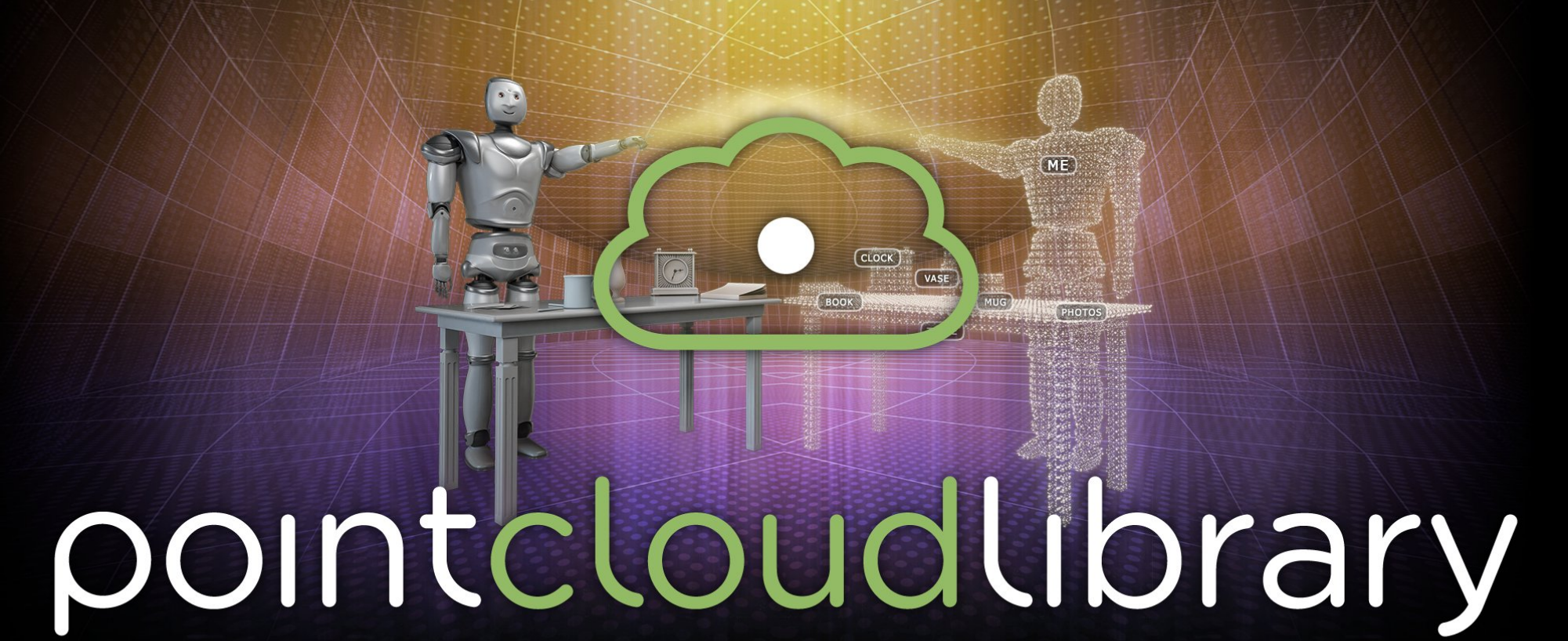


Point Cloud Library (PCL)

Radu B. Rusu

over 110 developers/contributors worldwide!... and counting

July 1, 2011

Outline

1. Why
2. How
3. PCL
4. Apps
5. Specifics

pointcloudlibrary

Why (1/16)

What is this all about?

pointcloudlibrary

Why (2-10/16)

Before we begin...

1. name is not important (!!!). We can call this:

pointcloudlibrary

Why (2-10/16)

Before we begin...

1. name is not important (!!!). We can call this:
   - 3D Library

pointcloudlibrary

Why (2-10/16)

Before we begin...

1. name is not important (!!!). We can call this:
   - 3D Library
   - 3D Toolkit

pointcloudlibrary

Why (2-10/16)

Before we begin...

1. name is not important (!!!). We can call this:
   - 3D Library
   - 3D Toolkit
   - 3D Point Cloud Toolkit

pointcloudlibrary

Why (2-10/16)

Before we begin...

1. name is not important (!!!). We can call this:
   - 3D Library
   - 3D Toolkit
   - 3D Point Cloud Toolkit
   - Point Cloud Library

pointcloudlibrary

Why (2-10/16)

Before we begin...

1. name is not important (!!!). We can call this:
   - 3D Library
   - 3D Toolkit
   - 3D Point Cloud Toolkit
   - Point Cloud Library
   - Point Cloud Toolkit

pointcloudlibrary

Why (2-10/16)

Before we begin...

1. name is not important (!!!). We can call this:
   - 3D Library
   - 3D Toolkit
   - 3D Point Cloud Toolkit
   - Point Cloud Library
   - Point Cloud Toolkit
   - Poop 3D... ☺

pointcloudlibrary

Why (2-10/16)

Before we begin...

1. name is not important (!!!). We can call this:
   - 3D Library
   - 3D Toolkit
   - 3D Point Cloud Toolkit
   - Point Cloud Library
   - Point Cloud Toolkit
   - Poop 3D... ◡̈
2. what matters: we need 3d infrastructure for the masses!

pointcloudlibrary

Why (2-10/16)

Before we begin...

1. name is not important (!!!). We can call this:
   - 3D Library
   - 3D Toolkit
   - 3D Point Cloud Toolkit
   - Point Cloud Library
   - Point Cloud Toolkit
   - Poop 3D... ◡̈
2. what matters: we need 3d infrastructure for the masses!
3. analogy to Boost: modular, compact, efficient, fast, open

pointcloudlibrary

Why (11/16)

Goal: Solve n-D perception problems!

pointcloudlibrary

Why (12/16)

n-D Perception???

- What is he talking about? What is n-D perception?

- at the very least $n = 3$: x,y,z.
- + RGB = 4D
- + normal information = 7D
- + surface curvature = 8D
- + ... = n-D

pointcloudlibrary

Why (13/16)

The world is not 2D

*

Sony 3D TV Commercial © Sony Electronics Inc., USA

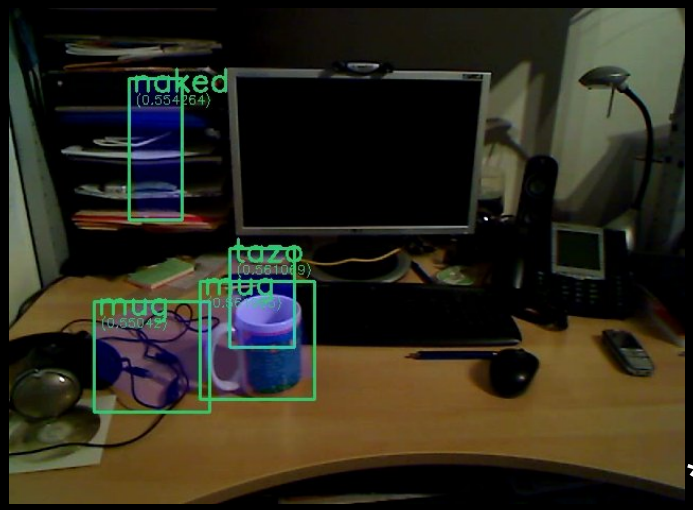
there's no robotics without 3D geometry

pointcloudlibrary

Why (14/16)

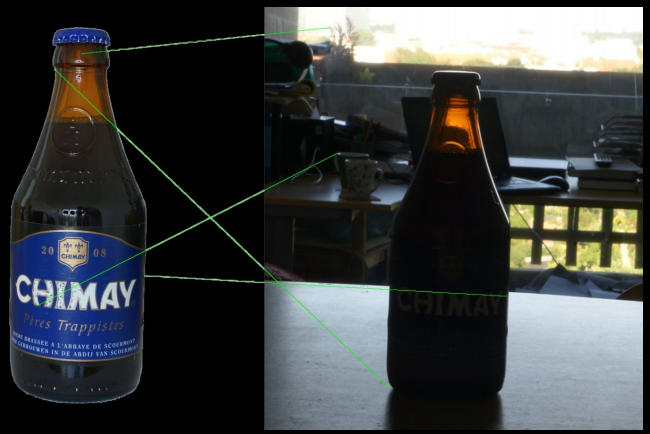
2D thinking might be slowing us down

- great progress in 2D computer vision



*

- still poised by false positives. not good enough for robotics.



pointcloudlibrary

Why (15/16)

3D has arrived!

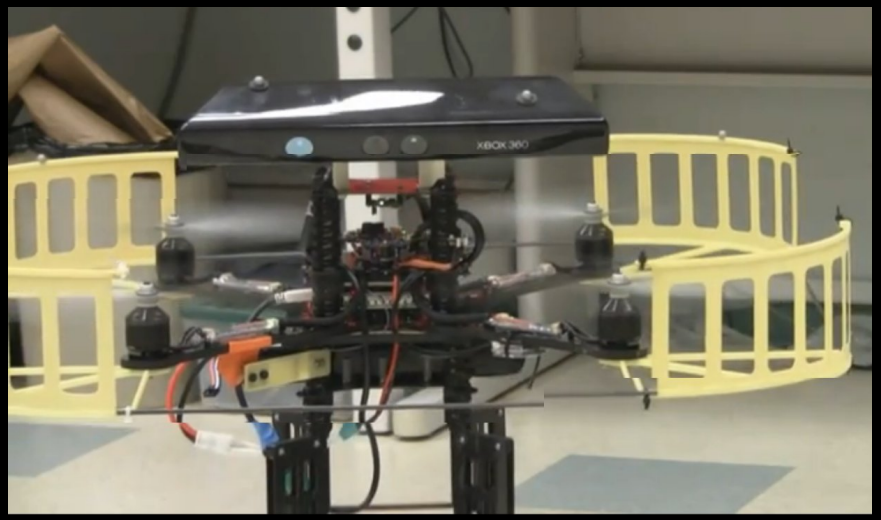
- Kinect was just the beginning
- Asus XtionPRo is next
- other devices coming soon

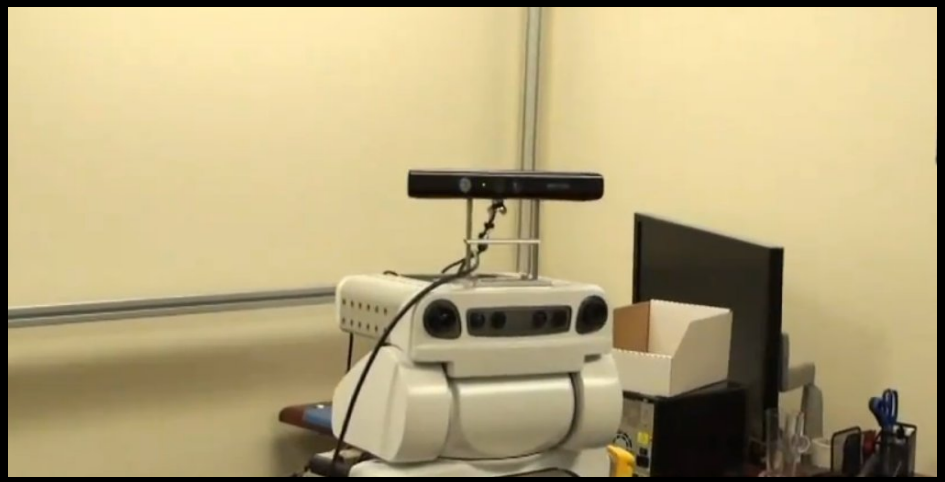
- 3D pocket cameras!
- high resolution, decent SNR
- potential for good close range 3D
- at some point => back to stereo!

pointcloudlibrary

Why (16/16)

Kinect enables robotics





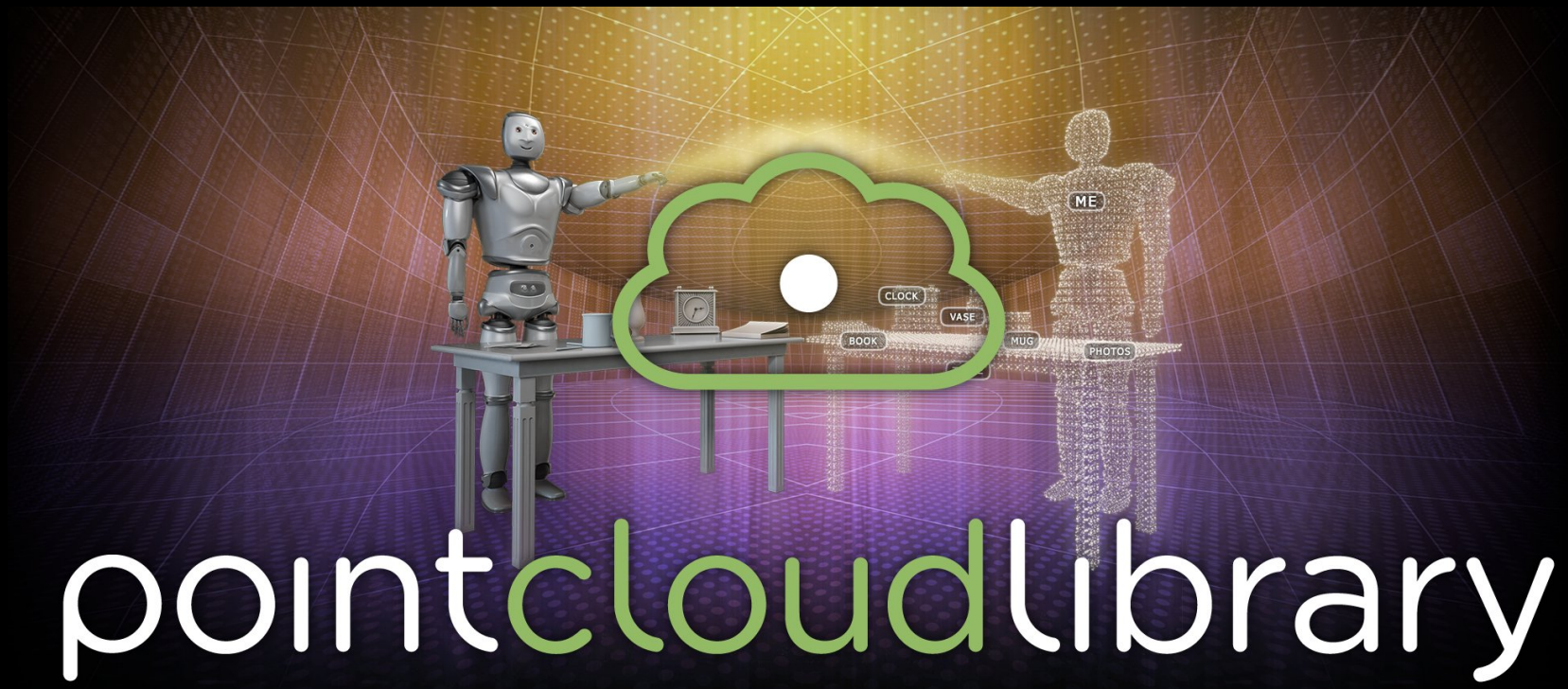
© UC Berkeley, MIT, Willow Garage

Outline

1. Why
2. How
3. PCL
4. Apps
5. Specifics

pointcloudlibrary

How (1/5)



http://pointclouds.org/

pointcloudlibrary

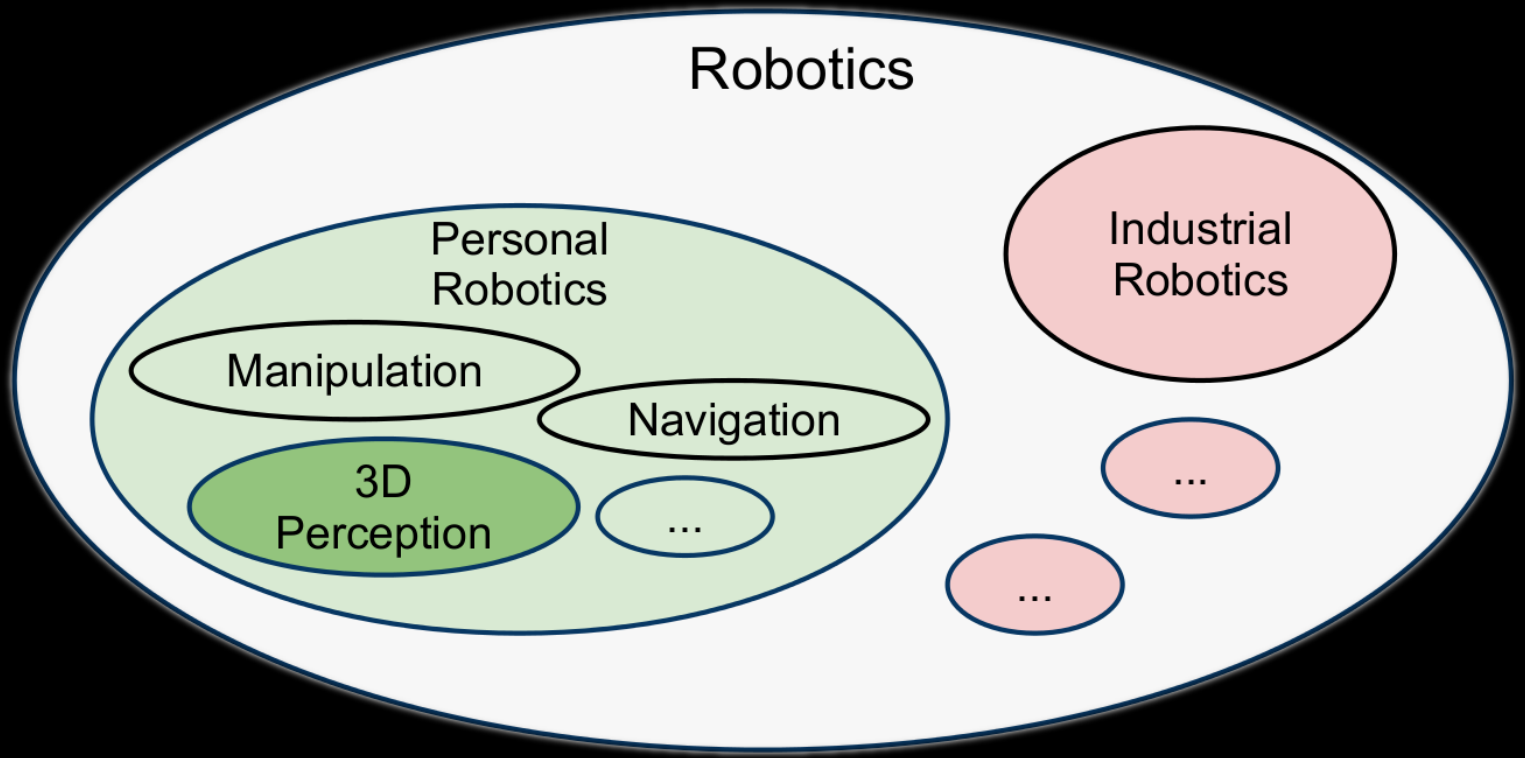
How (2/5)

What is PCL (Point Cloud Library)?

PCL is:

- a large scale, open project[1] for 3D point cloud processing

We are basically changing our attitude from this:



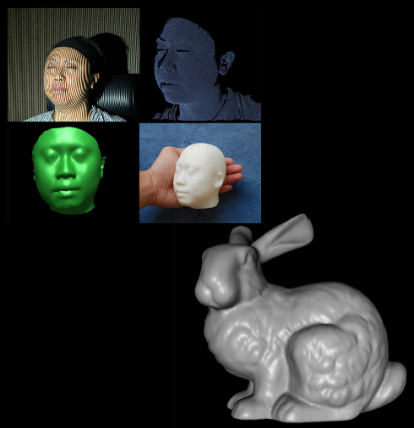
[1] source code is BSD licensed, management process is completely open

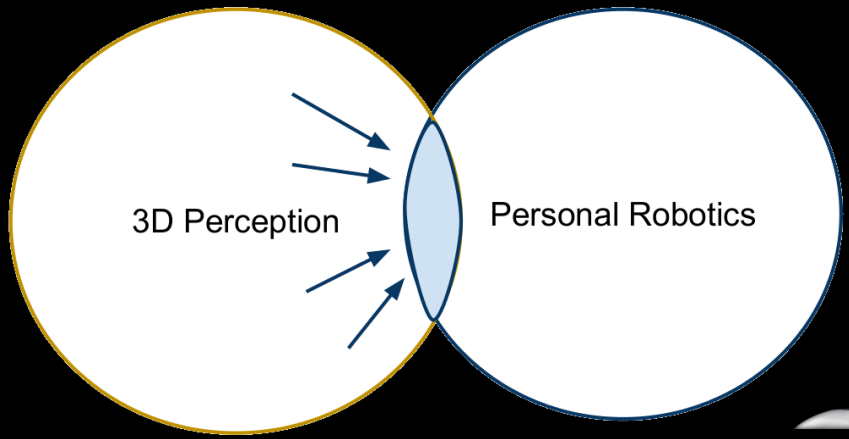
pointcloudlibrary

How (3/5)

What is PCL (Point Cloud Library)?

to this:





This means:

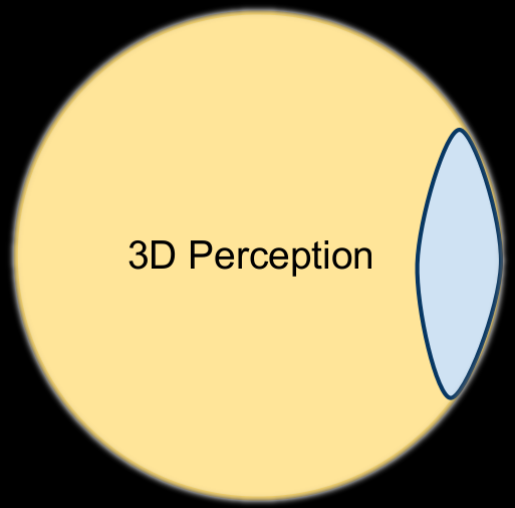
- their complexity is equally high
- many things in perception (that we do not think of) could help robotics

pointcloudlibrary

How (4/5)

What is PCL (Point Cloud Library)?

- leverage the broader (non-robot) 3D perception community
- help advance specific goals in personal robotics

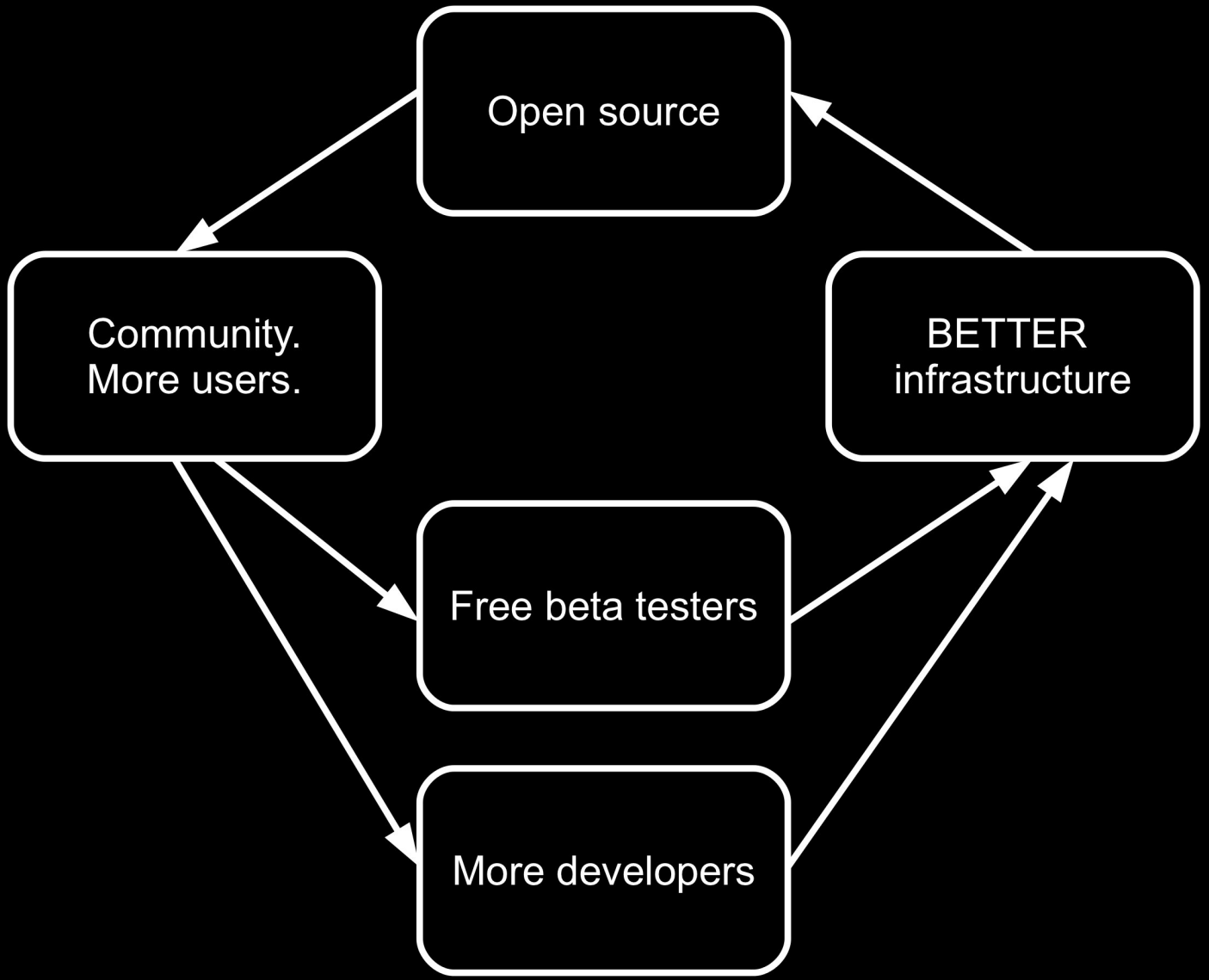


- many people don't care about robotics, but they care about perception
- bring them to robotics later

pointcloudlibrary

How (5/5)

What is PCL (Point Cloud Library)?



pointcloudlibrary

Outline

1. Why
2. How
3. PCL
4. Apps
5. Specifics

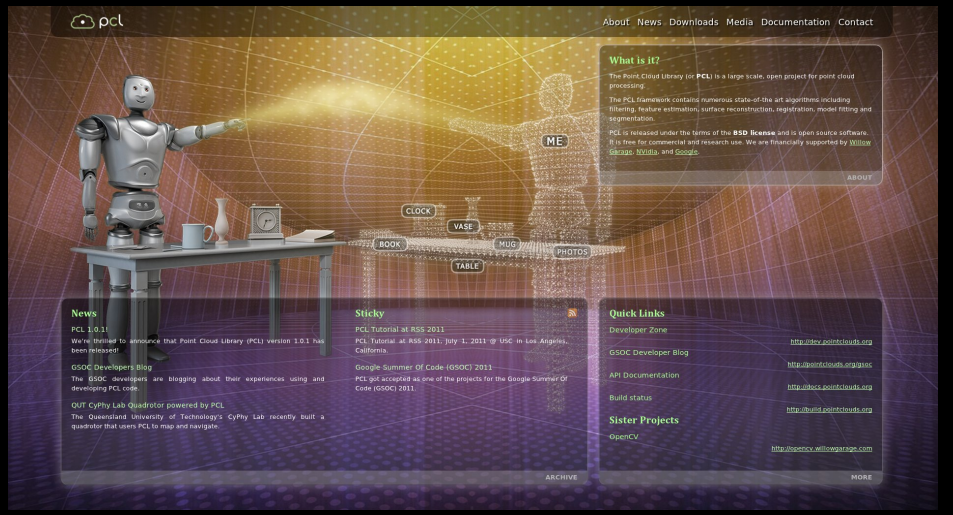
pointcloudlibrary

PCL (1/3)

Statistics

`www.pointclouds.org`

- News
- Videos and examples
- Tutorials and support
- Facebook/Twitter ◡̈



Statistics (3 months):

- over 500,000 views
- over 90,000 visits
- 6 continents
- 138 countries
- over 4,000,000 SVN repository hits

pointcloudlibrary

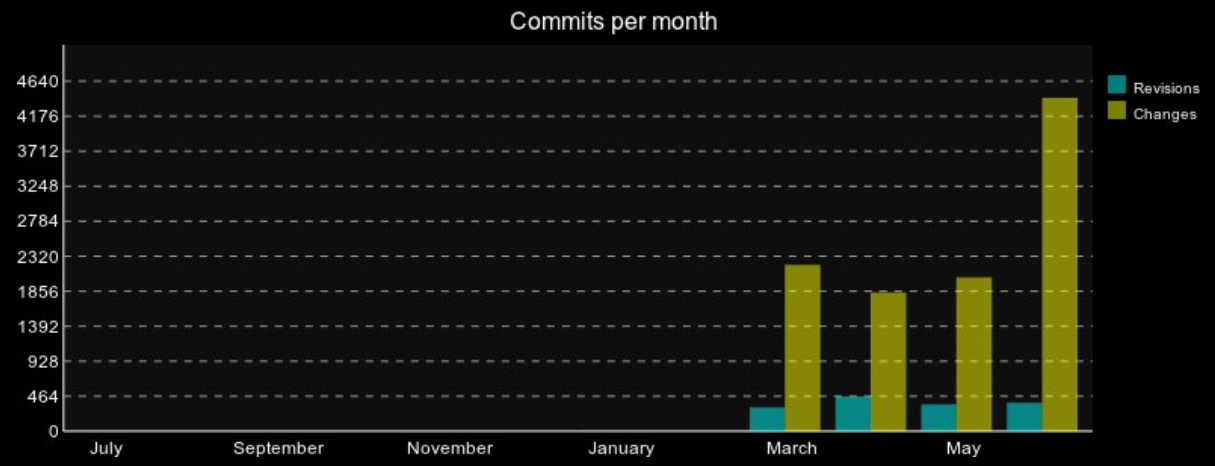
PCL (2/3)

An incredible community-driven push for 3D!

over 110 active developers and contributors world-wide (!)

pointcloudlibrary

PCL (3/3)



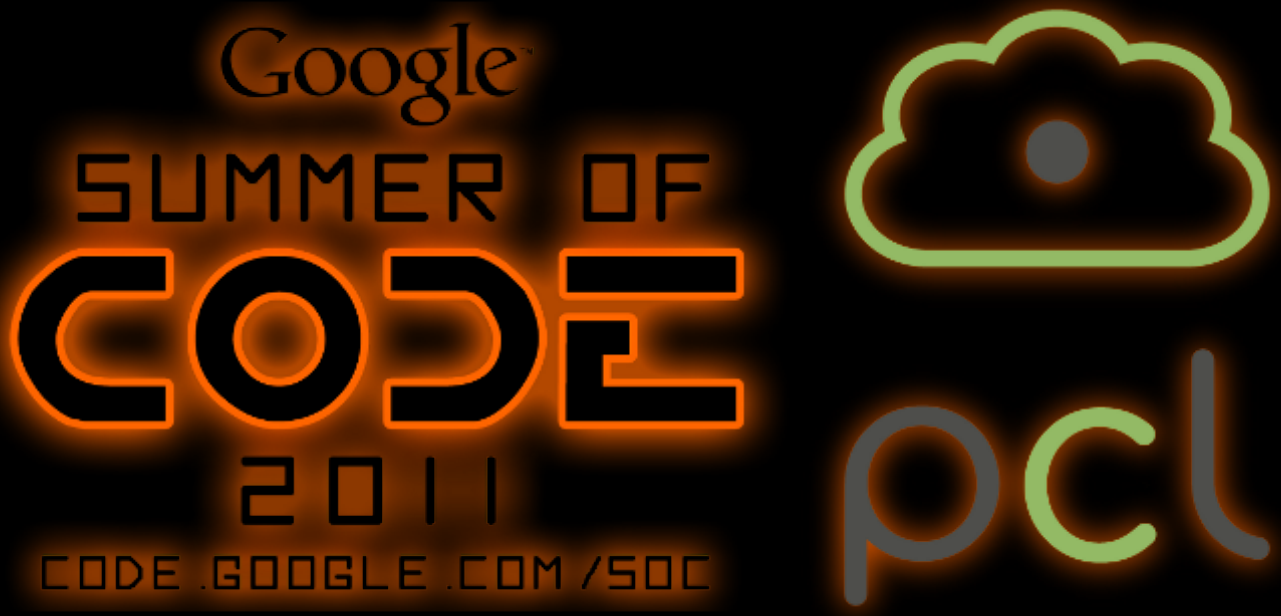
dev.pointclouds.org

- Source repository, Downloads
- Issue tracking
- Wiki
- Mailing lists: pcl-users@ & pcl-developers@

pointcloudlibrary

GSOC

- 175 out of 417 organizations accepted only (!)
- PCL accepted (!) ☺
- got 11 slots

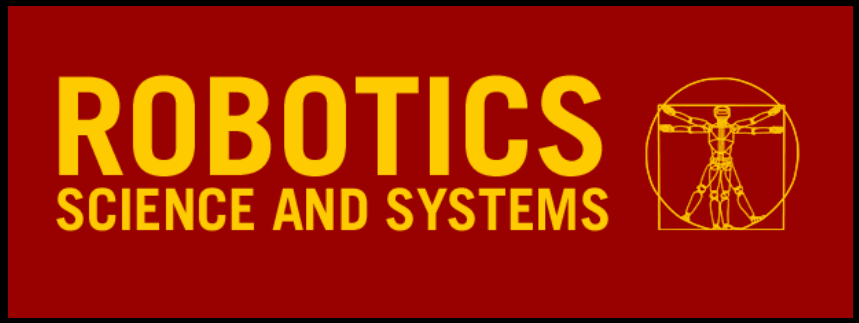


pointcloudlibrary

Tutorials

conferences: RSS, IROS, ICCV, CVPR

- RSS - Robotics Science and Systems :: July 1



- IROS - Intelligent Robots and Systems :: September 25

- ICCV - International Conference on Computer Vision

Outline

1. Why
2. How
3. PCL
4. Apps
5. Specifics

pointcloudlibrary

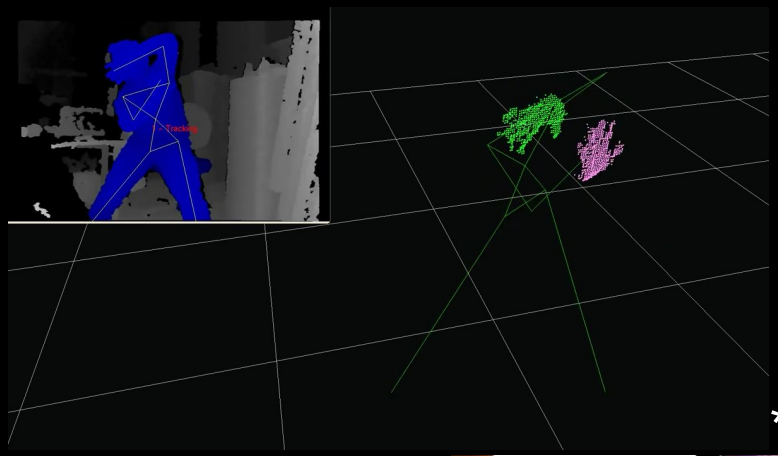
Demo Applications (1/5)

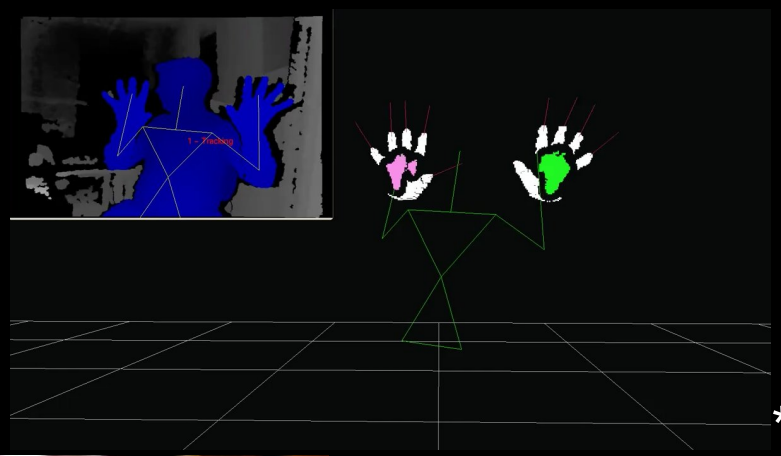
PCL Application Examples :: ROS3D contest

pointcloudlibrary

Demo Applications (2/5)

PCL Application Examples :: ROS3D contest





© Garratt Gallagher, MIT

pointcloudlibrary

Demo Applications (3/5)

PCL Application Examples :: ROS3D contest

*

© Albert-Ludwigs-Universität Freiburg

pointcloudlibrary

Demo Applications (4/5)

PCL Application Examples :: ROS3D contest

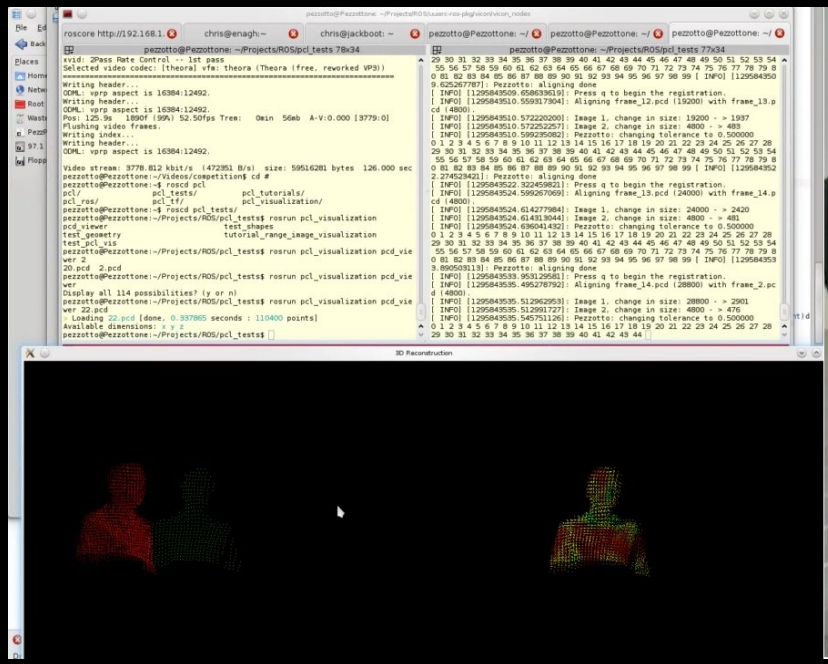
*

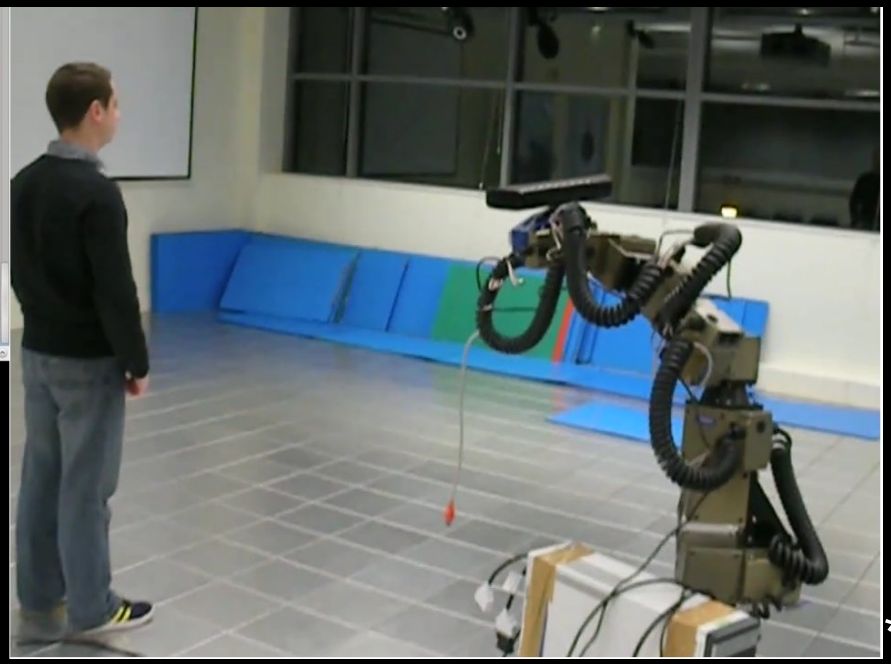
© UC Berkeley

pointcloudlibrary

Demo Applications (5/5)

PCL Application Examples :: ROS3D contest





*

© University of Ulster

pointcloudlibrary

Outline

1. Why
2. How
3. PCL
4. Apps
5. Specifics

pointcloudlibrary

PCL 1.0.1 (1/8)

1.0.1 released

pcl

About News Downloads Media Documentation Contact

PCL 1.0.1!

Posted on Jun 29, 2011 | Tags: release, 1.0.1

We're thrilled to announce that Point Cloud Library (PCL) version 1.0.1 has been released!

We're thrilled to announce that Point Cloud Library (PCL) version 1.0.1 has been released!

pcl 1.0.1

PCL 1.0.1 is a patch release, API compatible with 1.0. Here's a few of the release highlights:

- please note that version 1.0.0 had a flaw when creating ASCII pcd files. This version includes the tool pcd_convert_NaN_nan to fix this
- added VTK file visualization to pcd_viewer
- hiding the cminpack/FLANN headers, thus reducing compile time for user code
- fixed IntegralImageNormalEstimation
- tutorial updates and fixes + new tutorials. Changed tutorial structure to split CPP files from RST text.
- better doxygen documentation for many functions
- fixed a bug in ConditionalRemovalFilter where the keep_organized condition was reversed
- removed BorderDescription and Histogram<2> from the list of explicit template instantiations
- added PointXY point registration macros
- added ExtractIndicesSelf unit test
- fixed a lot of alignment issues on 32bit architectures
- PCD ascii files now have each individual line trimmed for trailing spaces
- internal changes for PCDReader/PCDWriter, where NAN data is represented as "nan"
- sped up compilation with MSVC by adding /MP for multiprocessor builds
- added a voxel grid command line tool filter
- issues fixed: #242, #207, #237, #215, #236, #226, #148, #214, #218, #216, #196, #219, #207, #194, #192, #183, #178, #154, #174, #145, #155, #122
- added support for PathScale pathcc compiler
- added support for Intel icc C++ compiler
- added support for GCC 4.6 C++ compiler
- added preliminary support for Clang C++ compiler
- FindPCL.cmake and PCLConfig.cmake completed

New to PCL? Visit our downloads page and give it a try. If you're looking for ideas on how to get started, be sure to check out our growing collection of tutorials.

Other news

pointcloudlibrary

PCL 1.0.1 (2/8)

PCL (Point Cloud Library) structure

PCL

- uses SSE optimizations for fast computations
- uses OpenMP for parallelization (Boost.MPI to come!)
- data passing between modules using shared pointers
- uses CUDA for GPGPU programming (trunk only atm)
- unit tests, examples, tutorials (!)
- apps: higher level + integration

pointcloudlibrary

PCL 1.0.1 (3/8)

PCL (Point Cloud Library) structure

...split into a collection of smaller, modular C++ libraries:

- libpcl_keypoints: nD interest points
- libpcl_features: nD feature descriptors
- libpcl_surface: surface meshing/reconstruction techniques
- libpcl_filters: point cloud data filters and smoothing
- libpcl_io: I/O operations, 3D camera drivers (e.g., Kinect)
- libpcl_kdtree: fast nearest neighbor operations
- libpcl_octree: downsampling, compression, change detection
- libpcl_range_image: efficient 3D operations
- libpcl_sample_consensus: RANSAC, MSAC, MLESAC, planes, spheres, etc
- libpcl_segmentation: model segmentation operations
- libpcl_registration: point cloud registration methods
- libpcl_visualization: 2D/3D visualization library

pointcloudlibrary

PCL 1.0.1 (4/8)

1.0.1 released

- full support for OpenNI (Kinect, PrimeSense, Asus XTion)

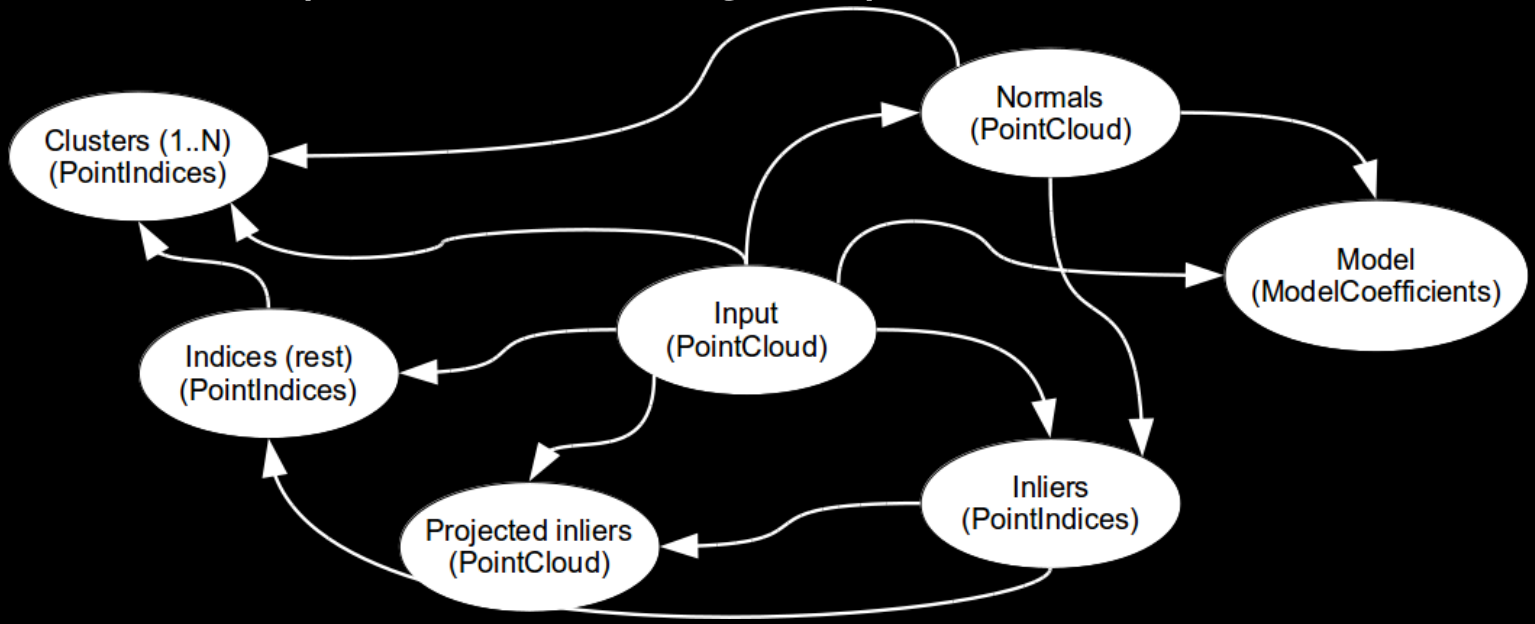
- Linux, Windows, MacOS support (next: Android)

pointcloudlibrary

PCL 1.0.1 (5/8)

PPG: Perception Processing Graphs

- Philosophy: *write once, parameterize everywhere*
- PPG: Perception Processing Graphs



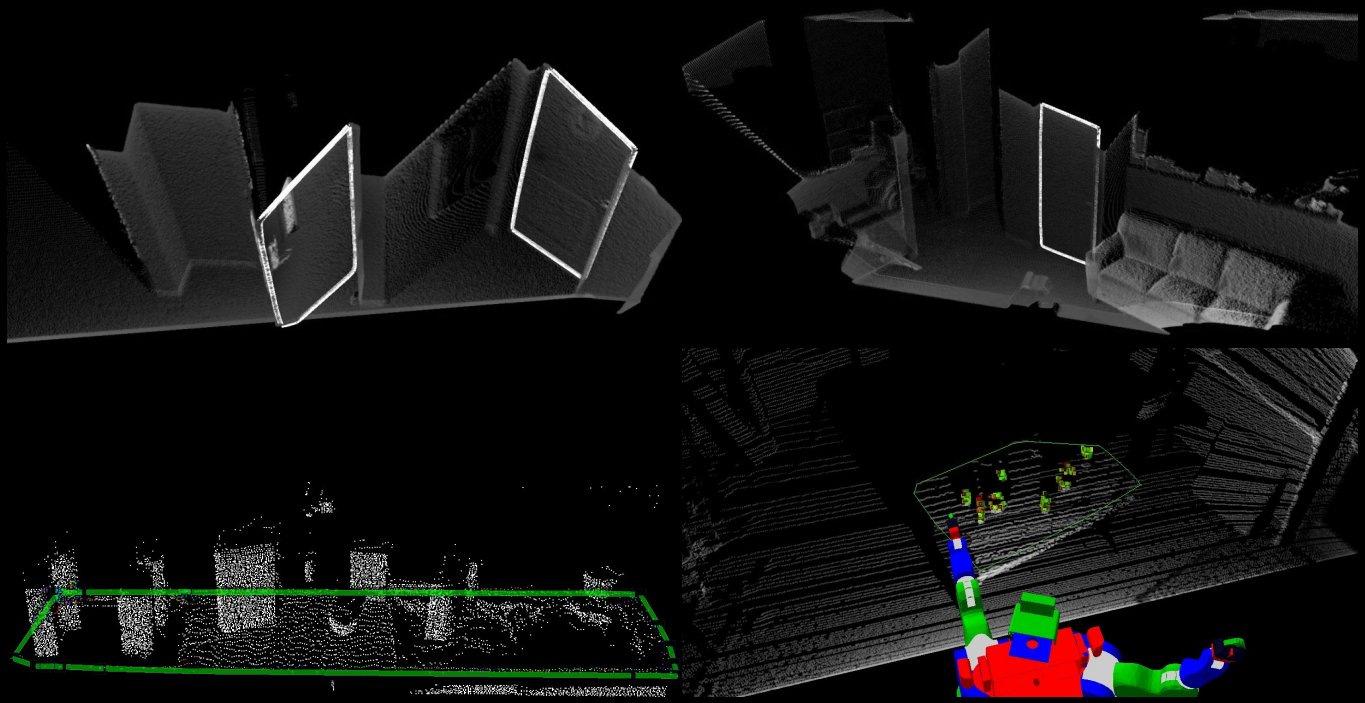
pointcloudlibrary

PCL 1.0.1 (6/8)

PPG: Perception Processing Graphs

Why PPG?

- Algorithmically:
  door detection = table detection = wall detection = ...
- the only thing that changes is: parameters (constraints)!

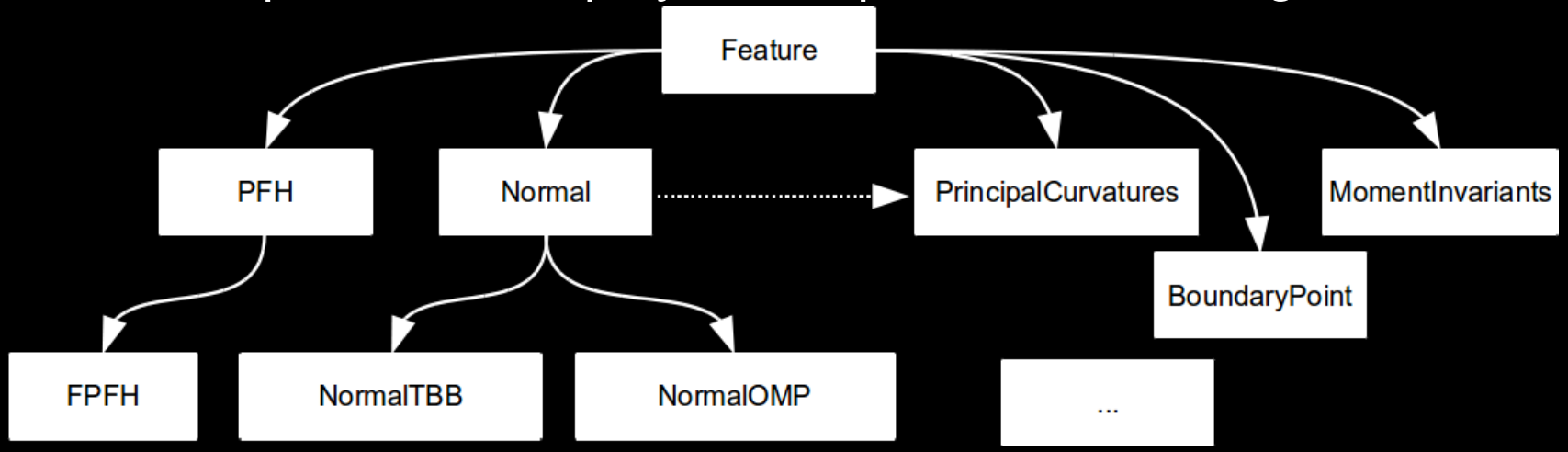


pointcloudlibrary

PCL 1.0.1 (7/8)

More on architecture

Good API practices simplify development and testing:

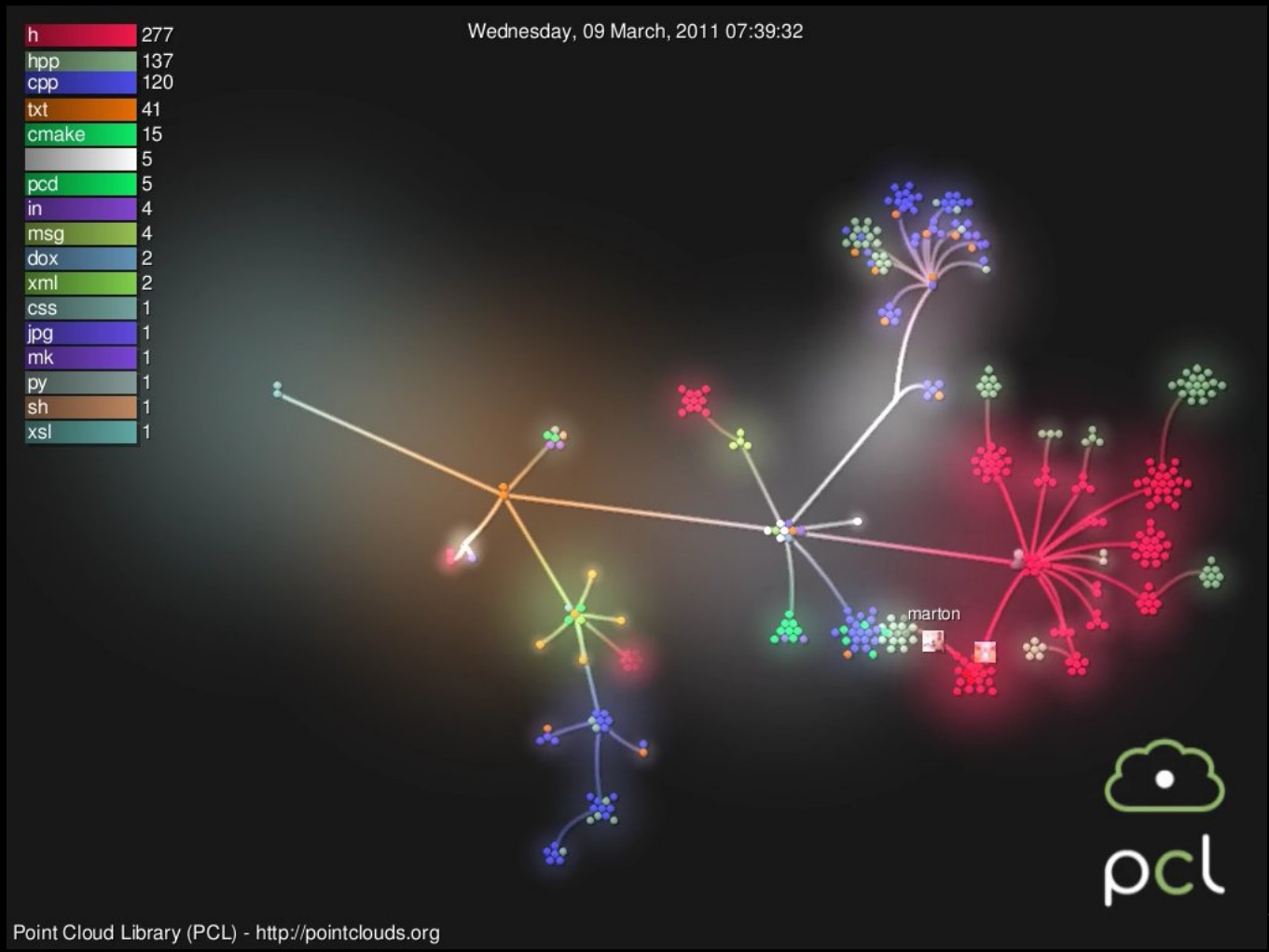



```
pcl::Feature<PointT> feat;
feat = pcl::Normal<PointT> (input);
feat = pcl::FPFH<PointT> (input);
feat = pcl::BoundaryPoint<PointT> (input);
...
feat.compute (&output);
...
```

pointcloudlibrary

PCL 1.0.1 (8/8)

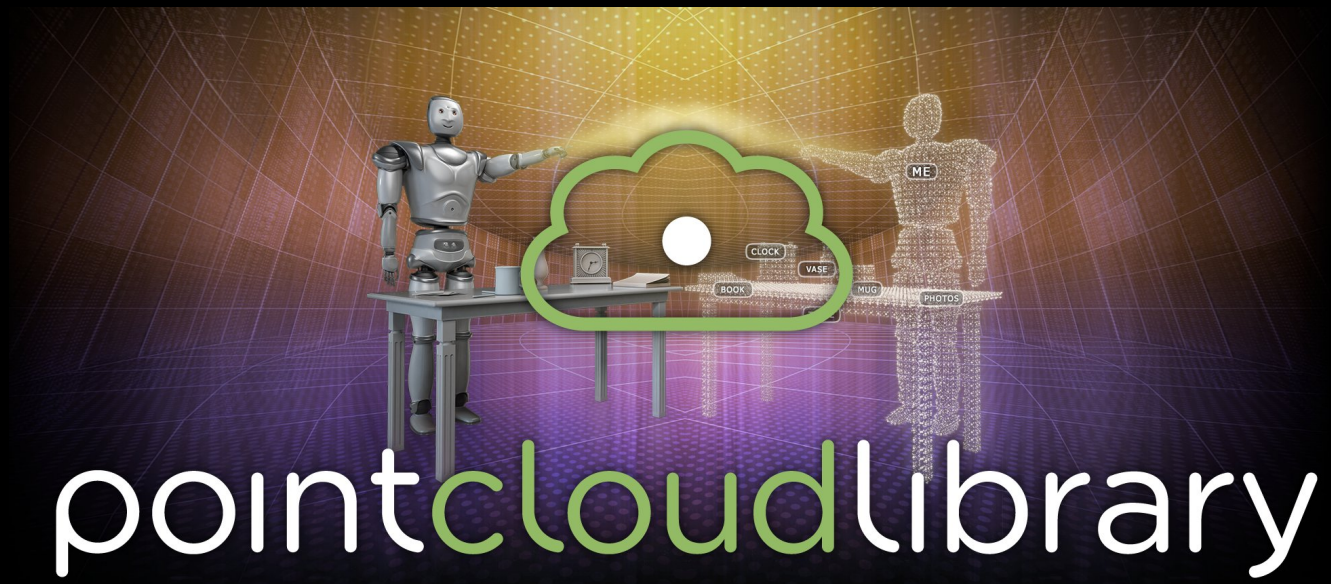
Project activity



*

pointcloudlibrary

Thanks!



*

http://pointclouds.org/