OpenPTrack: People Tracking for Heterogeneous Networks of Color-Depth Cameras

Matteo Munaro[1], Alex Horn[2], Randy Illum[2], Jeff Burke[2], and Radu Bogdan Rusu[3]

[1] Department of Information Engineering, University of Padova, Via Gradenigo 6B, 35131 - Padova, Italy,
munaro@dei.unipd.it,
[2] Center for Research in Engineering, Media and Performance, University of California Los Angeles, Los Angeles, CA 90095-1622, USA,
nano@remap.ucla.edu, randy@remap.ucla.edu, jburke@ucla.edu,
[3] Open Perception, Inc., Menlo Park, CA 94025, USA,
rusu@openperception.org

Abstract. This paper introduces *OpenPTrack*, an open source software for multi-camera people tracking in RGB-D camera networks. OpenPTrack provides real-time people detection and tracking algorithms from 3D data coming from Microsoft Kinect and Mesa SwissRanger. The software is able to track people at 30 Hz with minimum latency. A user-friendly calibration procedure is also provided, so that the camera network can be calibrated in few seconds by moving a checkerboard in front of the cameras and seeing the calibration results in real time. The algorithms for people detection are executed in a distributed fashion for every sensor, while tracking is done by a single node which takes into account detections from all over the network. Algorithms based on RGB or depth are automatically enabled while the system is running, depending on the luminance properties of the image. OpenPTrack is based on the Robot Operating System and the Point Cloud Library and has been tested on networks composed of up to six sensors.

Keywords: OpenPTrack, people tracking, RGB-D, open source, multi-camera, Microsoft Kinect, Mesa SwissRanger.

1 Introduction

The ability to detect and track people in real time is useful for a variety of applications: from video surveillance to robotics, from education to art. OpenPTrack[4] is an open source project launched in 2013 to create a scalable, multi-camera solution for person tracking that specifically aims to support applications in education, art, and culture.

[4] Website: http://openptrack.org, repository: https://github.com/OpenPTrack/open_ptrack.

With the advent of commercially available consumer depth sensors, and continued efforts in computer vision research to improve multi-modal image and point cloud processing, robust person tracking with the stability and responsiveness necessary to drive interactive applications is now possible at low cost. But the results of such research are not easy to use for application developers. Moreover, the extension of these algorithms to a multi-sensor scenario is not straightforward.

This project aims at enabling artists and creators to work with robust real-time person tracking in real-world projects. OpenPTrack aims to support *creative coders* in the arts, culture, and educational sectors who wish to experiment with real-time person tracking as an input for their applications. The library contains numerous state-of-the-art algorithms for RGB and/or depth-based people detection and tracking, and has been created on top of a modular node based architecture, to support the maximum re-use of code and a distributed implementation.

The system allows to use a network of imagers to track the moving centroids (center of mass) of people within a defined area. These data are also provided as a simple JSON-formatted stream via UDP, which can be incorporated into creative coding tools like Max/MSP, Touchdesigner, and Processing, as well as a variety of other software languages and environments.

OpenPTrack is built on other open source libraries: the Point Cloud Library (PCL [15]) and Robot Operating System (ROS [14]); it currently works with the Microsoft Kinect and Mesa Imaging Swissranger SR4500.

The remainder of the paper is organized as follows: In Section 2, we review the most recent works on RGB-D people detection and tracking and the available open source implementations, while in Section 3 we describe the procedure we implemented in OpenPTrack for calibrating a network of RGB-D sensors in real time. Section 4 gives an overview of our detection algorithms while Section 5 details the tracking node. Some results are reported in Section 6, while conclusions are drawn in Section 7.

2 Related Work

Since the introduction of low-cost RGB-D sensors, a number of people detection and tracking algorithms which exploit combined color and depth information have been proposed. Most of these approaches apply a sliding window technique to RGB and/or depth images ([6], [7]), thus requiring high parallelization with GPUs to obtain real time tracking.

Recent works ([2], [8], [11], [10], [5]) allow to avoid to apply machine learning classification to thousands of detection windows per image. They exploit the assumption that people stand/walk on a ground plane and cluster algorithms on depth data to find a small number of Regions Of Interest (ROIs) which are candidates to contain people and are then classified with more robust and computational demanding algorithms.

An open source implementation of the people detection algorithms in [8] and [11] is available in the Point Cloud Library [15]. In ROS-Industrial *Human Tracker* repository[5], these algorithms are combined with other people detectors in a people detection cascade which obtain even more computational efficiency and which is combined with the tracking algorithm described in [10]. Also the code for the GPU-based and the CPU-based versions of the people detection and tracking software in [5] has been recently released[6].

All of these softwares are targeted to track people from a single RGB-D camera which can move onboard of a wheeled robot. However, none of these provides the possibility to perform people tracking in a distributed fashion by exploiting multiple cameras.

OpenPTrack addresses the problem of people tracking in a distributed network of RGB-D sensors. For this purpose, it extends state-of-the-art approaches in terms of scalability and ease of use in a multi-camera scenario, while maintaining comparable tracking accuracy.

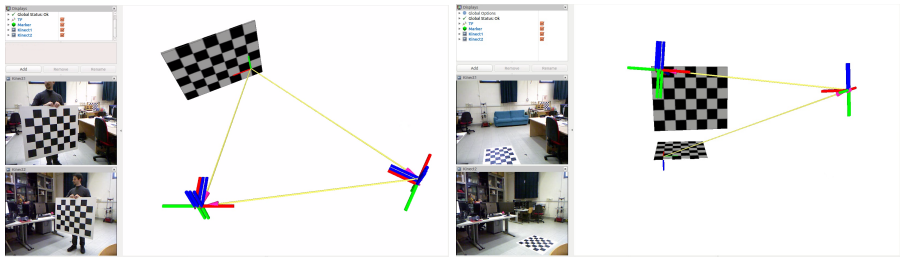
3 Real-Time Multi-Camera Calibration

In order to fuse people detections coming from every sensor, all camera poses should be known with respect to a common reference frame. We developed a calibration procedure which relies on ROS communication and networking capabilities and that can be executed in little time by simply moving a checkerboard in front of the cameras.

In particular, once the network of computers and sensors is set up, the user is requested to move a checkerboard in front of the cameras. Once the checkerboard is seen by a pair of cameras the relative pose between the two is estimated in real time and the resulting reference axes are drawn within the ROS visualizer, as shown in Fig. 1a. Then, every time a new camera sees the checkerboard at the same time of one of the already calibrated cameras, the new camera pose is found and added to visualization. After all cameras have been calibrated, the user is requested to place the checkerboard on the ground so that the first calibrated camera can see it, as in Fig. 1b. With this, the position of the ground plane with respect to each camera is calibrated.

This calibration procedure relies on the `calibration_toolkit` package and the *Google Ceres* library for optimizing the camera poses in a bundle adjustment fashion when more than two cameras see the checkerboard or when more frames are collected over time. In fact, the camera poses are estimated and refined in real time as soon as new images become available. Further details about the optimization algorithm can be found in [1]. When all cameras have been calibrated the user is prompted to save the results. At this point, a tree containing the relative transformations between the cameras is produced. Moreover, the first transformation of the tree is the one connecting the ground reference frame (`world`) with the first calibrated camera.

[5] `https://github.com/ros-industrial/human_tracker/tree/develop`.
[6] `http://www.vision.rwth-aachen.de/software/realpdt/realpdt`.



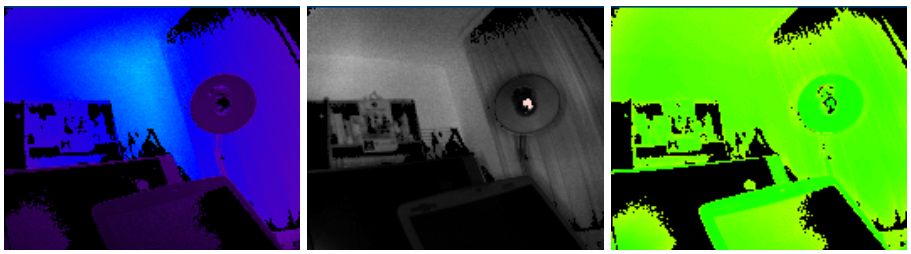
(a) Pairwise calibration (b) Ground plane calibration

Fig. 1. Screenshots from the real-time calibration procedure provided with OpenPTrack.

This calibration procedure has been made generic so as to work with heterogeneous sensors. At the time of writing, Microsoft Kinect and Mesa SwissRanger SR4500 can be directly used, but we envision to support also custom stereo cameras in the future.

3.1 Calibrating SwissRanger sensors within the network

The Mesa SwissRanger SR4500[7] is a time-of-flight sensor which produces matricial depth data at 176x144 pixel resolution. In particular, it also provides an infrared intensity image and a confidence value for every pixel stating the confidence of depth data. An example of this kind of data is reported in Fig. 2.



(a) Depth (b) Intensity (c) Confidence

Fig. 2. Data produced by the Mesa SwissRanger SR4500.

Depth estimation is less sensitive to external infrared illumination than for Kinect, but the depth sensor is not coupled with a color camera. Since the basis of our extrinsic calibration method is an algorithm which finds checkerboards within color/grayscale images[8], we use SwissRanger intensity images for calibra-

[7] `http://www.mesa-imaging.ch/products/sr4500`.
[8] `findChessboardCorners` method in OpenCV.

tion at the same way we use Kinect color images. In order for the checkerboards to be detected on those intensity images, we perform a proper rescaling of the intensity values to spread them between 0 and 255 and an image upsampling to fit the minimum image resolution expected by the checkerboard finding algorithm.

In Fig. 3, color images from three Kinects and rescaled intensity images from three SR4500 are shown, together with the output of our extrinsic calibration procedure visualized as reference frames of every sensor with respect to the ground reference frame.

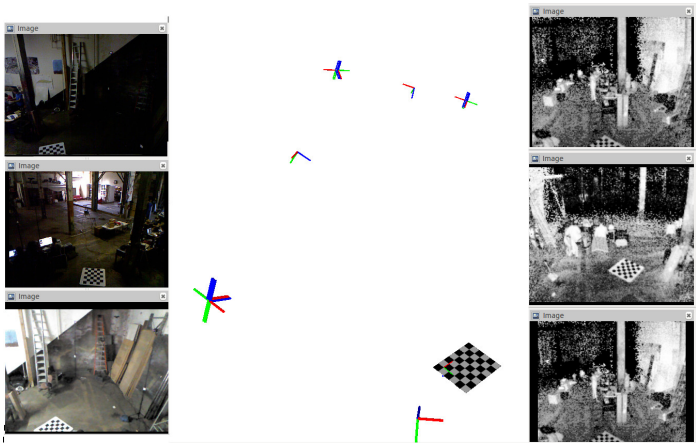
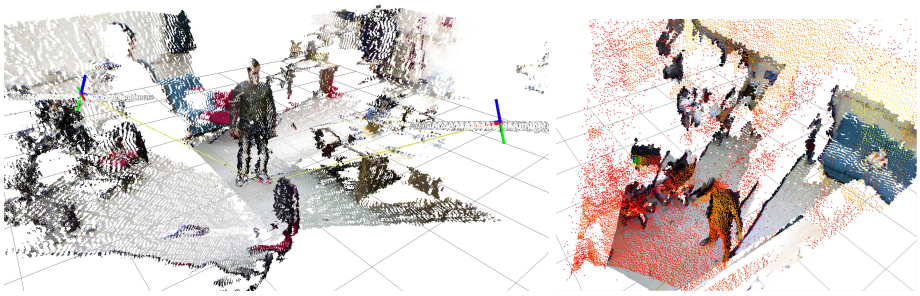


Fig. 3. Extrinsic calibration results for a network composed of three Kinects (images on the left) and three SwissRangers SR4500 (images on the right).

In Fig. 4, we propose a qualitative evaluation of two pairwise calibrations. In Fig. 4a, the point clouds obtained by two Kinects are shown when referred to a common reference frame obtained with calibration. The two Kinects were observing the scene with a difference in the point of view of about 120°. The accuracy in the point clouds alignment can be better appreciated if looking at the person's legs. In Fig. 4b, instead, we report in false colors the point cloud obtained with a SwissRanger and, in RGB, the point cloud obtained with a Kinect. It can be noticed how our procedure allows to align data coming from these two different sensors. However, since every sensor has a different intrinsic distortion in estimating the depth, better results could be obtained if taking into account a correction of the depth maps as described in [3].

4 Distributed People Detection

OpenPTrack allows to perform people tracking within a camera network by distributing people detection and centralizing the tracking process, as proposed in [12]. As depicted in Fig. 5, the sensors are directly attached to a computer which



(a) Point clouds from two Kinects

(b) Point clouds from a Kinect and a SwissRanger

Fig. 4. Point cloud alignment based on extrinsic calibration obtained with OpenPTrack.

analyzes the data stream and performs people detection. Only the detections are sent through the network, in order to be fused at the tracking level after being referred to a common reference frame by means of calibration data.

4.1 Ground plane estimation

The people detection algorithms implemented in OpenPTrack rely on the ground plane assumption. Thus, the ground plane equation is automatically estimated at startup. At first, all the planes are found by means of the `OrganizedMultiPlaneSegmentation` algorithm [16] in the Point Cloud Library. Then, the ground plane is chosen among them in order to be fairly horizontal and to be placed under any other plane. As an alternative to the automatic selection, the user can manually select the ground by clicking on three point cloud points (manual selection), or he can select the ground plane among those produced by the automatic multi plane segmentation (semi-automatic selection). In Fig. 6a, a sample point cloud is shown, while in Fig. 6b the planes resulting from the segmentation are shown with different colors. Finally, the plane selected by the automatic procedure is highlighted in red in Fig. 6c.

It is worth noting that all cameras within a network share the same ground plane equation and that this can be derived from calibration, without the need to estimate it for every sensor. OpenPTrack exploits the following procedure for initializing the ground plane for every detection node:

1. the ground plane equation is estimated from point cloud data
2. the ground plane equation is also estimated from calibration data
3. if the ground estimated from the point cloud is not valid or it is too distant from the plane derived from calibration, the latter is kept as valid
4. if the ground estimated from the point cloud is close to the one derived from calibration, the former is used. In fact, the ground estimated with

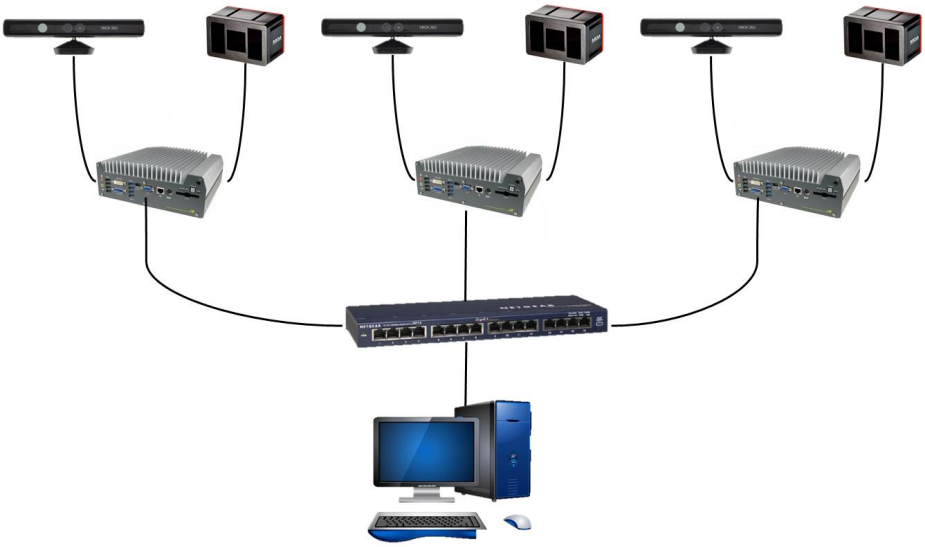
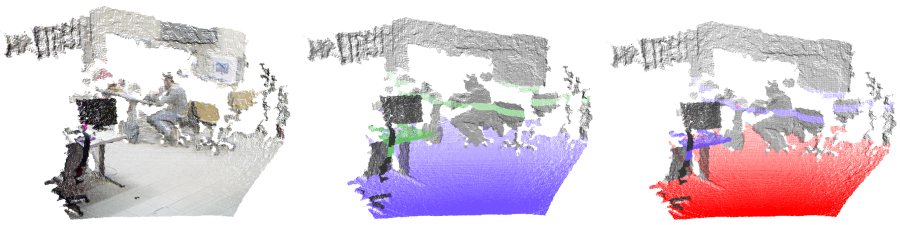


Fig. 5. Example of camera network composed by three Kinects and three SwissRangers. Sensors are directly connected to distributed PCs which perform people detection. Then detections are sent through the network and read/used by the PC running the tracking algorithm.



(a) (b) (c)

Fig. 6. Automatic ground plane estimation: (a) sample point cloud, (b) output of the multi-plane segmentation, (c) output of the automatic ground plane selection (in red the chosen plane).

calibration could be less precise than that obtained directly with detection from the point cloud because the former suffers from the problem of error propagation in the camera poses.

4.2 RGB-depth algorithms

People detection in OpenPTrack is performed with a mixed color-depth approach which is based on the algorithms presented in [11] and [10]. In particular:

1. as a first step, the depth-based clustering proposed in [11] is applied on the point cloud in order to determine clusters of points candidate to contain a person, which are then processed by texture-based and/or disparity-based classification algorithms;
2. if the mean luminance of the image is over a threshold, a HOG-based people detector ([11]) is applied to the RGB image;
3. then, an Adaboost classifier applied to the disparity map ([10]) further improves the results.

This pipeline allows to exploit both RGB and depth information for obtaining the best results when the color image is good, while using only depth data if the color image is too dark. Thus, OpenPTrack is also able to detect and track people in full darkness, even if the best results are obtained if the color image can be exploited.

4.3 People detection on SwissRanger intensity image

The same people detection algorithm used for Kinect is also applied to SwissRanger but the rescaled infrared intensity image is used in place of Kinect color image. In fact, after rescaling the intensity values, this image contains the proper edges for the HOG descriptor to work well. Moreover, unlike for Kinect, the HOG-based classification ([11]) is always used on the infrared intensity image, since its quality is independent from external illumination.

It is also worth mentioning that the point cloud points associated to low confidence values are filtered out before applying our depth-based clustering algorithm, since they are likely to be due to noise.

In Fig. 7, some people detection results applied to SwissRanger data are reported. From our experience, even if the SwissRanger SR4500 has considerably lower resolution than Kinect, our software allows to obtain comparable results.

5 Centralized Tracking Node

The tracking node is a centralized algorithm which receives detections from all over the network and performs data association every time a new set of detections arrives. For data association, a lightweight version of the tracker in [11] which only relies on motion information is used in order to reduce at a minimum the information flow over the network.

(a) (b) (c) (d)

Fig. 7. People detection results on SwissRanger infrared intensity images.

In particular, we compute the Mahalanobis distance between track i and detection j as

$$D_M^{i,j} = \tilde{\mathbf{z}}_k^{\mathrm{T}}(i,j) \cdot \mathbf{S}_k^{-1}(i) \cdot \tilde{\mathbf{z}}_k(i,j) \tag{1}$$

where $\mathbf{S}_k(i)$ is the covariance matrix of track i provided by a filter and $\tilde{\mathbf{z}}_k(i,j)$ is the residual vector between measurement vector based on detection j and output prediction vector for track i:

$$\tilde{\mathbf{z}}_k(i,j) = \mathbf{z}_k(i,j) - \hat{\mathbf{z}}_{k|k-1}(i). \tag{2}$$

The values we compare with the Mahalanobis distance represent people positions and velocities in ground plane coordinates. Given a track i and a detection j, the measurement vector $\mathbf{z}_k(i,j)$ is composed by the position of detection j and the velocity that track i would have if detection j were associated to it.

An Unscented Kalman Filter is exploited to predict people positions and velocities along the two ground plane axes (x, y). As people motion model we chose a constant velocity model because it is good at managing full occlusions, as described in [4]. Given that the Mahalanobis distance for multinormal distributions is distributed as a chi-square [13], we use this distribution for defining a gating function for the possible associations.

People detection confidence is used for defining a policy for initializing, updating and removing tracks, as in [11].

As a future work, we envision to compute also compact yet effective signatures of people, similar to those in [9], which could allow for better data association while preserving the scalability properties of the system.

6 Results

So far, the OpenPTrack library has been used in real world scenarios for tracking people within networks composed of up to six sensors. In Fig. 8, an example of tracking output from a single Kinect camera is reported: on the left, the tracks of four people are drawn with different colors, while, on the right, people detection results are shown for the current frame. It is worth noting that these results refer to a scenario where the cameras were four meters high and considerably tilted. Moreover, the background was very cluttered, but our combination of RGB and depth algorithms allowed to obtain robust detection results.

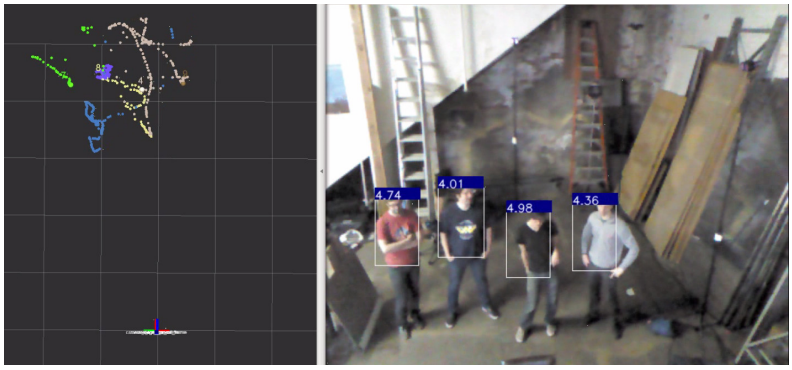
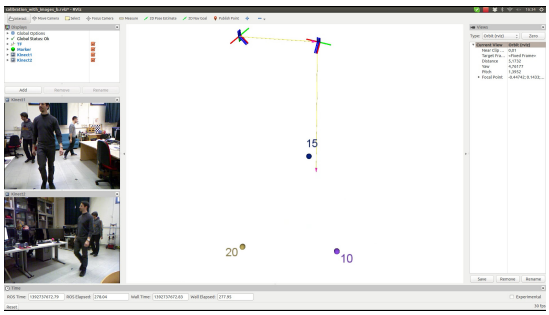
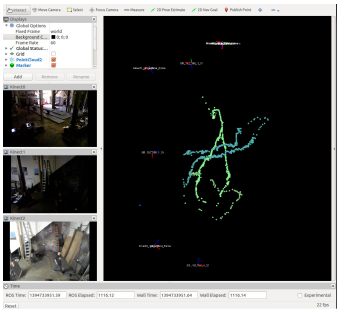


Fig. 8. People tracking results: tracks reported with different colors on the left, detections at the current frame on the right.

In Fig. 9a, the position of three people seen from two Kinect cameras are reported, while, in Fig. 9b, we visualize the tracks produced by our algorithm when tracking two people from three Kinects with overlapping field of views. Color images are reported on the left as a reference.



(a) Three people tracked with two Kinects.

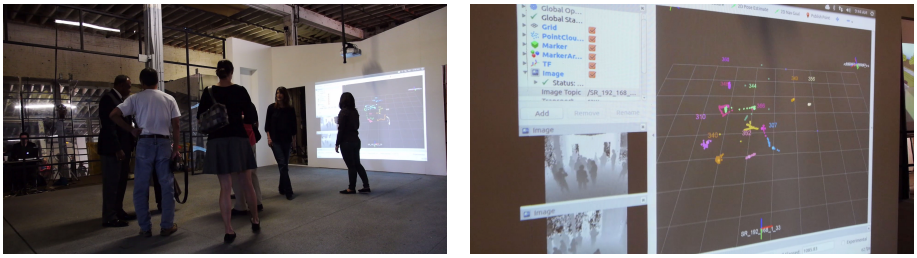


(b) Two people tracked with three Kinects four meters high.

Fig. 9. Some tracking results obtained with the OpenPTrack library.

The OpenPTrack library has been also validated in three real world scenarios. The first was in March 2014 in Los Angeles, where three Kinects and three SwissRangers were mounted on top of a 8x6x4 meters indoor pavilion and used for cooperative people tracking (Fig. 10) by the UCLA Interpretive Media Laboratory. The tracking output drove a interactive digital mural application showing images of the Los Angeles State Historic Park, which were selected and manipulated based on people's real-time position in the pavilion. Our calibration procedure enabled us to easily calibrate this complex network in one minute with

real time visualization of the calibration results and without the need for storing data to be processed offline. For this installation, we connected one Kinect and one SwissRanger to each of three PCs used for people detection. These PCs were fanless industrial PCs with Intel i7-3612QM @ 2.10GHz CPU. We obtained a tracking frame rate of about 30 fps for Kinect and 13 fps for the SwissRanger. However, the SwissRanger frame rate was limited by the exposure time chosen in order to have better depth estimates.



(a) (b)

Fig. 10. Pictures from the OpenPTrack installation in Los Angeles on March 2014. Three Kinects and three SwissRangers were mounted on top of a pavilion and used for cooperative people tracking. The tracking output was then used for driving an interactive mural application showing pictures of Los Angeles State Historic Park.

Subsequent installations in May and June 2014 at Indiana University's Rogers Elementary School and the UCLA Lab School tested OpenPTrack in a 5-sensor, 3-CPU configuration that enabled elementary school students to control a science simulation based on their body position and movement, as part of the NSF-supported Science through Technology Enhanced Play (STEP) research project. The childrens' motion controlled on-screen avatars as they explored an embodied simulation of states of matter. These deployments provided practical experience in parameter adjustment for children (80-100 cm tall), calibration by end-users, and how to best support third-party applications using the data in the future.

The OpenPTrack team plan to set up permanent installations at the UCLA Lab School and Interpretive Media Laboratory in the summer of 2014, to continue testing and evaluation.

7 Conclusions

In this work, we presented OpenPTrack, a library for real time people detection and tracking within heterogeneous networks of color and depth sensors, such as Microsoft Kinect and Mesa SwissRanger SR4500. People detection is done in a distributed fashion to allow for system scalability, while tracking is lightweight and is done by a centralized algorithm. We developed a user-friendly calibration procedure which allows to calibrate a camera network in real time,

together with the world reference frame which is then used by the tracking algorithm. OpenPTrack is able to automatically switch between depth-based and color-based people detection algorithms which make the system work also in full darkness, while exploiting color information when valid. It also works with data from SwissRanger sensors by exploiting the infrared intensity image in place of the color image. The obtained results are comparable to those with Kinect, even if SwissRanger resolution is considerably lower.

This library has been already tested with success in real world scenarios. As a future work, we plan to allow tracking parameters tuning by means of a graphical user interface and to refine extrinsic calibration by taking into account the depth maps correction as in [3]. Moreover, we plan to support also stereo cameras as new sensors which could be used to perform tracking.

8 Acknowledgements

OpenPTrack has been sponsored by UCLA REMAP and Open Perception. Key collaborators include the University of Padova and Electroland. Portions of the work have been supported by the National Science Foundation (IIS-1323767).

References

1. F. Basso, R. Levorato, and E. Menegatti. Online calibration for networks of cameras and depth sensors. In *In 12th Workshop on Non-classical Cameras, Camera Networks and Omnidirectional Vision (OMNIVIS 2014), Hong Kong, China*, 2014.
2. F. Basso, M. Munaro, S. Michieletto, E. Pagello, and E. Menegatti. Fast and robust multi-people tracking from rgb-d data for a mobile robot. In *12th Intelligent Autonomous Systems Conference (IAS-12)*, pages 265–276, Jeju Island, Korea, June 2012.
3. F. Basso, A. Pretto, and E. Menegatti. Unsupervised intrinsic and extrinsic calibration of a camera-depth sensor couple. In *In Proceedings of IEEE International Conference on Robotics and Automation (ICRA'14), Hong Kong, China*, 2014.
4. Nicola Bellotto and Huosheng Hu. Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters. *Autonomous Robots*, 28:425–438, May 2010.
5. O. Hosseini Jafari, D. Mitzel, and B. Leibe. Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras. In *In Proceedings of IEEE International Conference on Robotics and Automation (ICRA'14), Hong Kong, China*, 2014.
6. Matthias Luber, Luciano Spinello, and Kai O. Arras. People tracking in rgb-d data with on-line boosted target models. In *International Conference On Intelligent Robots and Systems (IROS) 2011*, pages 3844–3849, 2011.
7. D. Mitzel and B. Leibe. Real-time multi-person tracking with detector assisted structure propagation. In *International Conference on Computer Vision (ICCV) Workshops 2011*, pages 974–981. IEEE, 2011.
8. M. Munaro, F. Basso, and E. Menegatti. Tracking people within groups with rgb-d data. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, pages 2101–2107, Algarve, Portugal, October 2012.

9. M. Munaro, S. Ghidoni, D. Tartaro Dizmen, and E. Menegatti. A feature-based approach to people re-identification using skeleton keypoints. In *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong (China)*. Elsevier, June 2014.
10. M. Munaro, C. Lewis, D. Chambers, P. Hvass, and E. Menegatti. Rgb-d human detection and tracking for industrial environments. In *In Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13), Padua, Italy*, 2014.
11. M. Munaro and E. Menegatti. Fast rgb-d people tracking for service robots. *To appear in Autonomous Robots Journal*, 2014.
12. Matteo Munaro, Filippo Basso, Stefano Michieletto, Enrico Pagello, and Emanuele Menegatti. A software architecture for rgb-d people tracking based on ros framework for a mobile robot. In *Frontiers of Intelligent Autonomous Systems*, volume 466, pages 53–68. Springer, 2013.
13. C. L. Naberezny Azevedo. The multivariate normal distribution [online]. `http://www.ime.unicamp.br/~cnaber/mvnprop.pdf`.
14. Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *International Conference on Robotics and Automation (ICRA)*, 2009.
15. Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *International Conference on Robotics and Automation (ICRA) 2011*, pages 1–4, Shanghai, China, May 9-13 2011.
16. A Trevor, Suat Gedikli, R Rusu, and H Christensen. Efficient organized point cloud segmentation with connected components. In *3rd Workshop on Semantic Perception Mapping and Exploration (SPME), Karlsruhe, Germany*, 2013.