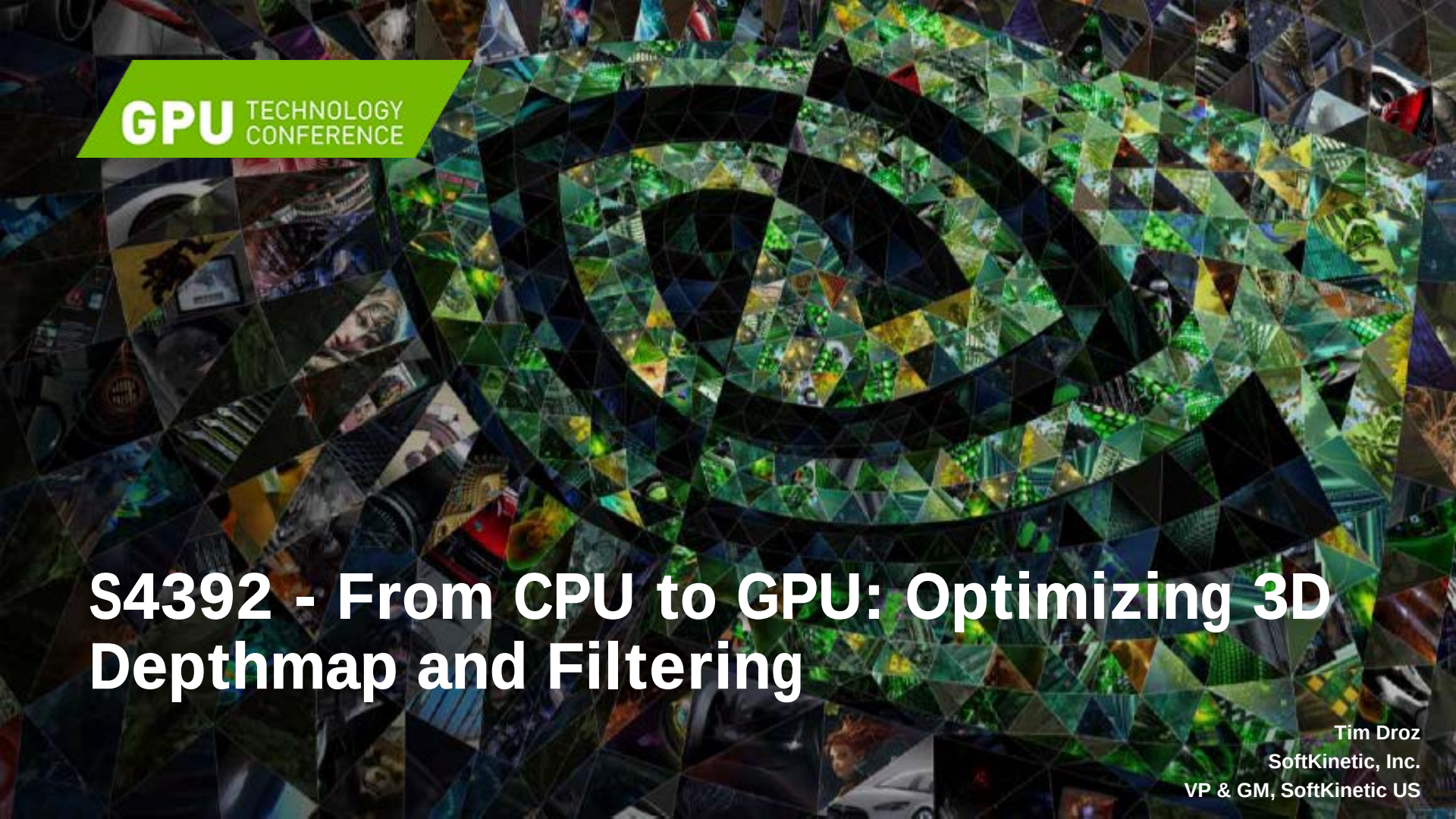


S4392 - From CPU to GPU: Optimizing 3D Depthmap and Filtering

Tim Droz
SoftKinetic, Inc.
VP & GM, SoftKinetic US

SoftKinetic 3D Vision Technology

Sensors | Cameras | Software

SoftKinetic , Founded in 2007 with offices in Belgium, China, USA, and Korea, is the only independent end-to-end provider for 3D and natural gesture solutions

3D Sensors & Cameras

- ☑ Patented Time-of-Flight (TOF) sensors
- ☑ Integrated 3D systems
- ☑ Close- & long-range, mobile, automotive, scanning

iisu™ middleware ("the interface is you")

- ☑ Whole body recognition (full skeleton)
- ☑ Pioneer in close interaction
  - 2 Hand, 10 finger tracking
- ☑ Powerful SDK available for developers

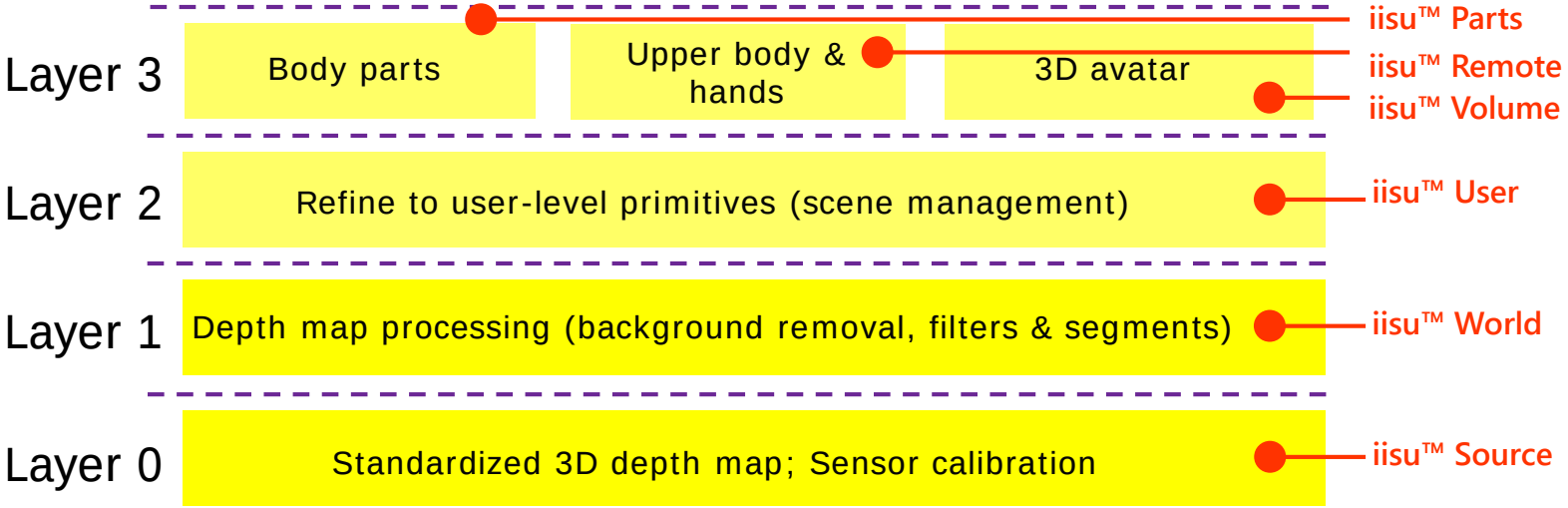
Image Processing Software

- ☑ Optimized for application use cases

©2014 SoftKinetic. All Rights Reserved.

iisu Architecture

Gesture recognition middleware



©2014 SoftKinetic. All Rights Reserved.

Problem Overview
Image processing CPU bottlenecks

- High processing times that result in movement ‘lag’ is non-optimal and can ruin gameplay
- Demand on embedded processing is taxed due to heavy filtering requirements that eat up as much as ¼ of the available resources

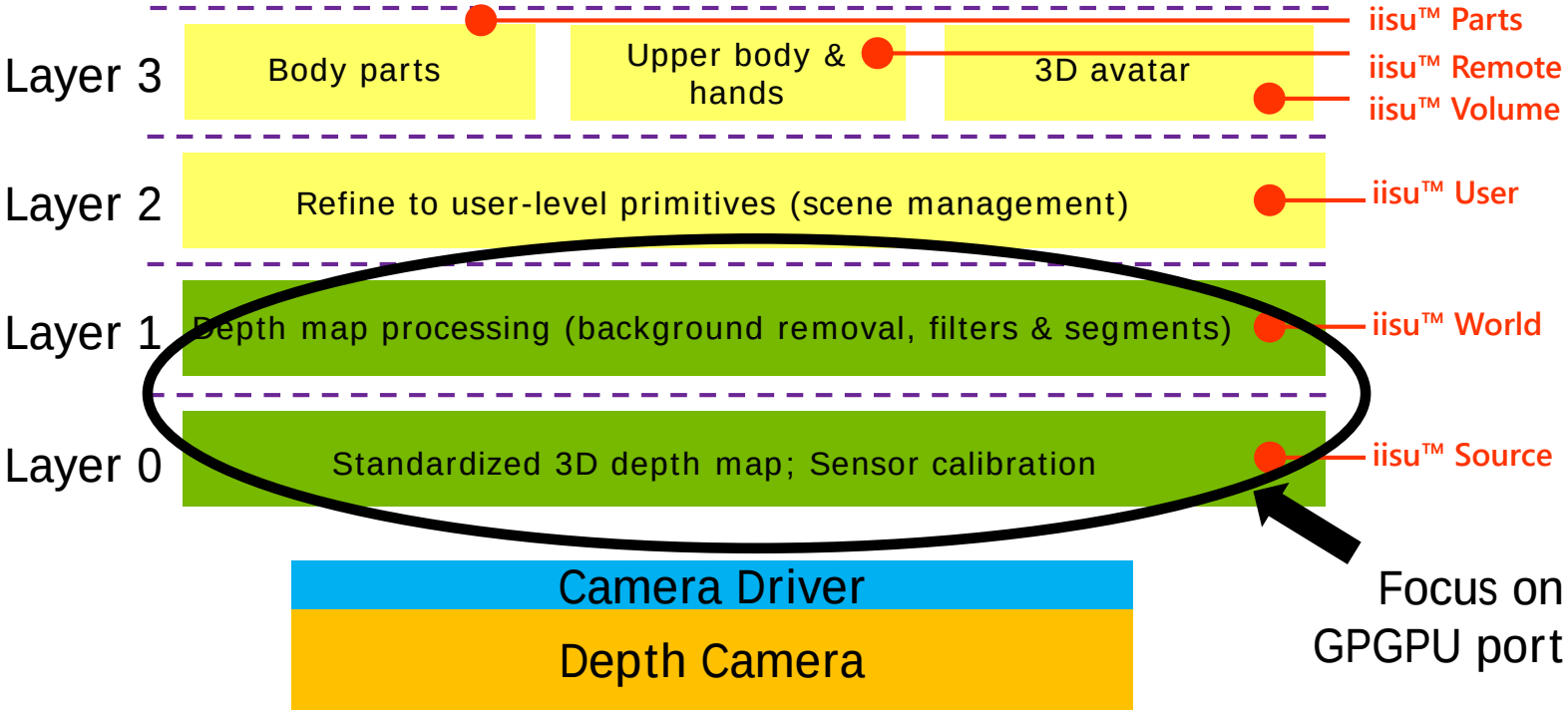
Our plan

- Offloading filtering and 3D depthmap generation to the GPU can reduce lag
- While a simple concept, it requires near perfect synchronization, making implementation complex and delicate, even in a GPU-optimized flow

©2014 SoftKinetic. All Rights Reserved.

iisu Architecture

Gesture recognition middleware



©2014 SoftKinetic. All Rights Reserved.

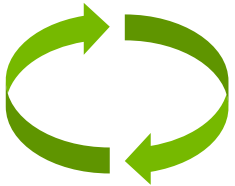
"Bathtub" Filter Details
Moving CPU resources to GPGPU

Assumptions

- Executing 60 iterations at each frame
- All floating point operations

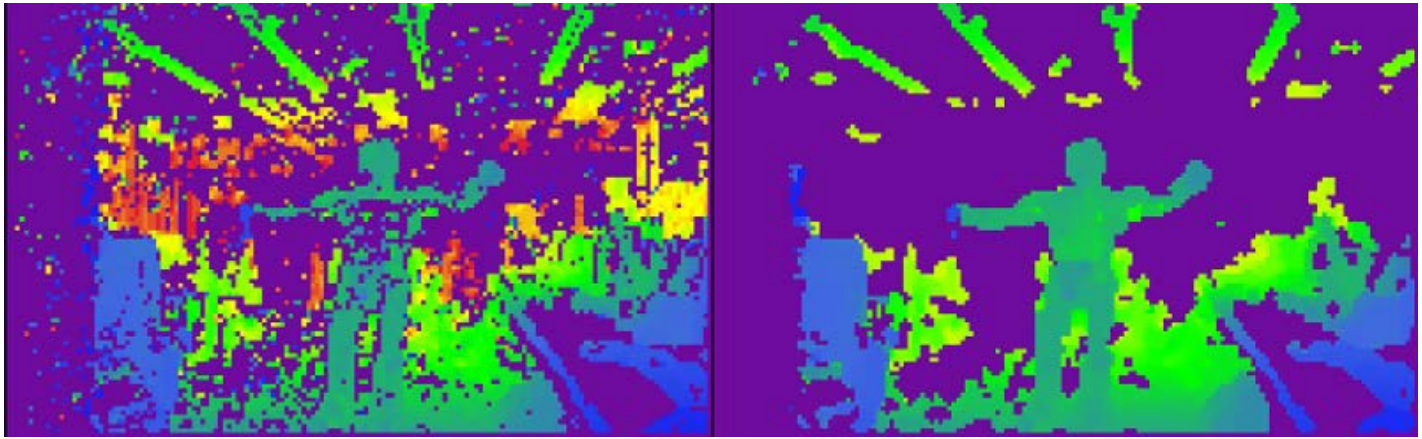
Algorithm flow – 2 passes at each iteration:

- 1st pass: from the input, and the output of the previous iteration two intermediate outputs are computed
- Our first ***sync point*** is now done
- 2nd pass: from the intermediate outputs computed in the 1st pass, the new output is computed
- Another ***sync point***
- This 2-pass cycle is then iterated until convergence, ~60 iterations



©2014 SoftKinetic. All Rights Reserved.

Results
Significant performance increase



Depthmap from the camera

Depthmap after the bathtub filter

Speedup is in the order of 5-6x compared to the SIMD implementation of the same algorithm

©2014 SoftKinetic. All Rights Reserved.

Lessons Learned/Observations

Synchronization and memory transfers

- Synchronization – Each pass uses intermediate results of previous pixels, pixels that might be processed by a different GPU thread on a different compute unit
  - Hence, the "***sync point***" operations that must be done after each pass
  - Efficiency is key, as 2*60 sync points could be a lot of cycles
- Memory transfers – Offloading computation to the GPU, the cost of transferring data from CPU memory to GPU memory is key
  - Our first system ported to required optimizations & tradeoffs to avoid the fixed cost of transferring data
  - Exploring unified memory model on Tegra K1/CUDA 6.0

©2014 SoftKinetic. All Rights Reserved.

What's Next
Continued GPGPU porting

- Analyzing and porting other layers to GPGPU
- Moving computations in the camera device driver to GPGPU
- Continued porting and optimization on K1
- Power optimization as we move from desktop to mobile

©2014 SoftKinetic. All Rights Reserved.

Contact:

Tim Droz
tdr@softkinetic.com

www.softkinetic.com
www.youtube.com/SoftKinetic

©2014 SoftKinetic. All Rights Reserved.