

# Real-Time Pointcloud Fusion and Mesh Generation

Remo Meyer & Jan Rüegg



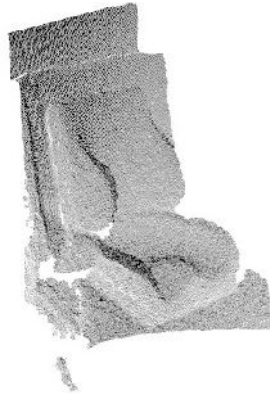
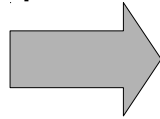
# Pipeline



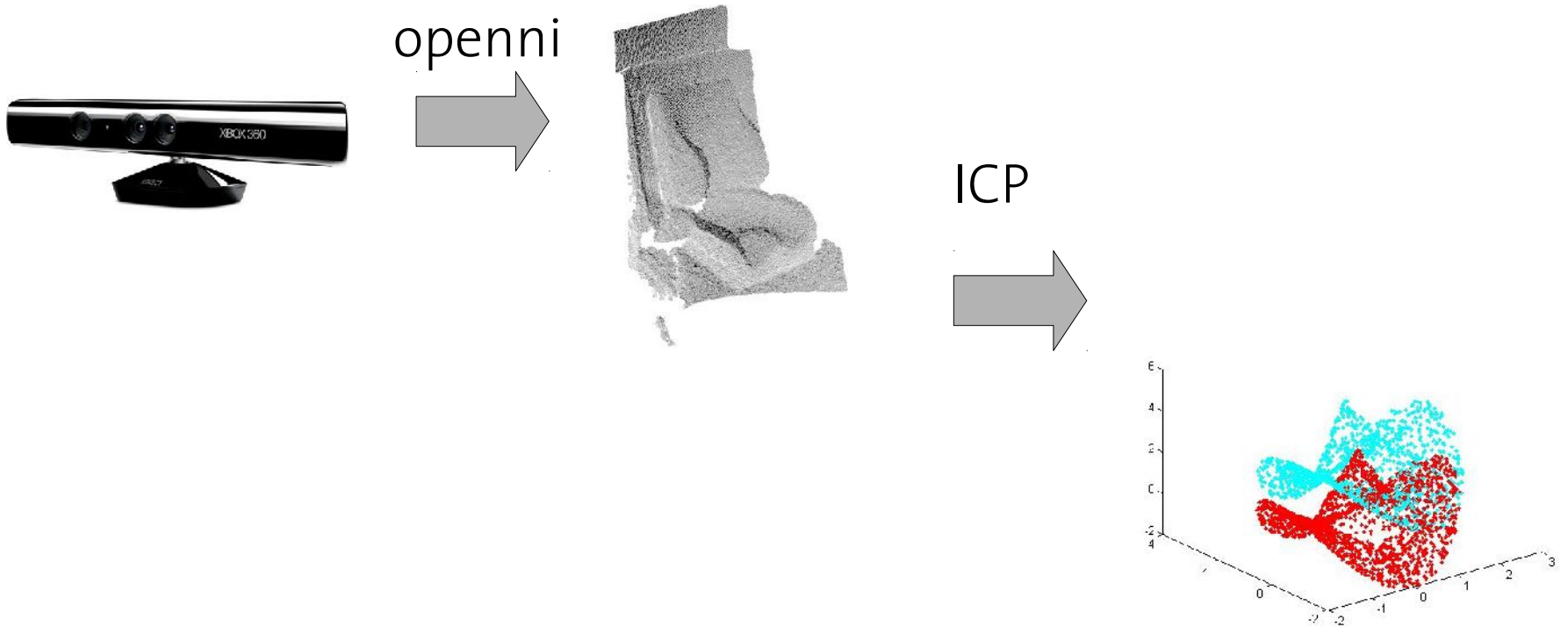
# Pipeline



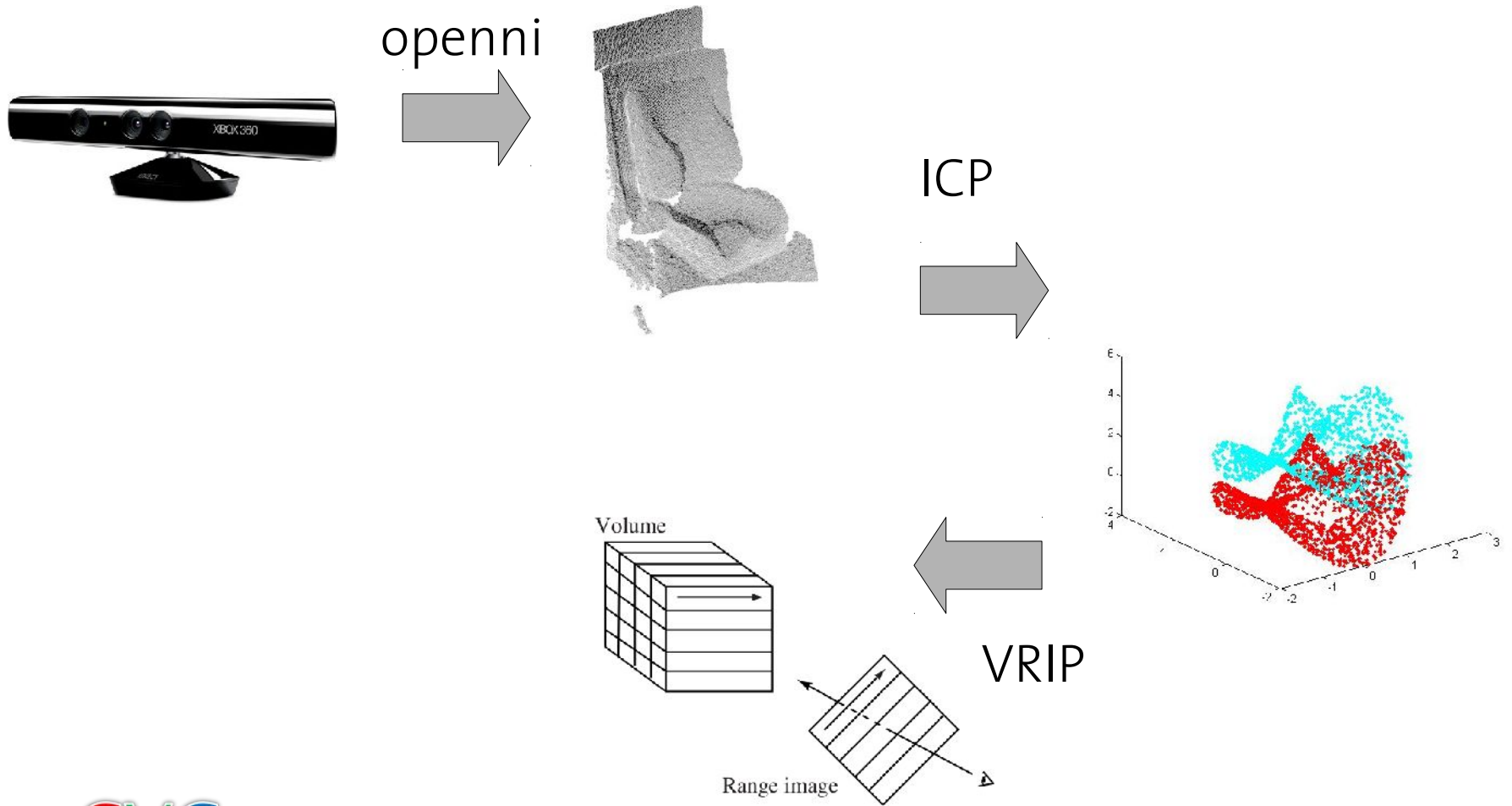
openni



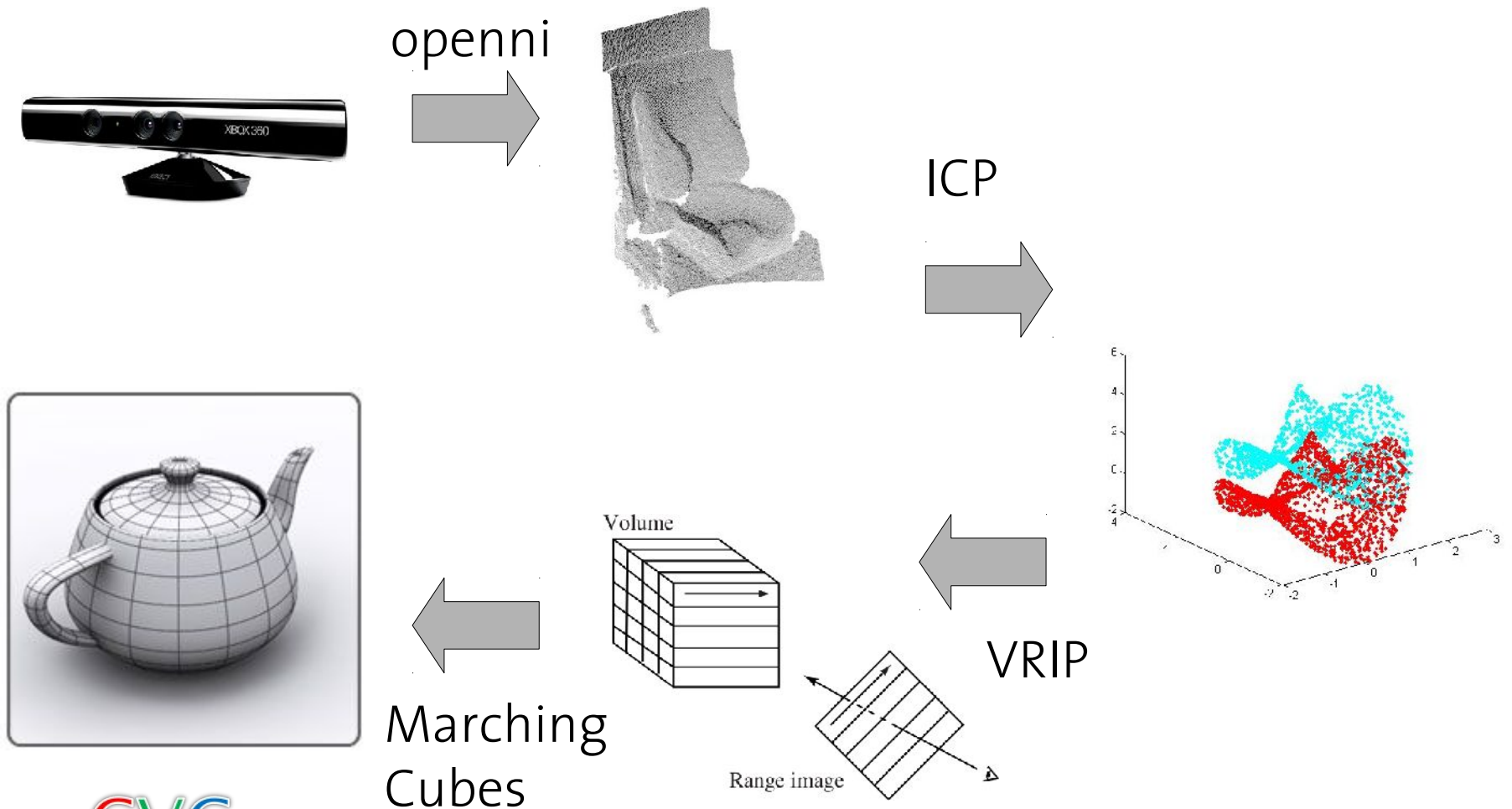
# Pipeline



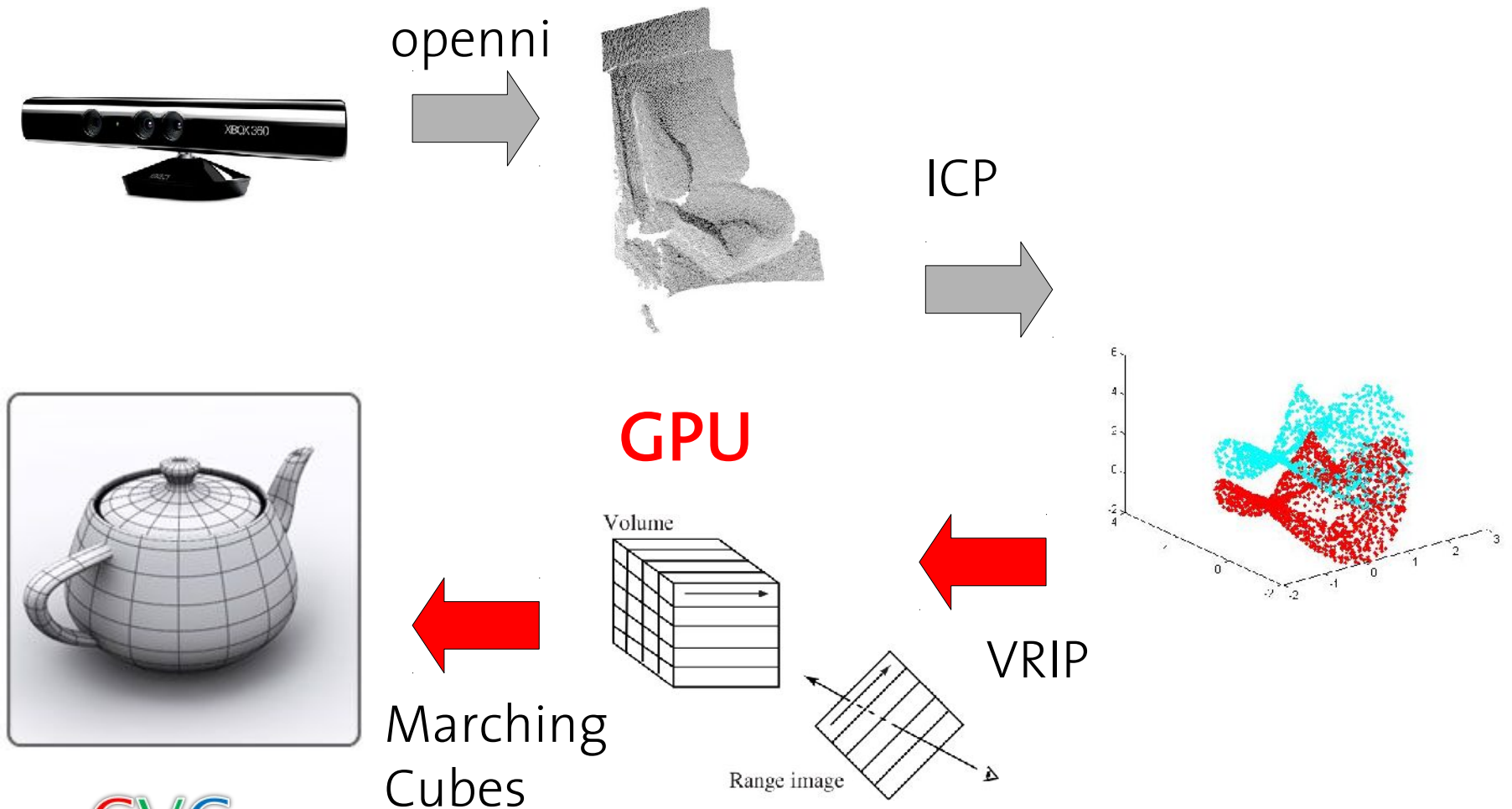
# Pipeline



# Pipeline



# Pipeline



# Part 1: ICP

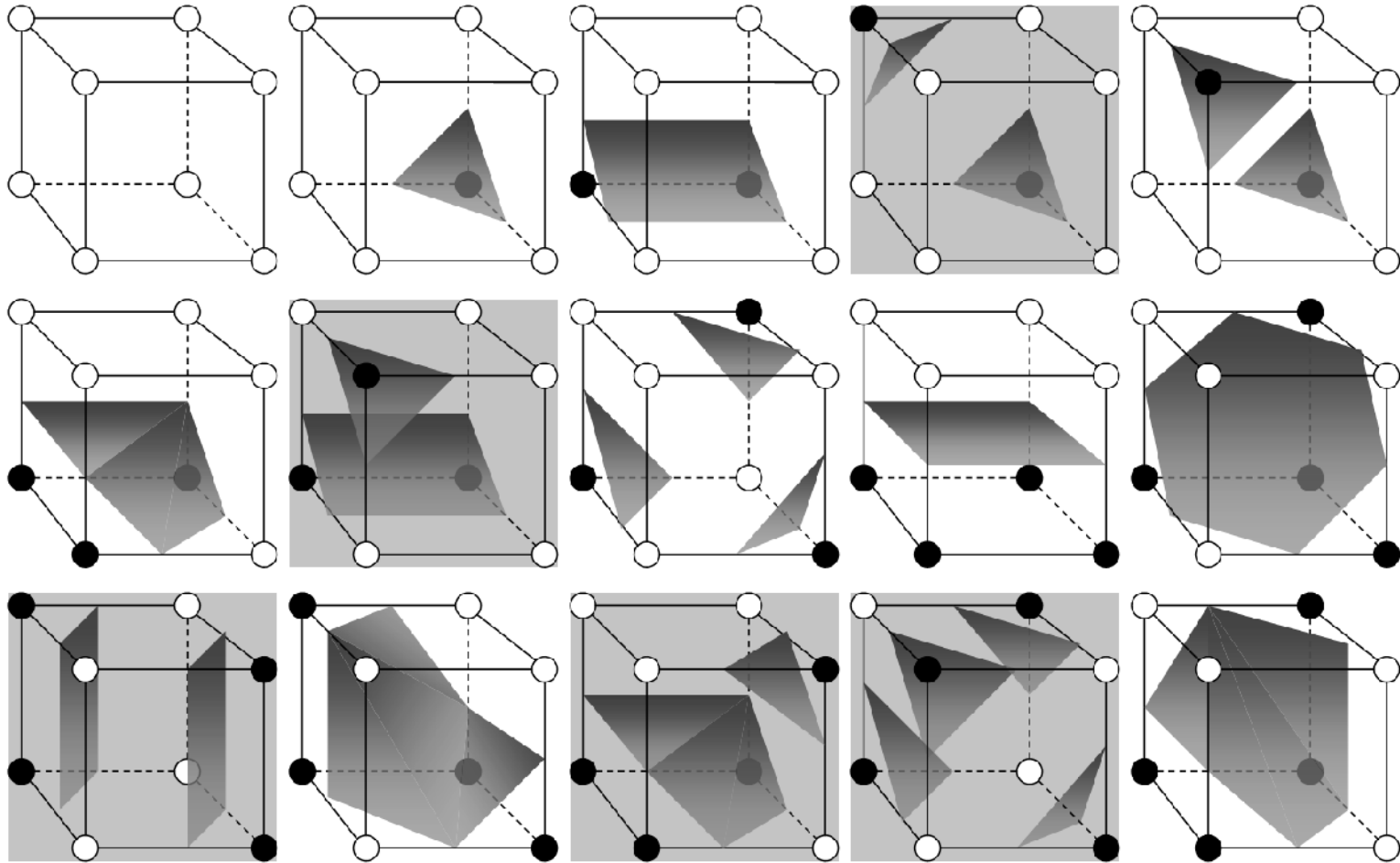
- 3D – Photography Course
- Reading Kinect Data
- 6-Step ICP Pipeline
- Robot Operating System



# Part 2: VRIP

- *A Volumetric Method for Building Complex Models from Range Images*
- Brian Curless & Marc Levoy, Stanford University, SIGGRAPH '96
- *From Scratch* Implementation

# Marching Cubes



# Marching Cubes

- Reference Implementation
  - <http://www.thebigblob.com/tag/marching-cubes/>
- Ported to OpenCL
- Prefix-sum to decrease memory requirement



# Technical Details

- ROS
- *OpenCL*
- Multiple *Threads*  
*(1 for ICP, 1 for VRIP / Marching Cubes)*
- Realtime
- Incremental



OpenCL

# Setup

## MacBook Pro 13“

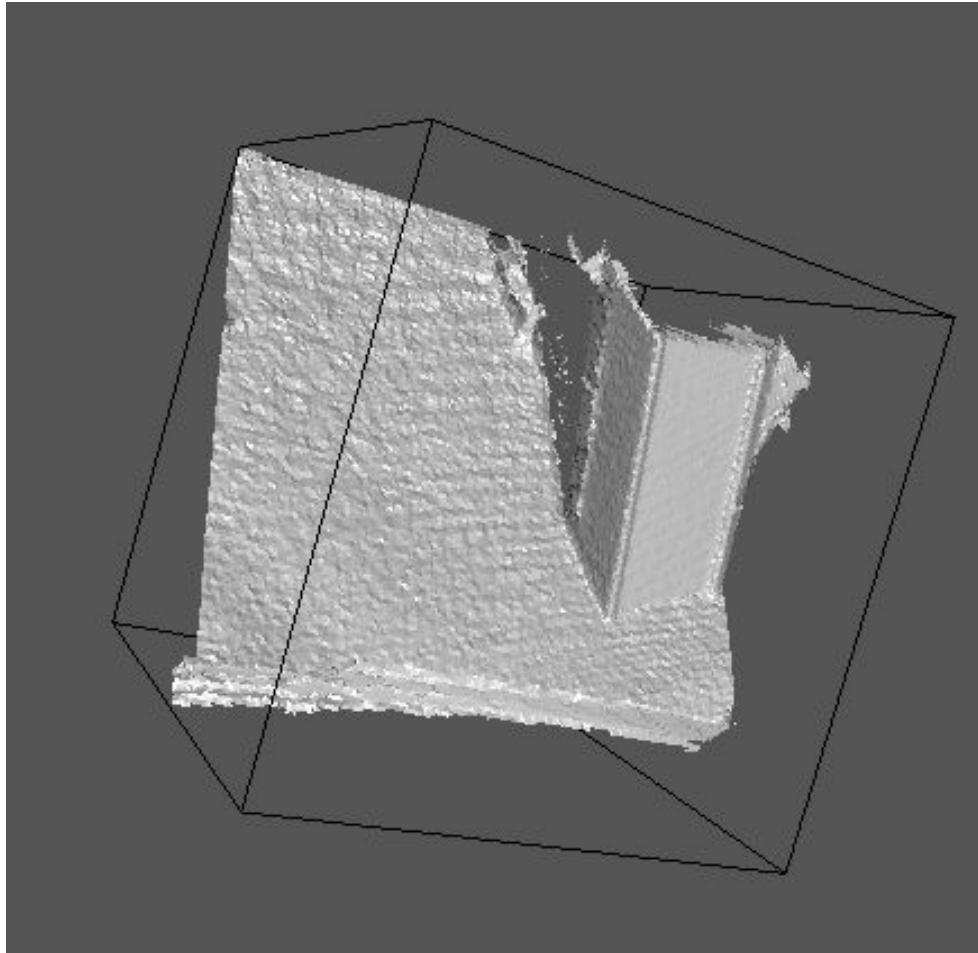
- *GPU*: Gforce 320M
- *CPU*: Core2 Duo @2.4GHz
- Grid Size: 128x128x128



# Speedup

	Vrip	Marching Cubes
CPU	44.3 ms	126 ms
GPU	5.2 ms	91 ms
Speedup	8.5	1.38

# Demo!



# Outlook

- Realtime **Viewer**
- More Memory Efficiency
- Finer Volumetric Grid





# Our Repository

- [https://github.com/SgtPepper123/ICP\\_3dPhoto](https://github.com/SgtPepper123/ICP_3dPhoto)
- *Important files in the project Folder (kinect\_icp):*
  - *src/vrip.cpp: Vrip and Marching Cubes Subscriber*
  - *src/point\_cloud\_fusor.cpp: ROS interface*
  - *fuse.cl: Vrip and Marching Cubes OpenCL Kernel*
  - *ScanLargeArrays\_Kernels.cl: PrefixSum kernel (not written by us)*