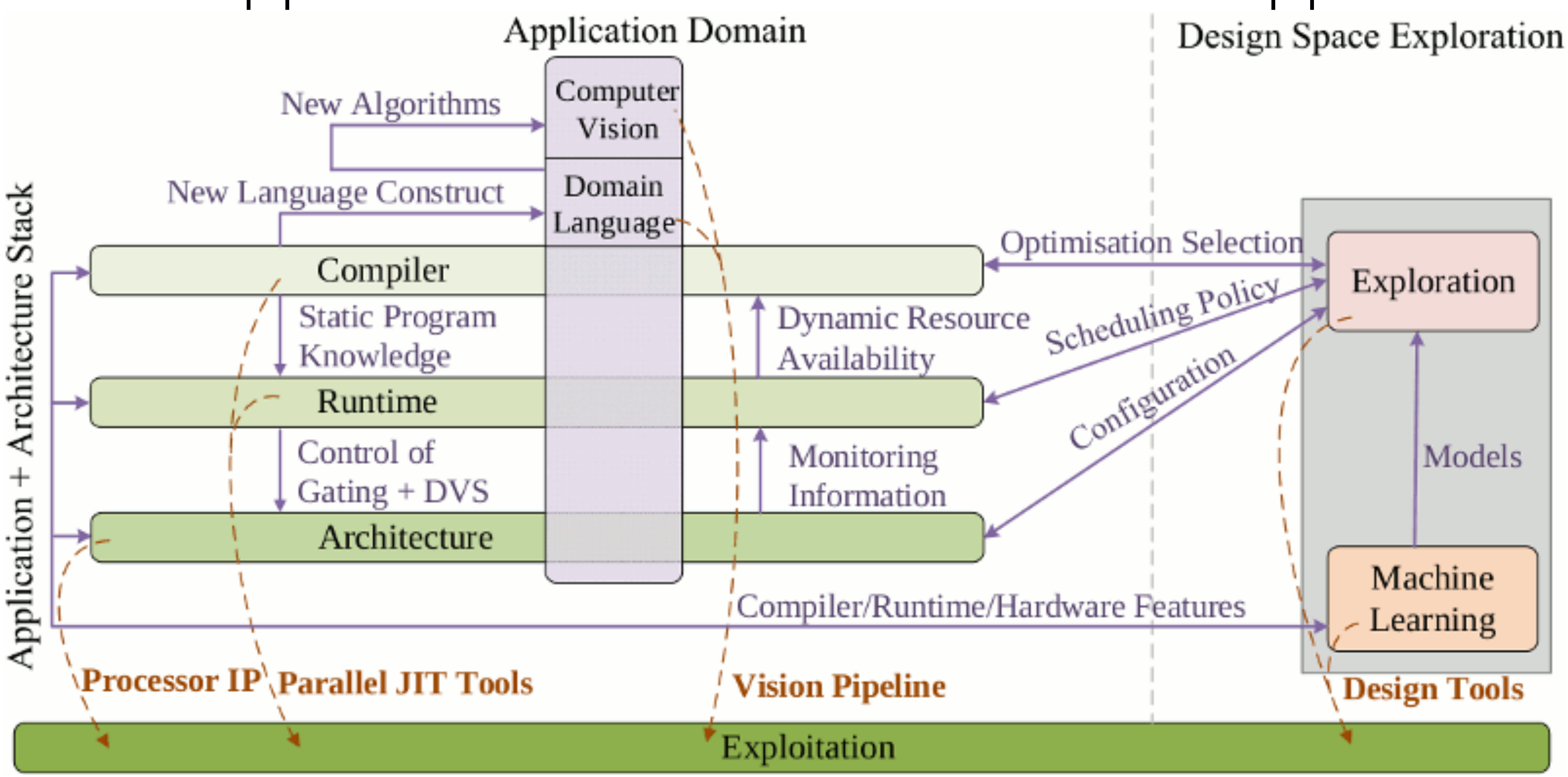
Luigi Nardi, PhD
Software Performance Optimisation group
@CSW Porto
April 21st 2016

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

MANCHESTER 1824 The University of Manchester
The University of Edinburgh
Imperial College London
EPSRC Engineering and Physical Sciences Research Council
PAMELA

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, M. Luján, A. J. Davison, G. Riley, N. Topham and S. Furber

MANCHESTER 1824 The University of Manchester

The University of Edinburgh

Imperial College London

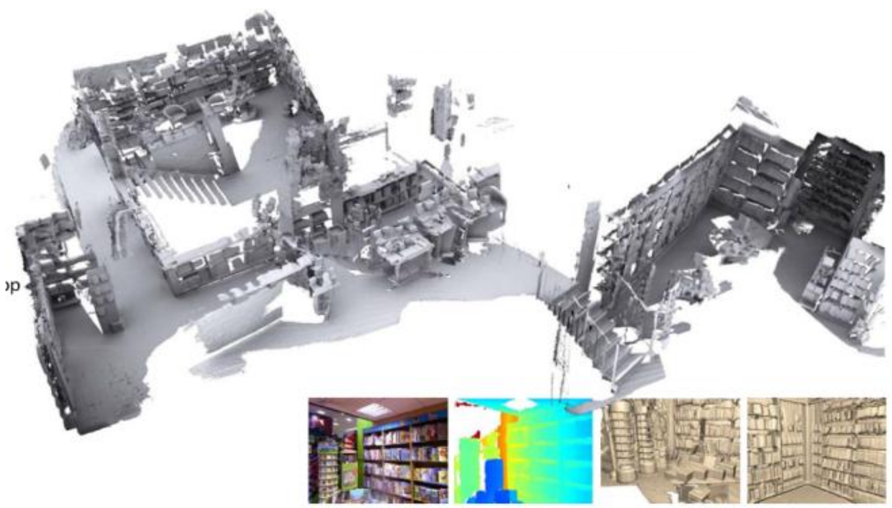
EPSRC Engineering and Physical Sciences Research Council

PAMELA

The three R's of vision: Spectrum of Computer Vision Research

Reconstruction

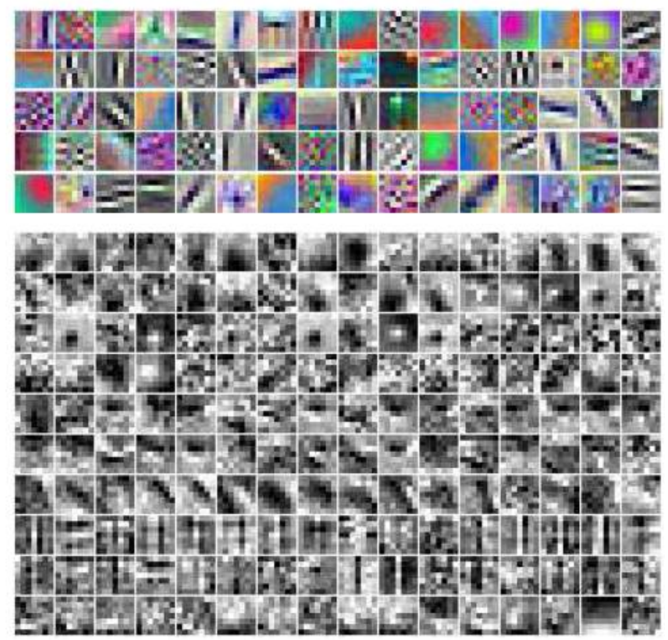
Scalable Kinect Fusion (2013)

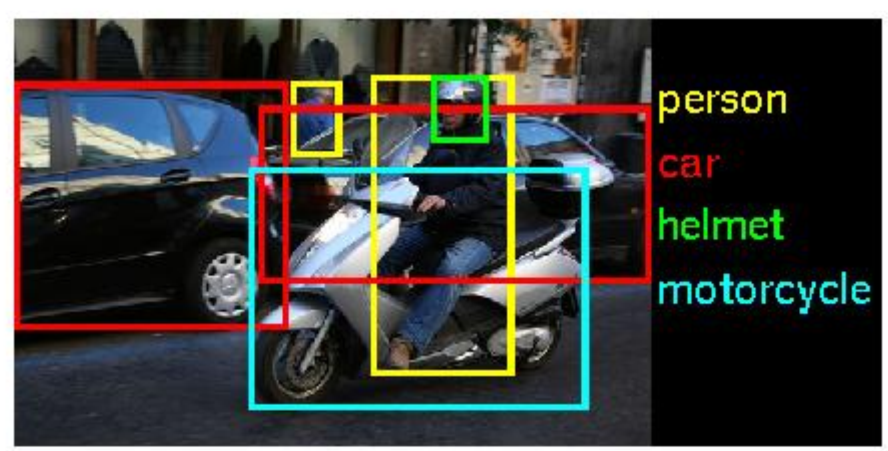


Building Rome on a cloudless day (2010)

Recognition

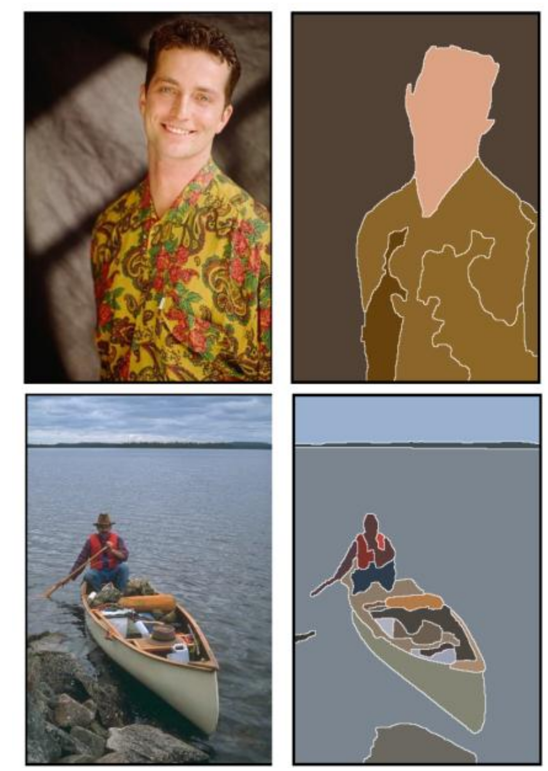
Deep learning for scalable Object class detection (2014)





Reorganisation or Grouping

Contour detection and segmentation (2011)

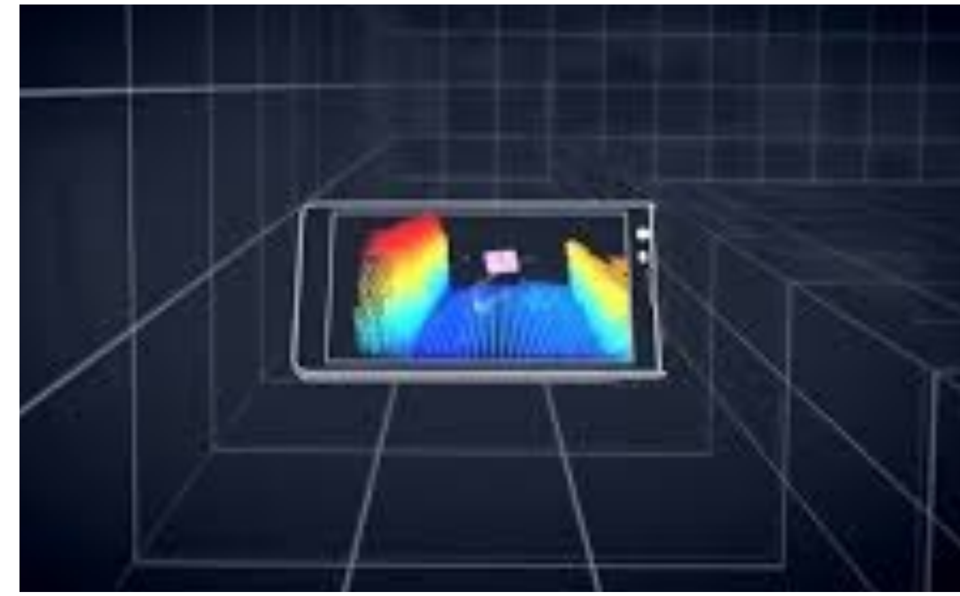


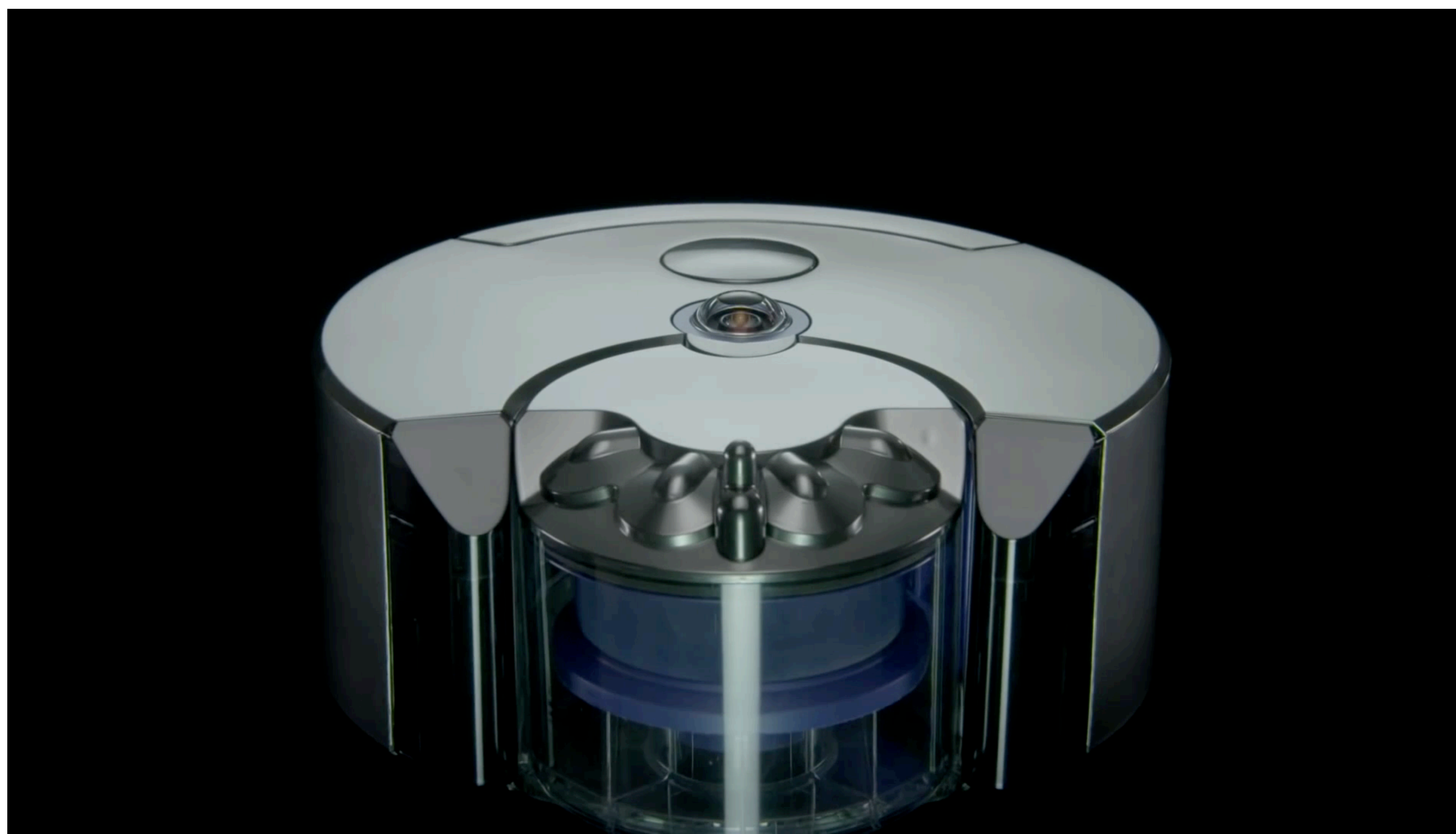
Credit: Zeeshan Zia

Luigi Nardi - Imperial College London

Simultaneous localisation and mapping (SLAM)

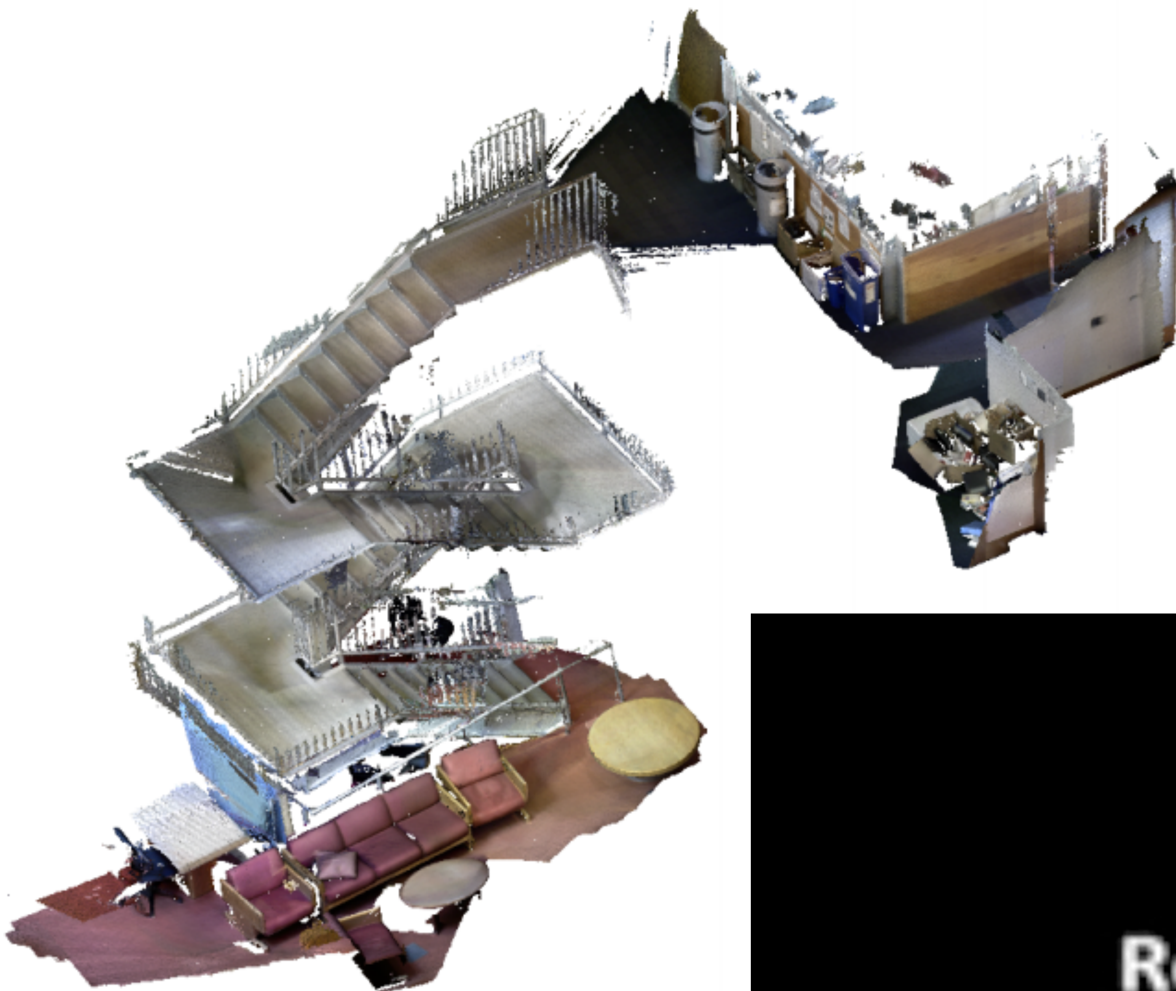
Build a coherent world representation and localise the camera in real-time



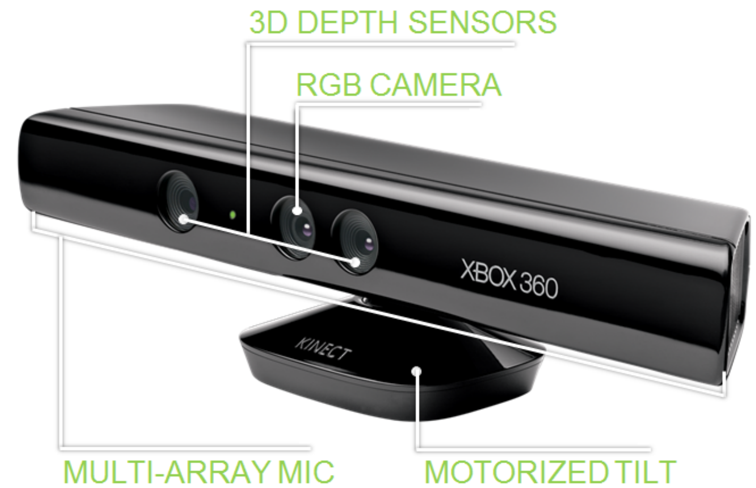


Video:
Dyson 360 Eye

Simultaneous localisation and mapping (SLAM)



[Whelan et al. 2012]

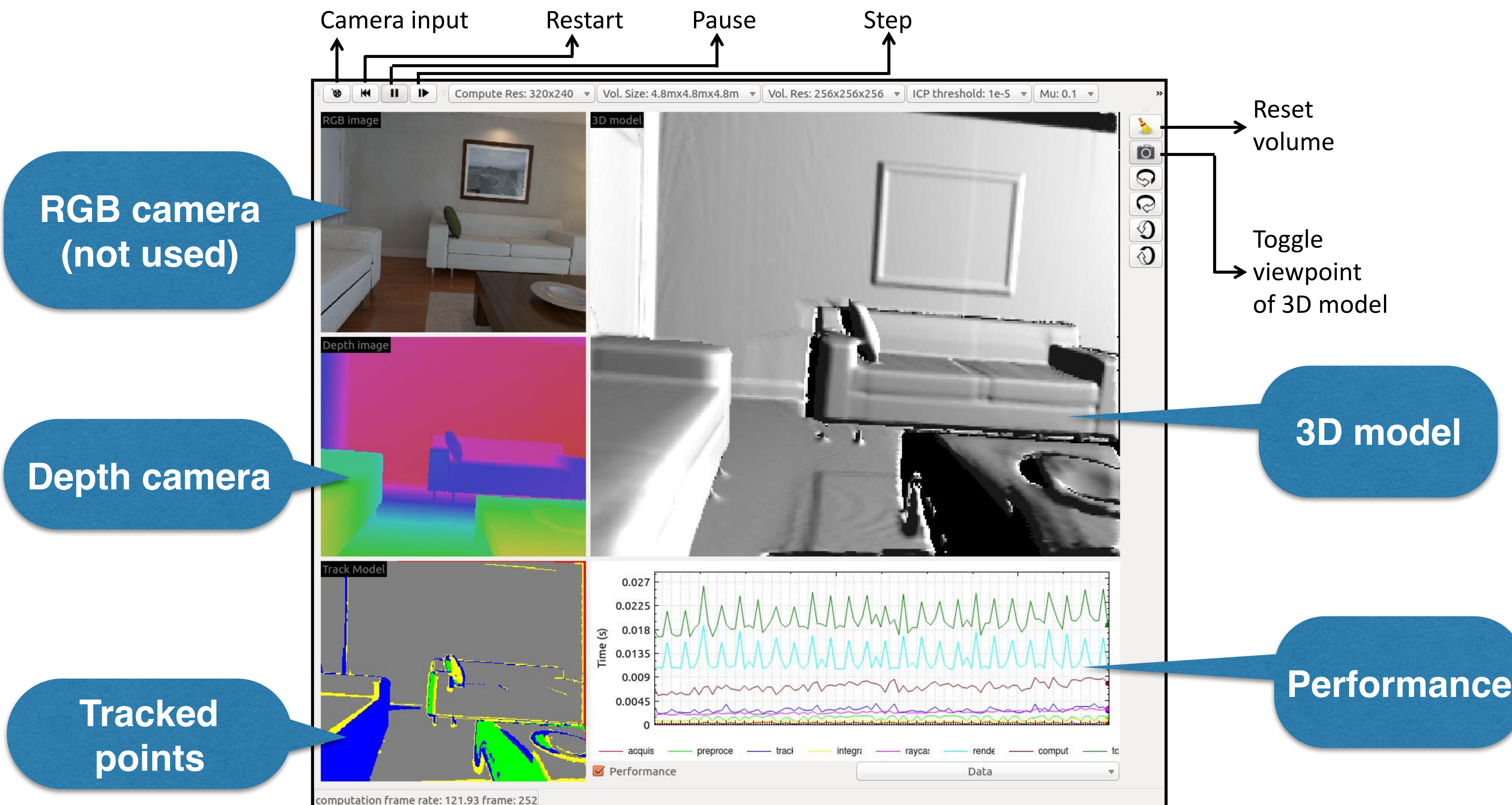


Video:
[Newcombe et al. ISMAR 2011]

Demo: 3D reconstruction of Albert

This slide has been added after the presentation and shows the live demonstration of the SLAMBench framework performed (see next slide for explanation)

SLAMBench GUI



What is "Performance"?

1. In several domains performance is **execution time**
2. In some domains performance is **accuracy**
3. What about **energy**?
4. But also memory consumption, temperature, robustness, etc.

<u>A modern system evaluation considers multiple metrics</u>:

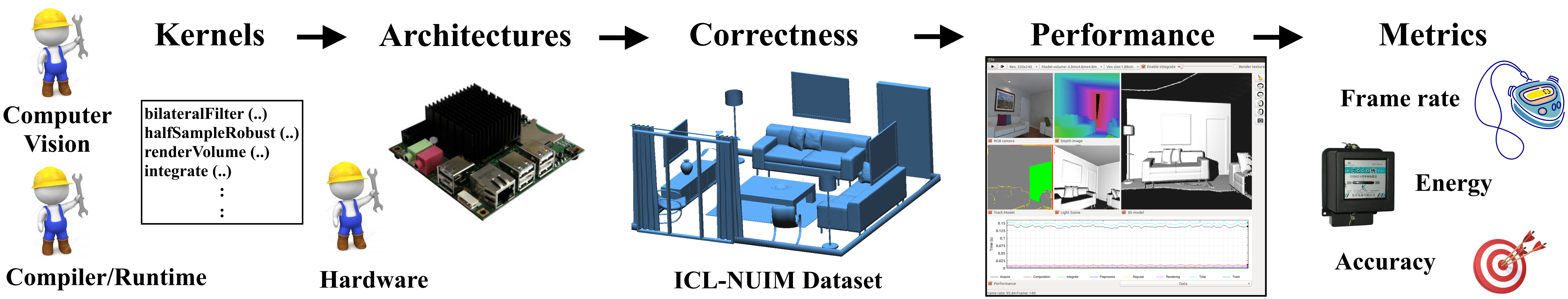
$$Performance = \begin{bmatrix} Runtime \\ Energy \\ Accuracy \end{bmatrix}$$

This defines a <u>multi-objective optimisation</u> problem: trade-off

Three “Performance” metrics

Holistic approach to SLAM “performance”:

SLAMBench



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

How to measure SLAM "Performance"?

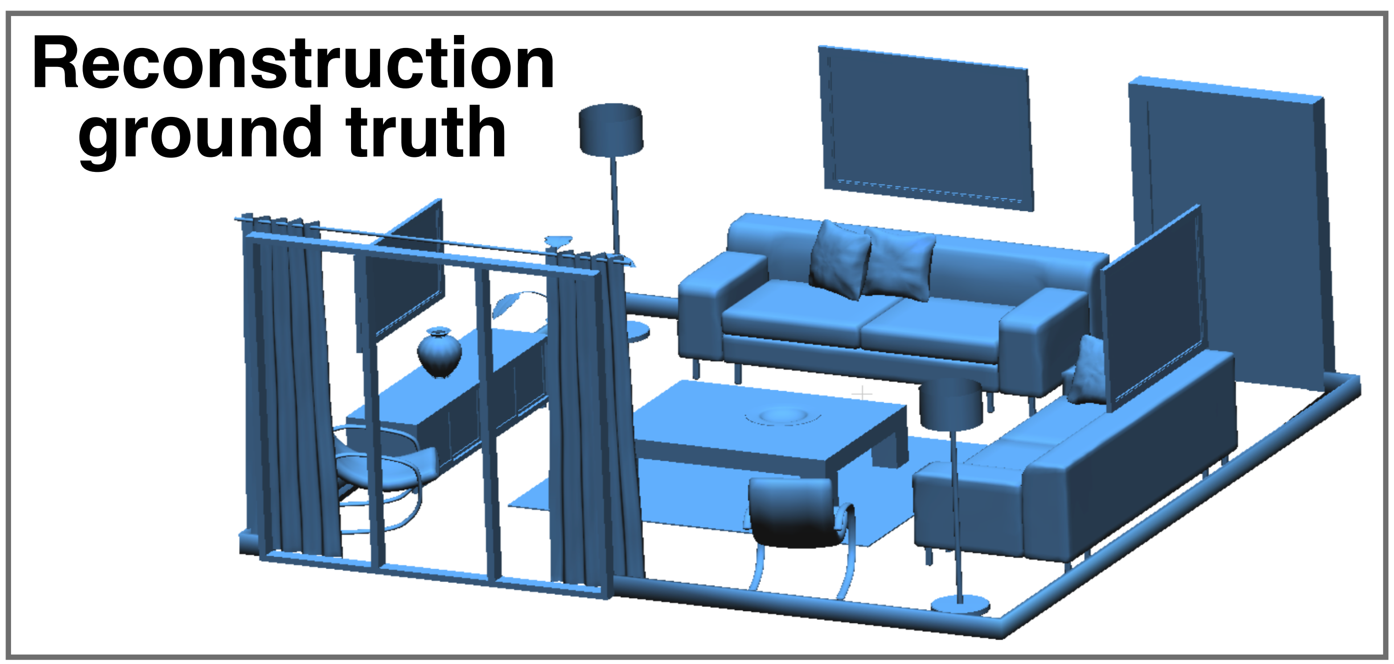
SLAM computation depends on:

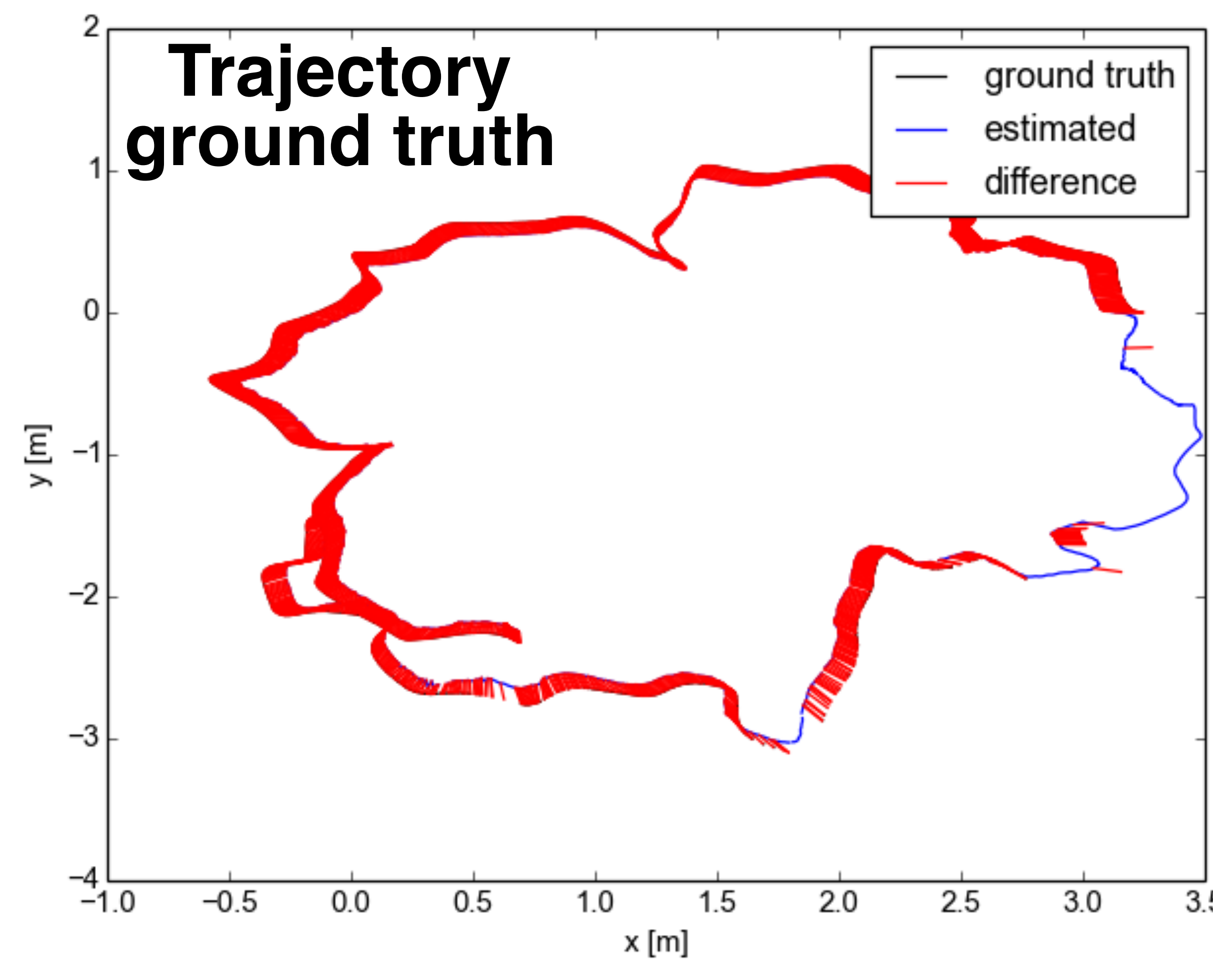
- Images acquired
- Way the camera is moved
- Numerical approximations
- Processing frame rate
  (<u>depends on hardware capability</u>)

Need for reproducibility and accuracy check
Pre-recorded scenes
Process-every-frame mode
ICL-NUIM and TUM RGB-D datasets

ICL-NUIM dataset

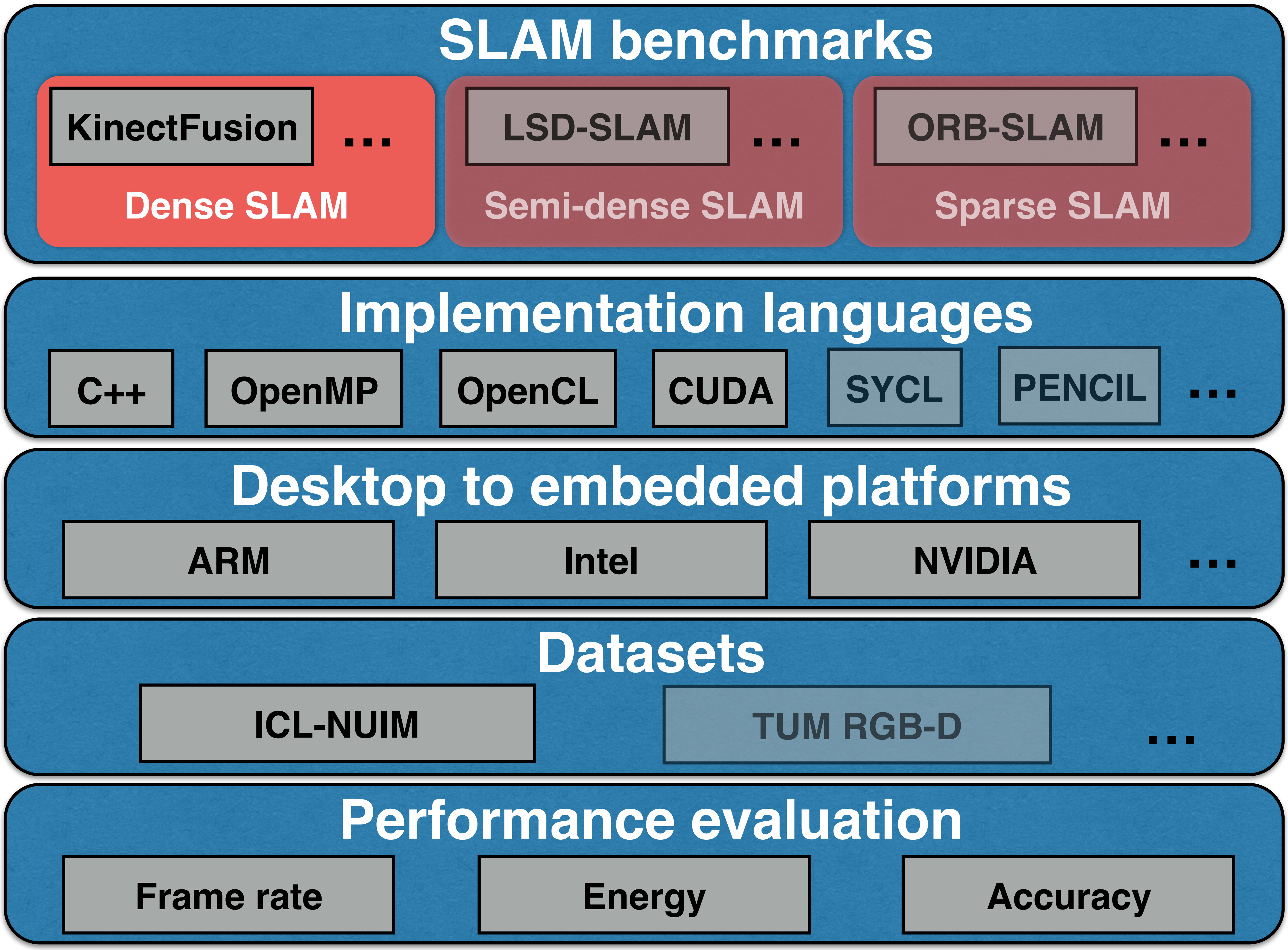
Luigi Nardi - Imperial College London





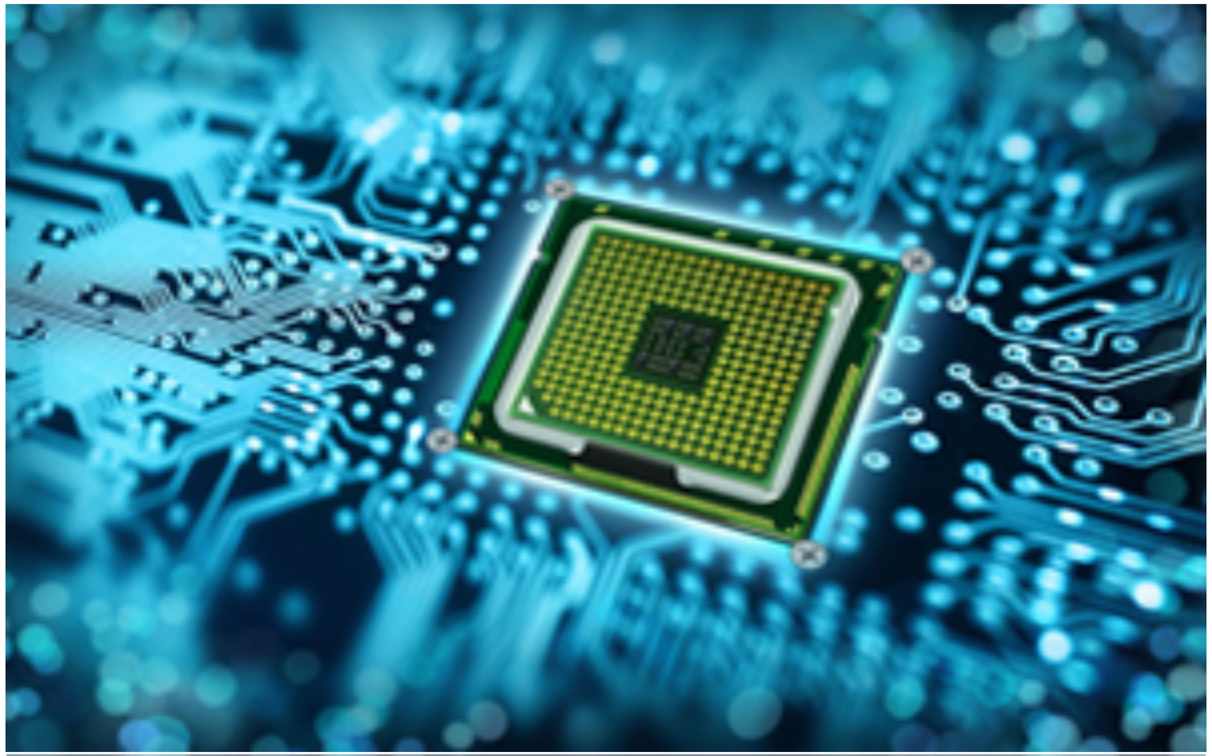
- ICL-NUIM synthetic dataset [Handa et al. 2014]
- 880 RGB-D frames at 30 FPS
- Absolute trajectory error (ATE) based on ground truth

SLAMBench framework



[Nardi et al., 2015]

SLAMBench opportunities



Chip design and simulation tools

SLAMBench opportunities

SLAMBench opportunities

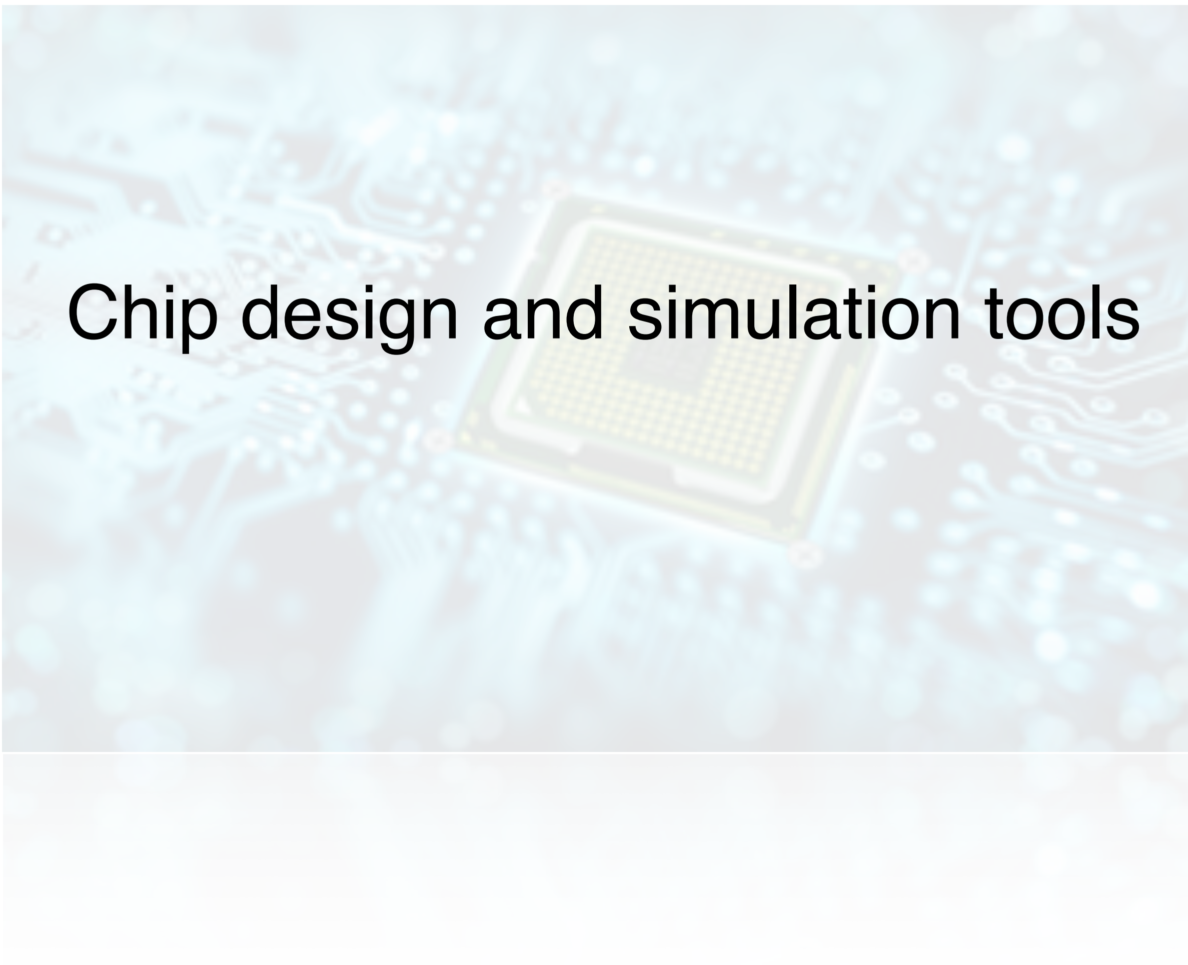
Chip design and simulation tools

SLAMBench evolution:

- Point fusion
- Octrees
- Semi-dense SLAM
- Feature-based SLAM

Kernels can be improved individually

SLAMBench opportunities



SLAMBench evolution:

- Point fusion
- Octrees
- Semi-dense SLAM
- Feature-based SLAM

Kernels can be improved individually

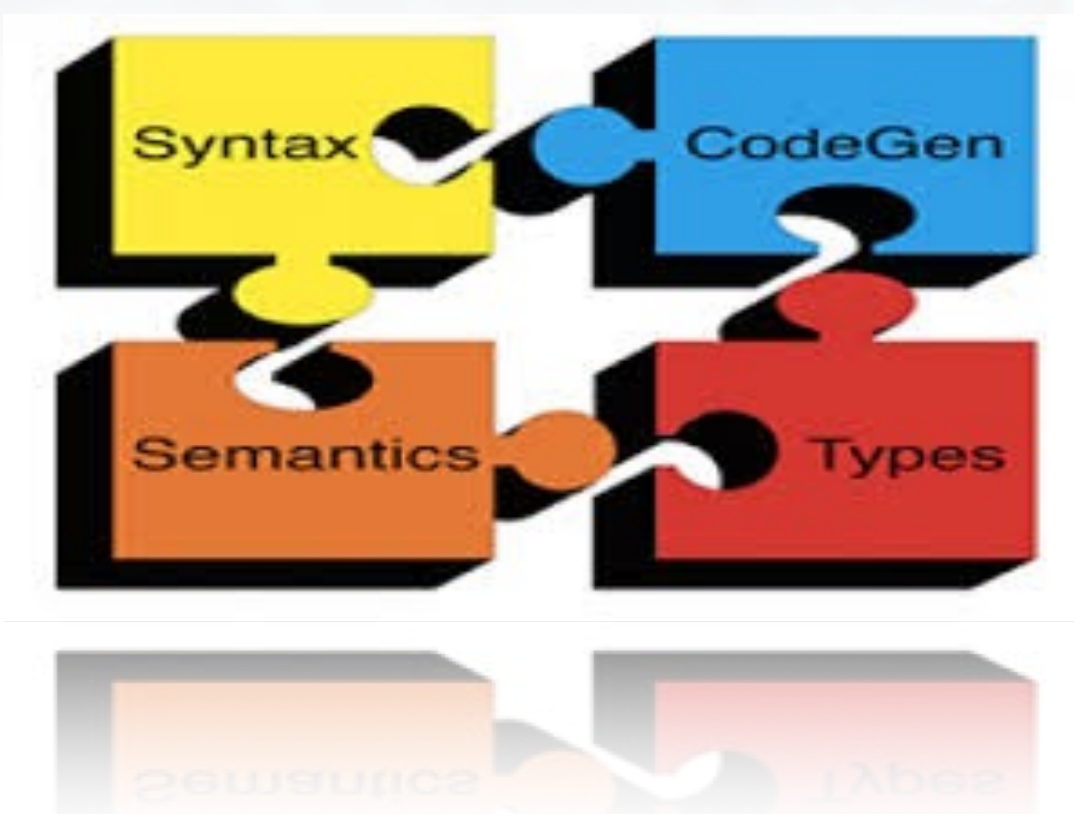
SLAMBench opportunities

Chip design and simulation tools

SLAMBench evolution:

- Point fusion
- Octrees
- Semi-dense SLAM
- Feature-based SLAM

Kernels can be improved individually



Domain-specific language (DSL) targeting high performance, low-power solutions

Design-space exploration, e.g. algorithmic, compiler and hardware parameters

SLAMBench kernels tuning, e.g. vectorisation, GPU, auto-tuning

SLAMBench opportunities

Chip design and simulation tools

Domain-specific language (DSL) targeting high performance, low-power solutions

Design-space exploration, e.g. algorithmic, compiler and hardware parameters

SLAMBench kernels tuning, e.g. vectorisation, GPU, auto-tuning

SLAMBench evolution:

- Point fusion
- Octrees
- Semi-dense SLAM
- Feature-based SLAM

Kernels can be improved individually

SLAMBench opportunities

Chip design and simulation tools

SLAMBench evolution:

- Point fusion
- Octrees
- Semi-dense SLAM
- Feature-based SLAM

Kernels can be improved individually

Domain-specific language (DSL) targeting high performance, low-power solutions

Design-space exploration, e.g. algorithmic, compiler and hardware parameters

SLAMBench kernels tuning, e.g. vectorisation, GPU, auto-tuning

SLAMBench opportunities

Chip design and simulation tools

SLAMBench evolution:

- Point fusion
- Octrees
- Semi-dense SLAM
- Feature-based SLAM

Kernels can be improved individually

Domain-specific language (DSL) targeting high performance, low-power solutions

Design-space exploration, e.g. algorithmic, compiler and hardware parameters

- CPU/GPU mapping/partitioning
- Just-in-time compilation

SLAMBench kernels tuning, e.g. vectorisation, GPU, auto-tuning

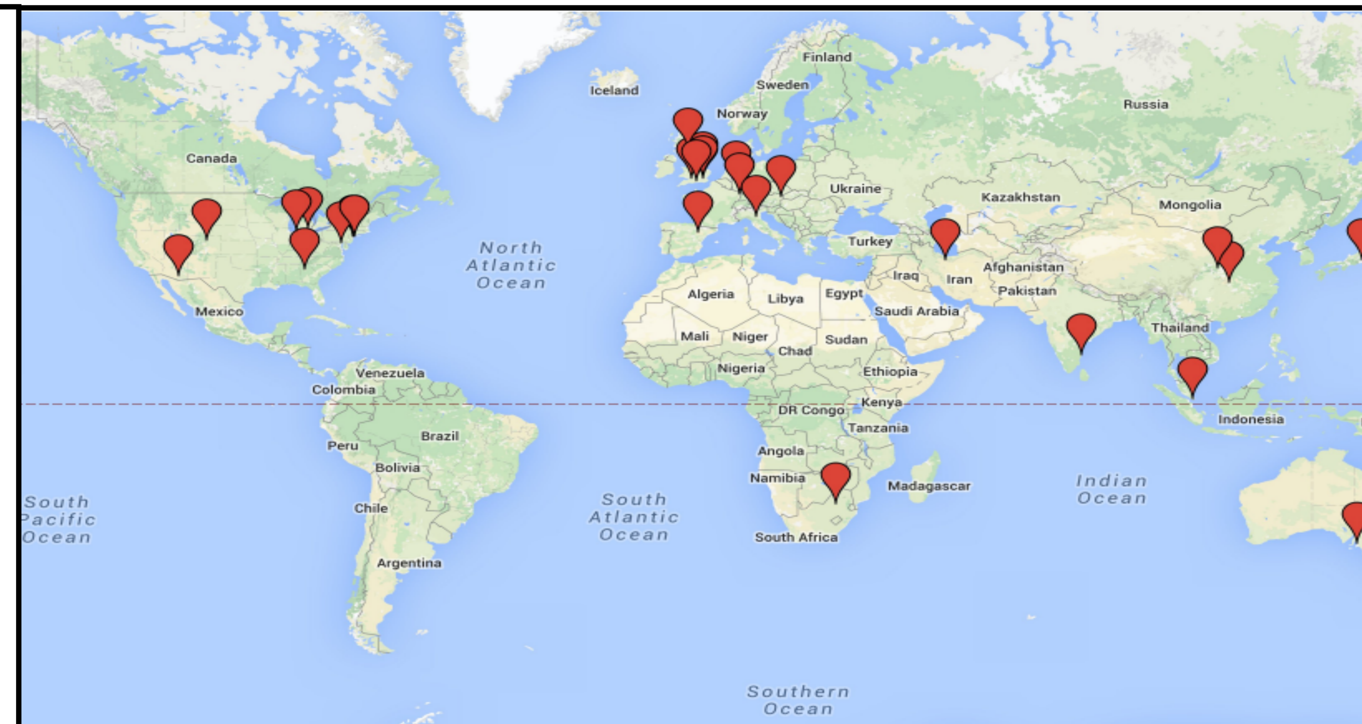
SLAMBench today

- Publicly released 13/11/2014 (1200+ downloads)

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

Company users

Academic users



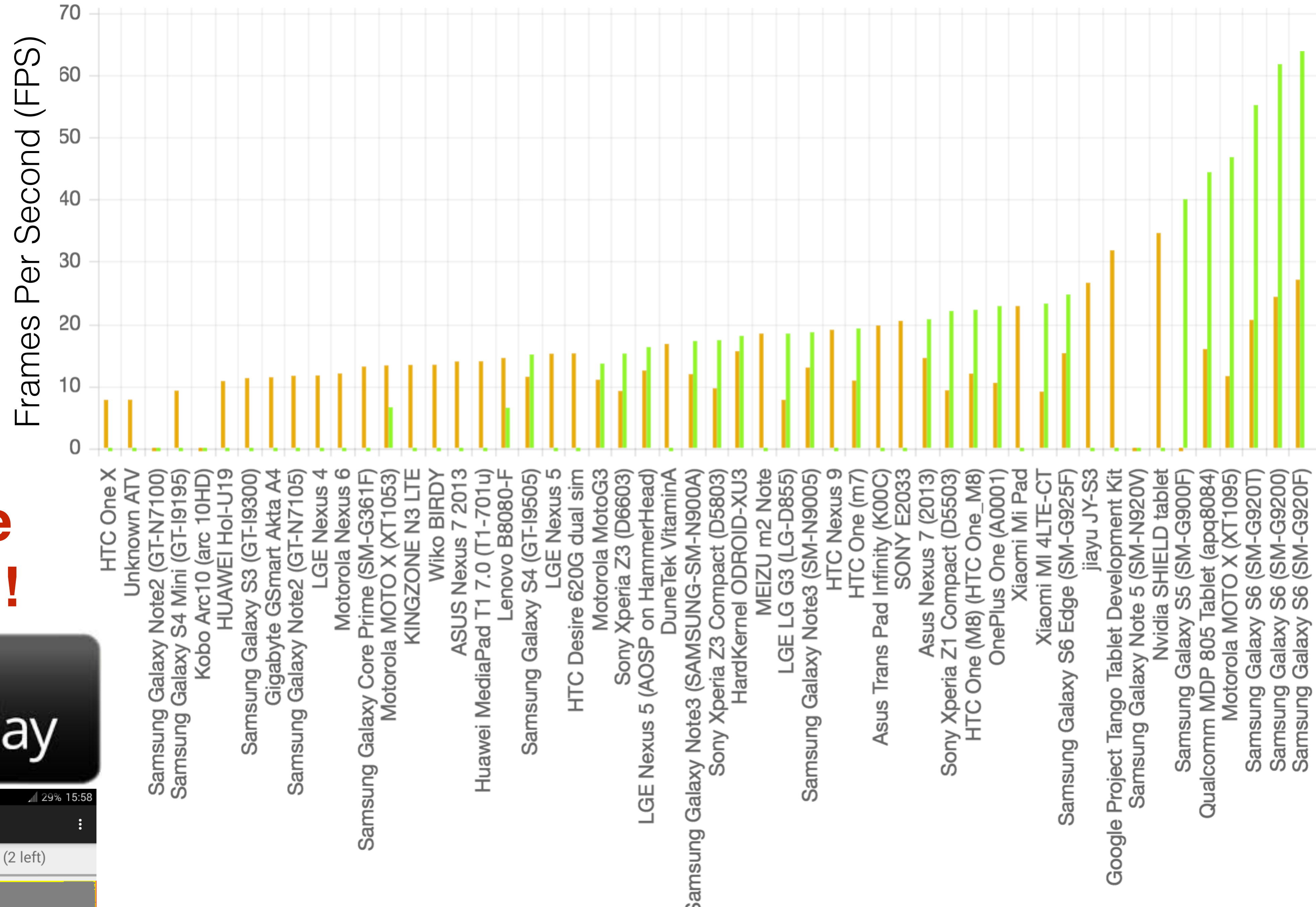
Web: apt.cs.manchester.ac.uk/projects/PAMELA/tools/SLAMBench/

- ***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

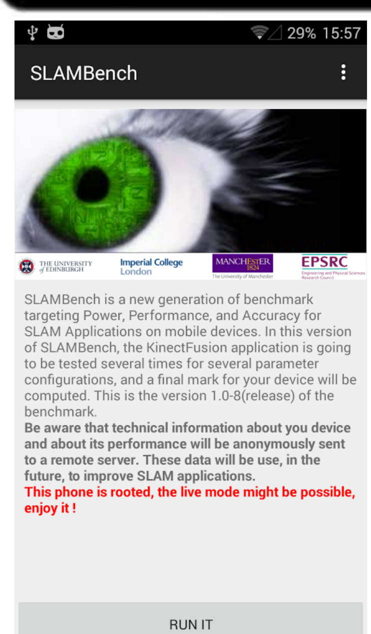
Crowdsourcing mobile Android SLAMBench

- SLAMBench OpenMP
- SLAMBench OpenCL



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

Get it on Google play



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

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

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

Space 2

2. Compilation:
   - 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

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

Space 2

2. Compilation:
    - 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

3. Architecture:
    - 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

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

Space 2

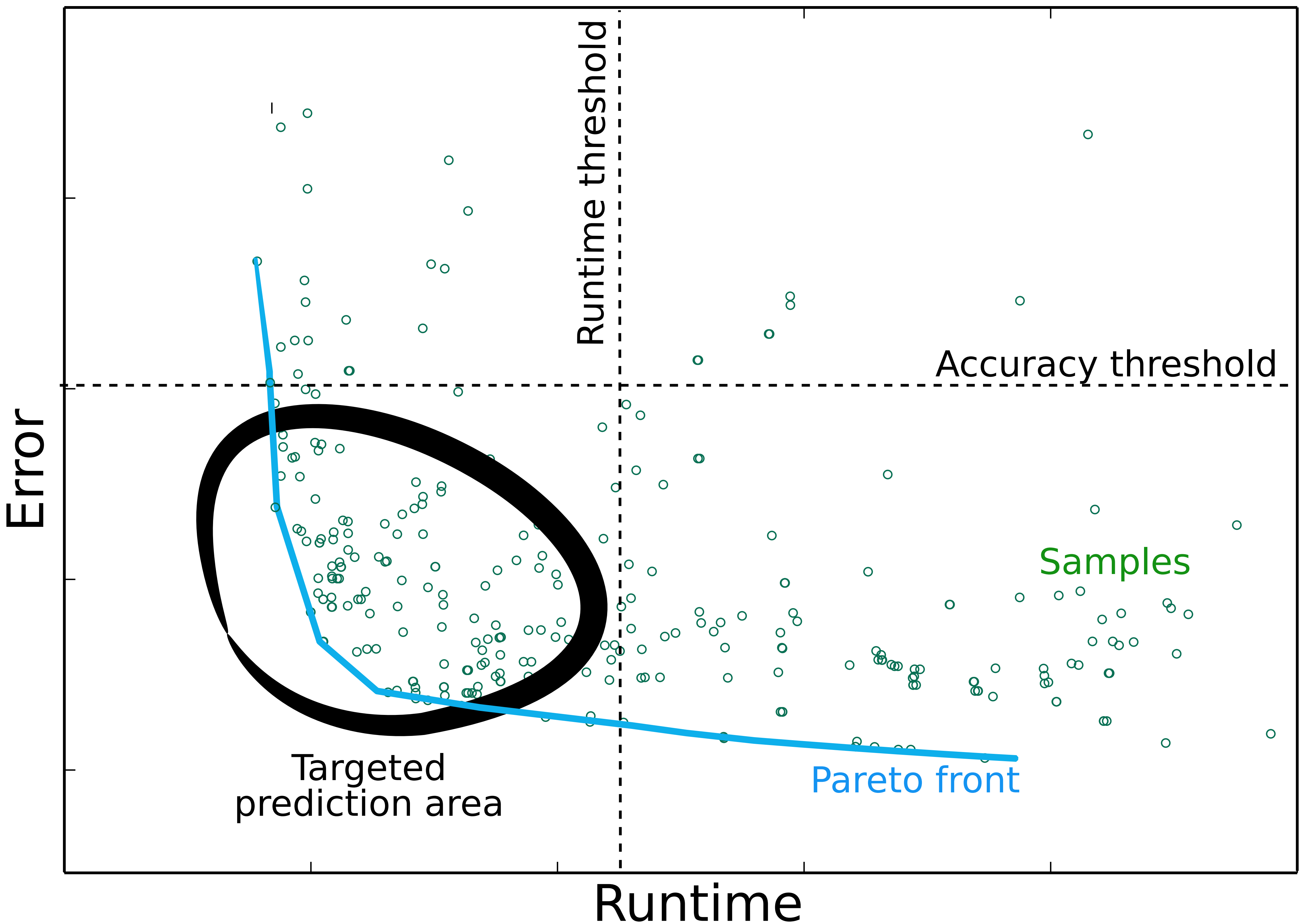
2. Compilation:
   - opencl-params: -cl-mad-enable,-cl-fast-relaxed-math, etc.
   - LLVM flags: O1, O2, O3, vectorize-slp-aggressive, etc.
   - Local work group size: 16/32/64/96/112/128/256
   - Vectorisation: width (1/2/4/8), direction (x/y)
   - Thread coarsening: factor (1/2/4/8/16/32), stride (1/2/4/8/16/32), dimension (x/y)

Space 3

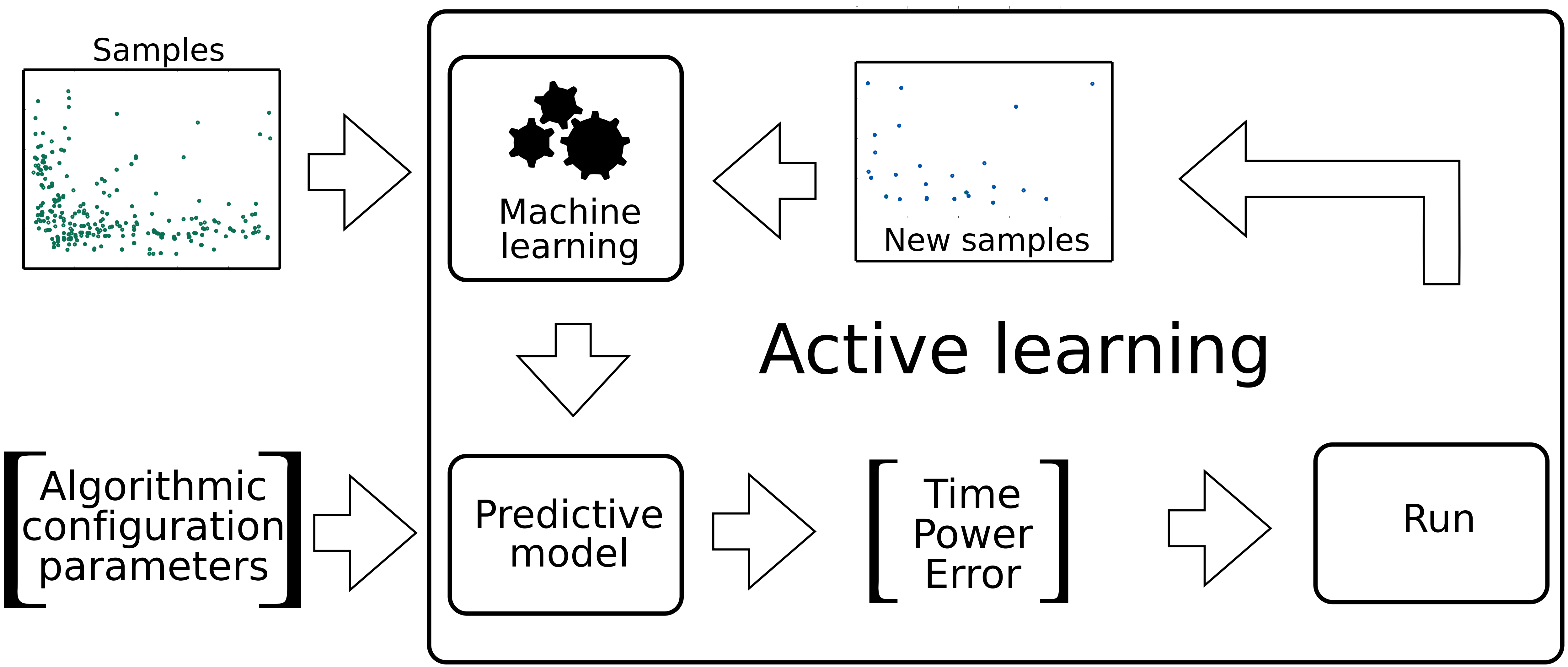
3. Architecture:
   - GPU frequency: 177/266/350/420/480/543/600/DVFS
   - # of active big cores: 0/1/2/3/4
   - # of active LITTLE cores: 1/2/3/4

Warning: huge spaces, impossible to run exhaustively

Exploration goal

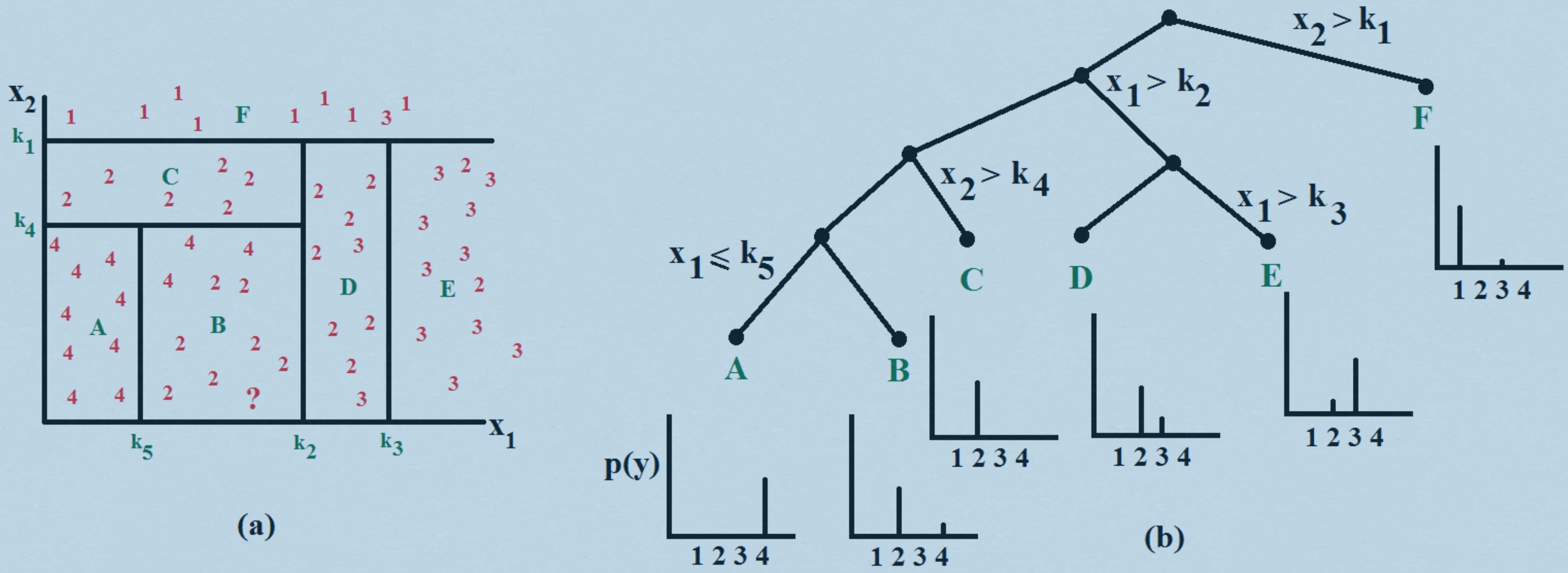


Algo design-space exploration (DSE)

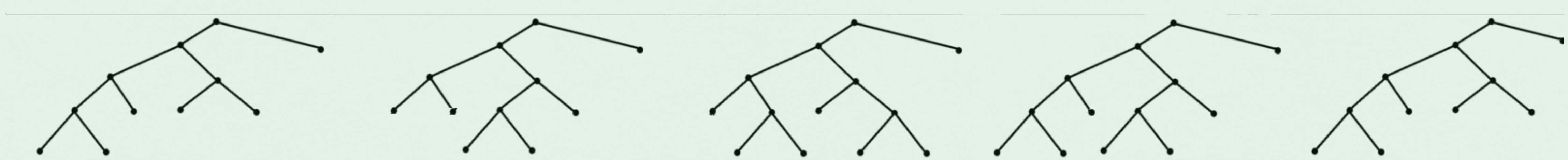


Machine learning methods used

Decision tree



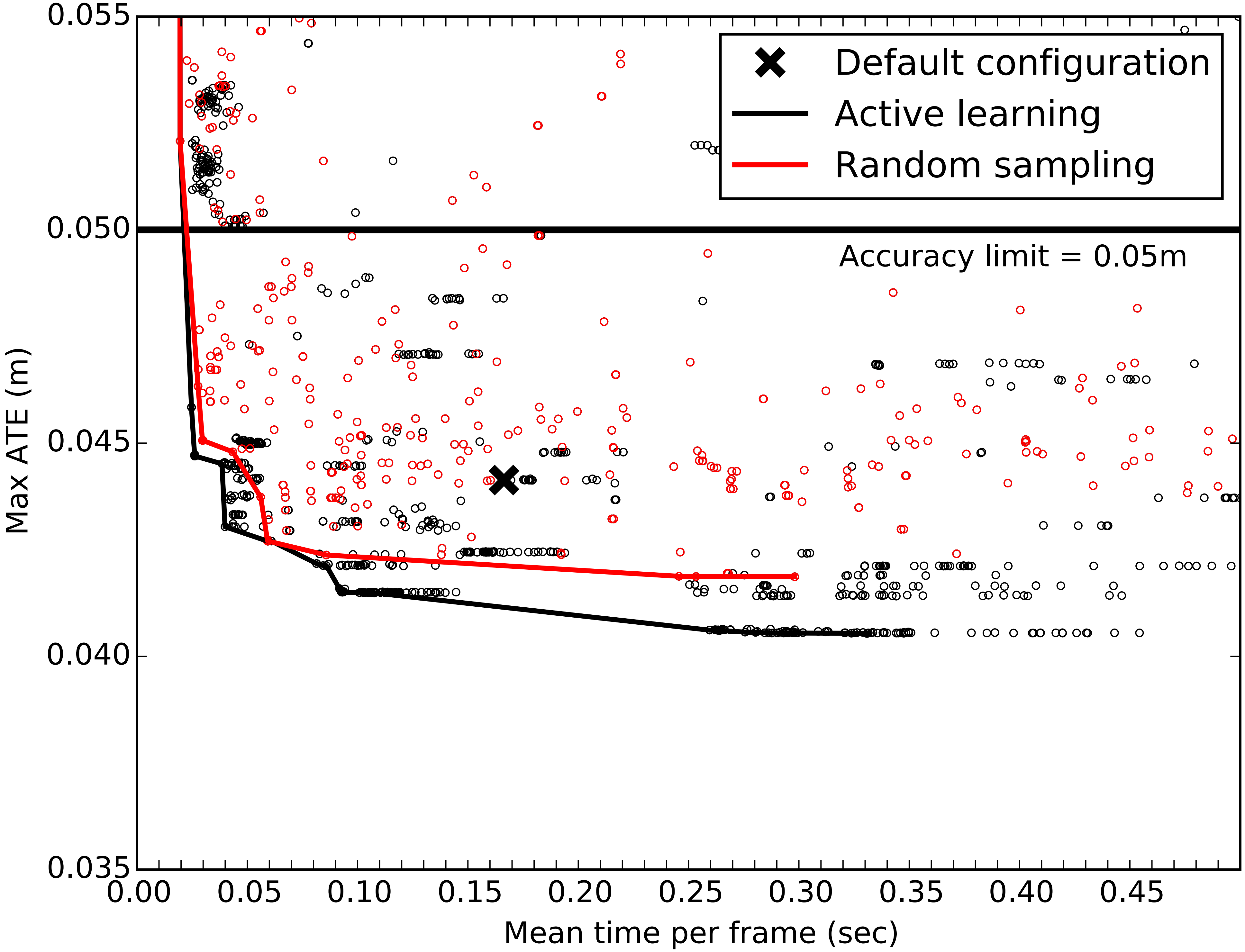
Random forest

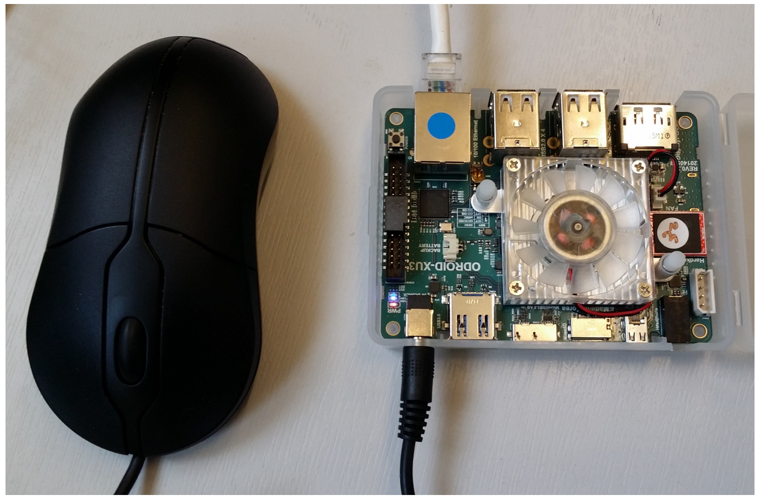


Luigi Nardi - Imperial College London

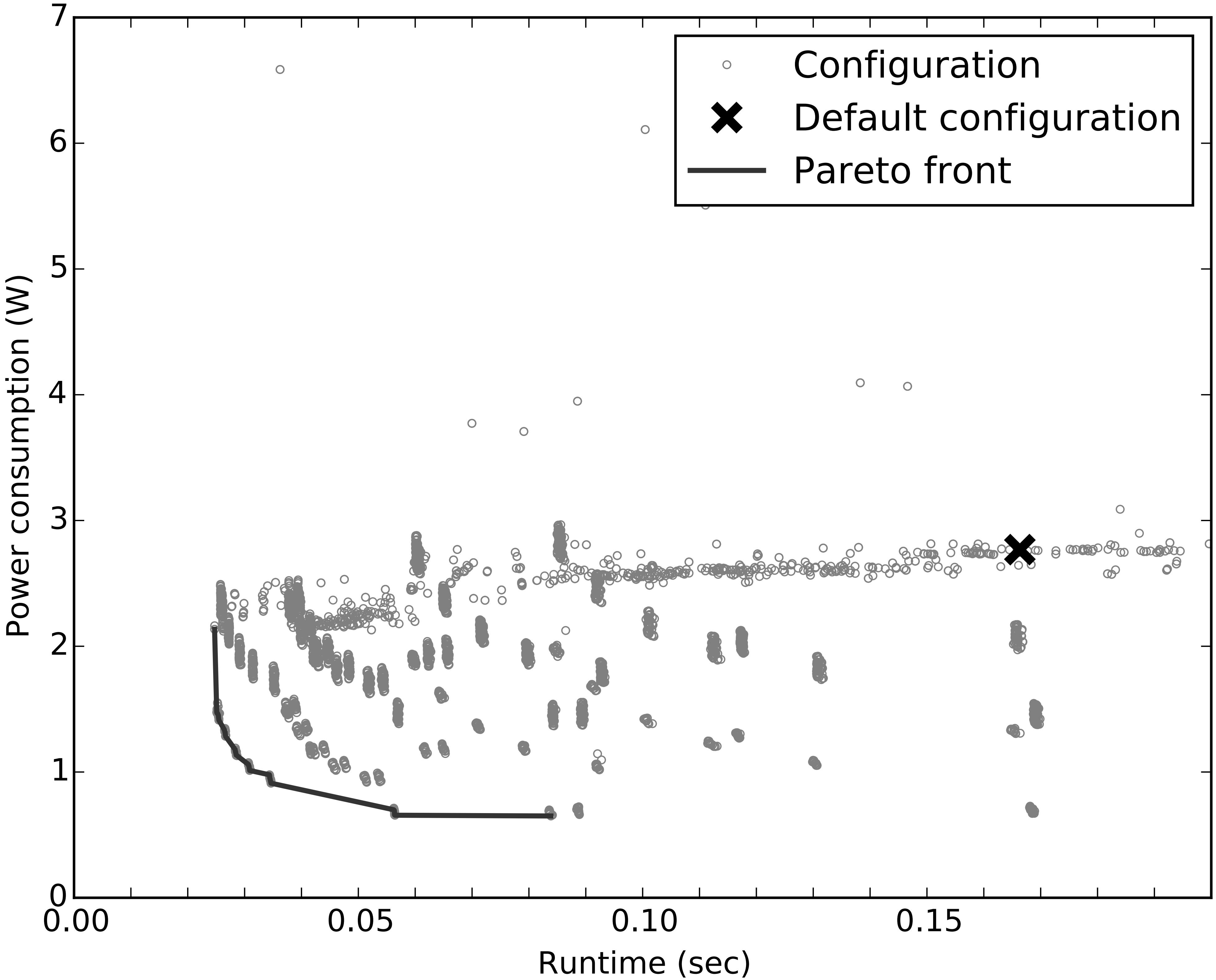
DSE on algorithmic parameters error/runtime

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





DSE architecture parameters power/runtime



DSE final results

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

- 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

SLAMBench vs other benchmarking frameworks: should HPC follow this model?

- Multi-objective optimisation: frame rate/power/accuracy
- Semantic accuracy check is very powerful:
  - enables non bit-wise accuracy check
  - scope for aggressive approx. computing and auto-tuning
- Several datasets:
  - not just 1 "big" and 1 "small"
- Multi-kernel benchmark:
  - enables benchmarking complex frameworks
- Multiple applications from the same domain (SLAM)
  - this is coming …

References I

- [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.
- [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.
- [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.
- [Chen et al. 2013] J. Chen, D. Bautembach, and S. Izadi, Scalable real-time volumetric surface reconstruction, in ACM Trans. Graph., 2013.
- [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.
- [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.
- [Reitmayr] G. Reitmayr. KFusion github 2011. https://github.com/GerhardR/kfusion
- [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.
- [Whelan et al. 2012] T. Whelan, M. Kaess, M. Fallon,H. Johannsson, J. Leonard, and J. McDonald. Kintinuous: Spatially extended kinectfusion. 2012.
- C. Jiawen, D. Bautembach, and S. Izadi. "Scalable real-time volumetric surface reconstruction." ACM TOG, 2013.
- Frahm, Jan-Michael, et al. "Building Rome on a cloudless day." Computer Vision–ECCV 2010. Springer Berlin Heidelberg, 2010.
- Erhan, Dumitru, et al. "Scalable object detection using deep neural networks." Proceedings of the IEEE CVPR. 2014.

References II

- Arbelaez, Pablo, et al. "Contour detection and hierarchical image segmentation." IEEE Pattern Analysis and Machine Intelligence, 2011.
- [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.

PAMELA

Backup slides

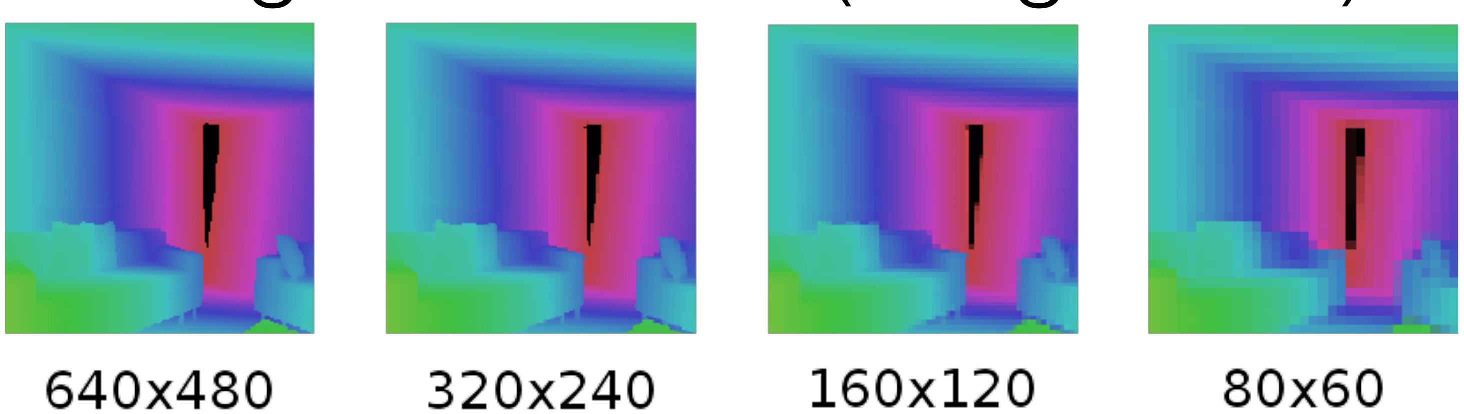
KinectFusion algorithmic features

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

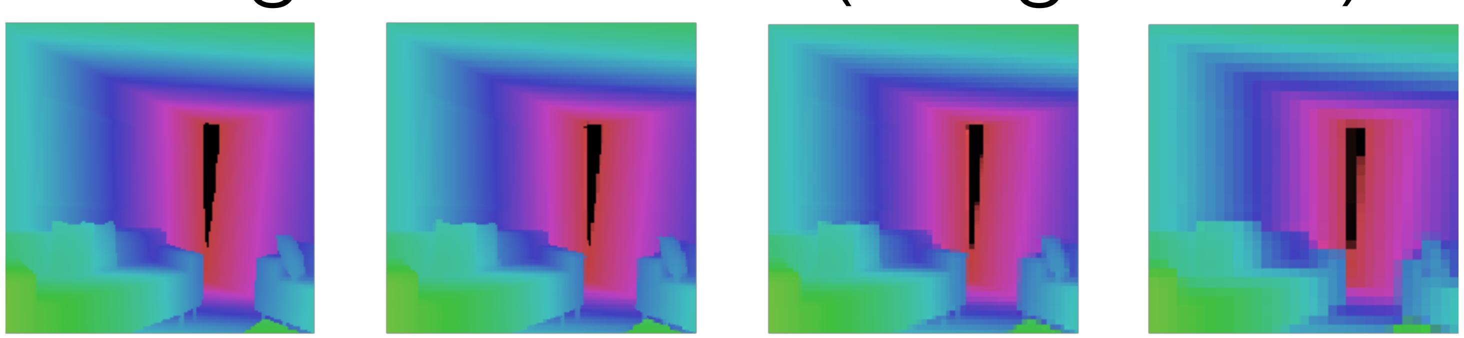
Image resolution (image ratio)



KinectFusion algorithmic features

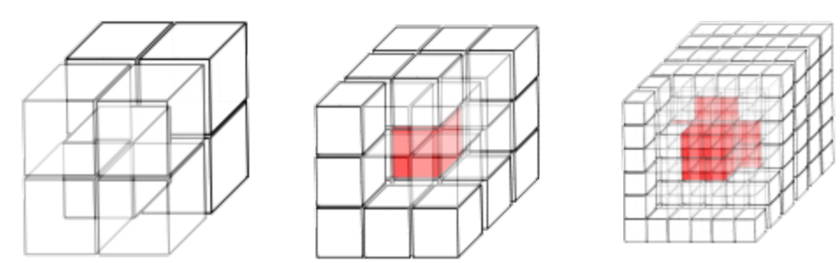
Features	Ranges
Volume resolution	64x64x64, 128x128x128, 256x256x256, 512x512x512
μ distance	0 .. 0.5
Pyramid level iterations (3 levels)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Image resolution (image ratio)	1, 2, 4, 8
Tracking rate	1, 2, 3, 4, 5
ICP threshold	10^{-6} .. 10^{2}
Integration rate	1 .. 30

Image resolution (image ratio)



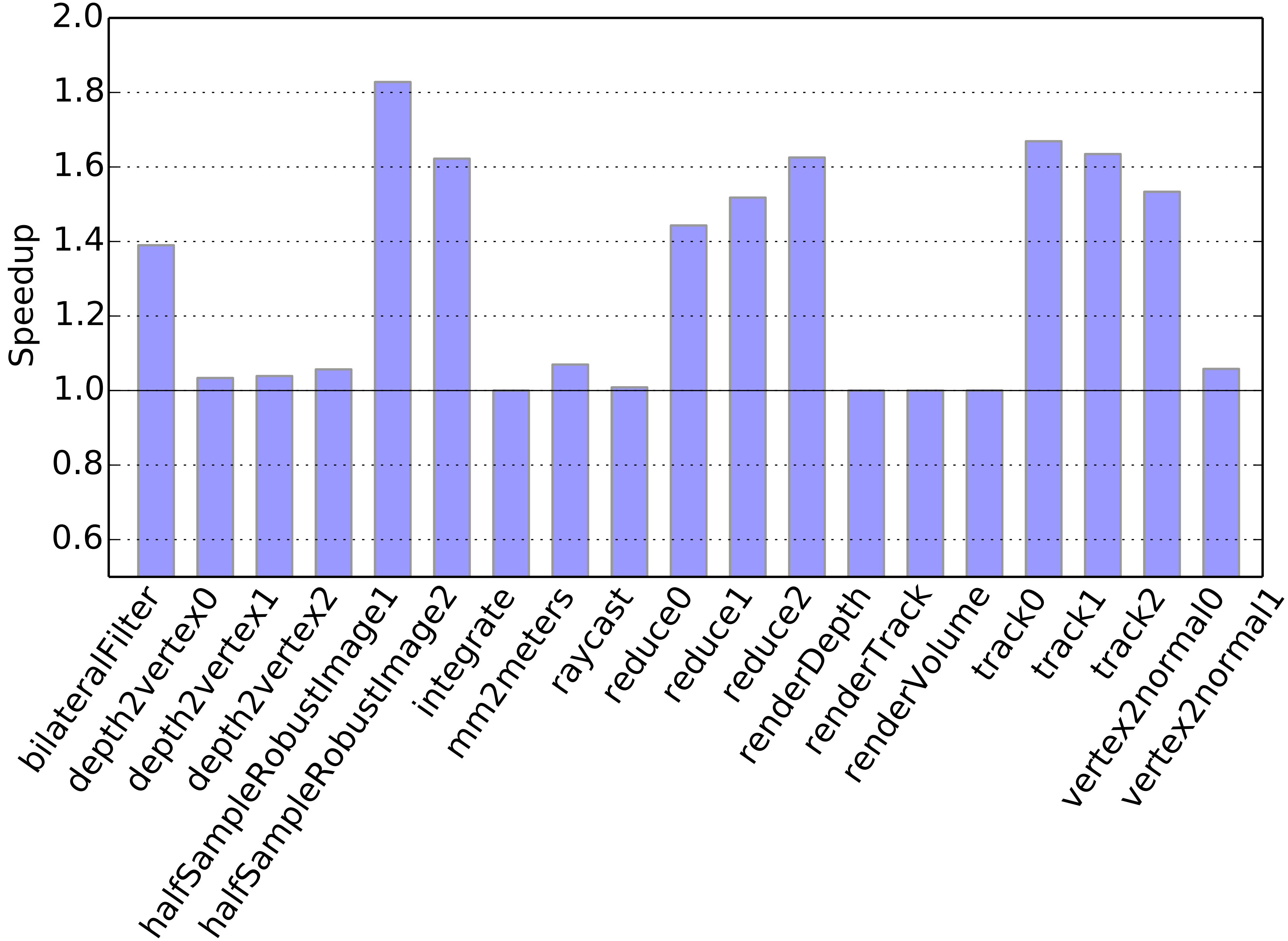
640x480 320x240 160x120 80x60

Volume resolution

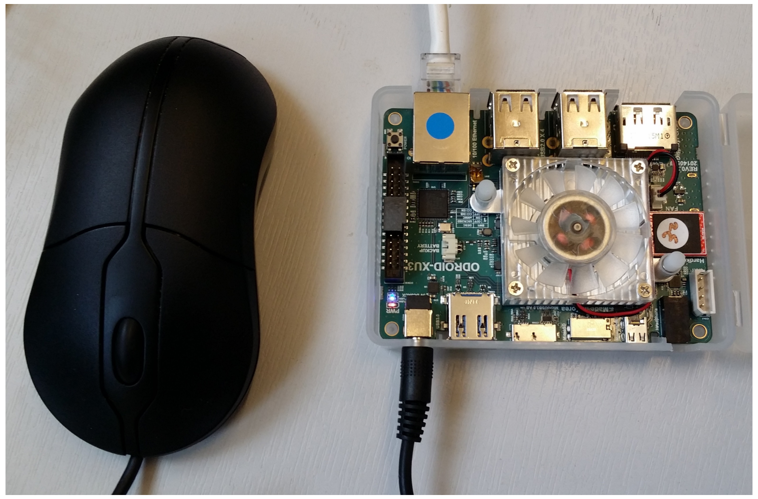


2x2x2 3x3x3 6x6x6

DSE compiler parameters speedup



"Performance" on SLAMBench



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

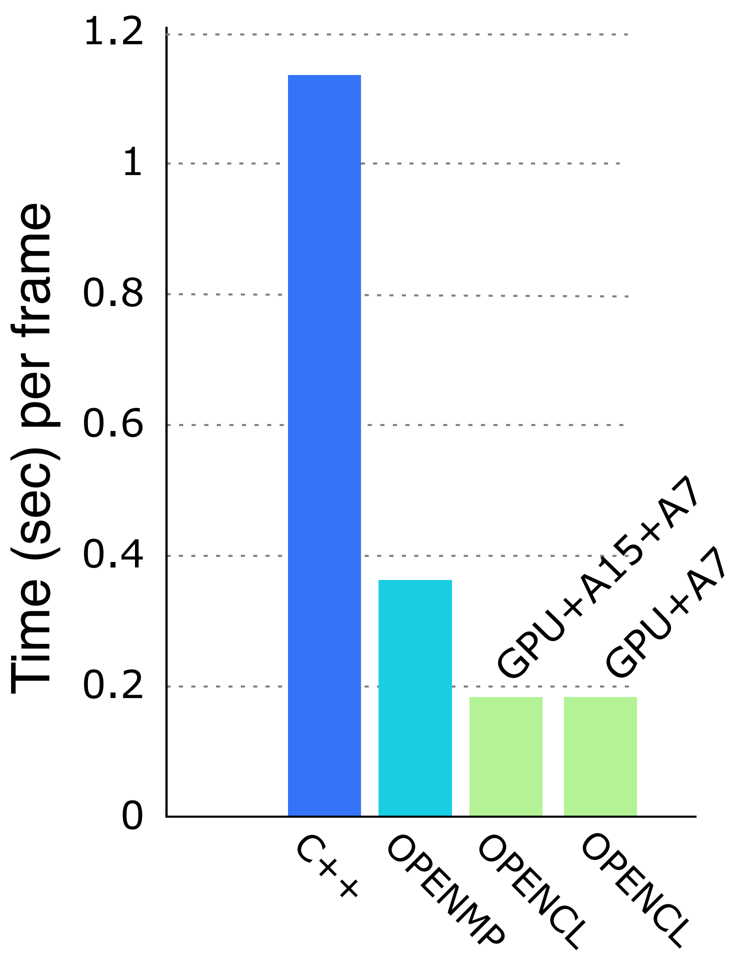
Machine	CPU	CPU name	CPU GFLOPS	CPU cores	GPU	GPU name	GPU GFLOPS	TDP Watts
Hardkernel ODROID-XU3	ARM A15 + A7	Exynos 5422	80	4 + 4	ARM	Mali-T628	60 + 30	10

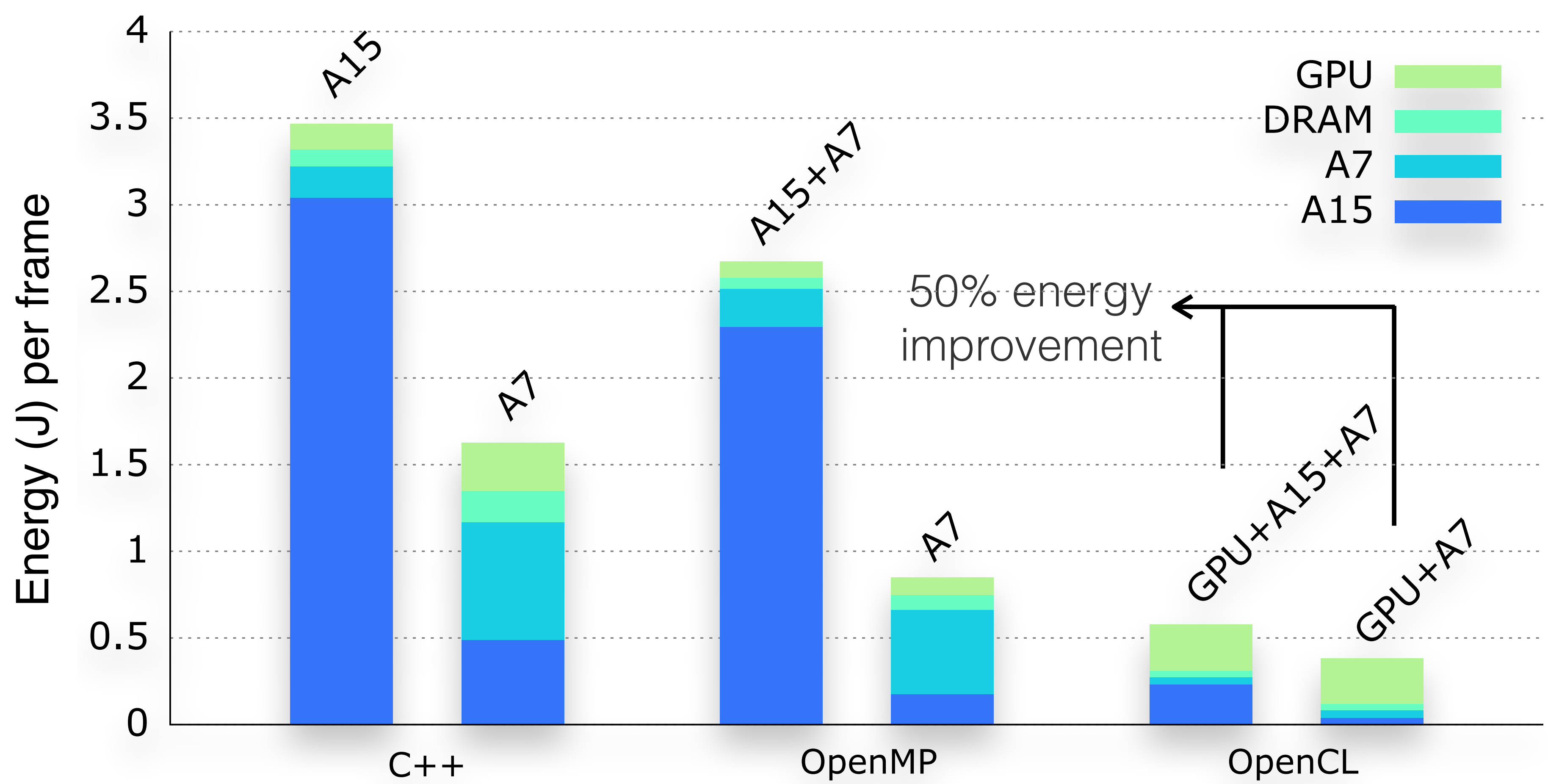
"Performance" on SLAMBench

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

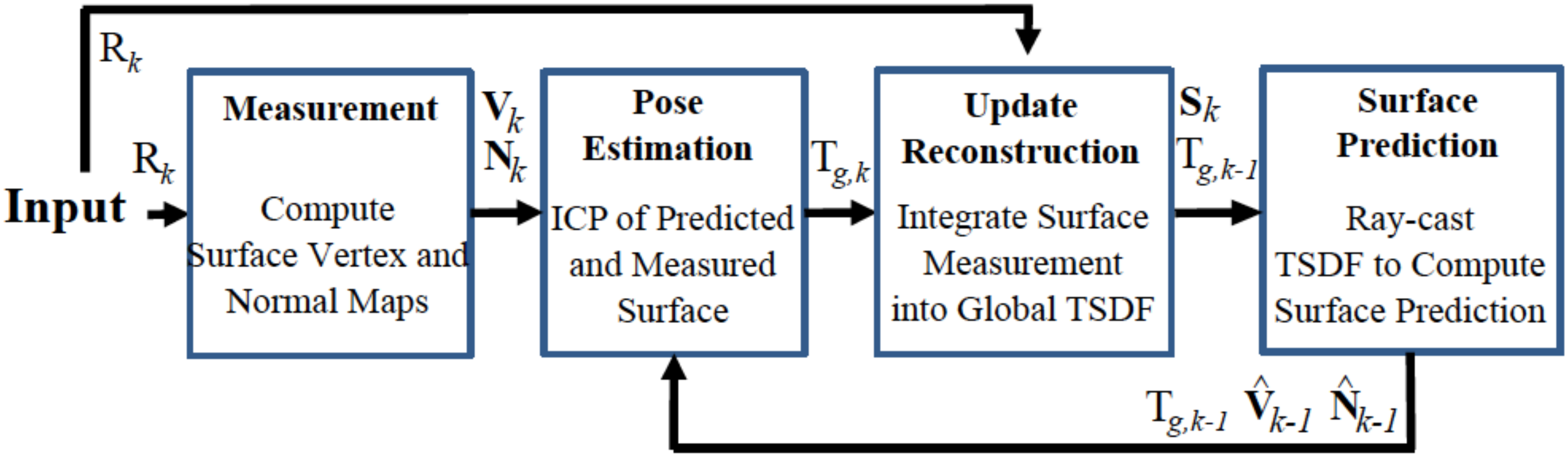
ATE in cm	
C++	2.06
OpenMP	2.06
OpenCL	2.01

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





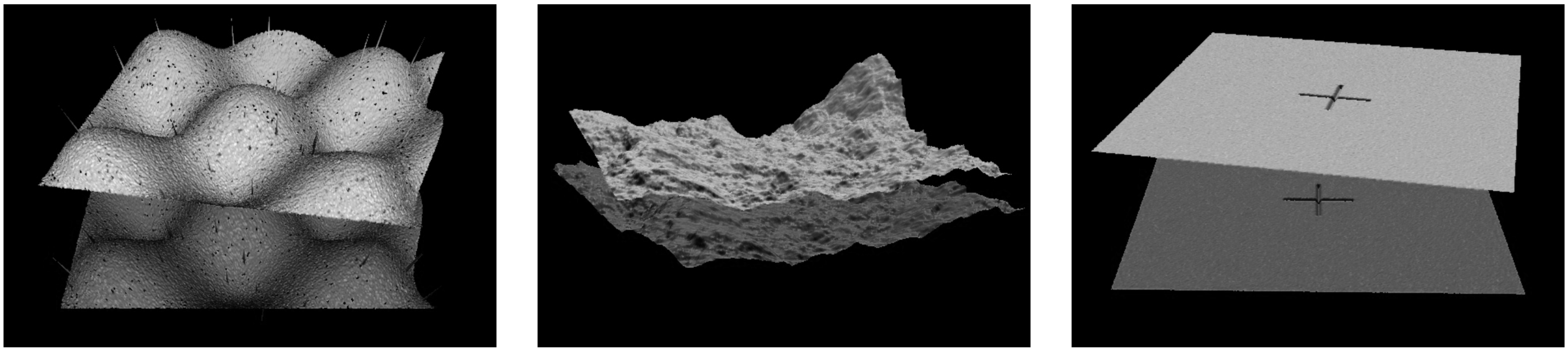
[Nardi et al., 2015]



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

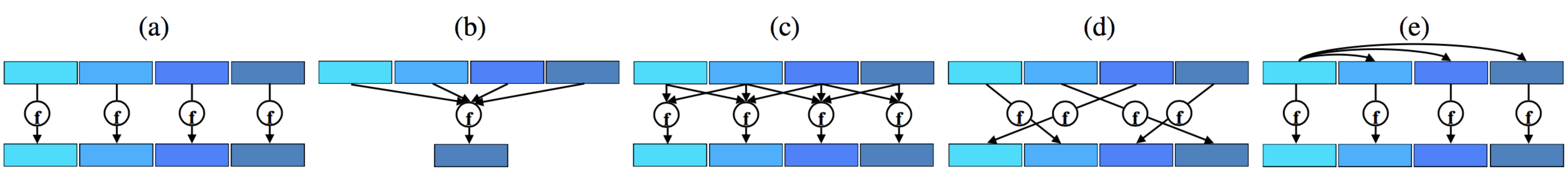
1	0.8	0.2	0.0	-0.2	-0.9	1	1
1	1	0.8	0.2	0.0	-0.2	-0.95	1
1	1	0.9	0.4	0.0	-0.1	-0.7	1
1	0.8	0.5	0.1	-0.1	-0.5	-0.9	1
1	1	0.8	0.4	0.0	-0.3	-0.7	1
1	1	0.9	0.5	0.1	0.0	-0.3	-0.5
1	1	1	0.9	0.8	0.2	0.0	-0.1
1	1	1	1	1	0.7	0.4	0.1

[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

SLAMBench kernels

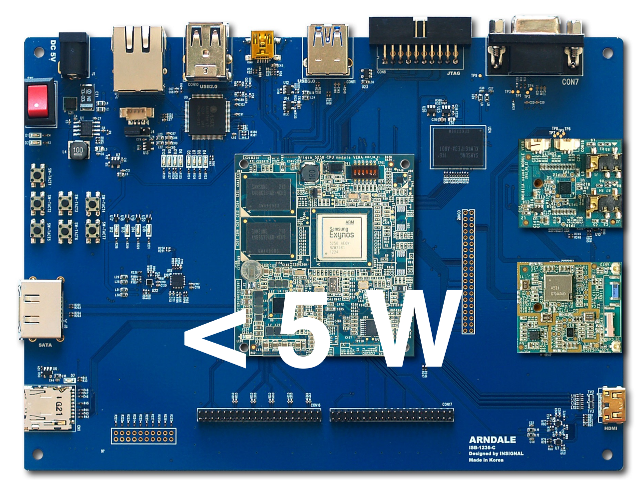
Kernels	Pipeline	Pattern	In	Out	%
acquire	Acquire	n/a	pointer	2D	0.03
mm2meters	Preprocess	Gather	2D	2D	0.06
bilateralFilter	Preprocess	Stencil	2D	2D	33.68
halfSample	Track	Stencil	2D	2D	0.05
depth2vertex	Track	Map	2D	2D	0.11
vertex2normal	Track	Stencil	2D	2D	0.27
track	Track	Map/Gather	2D	2D	4.72
reduce	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

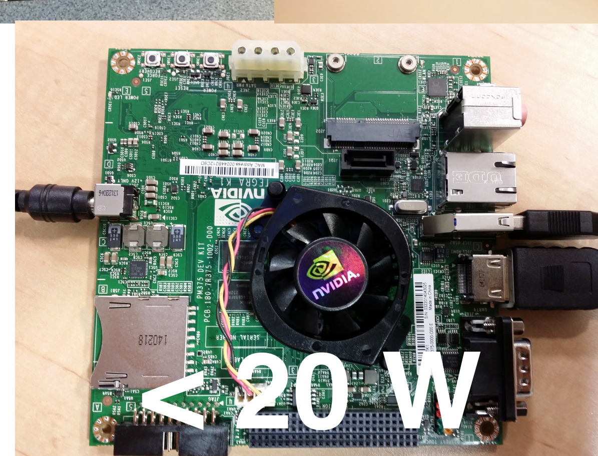


Parallel patterns: (a) Map, (b) Reduction, (c) Stencil, (d) Gather and (e) Search.

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

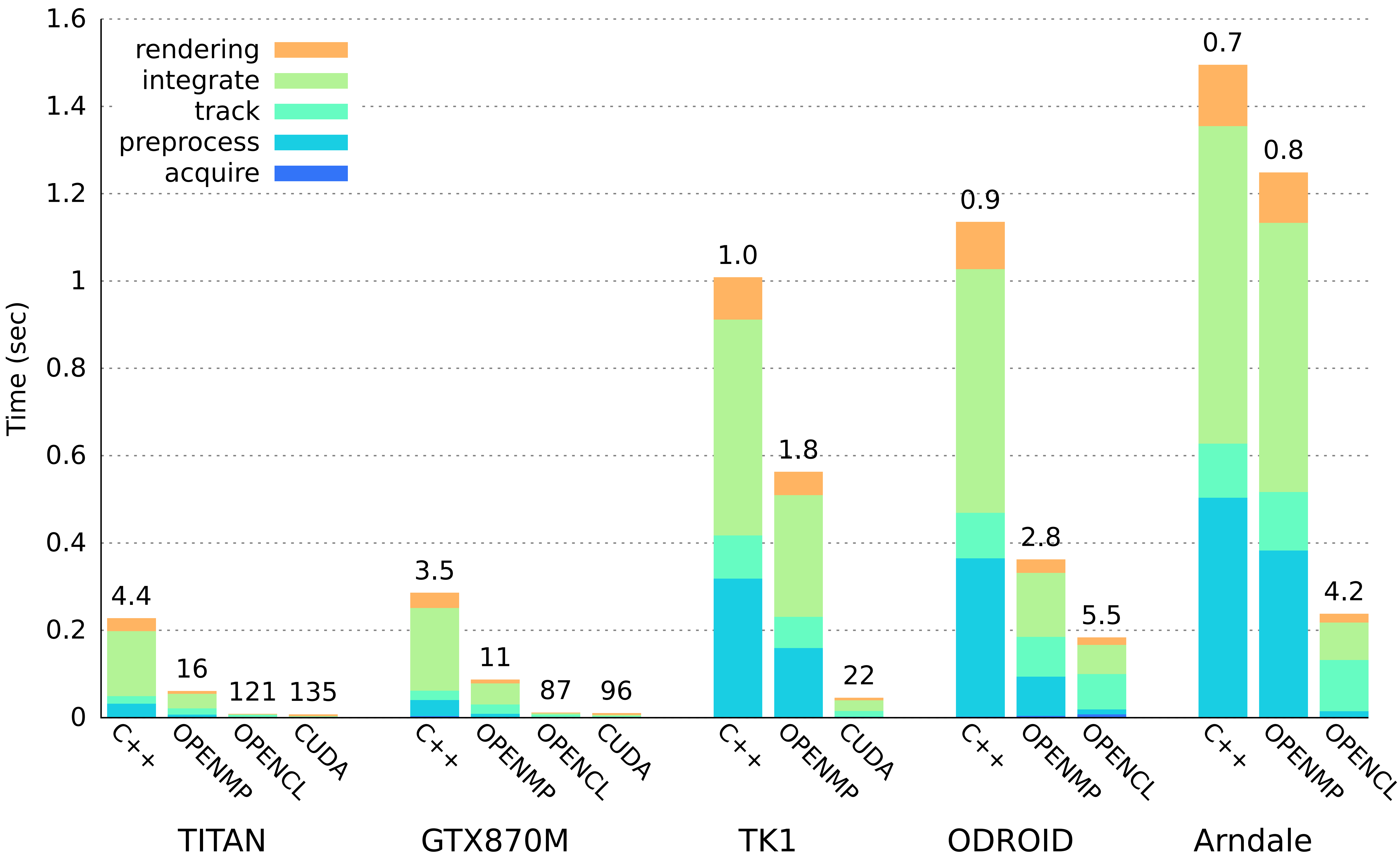
ATE in cm	TITAN	GTX870M	TK1	ODROID	Arndale
C++	2.07	2.07	2.06	2.06	2.06
OpenMP	2.07	2.07	2.06	2.06	2.06
OpenCL	2.07	2.07	n/a	2.01	2.07
CUDA	2.07	2.07	2.07	n/a	n/a

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

Luigi Nardi - Imperial College London

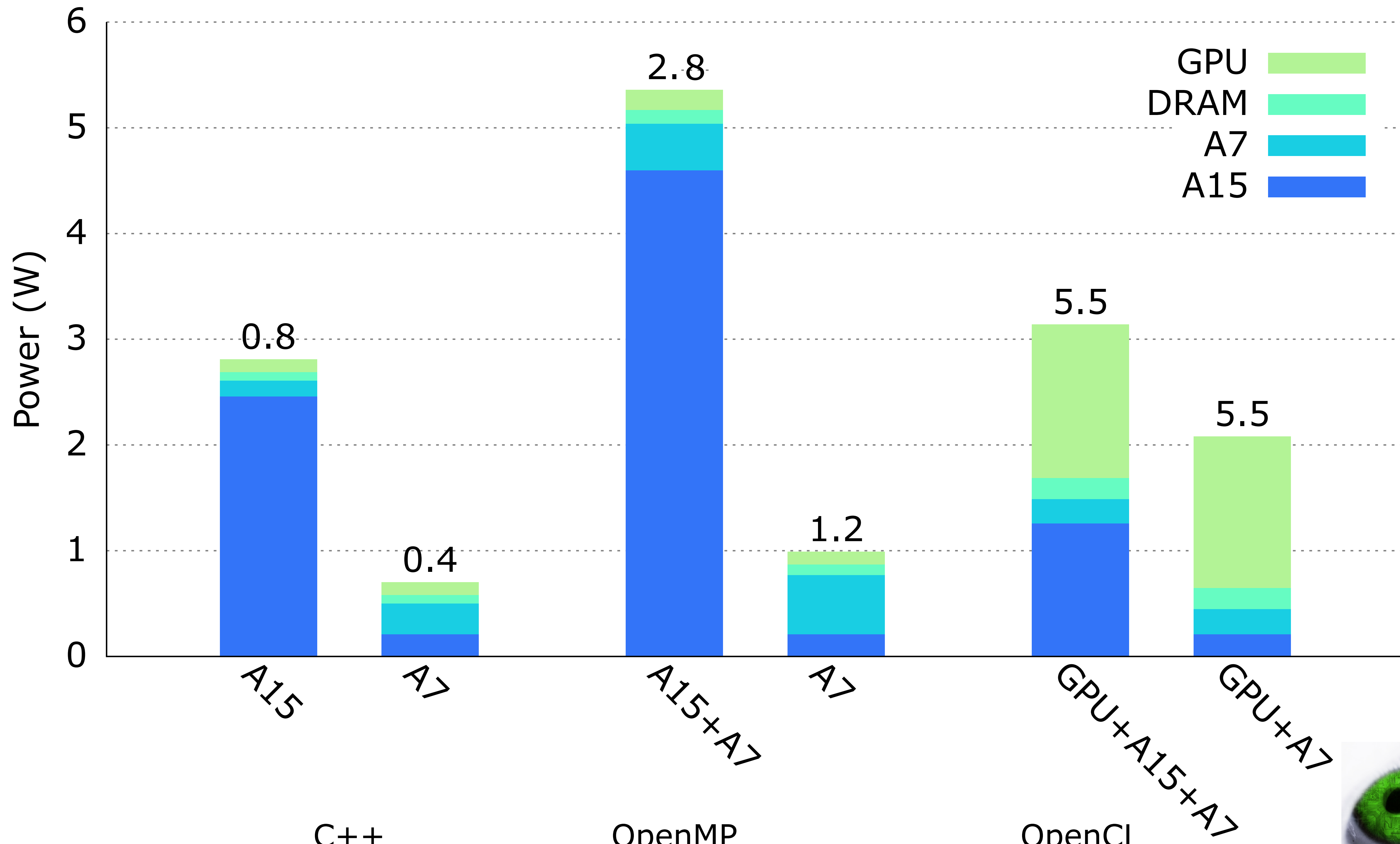
"Performance": execution time

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



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=OadhuICDAjk
- 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-darpas-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