pointcloudlibrary

PCL Hackfest Summer 2016

June 14th TU Munich, Germany

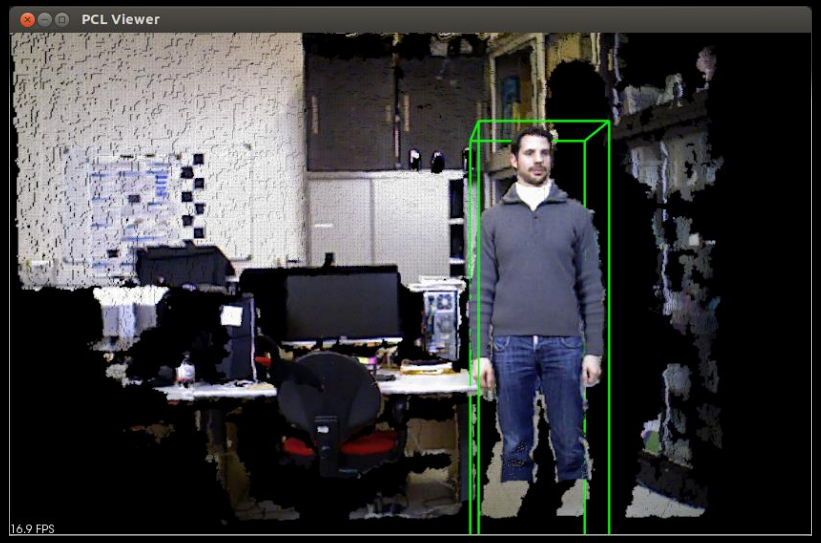
People Detection and Tracking in RGB-D Data

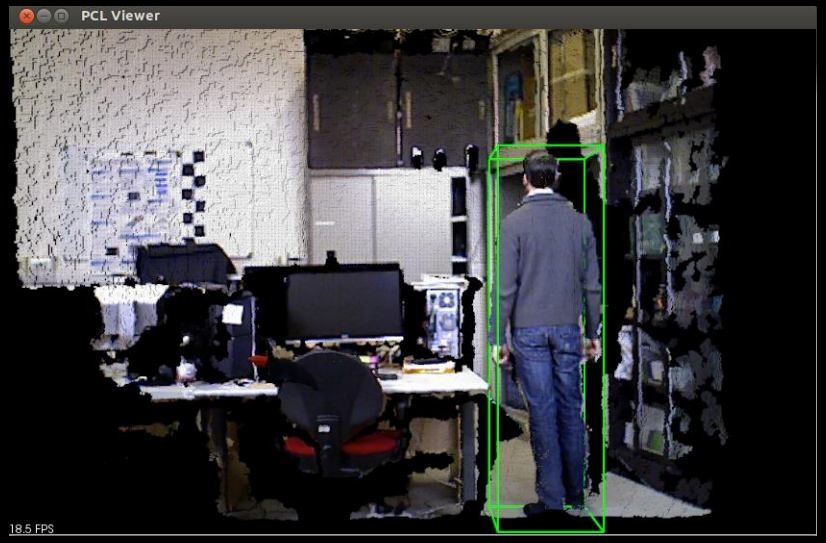
Matteo Munaro
matteo.munaro@dei.unipd.it
Intelligent Autonomous Systems Lab (IAS-Lab)
University of Padova

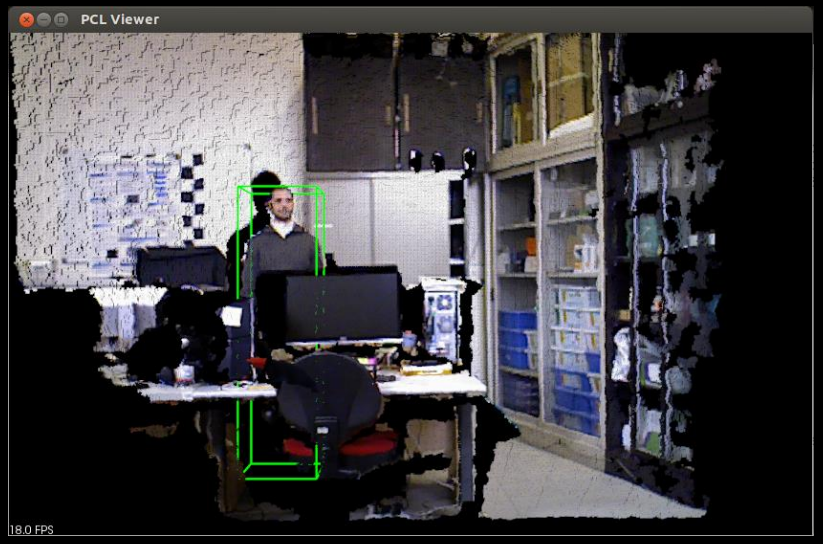
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

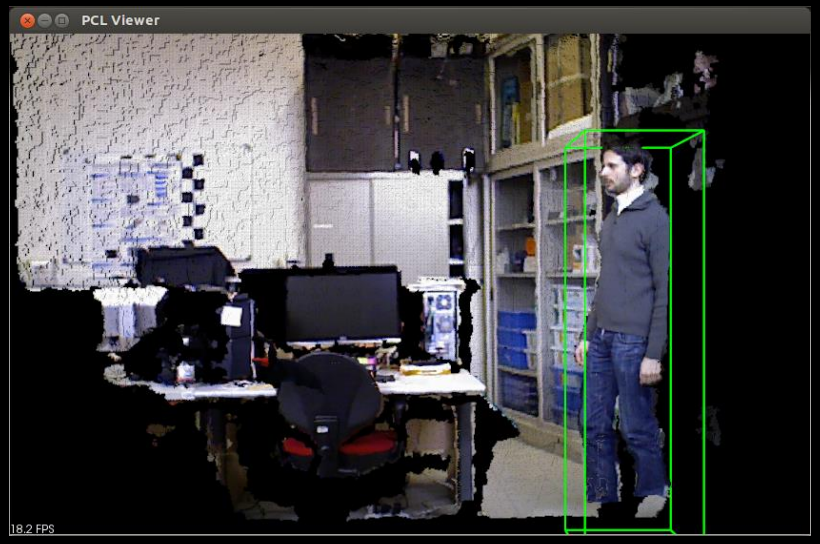
pointcloudlibrary

pcl::people





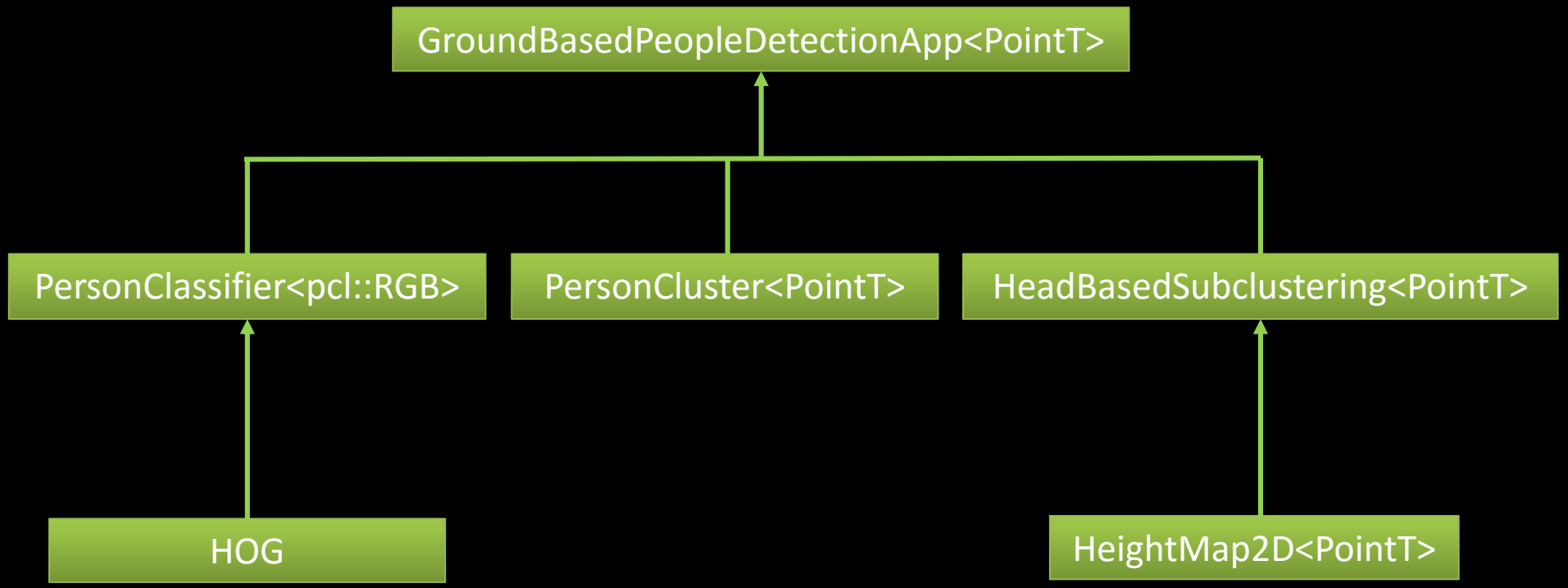




M. Munaro, F. Basso and E. Menegatti. Tracking people within groups with RGB-D data. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2012, Vilamoura (Portugal), 2012.

M. Munaro, E. Menegatti. *Fast RGB-D People Tracking for Service Robots*. Journal on Autonomous Robots, 2014.

pcl::people


GroundBasedPeopleDetectionApp<PointT>
PersonClassifier<pcl::RGB>
PersonCluster<PointT>
HeadBasedSubclustering<PointT>
HOG
HeightMap2D<PointT>

pcl::people

```
#include <pcl/people/ground_based_people_detection_app.h>
...
// Create classifier for people detection:
pcl::people::PersonClassifier<pcl::RGB> person_classifier;
person_classifier.loadSVMFromFile(svm_filename);       // load trained SVM
```

pcl::people

```
#include <pcl/people/ground_based_people_detection_app.h>
...
// Create classifier for people detection:
pcl::people::PersonClassifier<pcl::RGB> person_classifier;
person_classifier.loadSVMFromFile(svm_filename);       // load trained SVM

// People detection app:
pcl::people::GroundBasedPeopleDetectionApp<PointT> people_detector; // people detection object
std::vector<pcl::people::PersonCluster<PointT> > clusters; // vector containing persons clusters
people_detector.setIntrinsics(rgb_intrinsics_matrix);     // set RGB camera intrinsic parameters
people_detector.setClassifier(person_classifier);         // set person classifier
people_detector.setInputCloud(cloud);
people_detector.setGround(ground_coeffs);                 // set floor coefficients
people_detector.compute(clusters);                        // perform people detection
```

pcl::people

```
// Optional settings:
people_detector.setVoxelSize(voxel_size);                    // set the voxel size (0.06)
people_detector.setHeightLimits(min_height, max_height); // set height limits (1.3, 2.3)
people_detector.setDimensionLimits(min_points, max_points); // set dimension limits (30, 5000)
people_detector.setMinimumDistanceBetweenHeads(min_dist); // set minimum distance between persons' heads (0.3)
people_detector.setSensorPortraitOrientation(true); // set portrait/landscape orientation (false)
```

pcl::people

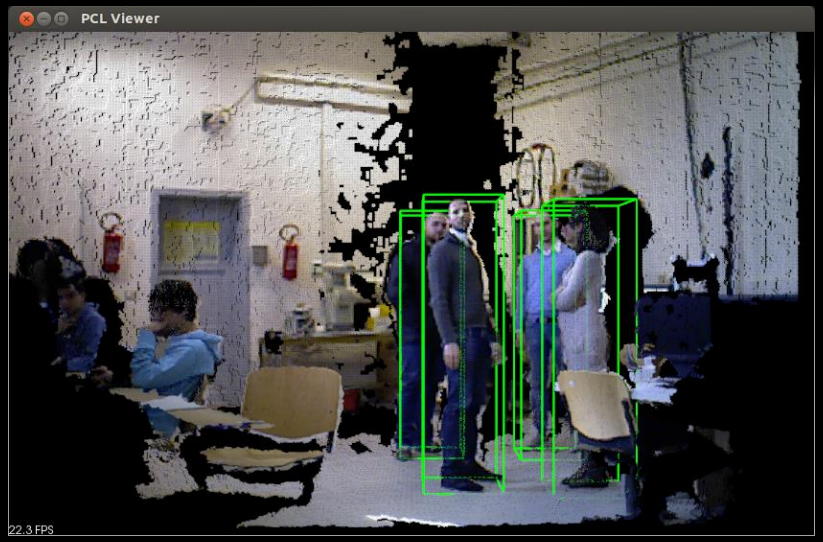
```
// Optional settings:
people_detector.setVoxelSize(voxel_size);                    // set the voxel size (0.06)
people_detector.setHeightLimits(min_height, max_height); // set height limits (1.3, 2.3)
people_detector.setDimensionLimits(min_points, max_points); // set dimension limits (30, 5000)
people_detector.setMinimumDistanceBetweenHeads(min_dist); // set minimum distance between persons' heads (0.3)
people_detector.setSensorPortraitOrientation(true); // set portrait/landscape orientation (false)
```

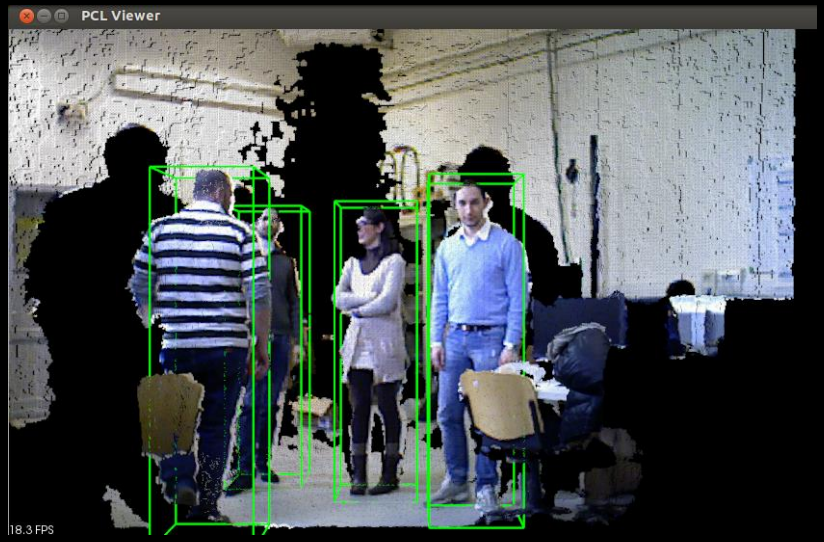
pcl::people

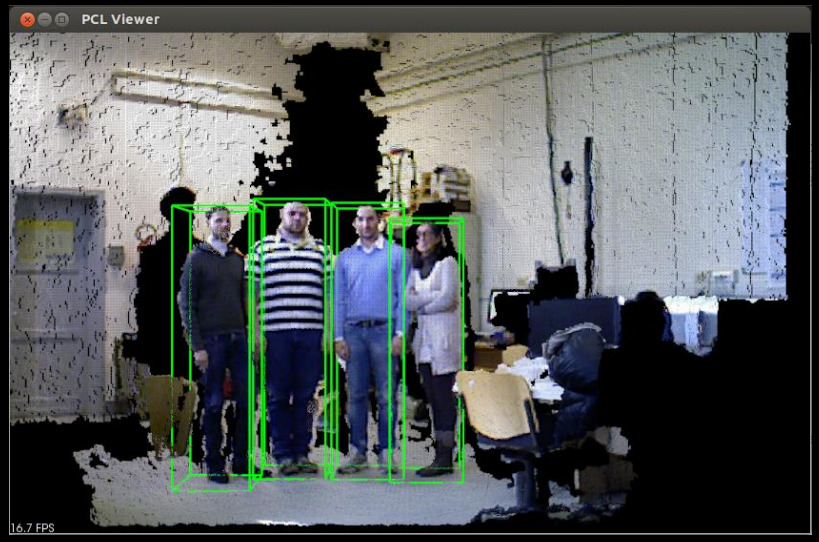
```
// Optional settings:
people_detector.setVoxelSize(voxel_size);                    // set the voxel size (0.06)
people_detector.setHeightLimits(min_height, max_height); // set height limits (1.3, 2.3)
people_detector.setDimensionLimits(min_points, max_points); // set dimension limits (30, 5000)
people_detector.setMinimumDistanceBetweenHeads(min_dist); // set minimum distance between persons' heads (0.3)
people_detector.setSensorPortraitOrientation(true); // set portrait/landscape orientation (false)


people_detector.setPersonClusterLimits(min_height, max_height, min_width, max_width);
```

pcl::people







M. Munaro, F. Basso and E. Menegatti. Tracking people within groups with RGB-D data. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2012, Vilamoura (Portugal), 2012.

M. Munaro, E. Menegatti. *Fast RGB-D People Tracking for Service Robots*. Journal on Autonomous Robots, 2014.

Input/Output

Input

- XYZRGB point cloud
- Ground coefficients

Output

- People clusters (centroid, points indices, …)

Framerate (on Intel Core2 @ 2.5GHz)

- About 15 fps (VGA resolution, data grabbing and visualization not included)
- About 20 fps (with QQVGA resolution) on the original ROS-based implementation

ROS-Industrial Human Tracker

- pcl::people basis for ROS-Industrial Human Tracker
- project in collaboration with Southwest Research Institute (**SwRI/ROS-I**) and National Institute of Standard and Technology (**NIST**) for people **tracking** in **industrial environments**
- a RGB-Depth dataset has been collected for testing in an industrial environment

M. Munaro, C. Lewis, D. Chambers, P. Hvass and E. Menegatti. *RGB-D Human Detection and Tracking for Industrial Environments*. In Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13), Padova, Italy, July 2014 (ORAL).

OpenPTrack: multi-camera tracking

OpenPTrack

http://openptrack.org

- open source solution for **scalable**, **multi-imager** person tracking for education, arts, and culture applications
- **OpenPTrack** started in 2013 from a collaboration with UCLA REMAP and Open Perception. Code is available under a BSD license. Portions of the work are supported by the National Science Foundation (IIS-1323767).

ELECTROLAND

Università degli Studi di Padova

M. Munaro, F. Basso, E. Menegatti. *OpenPTrack: Open Source People Tracking for RGB-D Camera Networks*. Robotics and Autonomous Systems Journal, vol. 75, part B, pp. 525-538, ISSN: 0921-8890, 2016.

M. Munaro, A. Horn, R. Illum, J. Burke and R. B. Rusu. *OpenPTrack: People Tracking for Heterogeneous Networks of Color-Depth Cameras*. In IAS-13 Workshop Proceedings: 1st Intl. Workshop on 3D Robot Perception with Point Cloud Library, pp. 235-247, Padova, Italy, 2014.

F. Basso, R. Levorato, M. Munaro, and E. Menegatti. A Distributed Calibration Algorithm for Color and Range Camera Networks. In book Robot Operating System - Studies and Systems, Decision and Control, Springer, 2016.

pointcloudlibrary

Technology comparison for people tracking

	OpenPTrack	Single Kinect (w/ Microsoft SDK or OpenNI/NiTE)	Blob Tracking (e.g., Community Core Vision)	Augmented Reality (e.g., AR Toolkit)	Motion Capture Marker-based (e.g., Vicon)	RF Tracking (e.g., Zebra)
Target audience	Education, Arts, Culture	Various	Various	Various	High-end Production	Industrial sensing
Core technology	Networked 3D imagers	3D Imager	2D Camera(s)	2D Camera	2D Cameras	Radio Frequency
Output Type	ID, 3D Centroid	ID, Skeletal Data	ID, 2D Centroid	ID, 3D Position, Orientation	ID, Dense Skeletal Data	ID, 3D Position
Max. Tracking Volume	Large	Small	Small to Medium	Small to Medium	Can be very large	Can be very large
Fusion of multiple views	Intrinsic	N/A	Up to developer	Up to developer	Intrinsic	N/A
Typical refresh rate (Hz)	30-60	30	15-30	30-60	60-120+	20-50
Lag (perceptual)	Low	Low	Low	Low	Very Low	Medium
Maximum people tracked	Many	Typically 4	Tradeoff with volume	Tradeoff with volume	Many	Many
Person detection	Yes	Yes	Not usually	N/A	Yes	N/A
ID Stability	Medium	Medium to High	Low	High	High	Very High
3D Tracking	Yes	Yes	No	Yes	Yes	Yes
Skeletal Tracking	No	Yes	No	No	Yes	No
Must carry / wear something	No	No	No	Yes	Yes	Yes
Position accuracy	High	High	Varies greatly	Medium to High	Very high	Medium
Occlusion resistance	High in multi-imager nets	Low	Low	Low	High	Requires multiple tags
Visible light sensitivity	Minimal	Minimal	Yes	Yes	Some	None
IR light sensitivity	Depends on imager	Yes	Depends on imager	Depends on Imager	Often	No
Costume sensitivity	Must be humanoid	Must be humanoid	None	Must show tag	Must wear marker suit	None
Multiple imager types	Yes	No	Yes	Yes	Yes	n/a
Typical setup time	Medium	Very low	Low to Medium	Low to Medium	Medium to High	Medium to High
App integration complexity	Low	Low	Low to Medium	Low to Medium	Medium to High	Low to Medium
Open source software	Yes	Usually	Usually	Usually	No	No
Off-the-shelf parts	Yes	Yes	Yes	Yes	No	No
Typical system cost	$$	$	$	$	$$$$	$$$

pointcloudlibrary

OpenPTrack: supported sensors

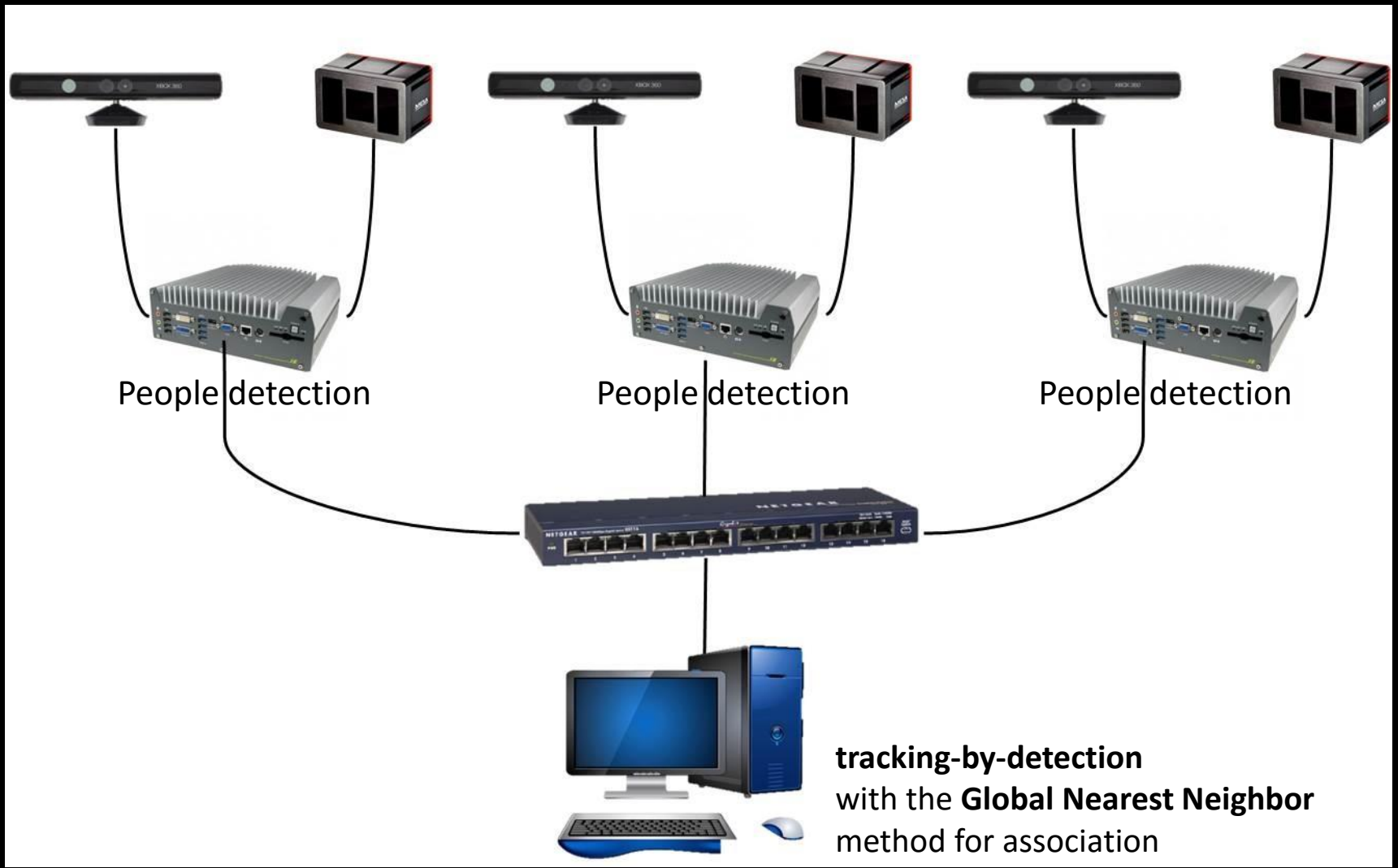




	Kinect v1	Kinect v2	SR4500	Stereo
Depth resolution	320x240	512x424	172x144	variable
Depth range [m]	0.8 - 8	0.4 - 10	0.4 - 9	variable
Intensity resolution	640x480	1920x1080	172x144	variable
Type of intensity	color	color/infrared	infrared	color/mono/infrared
Error VS distance	quadratic	nearly constant	nearly constant	quadratic
Crosstalk	yes	no	yes	no
Visible light sensitivity	no	no	no	high for color cam
Infrared light sensitivity	high	low	low	high for IR cam
Optimal frame rate (fps)	30	30	13	variable

- Networks can be composed of **different types** of sensors
- Routines for **intrinsic** and **extrinsic calibration**

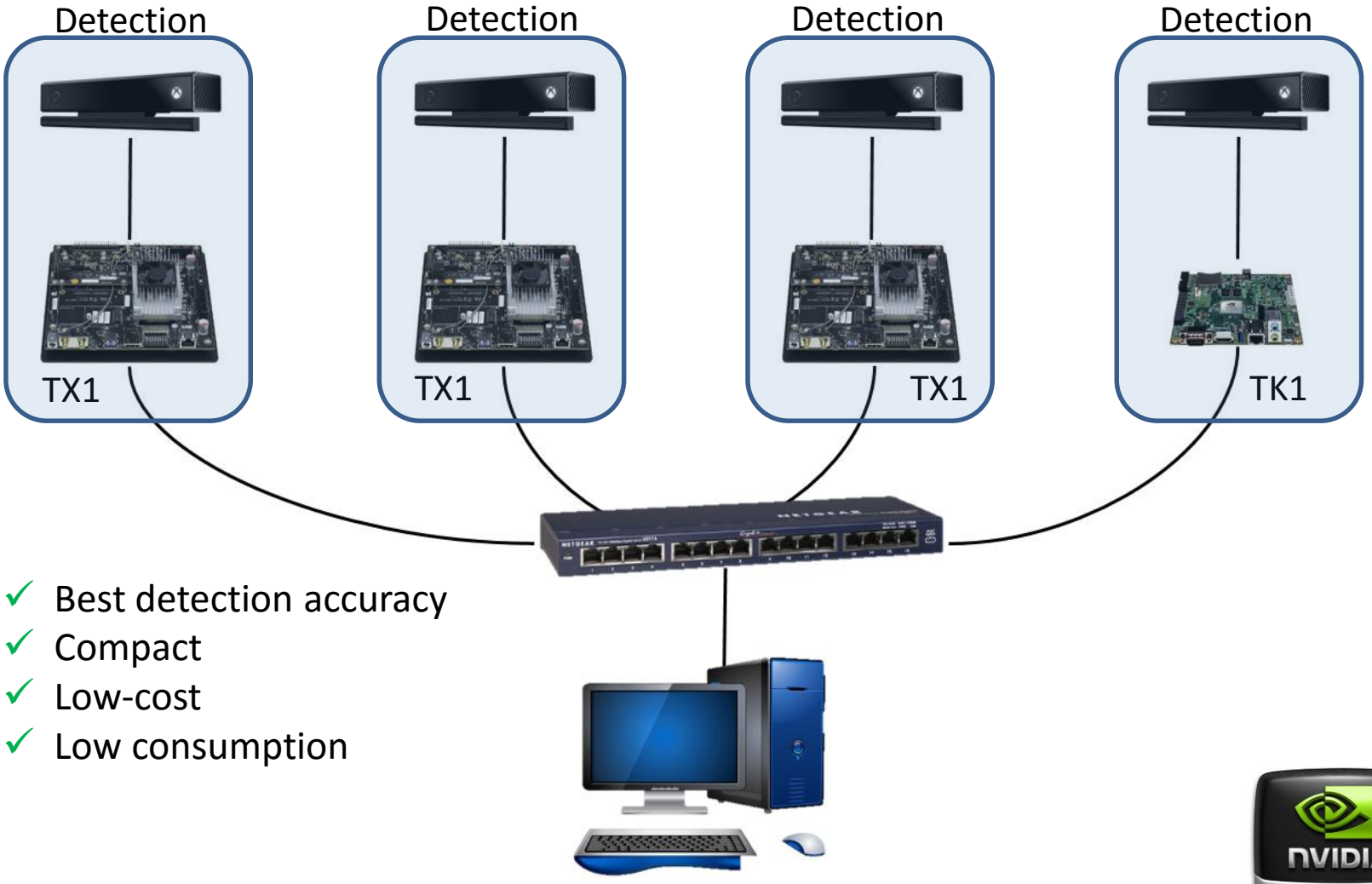
Distributed Multi-camera Tracking



tracking-by-detection
with the **Global Nearest Neighbor**
method for association

pointcloudlibrary

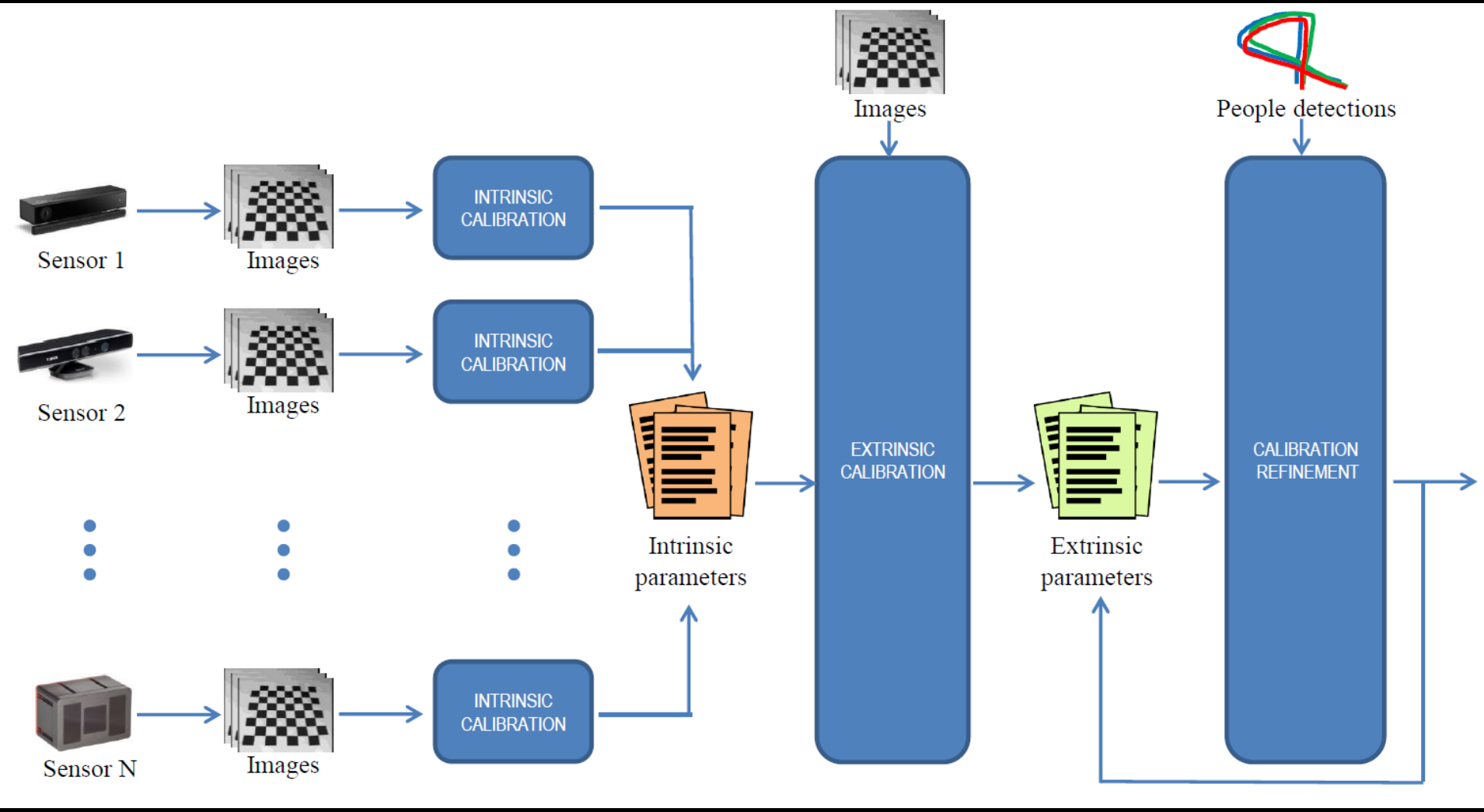
Detection node: Jetson TK1/TX1 + Kinect v2



- ✓ Best detection accuracy
- ✓ Compact
- ✓ Low-cost
- ✓ Low consumption

NVIDIA TEGRA

OpenPTrack: Multi-camera calibration



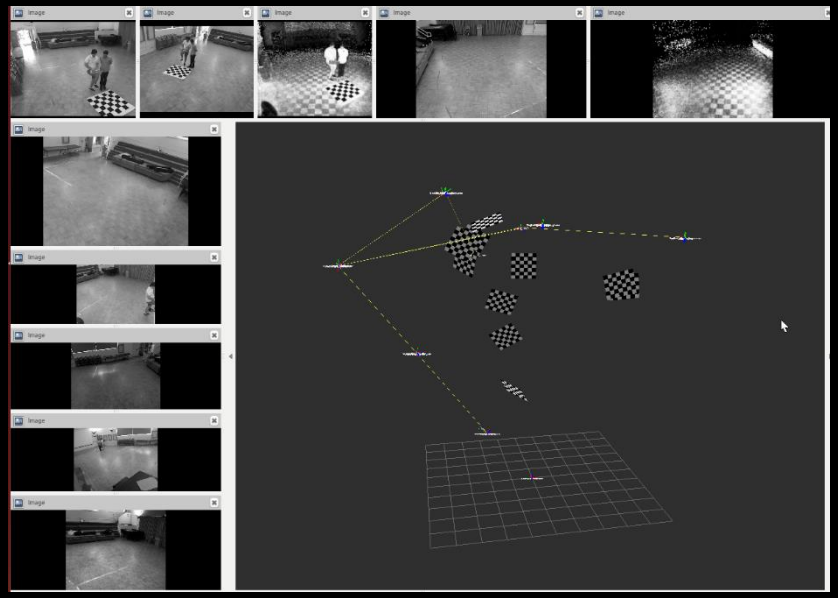
pointcloudlibrary

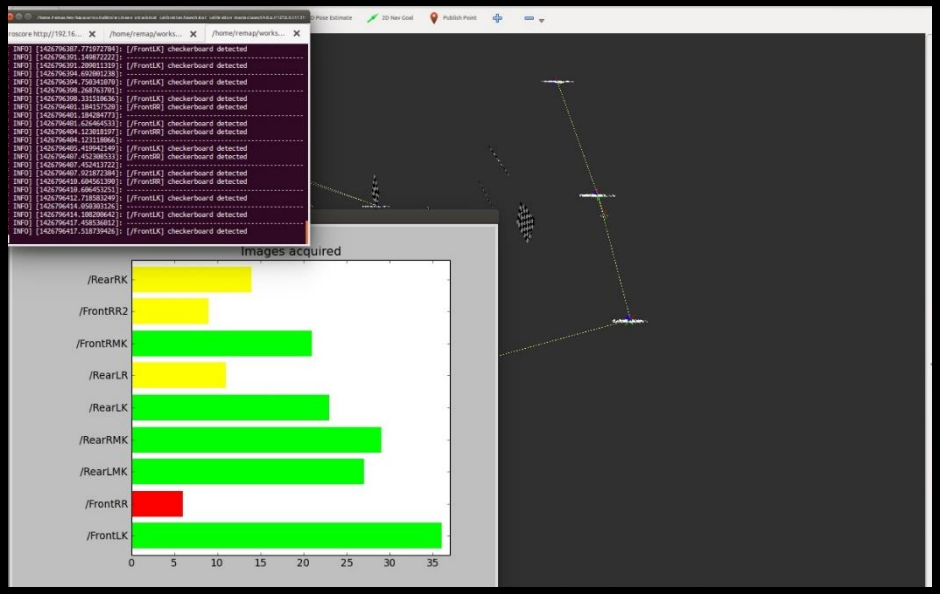
Calibration: Estimating camera poses

- Estimate the **extrinsic parameters** of a camera sensor network
- Provide a **general calibration** procedure for **heterogeneous sensor networks**
- Have an **online feedback**
- **ROS implementation** for large community development
- Possibility to define a **«user» reference frame** for tracking
- **Multi-frame** procedure for optimizing camera poses and **ground** plane **estimation**

$$\mathrm{e} = \sum_{k=1}^{K} \left[\frac{1}{\sigma_{\mathrm{cam}}^2} \sum_{n=1}^{N} u_{nk} \cdot \mathrm{e}_{\mathrm{cam}}({}^{\mathcal{W}}\mathcal{C}_n, {}^{\mathcal{W}}\mathcal{B}_k) \right]$$

$$\mathrm{e}_{\mathrm{cam}}({}^{\mathcal{W}}\mathcal{C}, {}^{\mathcal{W}}\mathcal{B}) = \sum_{\mathbf{b} \in \mathbf{B}} \left\| \mathrm{repr}_{\mathrm{C}}({}^{\mathcal{C}}\mathbf{b}) - \hat{\mathbf{b}} \right\|^2$$

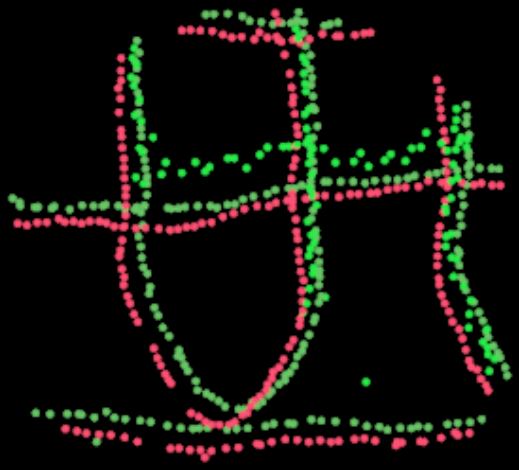




Calibration: Refining camera poses

- Camera poses are corrected so that **detections** coming from different sensors are **aligned**
- The procedure requires **one** person to walk within the tracking space
- It exploits the **pcl::registration** module

Detections before refinement



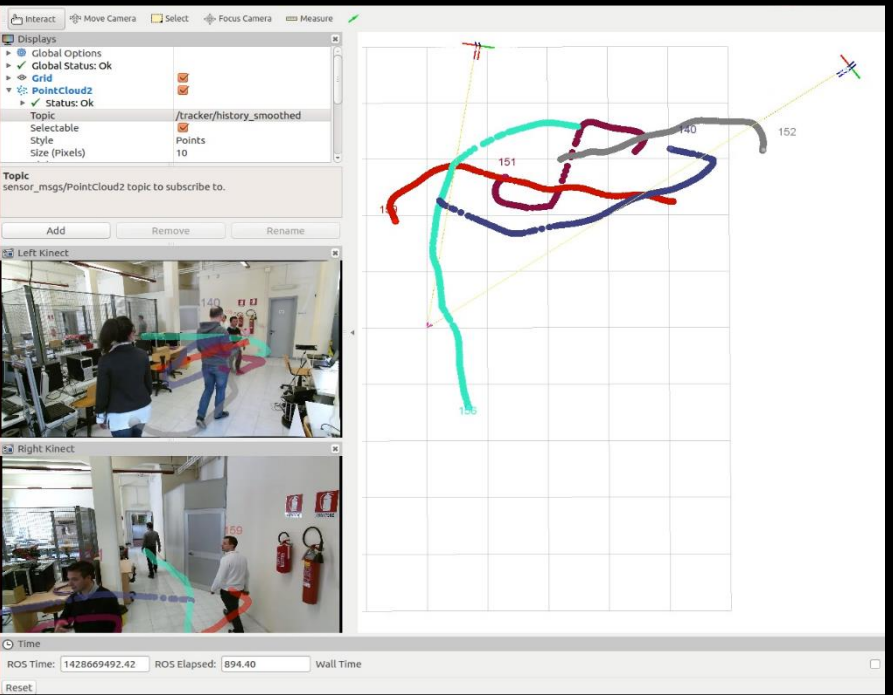
Detections after refinement

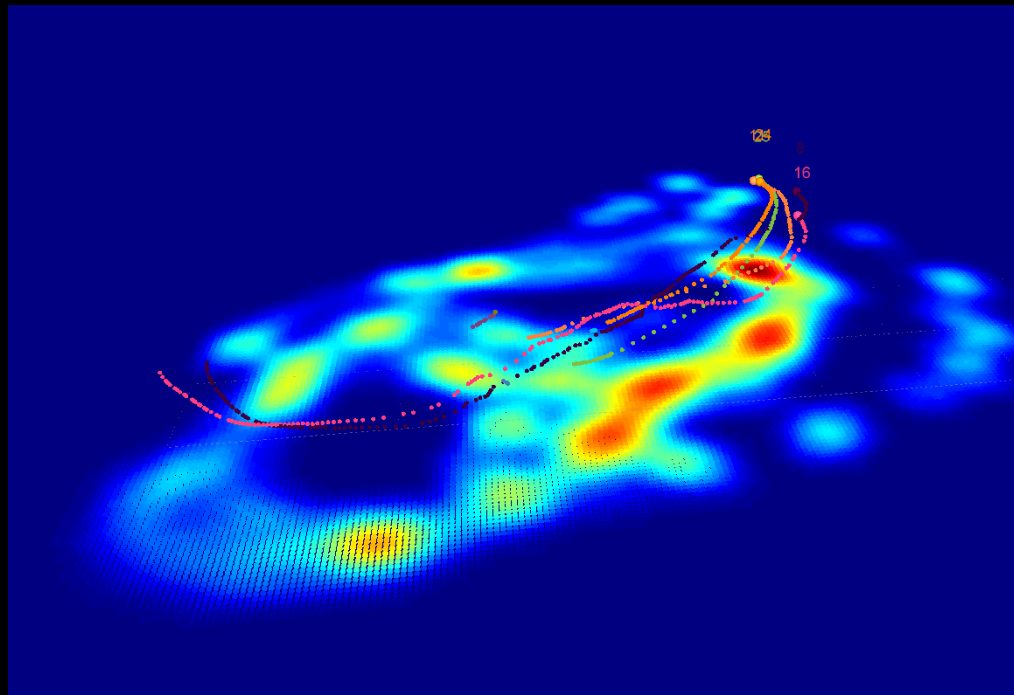
M. Munaro, F. Basso, E. Menegatti. *OpenPTrack: Open Source People Tracking for RGB-D Camera Networks*. Robotics and Autonomous Systems Journal, vol. 75, part B, pp. 525-538, ISSN: 0921-8890, 2016.

pointcloudlibrary

How tracking works?

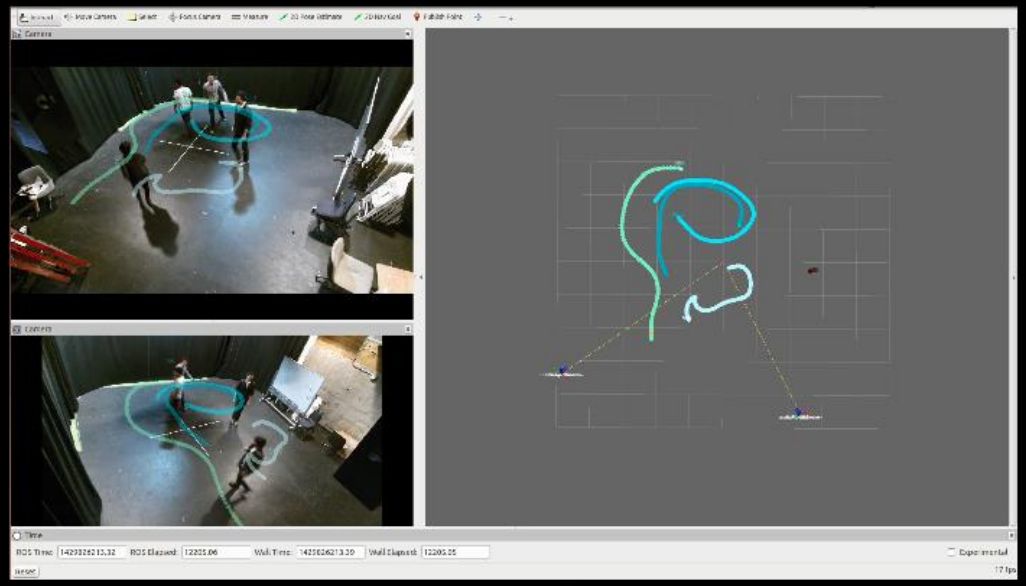
- Detections from all the sensors are referred to a common reference frame and merged into **tracks** based on **motion consistency**
- Detections <-> tracks association is globally optimized (**Global Nearest Neighbor**)
- Tracking messages contain:
  - ID number
  - Centroid position and height
  - Bounding box
  - Track age and status

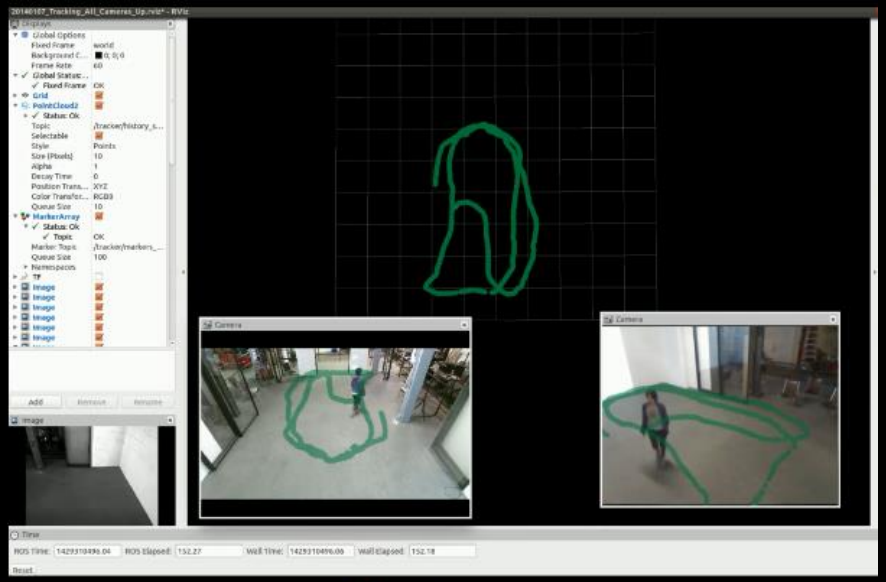




pointcloudlibrary

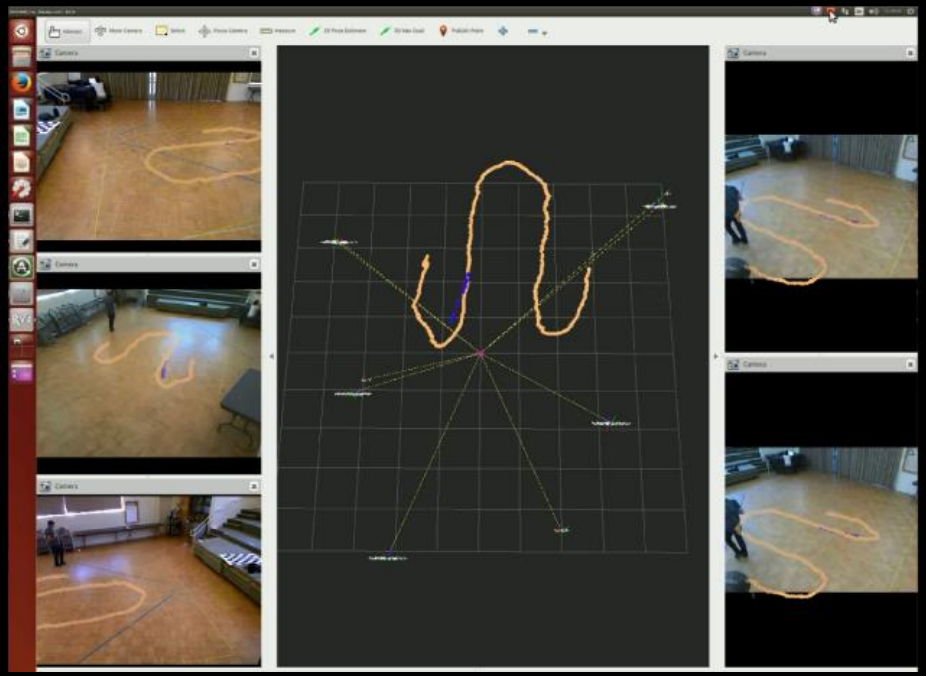
OpenPTrack installations





UCLA School of Theater, Los Angeles (CA)

- Two Kinect v2
- Film lighting



Chiparaki, Los Angeles (CA)

- Six cameras (three Kinects, thr
- 26*20*13 ft Pavilion

UCLA Lab School, Los Angeles

- Six Kinect v1, three SR4500, 1 stereo
- Six PCs
- 1200 sq/ft
- Applied to children

OpenPTrack applications

ARTS: «Whorl» by Damon Seeley

EDUCATION: UCLA Lab School

CULTURE: UCLA IMLab Cybermural for LA State Historic Park

Live Demo!