Embedded real-time people detection and tracking with time-of-flight camera

Conference Paper · April 2021

DOI: 10.1117/12.2586057

2 authors, including:

Levente Tamas
Universitatea Tehnica Cluj-Napoca
77 PUBLICATIONS **348** CITATIONS

SEE PROFILE

Embedded real-time people detection and tracking with time-of-flight camera

Levente Tamas[a] and Andrei Cozma[b]

[a]Technical Univeristy of Cluj-Napoca, Romania
[b]Analog Devices, Romania

ABSTRACT

People recognition is a relevant subset of the generic image based recognition task with many possible application areas such as security, surveillance, human-robot interaction or recently the social security in a pandemic context. In this work we present a light-weight recognition pipeline for time-of-flight cameras based on deep learning techniques tailored to this specific type of camera with registered infrared and depth images. By combining the maturity of the 2D image based recognition techniques with the custom depth sensing we achieved effective solutions for a number of relevant industrial applications. In particular, our focus was on automatic door-control and people counting applications.

Keywords: people recognition, embedded computing, ToF camera

1. INTRODUCTION

People recognition is a relevant subset of the generic image based recognition task with many possible application areas such as security, surveillance, human-robot interaction or recently the social security in a pandemic context. In this work we present a light-weight recognition pipeline for time-of-flight cameras based on deep learning techniques tailored to this specific type of camera with registered infrared and depth images. By combining the maturity of the 2D image based recognition techniques with the custom depth sensing we achieved effective solutions for a number of relevant industrial applications. In particular, our focus was on automatic door-control and people counting applications.

Our processing pipeline has two main branches: one for the IR image based custom people detection and one for the 3D pointcloud processing. The conceptual distance measurement with the Analog Devices (ADI) ToF CCD camera camera is shown in Figure 1: the flight time for the closer objects is lower, while for the further ones is increasing, hence we can measure in discretized space the distances from the camera to the surrounding world.

The 2D image processing part is a customized IR image module based on transfer learning for bounding box estimation, skeleton extraction and hardware specific model translation. The latter is relevant in order to have a light-weight embedded solution running on limited floating-point precision hardware platforms such as Jetson Nvidia Family. As the existing deep-learning models are mainly with the focus on RGB images, we had used transfer learning as a method to finetune the existing state of art models such as SSDMobilenet for the IR images from the ToF camera. This solution seemed to be effective in terms of precision and runtime on embedded devices (e.g Jetson Nano). For the skeleton detection part we relied on the real-time tensorflow optimized module for the Jetson product family, however for the generic GPU enabled devices we had to tailor our models since these are custom solutions. In order to reduce the initial parameter set we had to finetune the model in such a way that we preserve the accuracy of the predictions with reduced memory footprint for the Jetson platform. Also in this case we further customized the output from the existing 16 keypoints for the skeletons to 18 in order to have a better representation of the head part of the skeleton.

For the 3D pointcloud processing pipeline we considered a number of custom tailored filtering blocks enabling the effective extraction of 3D region-of-interests relevant to the use-case specific applications. As the first step for

Corresponding author: Levente Tamas
E-mail: Levente.Tamas@aut.utcluj.ro, Telephone: +40-264401586

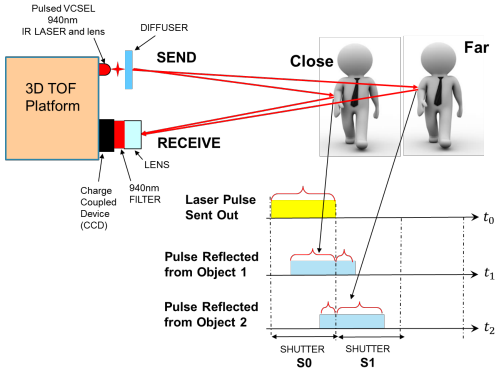

Figure 1. ToF camera functional overview

the pointcloud generation, we considered the undistorted depth images, from which we generated the discretized voxel representation of the 3D space by applying the voxel filter. In the next step we searched for the planar regions in order to detect potential large segments from the initial pointcloud, i.e. floor or wall candidates in order to reduce further the pointcloud size needed to be processed by the remaining processing chain. Next we perform a passthrough filtering, which can be seen as a region of interest selection in the 3D space in order to reduce furthermore the pointcloud size. For the spurious noise removal we applied a statistical outlier removal, customized for the specific noise distribution from a ToF camera. Finally, based on the 2D bounding boxes estimated from the IR images we reprojected these into the 3D space and computed oriented bounding boxes in order to have the regions around the human in the pointcloud as well. The later output is used for the use-case specific applications such as automatic door control, people counting or social distance monitoring.

For each of the presented applications the details of the implementation are given as well as real-life experimental results in order to facilitate the reproducible research approach.

1.1 Related work

The 3D scene parsing has a long history: the recognition part most often is included in a semantic scene parsing.[1] Early variants for the scene parsing and object detection including tracking for RGB-D data can be found in.[2,3] A good overview of the 3D data representation for the RGB-Depth data with the recent learning based techniques can be found in.[4] The data representation is essential for the processing speed: for the spatial data with the introduction of the pointnet[5] architecture was a milestone for the 3D data processing. After this several fast recognition pipelines were developed.[6] This is essential for embedded applications such as wearable[7] or mobile devices.[8] A relevant overview for the multimodal data based tracking is presented in the work.[9]

An essential part of the recognition and tracking pipeline is related to the point cloud preprocessing: this highly influences the quality of the detection and tracking. A good overview of the 3D data filtering and preprocessing can be found in.[10] We considered also learning based variants,[11] but due to their limited tuning capabilities we voted to classical approaches.[12]

2. OUR APPROACH

In our approach for the human detection and tracking for embedded devices we used a lightweight CNN for the IR images based on the SSD MobileNetV2[13] architecture trained on COCO dataset using transfer learning techniques for robust detection results. The detected 2D region of interest (ROI) used reprojected into the 3D space in order to localize in the metric space the detected objects. In order to have an efficient 3D data processing on the embedded devices, we used a point cloud preprocessing task to filter and compress the 3D data. Further on, for the tracking of the detected regions we considered the covariance based tracker[14] adopted to the special case of the IR images received from the ToF camera. Each of these modules were integrated into the already existing Analog Devices ToF software development kit. The schematic overview of this integration is presented in Figure 2.

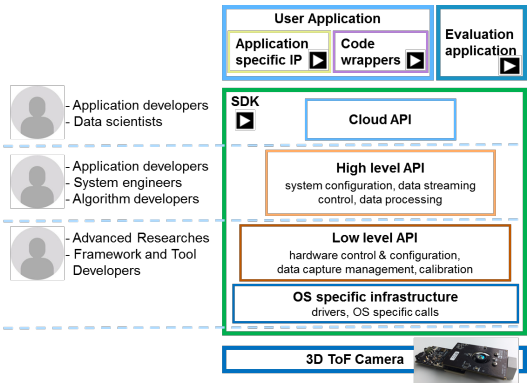

Figure 2. Overview of the used SDK for the camera

In the following parts we present the details of each subsection for the the above mentioned applications.

2.1 Depth image preprocessing

The main role of the depth image preprocessing part is the filtering and bounding box estimation for the 3D ROI. The filtering is essential for the embedded device in order to reduce the computational overload. For the filtering pipeline we considered three interconnected filters: voxel, pass-through and outlier filter as this is visible in Figure 3. All these implementations are open source library based variants.

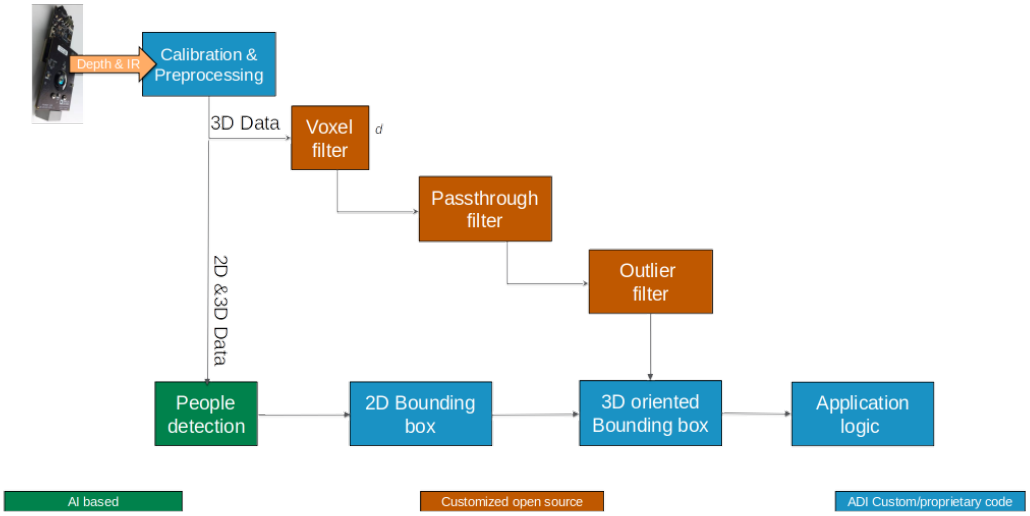

Figure 3. Filtering pipeline used in the application

The voxel filter for the point cloud is a spatial down-sampling, i.e. replaces the points in the voxel specified leaf size with a single point. This is relevant especially if we have noise data from the sensor, e.g. in strong light on rainy weather conditions.[15]

The next filter is the pass-through filter which limits the point clouds on the axes, i.e. only the those points pass through this filter which are inside the interval specified as parameter for the filter. This is a relevant step is some parts of the scene (e.g. ceiling) can be excluded from the ROI zones, thus reducing the overall number of processed points.

The last filtering stage deals with the spurious noise. The outlier removal filter is comparing the normal covariance of the points in order to remove the ones which are noise point candidates. For the ToF cameras this is essential, as usually the level of noise is rather high for sunny images.

2.2 Joint 2D-3D recognition

The main idea behind the joint 2D-3D recognition is to reuse the already existing machine learning algorithms from the 2D domain and combine with the 3D application specific region focusing. The sketch of the proposed approach is visible in Figure 4: after the data acquisition and raw data processing, the 2D IR data is fed into a CNN based detection algorithm, while in parallel the 3D data is preprocessed for the late stage fusion.[16]

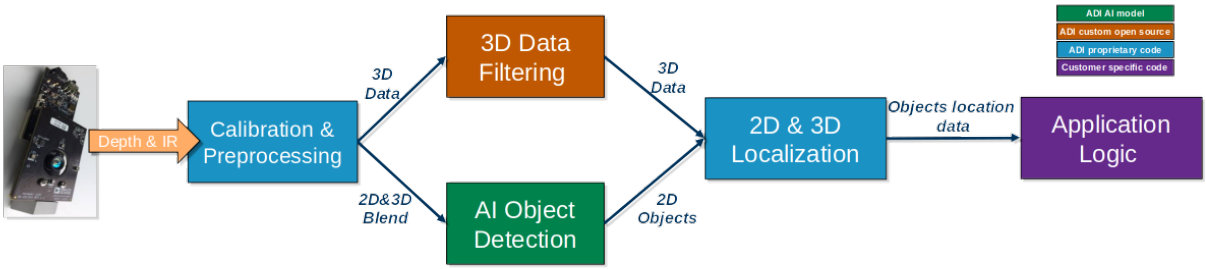

Figure 4. 2D-3D data fusion pipeline

The calibration part includes in our configuration setup the depth image validation for based on the received reflection's values, the image undistortion and the frame synchronization among the depth and IR image pairs.

The 3D filtering part included the steps summarized in the subsection 2.1.

The AI Object detection is an optimized variant for the SSD MobilenetV2 implementation on embedded GPU. We used the Jetson Nano board as primary embedded target, which has a native Int16 support for the tensor operations. Thus we pruned our initial custom IR image trained network for this specific architecture in order to optimize the size and speed of the network on this architecture. We also tested our pruning on Float32 optimized variants running on AGX device.

The last block of this processing chain relates to the application specific layer: here localization and pose estimation,[17] tracking or volumetric measurement can be adopted. Each of this block contains application specific code, reusing the generic 2D and 3D processing parts. In this paper we focus on the people detection and tacking experiments, thus this is presented in more details.

2.3 Tracking the carried object

One of the most common problems with the automatic door control setups is the activation of the doors in presence of the wild life animals as well as the collision or no activation in presence of large carried objects by humans. The first problem pays off in terms of heating and ventilation, while the second one may result in personal injuries.

We addressed these issues in our practical application specific module in order to enhance the door control to handle the above mentioned corner cases.

The first problem related to the door activation by non-human presence is directly solved with the 2D IR image based detection: we consider only as active case the presence of humans in the proximity of the door. This works for specific cases when the humans carry different objects (e.g. scale, shopping carts, etc) as these cases often appear for the door control systems mounted in supermarkets. A typical output of the detected region in 2D and 3D is shown in Figure 5. As it can be seen in the left hand side of the image, the detection is performed

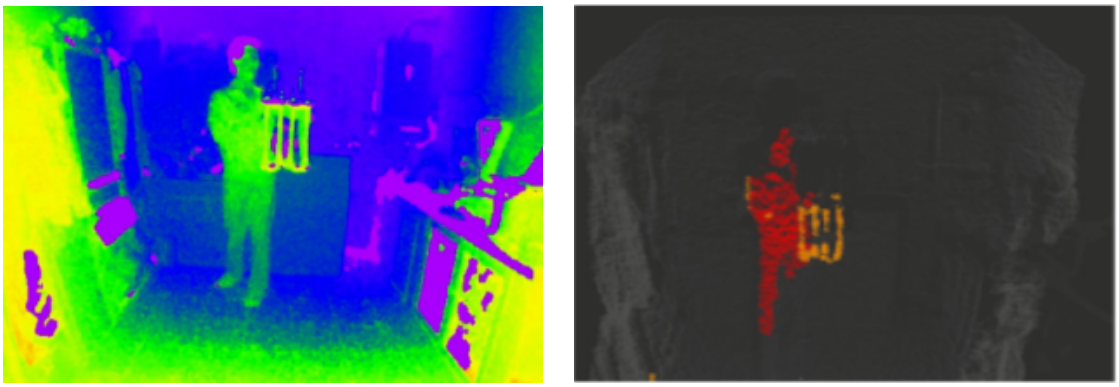

Figure 5. Tracking experimental result with ToF camera for humans caring objects

in the IR image (colorized with heat map for better visibility), while the 3D ROI is extracted based on the 2D detection and Euclidean clustering of the point cloud.

For the tracking part of the algorithm we considered the correlation based filter which is initialized with the 2D detection bounding box, and is running on the IR image using the CPU. We considered also the standard Kalman filtering approach for tracking using constant velocity model, but this seemed to be more sensible to the fast variations in the detection parts, making the re-identification of the same person difficult. For later development we will consider the GPU implementation of the correlation based filter, which should have better runtime performances on embedded GPU enabled devices.

3. EXPERIMENTAL RESULTS

For the experimental results we tested the algorithms is different light and weather conditions. The main aim of the tests was to tune spatial the person tracking algorithm in such a way that it remains robust against the environmental changes but runs with constraint runtime performances.

In order to validate the algorithm on different platforms, we tested against several embedded platforms the whole pipeline. The summary of the results are presented in Table 1.

Table 1. Summary of the runtime comparison for different devices with our method

Runtime comparison on different devices					
Device	Jetson Nano	Jetson NX	Jetson AGX	GTX 1060	Google Colab
Run Time $[s]$	0.54	0.31	0.21	0.047	0.11

As it can be seen from the experimental testing, the algorithm is able to run on $2-5$FPS on embedded Jetson platforms, which is usually sufficient for commercial solutions for door control applications.

4. SUMMARY

In this work we presented the implementation details of a generic 2D-3D processing pipeline for ToF cameras with application specific extension possibilities. The generic preprocessing and filtering part can be used for arbitrary setup, this being easy to integrate into user specific applications such as people tracking in a door control setup. The algorithms were tested and validated on a number of embedded platforms, showing good runtime performance metrics.

In the future we plan to make these modules to have more efficient run times, i.e. we plan to optimize the implementation especially for the filtering parts for embedded GPU platforms.

ACKNOWLEDGMENTS

The authors are thankful for the support of Analog Devices Romania, for the equipment list (cameras, embedded devices, GPUs) offered as support to this work. This work was financially supported by the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-III-P2-2.1-PTE-2019-0367.

REFERENCES

[1] Fooladgar, F. and Kasaei, S., "A survey on indoor RGB-D semantic segmentation : from hand-crafted features to deep convolutional neural networks," 4499–4524 (2020).

[2] Xue, H., Liu, Y., Cai, D., and He, X., "Tracking people in rgbd videos using deep learning and motion clues," *Neurocomputing* **204**, 70–76 (2016). Big Learning in Social Media Analytics.

[3] Zia, S., Yüksel, B., Yüret, D., and Yemez, Y., "Rgb-d object recognition using deep convolutional neural networks," in [*2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*], 887–894 (2017).

[4] Gezawa, A. S., Zhang, Y., Wang, Q., and Yunqi, L., "A review on deep learning approaches for 3d data representations in retrieval and classifications," *IEEE Access* **8**, 57566–57593 (2020).

[5] Qi, C. R., Yi, L., Su, H., and Guibas, L. J., "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," *Advances in Neural Information Processing Systems* **2017-Decem**, 5100–5109 (jun 2017).

[6] Liu, Z., Tang, H., Lin, Y., and Han, S., "Point-Voxel CNN for Efficient 3D Deep Learning," *arXiv* (jul 2019).

[7] Jafari, O. H., Mitzel, D., and Leibe, B., "Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras," in [*2014 IEEE International Conference on Robotics and Automation (ICRA)*], 5636–5643 (2014).

[8] Liu, H., Luo, J., Wu, P., Xie, S., and Li, H., "People detection and tracking using rgb-d cameras for mobile robots," *International Journal of Advanced Robotic Systems* **13**(5), 1729881416657746 (2016).

[9] Hou, L., Wan, W., Han, K., Muhammad, R., and Yang, M., "Human detection and tracking over camera networks: A review," in [*2016 International Conference on Audio, Language and Image Processing (ICALIP)*], 574–580 (2016).

[10] Han, X. F., Jin, J. S., Wang, M. J., Jiang, W., Gao, L., and Xiao, L., "A review of algorithms for filtering the 3D point cloud," *Signal Processing: Image Communication* **57**(February), 103–112 (2017).

[11] Rakotosaona, M. J., La Barbera, V., Guerrero, P., Mitra, N. J., and Ovsjanikov, M., "PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds," *Computer Graphics Forum* **39**(1), 185–203 (2020).

[12] Rusu, R. B. and Cousins, S., "3d is here: Point cloud library (pcl)," in [*2011 IEEE International Conference on Robotics and Automation*], 1–4 (2011).

[13] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , 4510–4520 (jan 2018).

[14] Lee, J. and Cho, J., "Effective covariance tracker based on adaptive changing of tracking window," in [*ICCAS 2010*], 1057–1058 (2010).

[15] Tamas, L. and Jensen, B., "Robustness analysis of 3d feature descriptors for object recognition using a time-of-flight camera," in [*22nd Mediterranean Conference on Control and Automation*], 1020–1025, IEEE (2014).

[16] Gao, J., Li, P., Chen, Z., and Zhang, J., "A Survey on Deep Learning for Multimodal Data Fusion," *Neural Computation* **32**, 829–864 (05 2020).

[17] Frohlich, R., Tamas, L., and Kato, Z., "Absolute pose estimation of central cameras using planar regions," *IEEE transactions on pattern analysis and machine intelligence* (2019).