GPU TECHNOLOGY CONFERENCE

S0088
Point Cloud Library (PCL) on CUDA

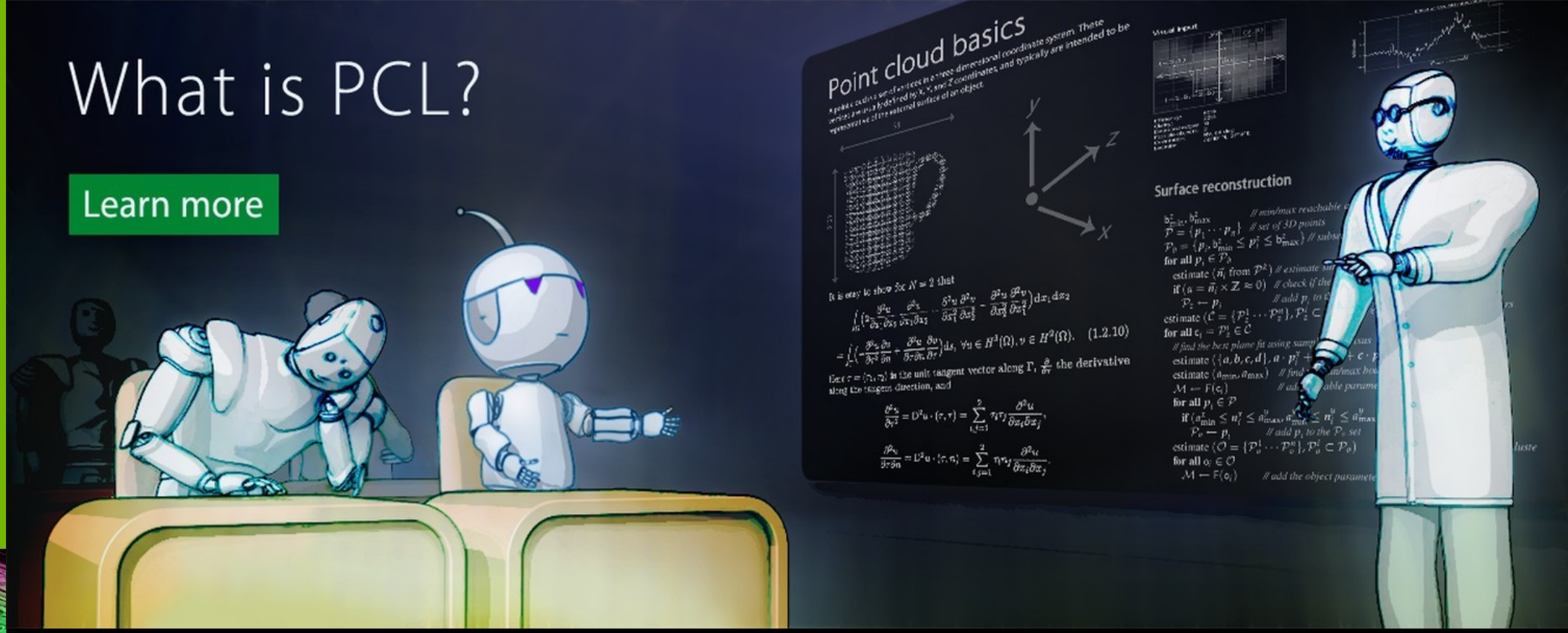
Radu B. Rusu @ Open Perception
Michael Dixon @ Willow Garage

PRESENTED BY NVIDIA.

GPU TECHNOLOGY CONFERENCE

What is PCL?

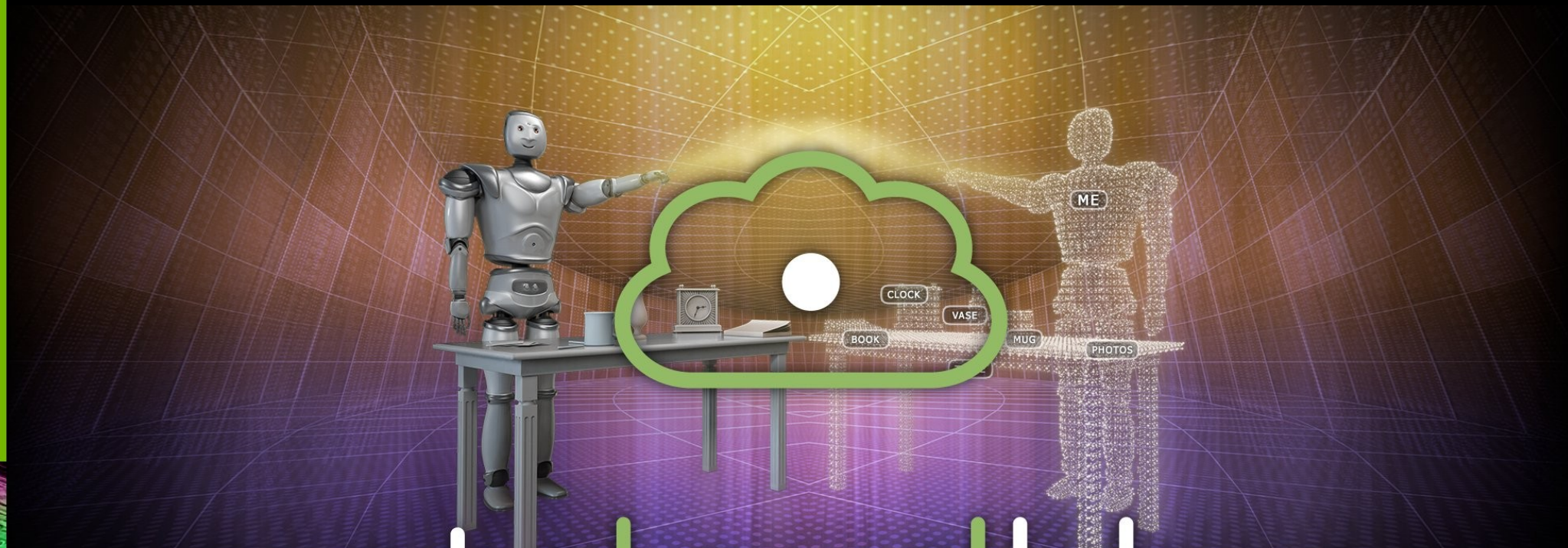
Learn more



- General overview on PCL – Radu
- CUDA optimizations for KinFu and KinSkel - Michael

pcl

GPU TECHNOLOGY CONFERENCE



pointcloudlibrary

pcl

GPU TECHNOLOGY CONFERENCE

Point Cloud Library (PCL)

- large scale, collaborative, open project for 2D/3D processing
- BSD licensing, free for commercial use
- 350 developers and contributors, many thousands of users

pcl

GPU TECHNOLOGY CONFERENCE

Why 3D?

pcl

Why 3D?

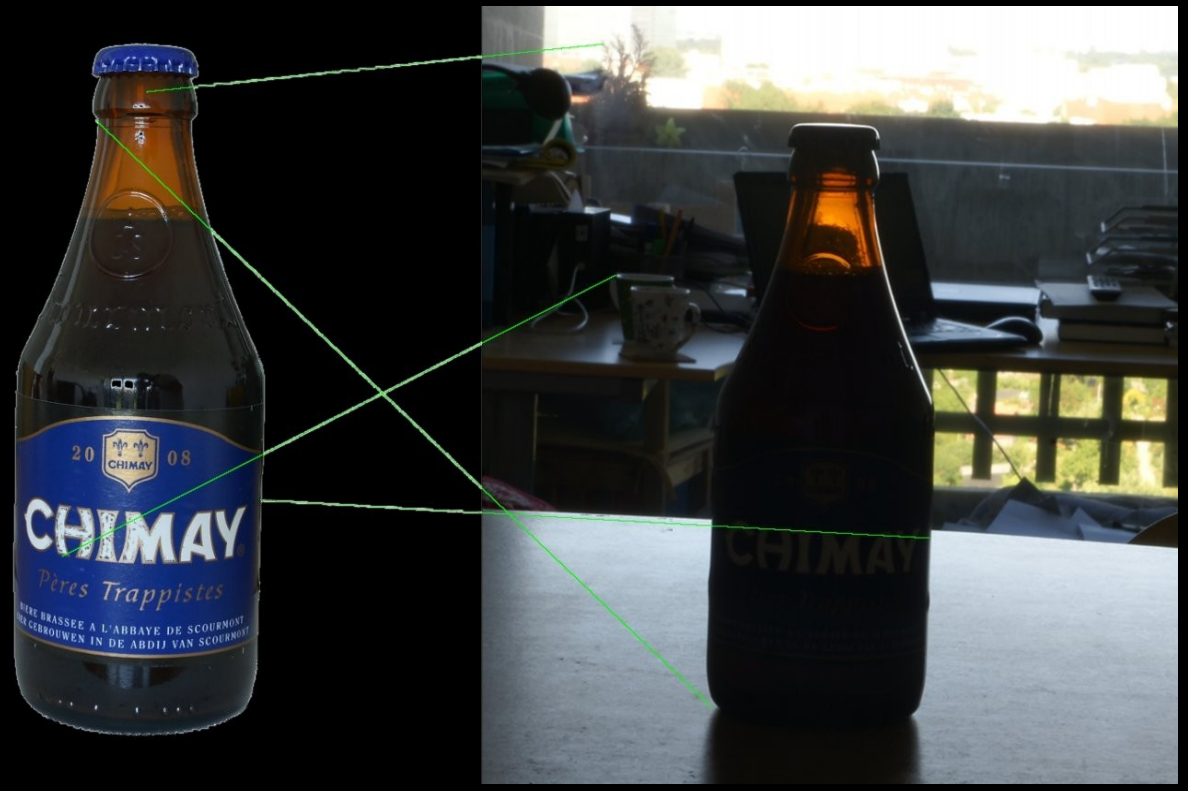
- Because the world is not 2D

Why 3D?

- Because the world is not 2D

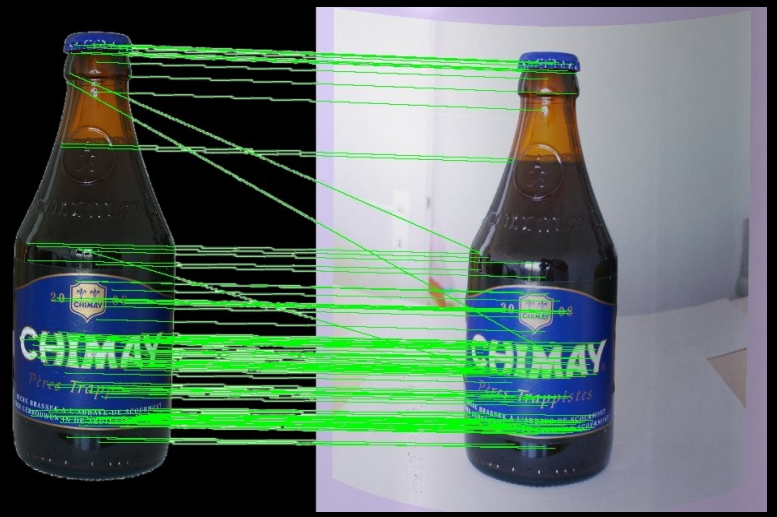
Why 3D?

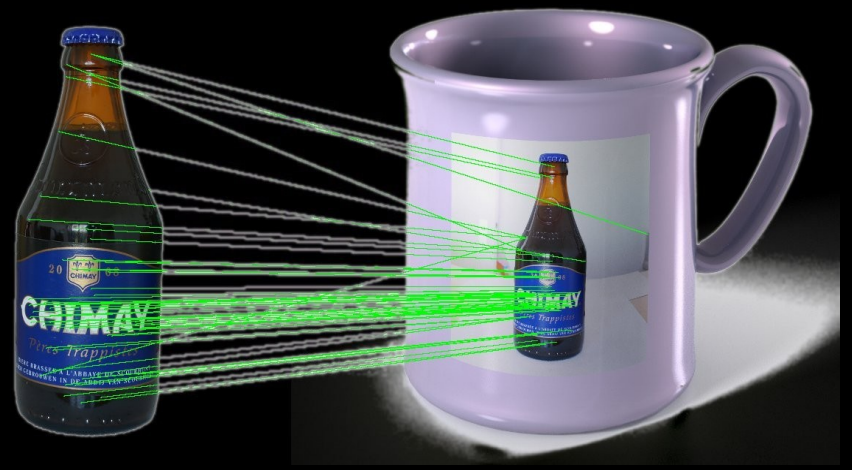
- Because 2D imagery is **useless** in certain conditions

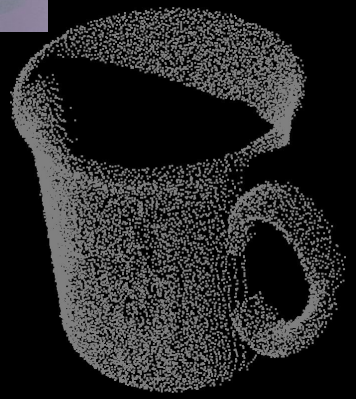


Why 3D?

- Because 2D imagery doesn't always infer good semantics

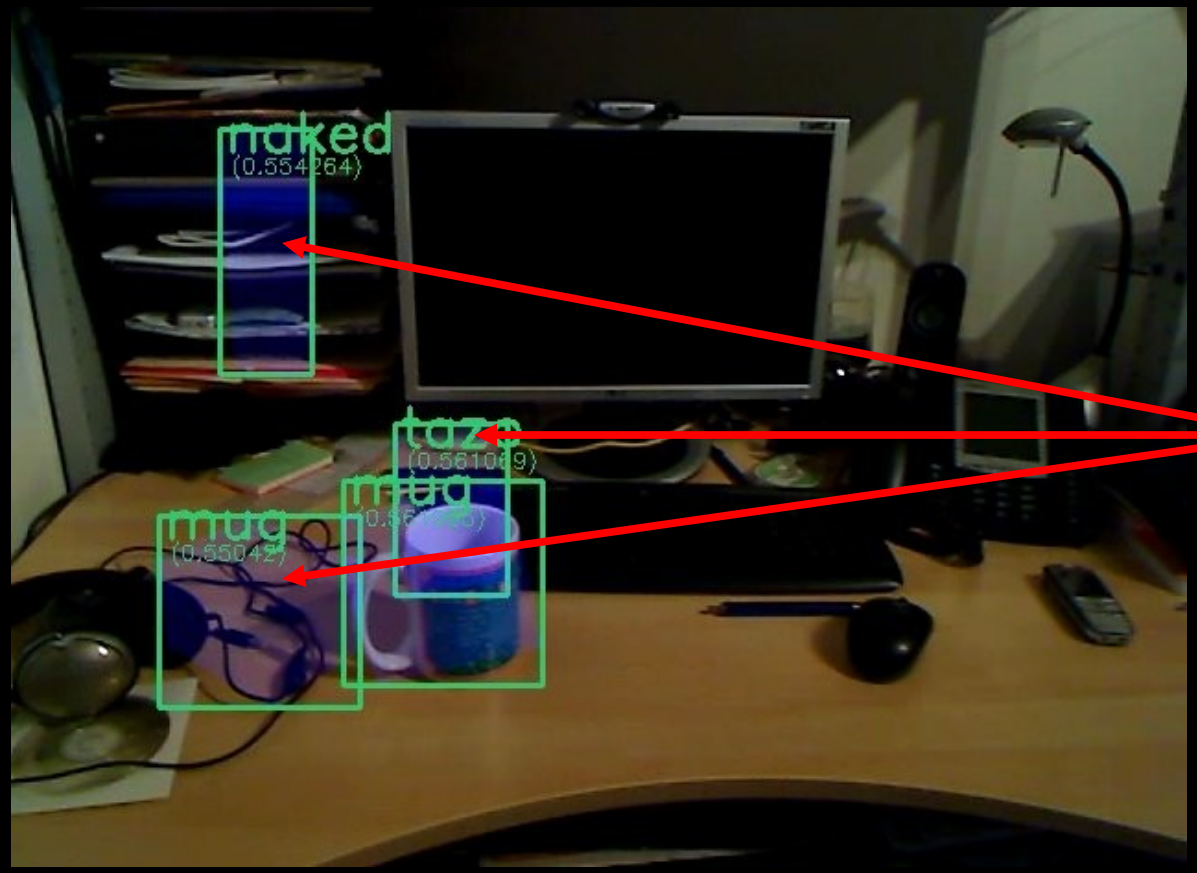






Why 3D?

- Because 2D imagery is just a … 2D projection



2D matching failures

GPU TECHNOLOGY CONFERENCE

3D

50% better than 2D

pcl

Three-dimensional data

- Point Cloud = collection of 3D points
  - 3D is really more like nD

GPU TECHNOLOGY CONFERENCE

What

can we do with Point Clouds?

pcl

Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems

pcl

Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems

pcl

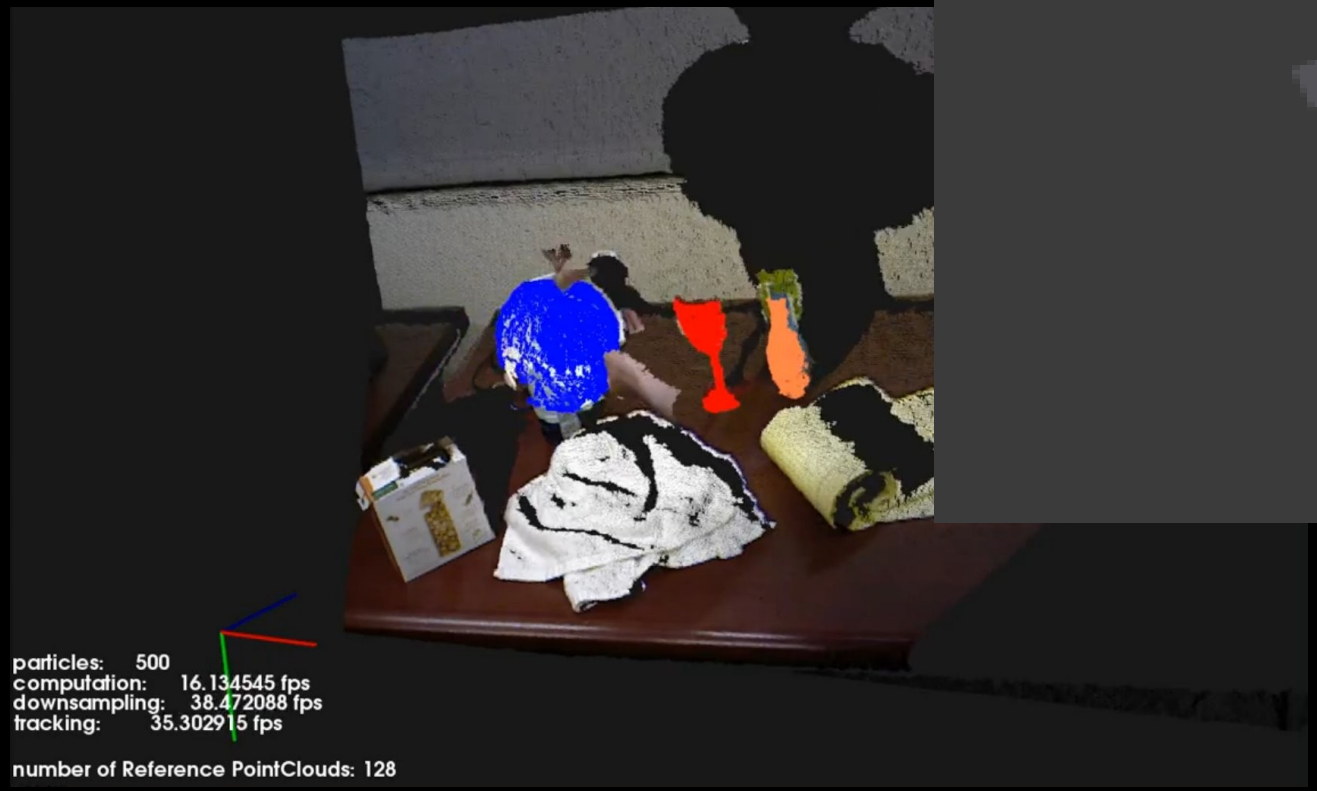
Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems

pcl

Applications of Point Clouds

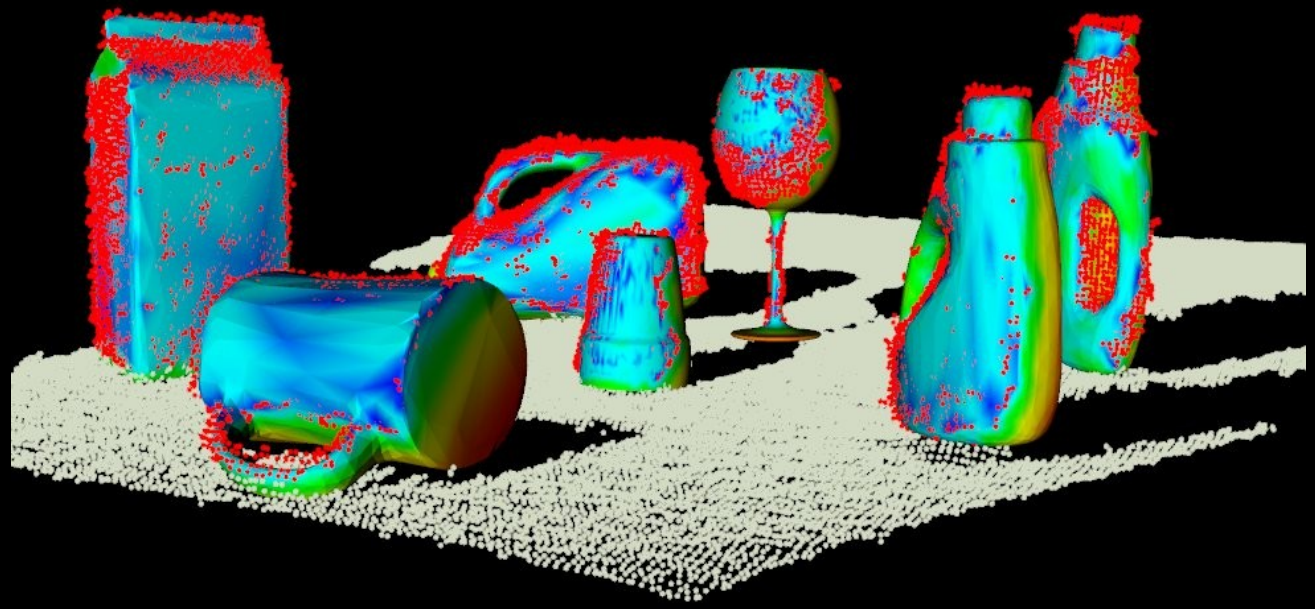
- Point clouds enable very cool applications
- We are trying to solve extremely hard problems



pcl

Applications of Point Clouds

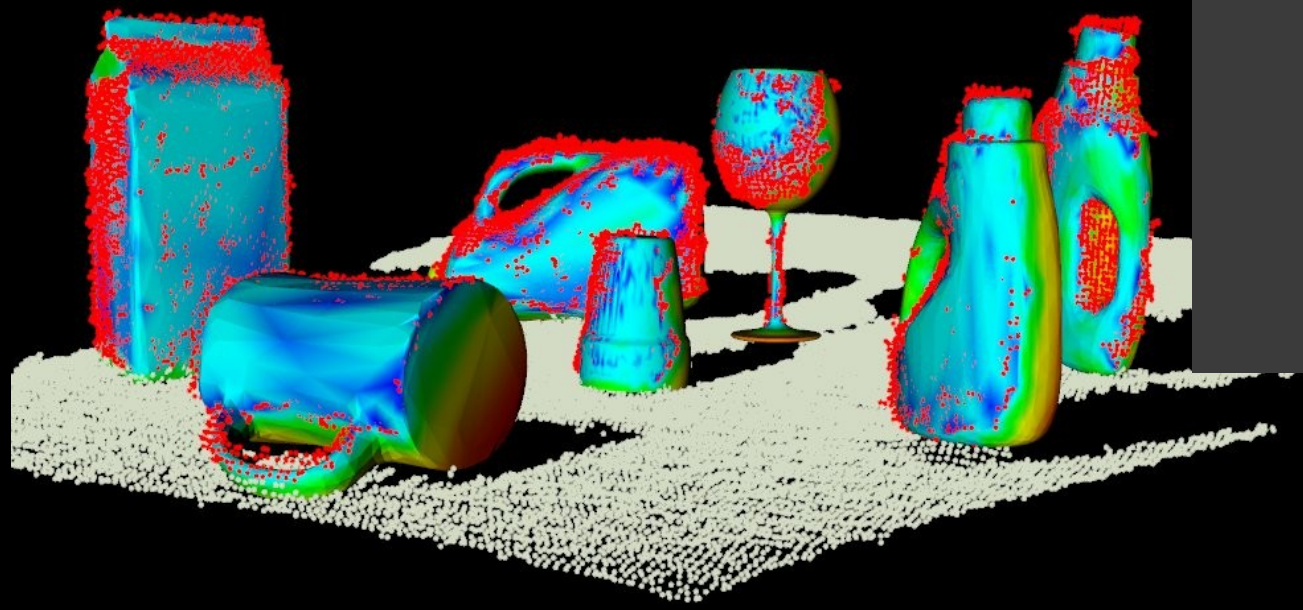
- Point clouds enable very cool applications
- We are trying to solve extremely hard problems



pcl

Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems



pcl

Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems

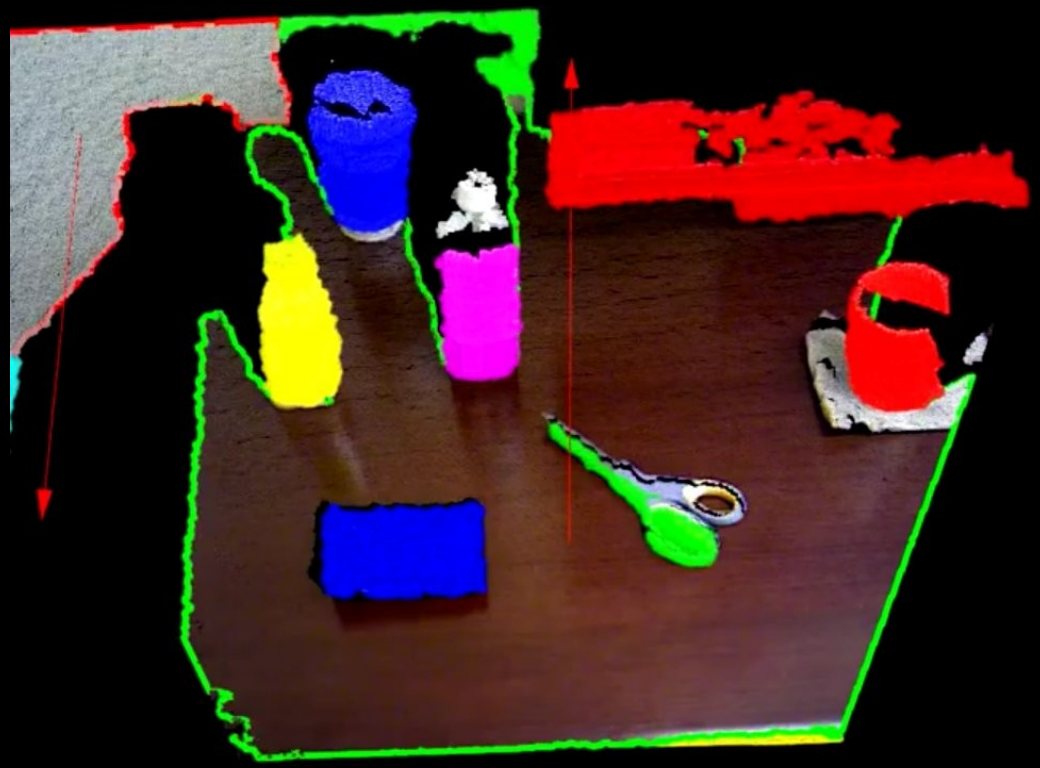
Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems

pcl

Applications of Point Clouds

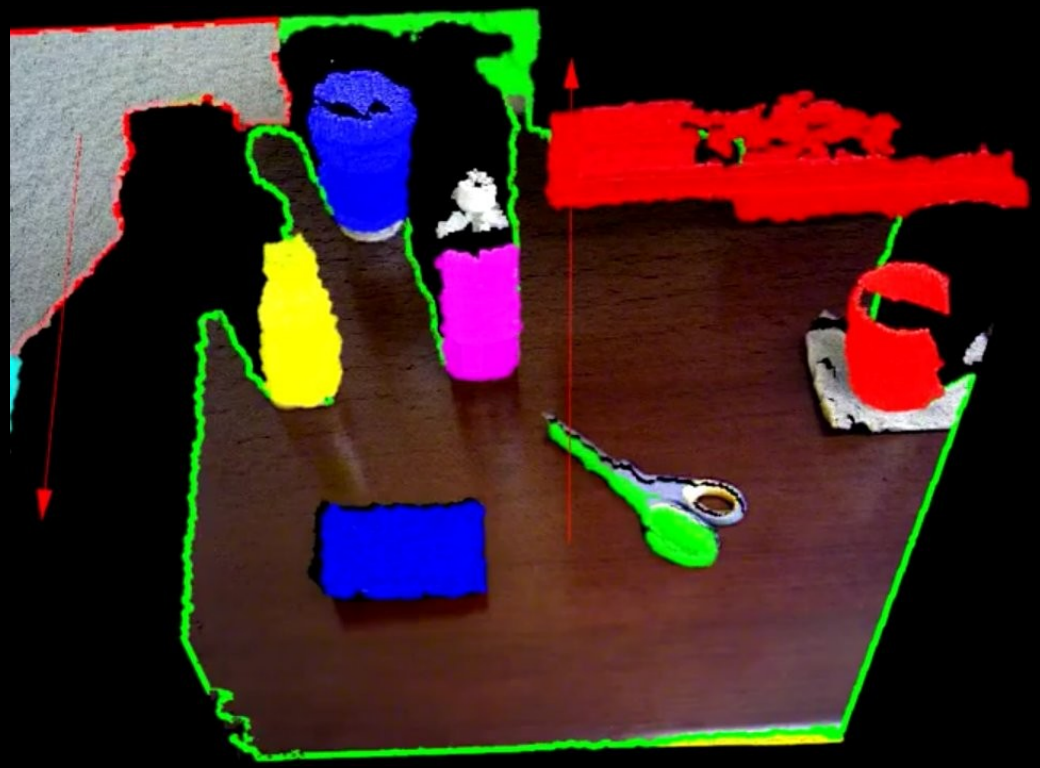
- Point clouds enable very cool applications
- We are trying to solve extremely hard problems



pcl

Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems



pcl

Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems

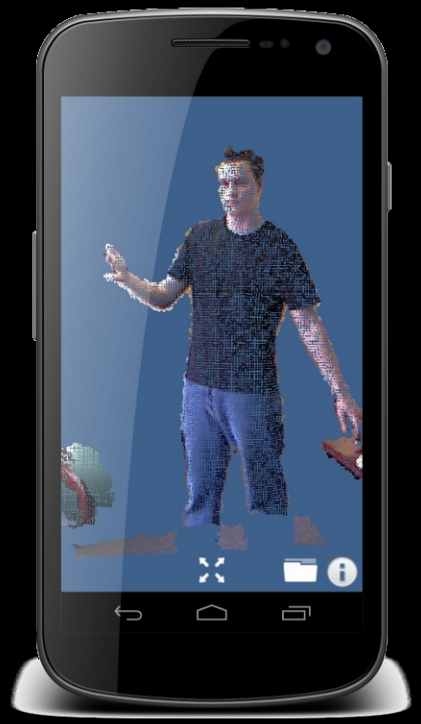
pcl

Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems

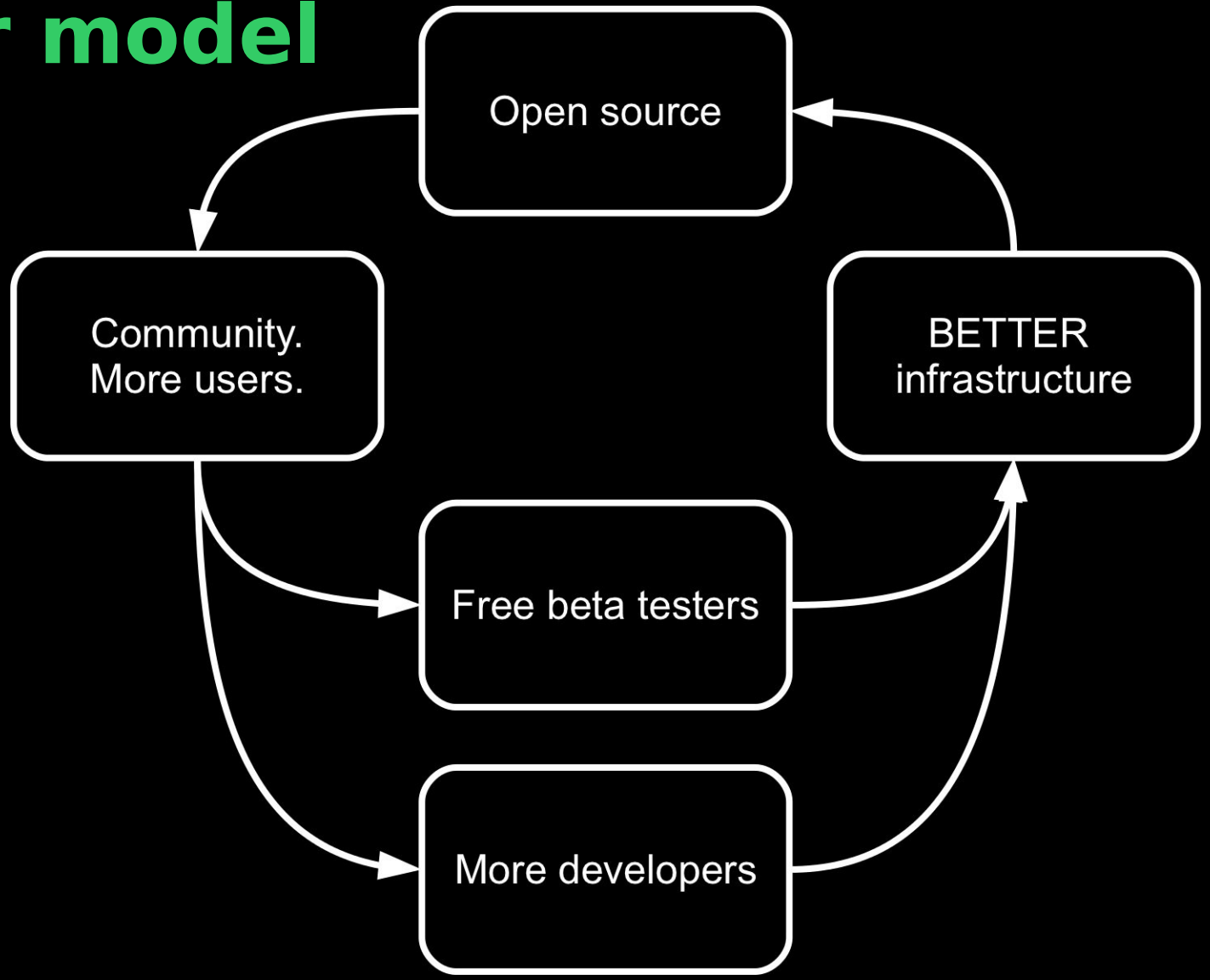
Applications of Point Clouds

- Point clouds enable very cool applications
- We are trying to solve extremely hard problems



pcl

Our model



GPU TECHNOLOGY CONFERENCE

Commercial partnerships

pcl

GPU TECHNOLOGY CONFERENCE

Current sponsors

Willow Garage

NVIDIA

Google Summer of Code 2012

Trimble

TOYOTA

Urban Robotics

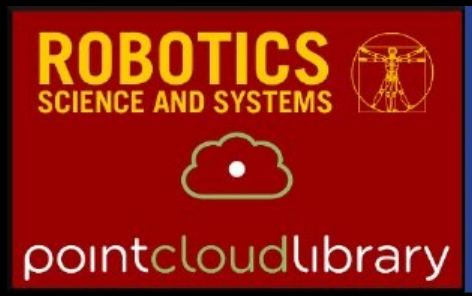
Sandia National Laboratories

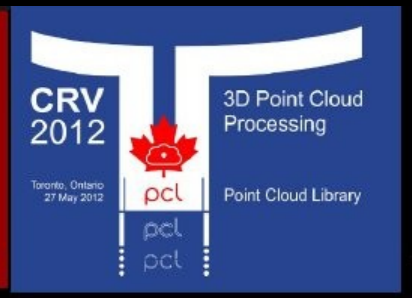
NASA

HRI

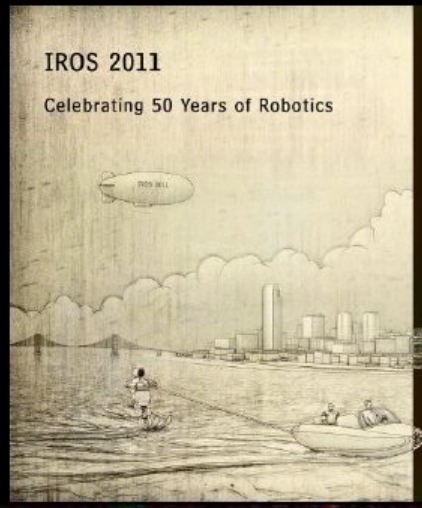
http://pointclouds.org/blog

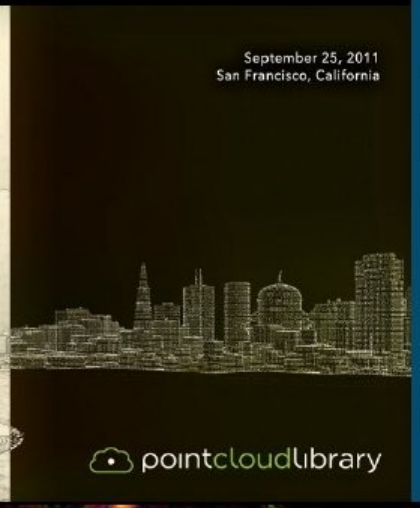
Tutorials

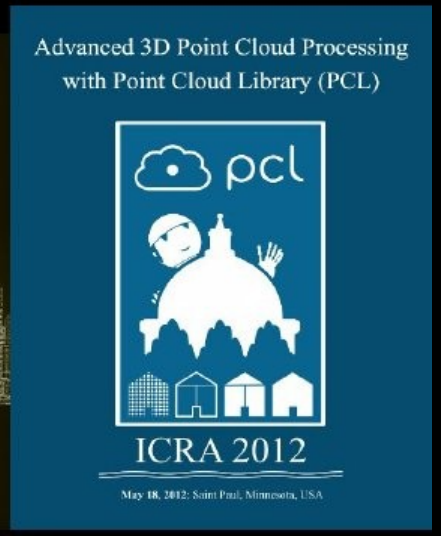




Advanced 3D Image Processing
with Point Cloud Library





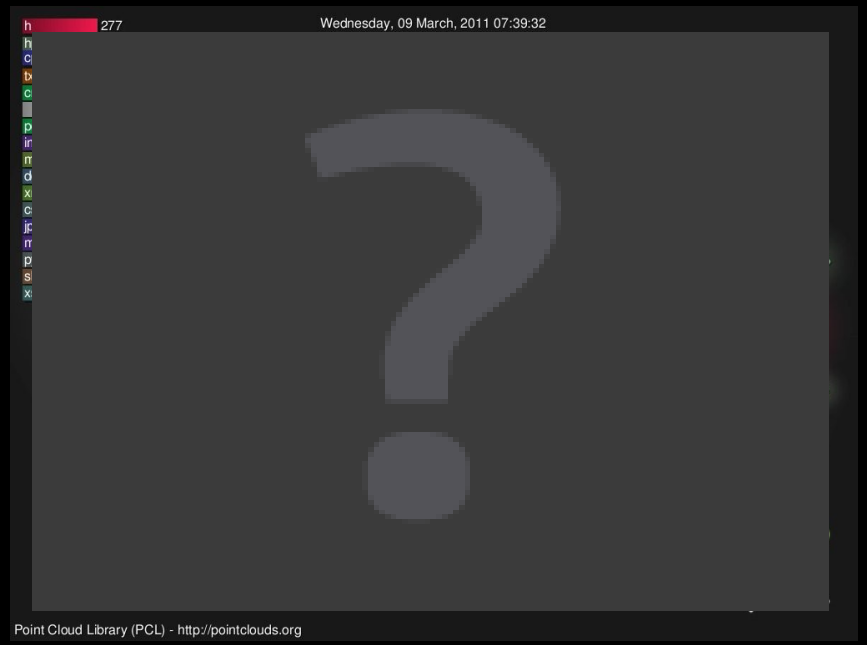


GPU TECHNOLOGY CONFERENCE

pcl

OSS 2011 challenge first prize

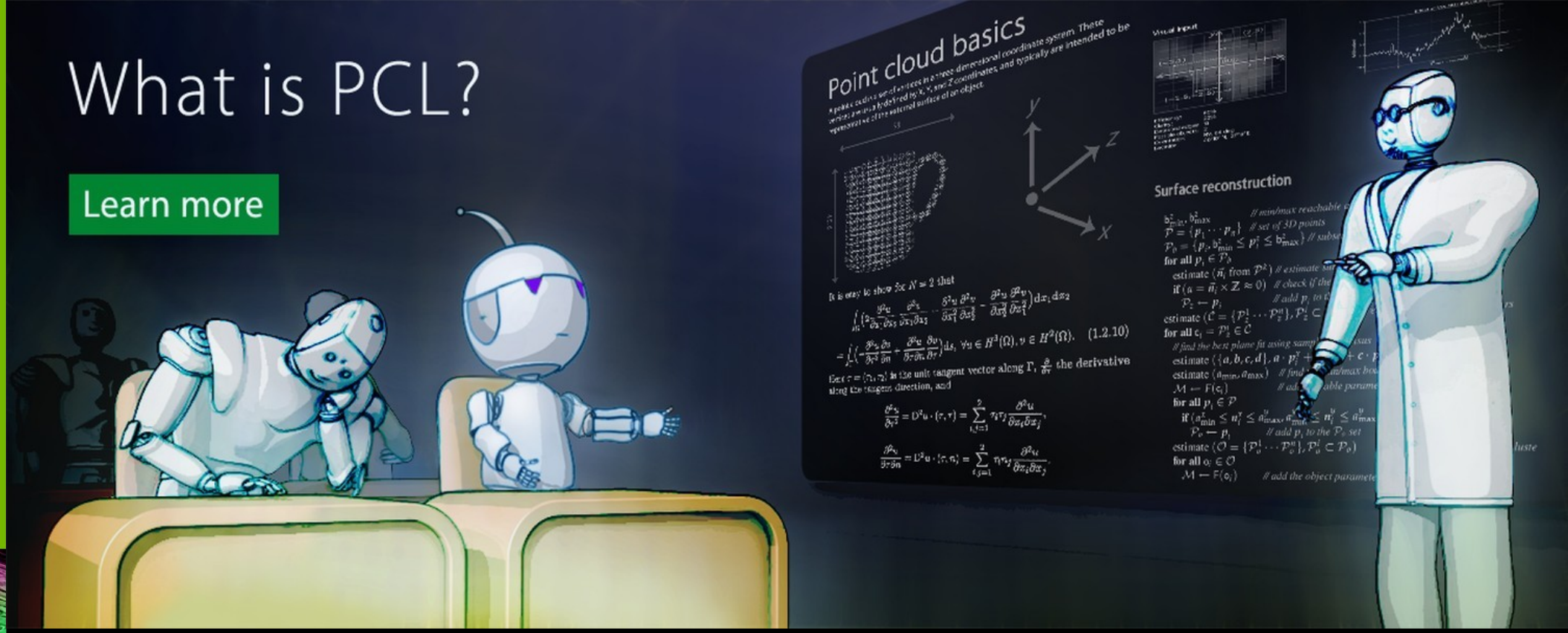
24/7 development



GPU TECHNOLOGY CONFERENCE

What is PCL?

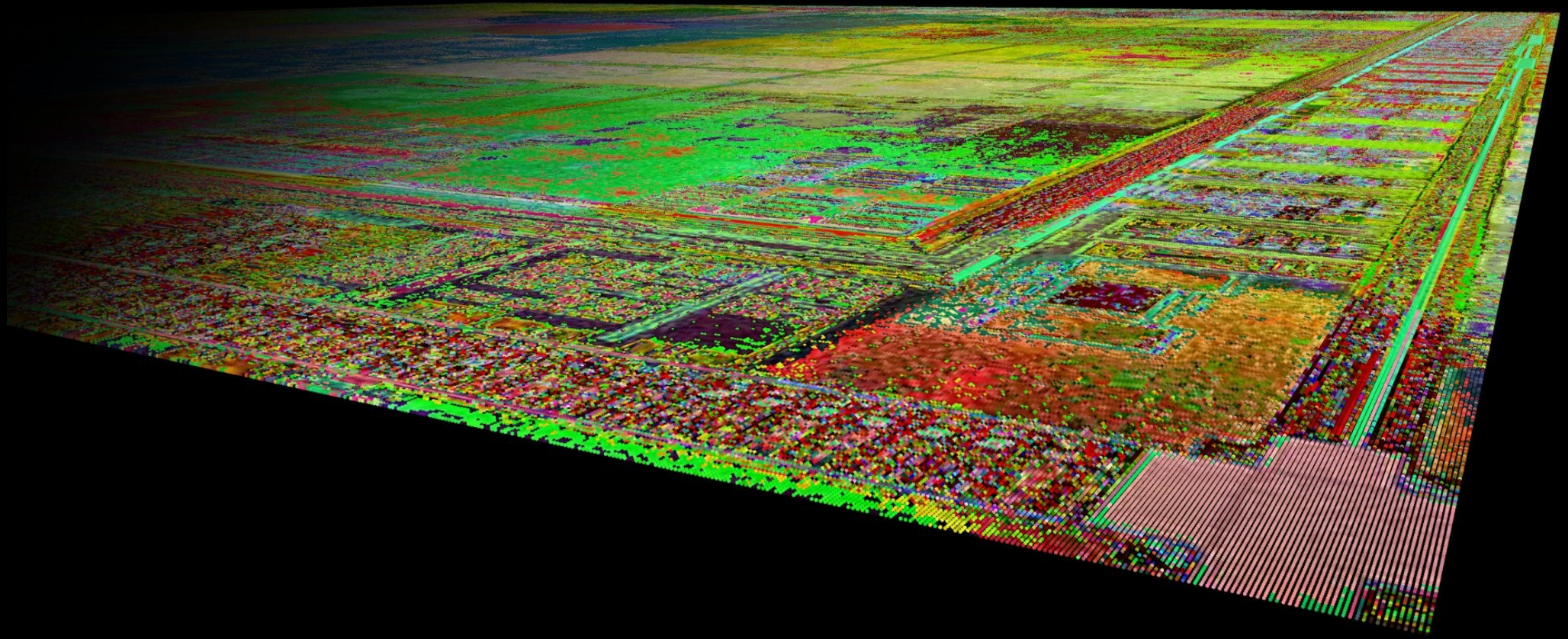
Learn more



- General overview on PCL – Radu
- CUDA optimizations for KinFu and KinSkel - Michael

pcl

GPU TECHNOLOGY CONFERENCE



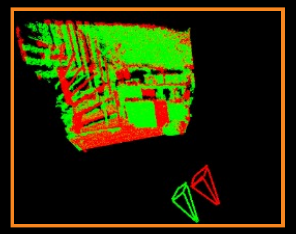
- Extremely large speedups with CUDA processing!
  - depth to cloud assembly: 10x
  - 3D feature estimation: 50-500x
- We are enabling research otherwise not possible!

pcl

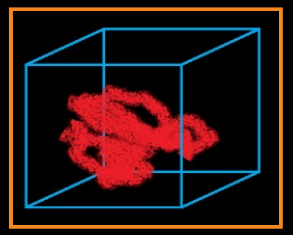
Kinect Fusion

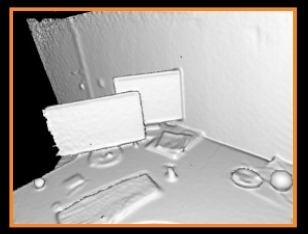
first open source implementation

Cloud & Normals → Camera tracking (frame-to-model) → Volumetric integration → Ray casting



model





References:

– R.Newcombe at all, “Kinectfusion: Real-time dense surface mapping and tracking”. In ISMAR, IEEE, 2011.

Kinect Fusion – camera tracking ICP

- Want to compute: α, β, ϒ, tx, ty, tz
- At each iteration
  - Find matching points between two frames in parallel
  - Cost: Sum of point-to-plane distances for all matching pixels

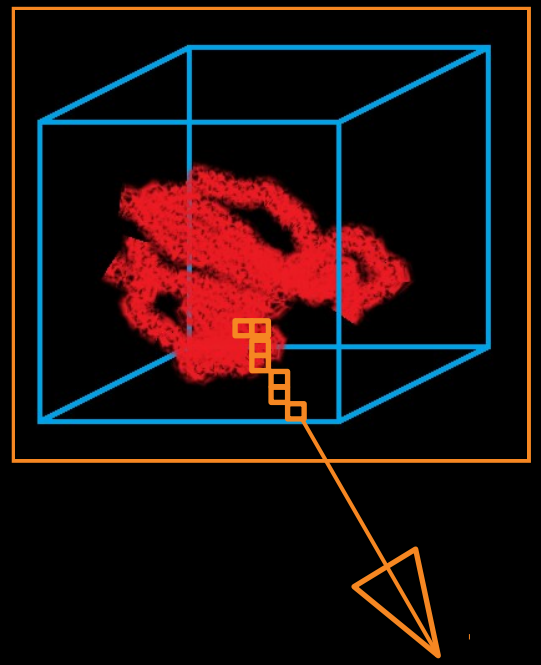
  - Lest squares problem Ax = b (6x6)
  - Coefficients are computed using standard GPU reduction
  - over 640x480 pixels, next linear system is solved on CPU

Kinect Fusion – volume integration

- Voxel grid 512x512x512 (voxels contain avg. distance to surface)

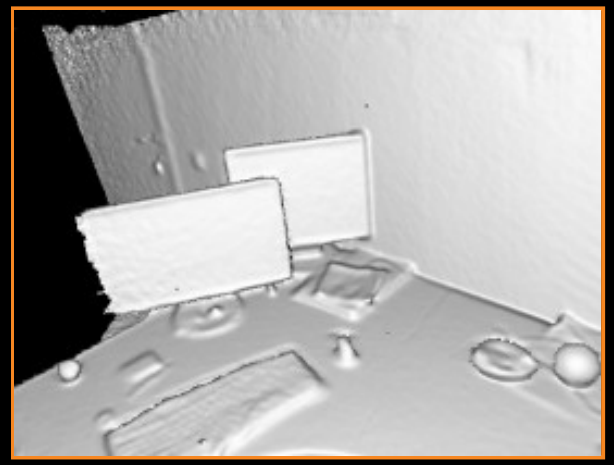
Know current distance measurements
- for all voxels in camera frustum

- For each voxel in parallel on GPU
  - Project voxel to Kinect image
  - Get depth from pixel and update voxel value (using running average)



Kinect Fusion – parallel raycasting

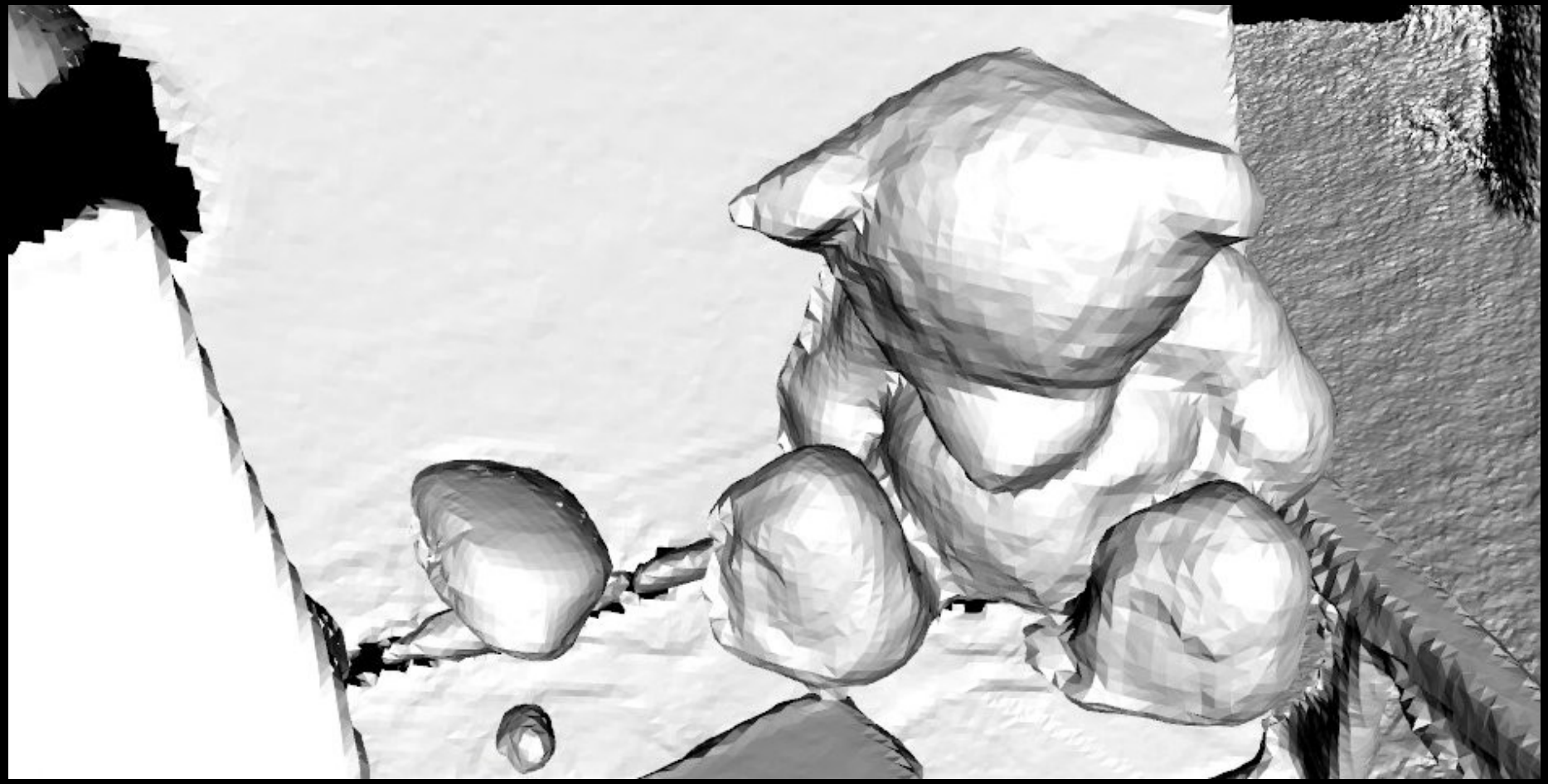
- Want to measure exact distance to surface for each pixel from current camera position.



- For each pixel in parallel on GPU
- Ray is traced through integration volume
  – until surface intersection

Output is noise-less and much more precise than frame from Kinect (because volume contains averaged data from different positions)

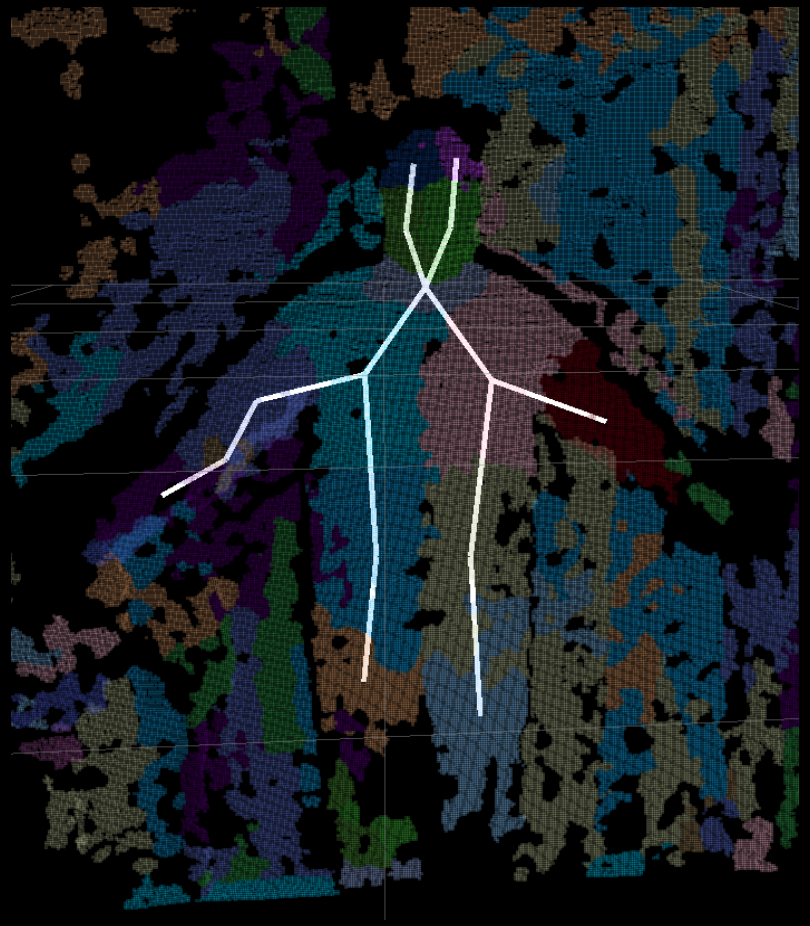
Kinect Fusion – demo



GPU TECHNOLOGY CONFERENCE

Kinect Skeleton Tracking

first open source implementation

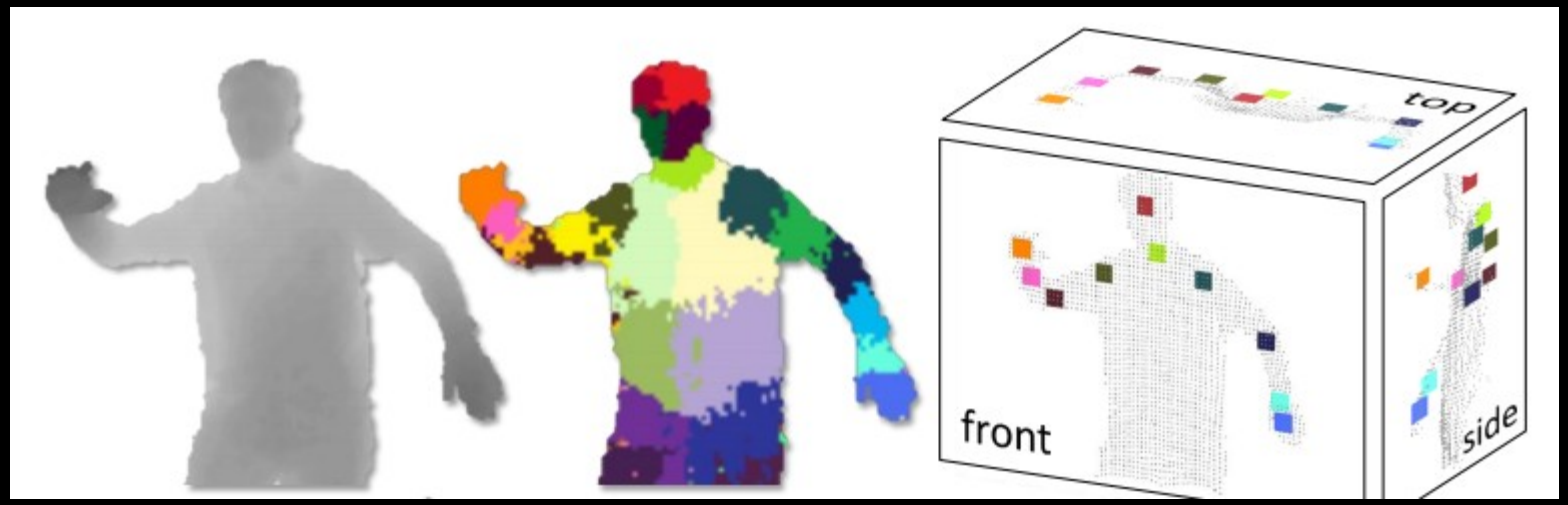


pcl

Algorithm overview

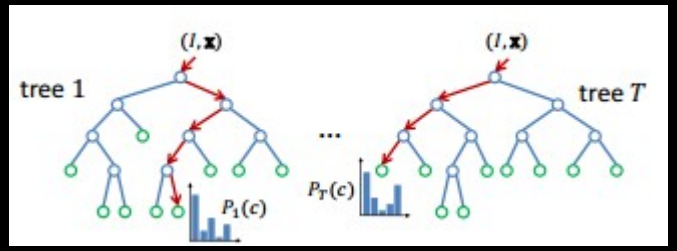
Given a depth image...

- **Segment** the person from the background
- **Label** each pixel with a body part
- **Cluster** part labels to locate skeletal joints

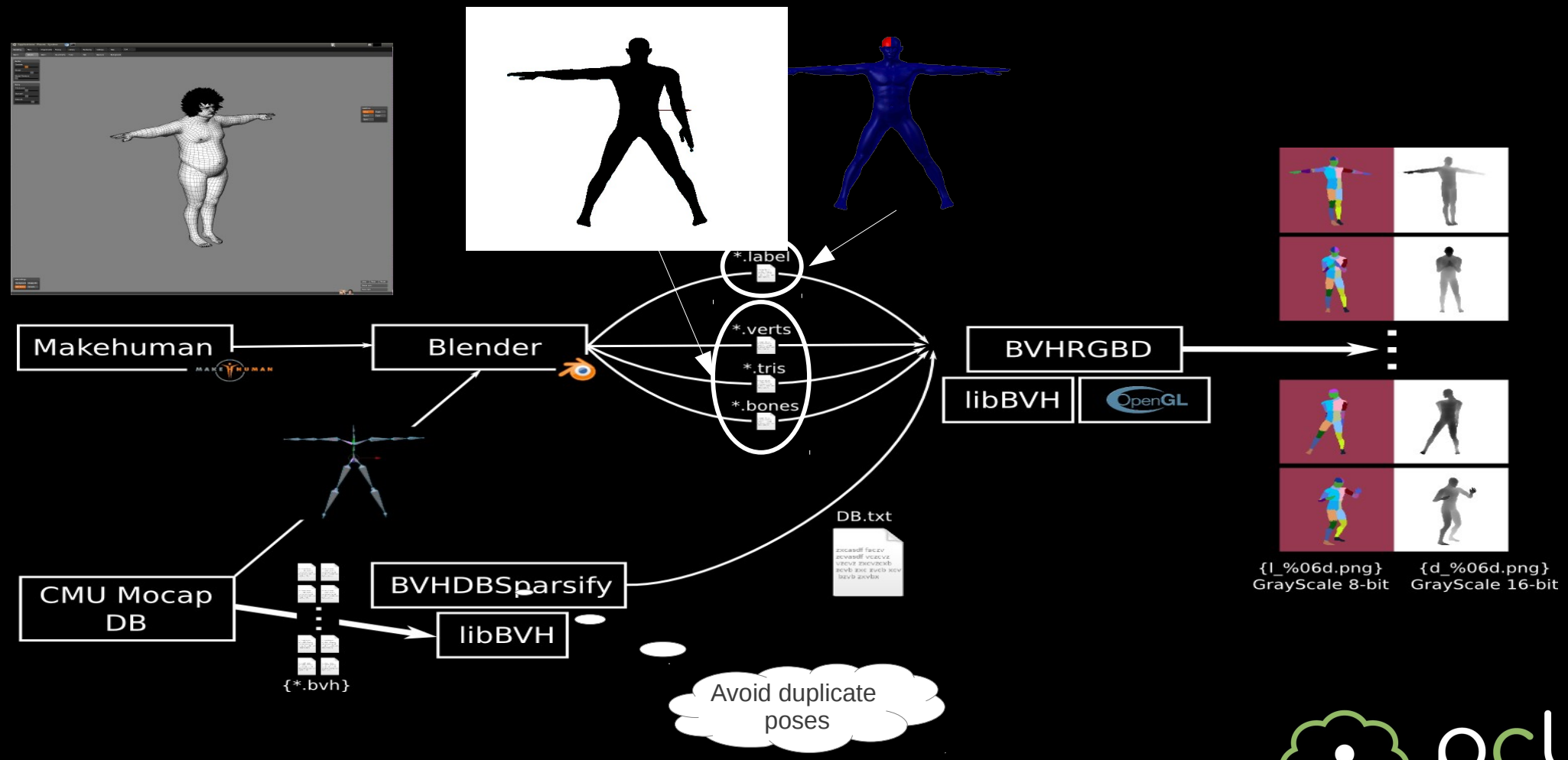


Part labeling

- Use a “forest” of decision trees to classify each pixel
- Train these decision trees using **lots** of synthetic data



Synthesizing training data

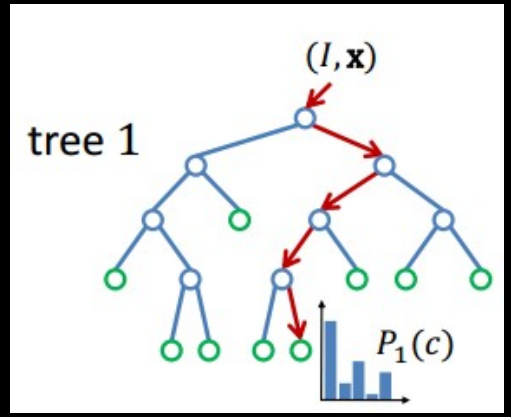


Training data

- Use virtual body to render lots of depth maps and labels
- Sample lots of small patches from each image
- Train decision trees to predict the label for each patch

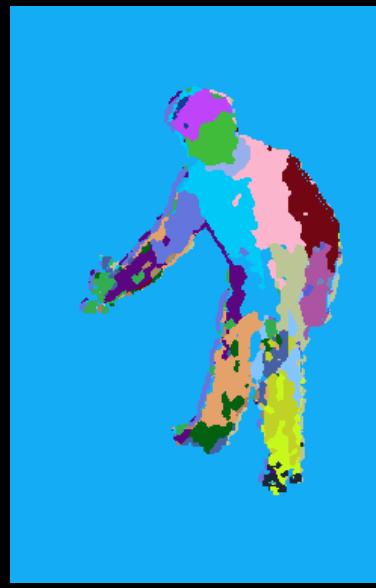
Decision tree structure

- Each node evaluates a simple local feature based on two pixel locations and a threshold (chosen during training)
- To evaluate: subtract the depth values at the given pair of pixels and compare to the difference to the given threshold
- If less than the threshold branch left, otherwise branch right
- Recurse until it reaches a leaf node
  - Leaf node contains the probability distribution of each body part label

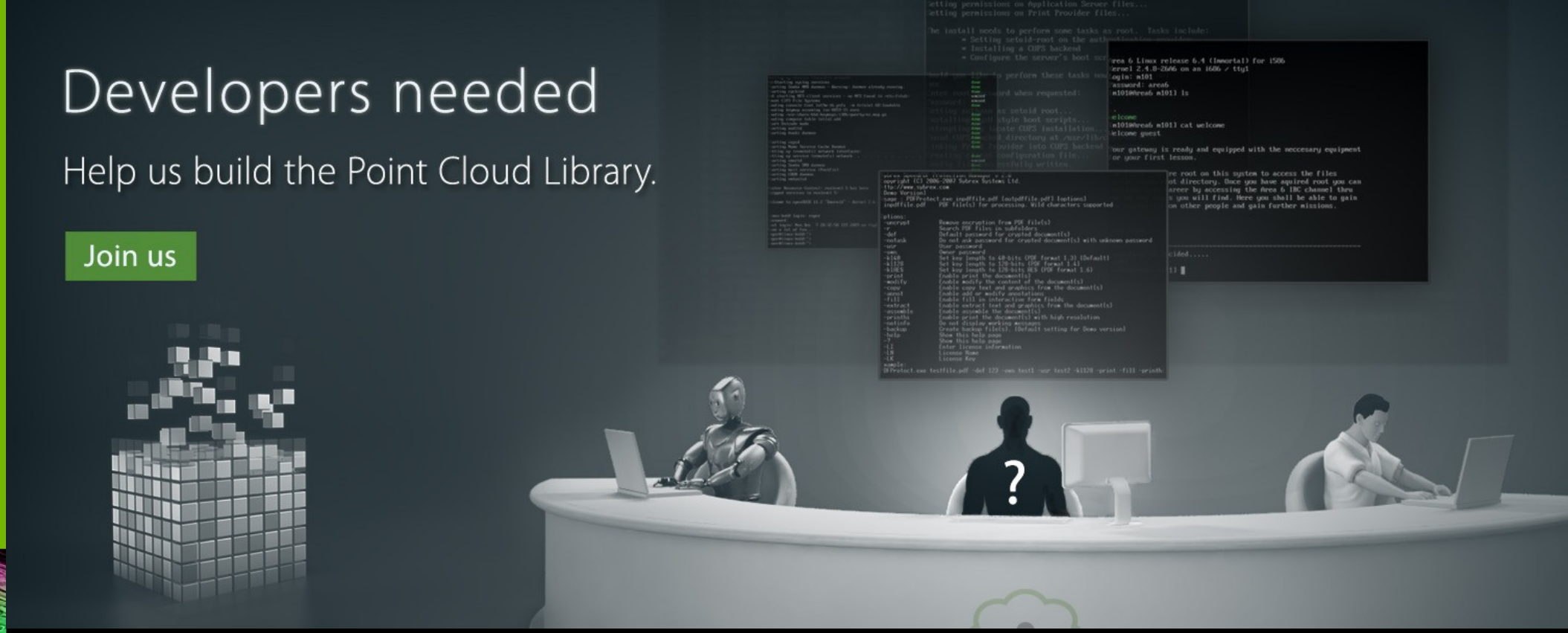


Skeleton tracking

- Remove the background
- Use each tree in the forest to label each pixel
  - Highly parallel
- Average the outputs from each tree into one label image
- Cluster each part label to find the joint position



GPU TECHNOLOGY CONFERENCE



Thanks to our amazing developer community!

(Aitor, Alex^2, Anatoly, Caroline, Francisco, Julius, Koen, Pat, Raphael, Raymond, Ryohei, Suat, Walter)

pcl