https://doi.org/10.2352/ISSN.2470-1173.2021.18.3DIA-105

Semantic 3D Indoor Reconstruction with Stereo Camera Imaging

Xin Liu[1], Egor Bondarev[1], Sander R. Klomp[1,2], Joury Zimmerman[2], Peter H.N. de With[1]
[1] Eindhoven University of Technology, SPS-VCA group of Electr. Eng.; Eindhoven, The Netherlands
[2] ViNotion B.V.; Eindhoven, The Netherlands

Abstract

On-the-fly reconstruction of 3D indoor environments has recently become an important research field to provide situational awareness for first responders, like police and defence officers. The protocols do not allow deployment of active sensors (LiDAR, ToF, IR cameras) to prevent the danger of being exposed. Therefore, passive sensors, such as stereo cameras or moving mono sensors, are the only viable options for 3D reconstruction. At present, even the best portable stereo cameras provide an inaccurate estimation of depth images, caused by the small camera baseline. Reconstructing a complete scene from inaccurate depth images becomes then a challenging task. In this paper, we present a real-time ROS-based system for first responders that performs semantic 3D indoor reconstruction based purely on stereo camera imaging. The major components in the ROS system are depth estimation, semantic segmentation, SLAM and 3D point-cloud filtering. First, we improve the semantic segmentation by training the DeepLab V3+ model [9] with a filtered combination of several publicly available semantic segmentation datasets. Second, we propose and experiment with several noise filtering techniques on both depth images and generated point-clouds. Finally, we embed semantic information into the mapping procedure to achieve an accurate 3D floor plan. The obtained semantic reconstruction provides important clues on the inside structure of an unseen building which can be used for navigation.

Introduction

Defence and/or police officers may enter unknown buildings and hostile environments for indoor inspection. Currently, the commander coordinates these officers by radio communication with limited or no visual observations. This provides low global situational awareness for the commander, which degrades the efficiency and safety of the inspection. On-the-fly 3D reconstruction of premises via on-body sensors and simultaneous localization and mapping (SLAM) can help to achieve global situational awareness, by offering a live view of the 3D model and officer locations to the commander in a remote application.

Existing work on SLAM can be divided into active [1] and passive SLAM [2, 3, 4], based on the applied sensor types. However, to prevent military officers from being exposed, active sensors (e.g. LiDAR, ToF and infrared camera) cannot be employed. Therefore, in this paper, a SLAM system deployment is investigated with passive sensors for generation of a global 3D floor plan.

The hardest challenge of stereo SLAM is the inaccurate estimation of depth images caused by a small baseline of a stereo camera or poor indoor illumination. Incorrect depth images lead to noisy 3D reconstruction. It is also difficult to extract a meaningful 3D floor plan from a noisy reconstruction.

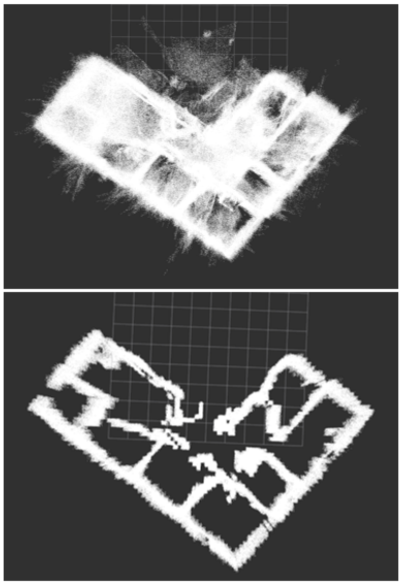


Figure 1. 3D Reconstruction from our stereo SLAM system. Top: point-cloud from SLAM baseline. Bottom: filtered point-cloud.

This paper proposes a stereo SLAM system, which provides an accurate 3D model (as shown in the example of Fig. 1) of an unknown indoor environment. We aim at providing a clean 3D reconstruction with walls, windows and doors. To achieve this, first, we propose several noise filtering techniques on both depth images and generated point-clouds. Second, we deploy all of our components into the robotic operation system (ROS). Finally, we generate and embed semantic information into the 3D mapping procedure.

State of the art

SLAM can be categorized into active and passive sensor SLAM. Our focus is on the passive sensor SLAM, which can be further divided into mono SLAM and stereo SLAM. Mono SLAM incurs low cost, but its performance limited due to lack of scale information [7]. The emergence of stereo cameras have solved this problem. A stereo camera is similar to the human eye, where the depth information is calculated by the difference between the left and right images [10].

The stereo SLAM system can build a sparse or dense 3D reconstruction, by utilizing the estimated depth image and visual

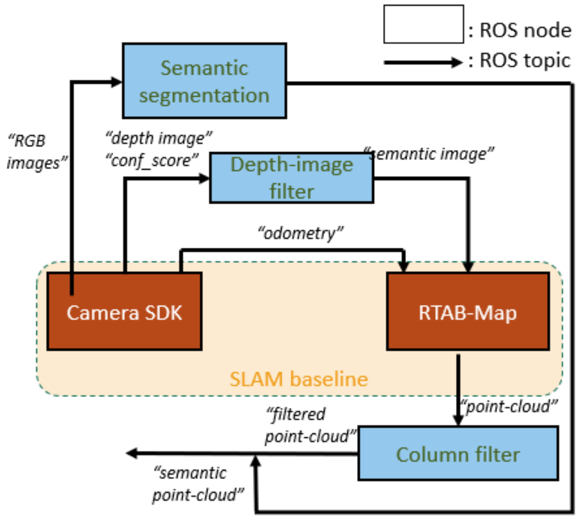


Figure 2. *Block diagram of the proposed system in ROS. The square and arrow represent an ROS node and an ROS topic, respectively. The blue ROS nodes compose our SLAM baseline. The camera SDK [6] is responsible for processing the stereo images and publishing topics, such as estimated depth images. RTAB-Map [3] is responsible for real-time pose tracking of the sensor and global point-cloud generation from the received topics.*

odometry. A sparse reconstruction cannot provide completeness in environment description, thus, it is normally used for localization purposes [2]. A dense reconstruction provides more complete data, but as the scanned environment becomes larger, the computation becomes extremely expensive [11]. An Open-source SLAM system RTAB-Map provides a proper memory management, which makes the reconstruction process independent of time and scale of the environment [3].

The 3D map reconstructed by stereo SLAM is usually limited and noisy, due to inaccurate depth estimation. In addition to the small camera baseline, the variable indoor lighting conditions and depth interpolation are the main influencing factors. Incorrect depth information can lead to a very noisy 3D reconstruction. Therefore, an accurately tuned indoor SLAM system with passive stereo sensors is required, where additional filtering and semantic data integration can help to recover from the deficiencies of the stereo data.

Method: Architecture and stages

Fig 2 presents the proposed ROS-based system architecture. Subsection A first introduces the SLAM baseline. Afterwards, Subsection B explains the preprocessing step of the estimated depth image. Then, the semantic segmentation training procedure is explained in Subsection C. Subsection D introduces the proposed point-cloud “column filtering” approach. Finally, Subsection E discusses the integration of semantic segmentation into the proposed system.

A. SLAM baseline

The SLAM baseline consists of two parts: 1) the SDK of the selected Zed Mini camera [6] and 2) the visual SLAM system known as Real-Time Appearance-Based Mapping (RTAB-Map). Both parts are integrated as ROS nodes.

The camera SDK [6] (Stereolabs Inc., San Francisco, USA) estimates and interpolates depth information from the left and right image pairs. The odometry is estimated by the embedded IMU. The depth and odometry information is published as an ROS topic for other ROS nodes.

The RTAB-Map node subscribes for the depth image and odometry topics to generate the 3D point-cloud of an unknown environment. The left part in Fig. 3 presents a 3D reconstruction of a building, based solely on the SLAM baseline. The radial noise is caused by the incorrectly estimated depth information. At such quality, the commander is not able to recognize that there are six rooms inside the building. Therefore, we improve this baseline with the three techniques (shown as yellow blocks in Fig. 2): depth-image filtering, semantic segmentation and column filtering.

B. Depth-image preprocessing

We observe that the depth error increases with distance. Each pixel in the depth image has a confidence score. As a SLAM preprocessing step, we discard pixels with long distances and low confidence scores based on empirically defined thresholds (in this study, the distance threshold is 8 meters, the confidence threshold is 0.95). As shown in Fig. 2, the node publishes the filtered depth images, serving as an input to the RTAB-Map node.

C. Semantic Segmentation

To pinpoint locations of interest on the map for the commander, such as windows, doors and stairs, a semantic segmentation network is trained with public datasets to classify these objects at the pixel level. Mapping of the pixel-level labels to either the 3D reconstruction or 2D floor map is then performed by back-projection.

For practical feasibility, the floor-plan creation system should operate at near real-time speed on a laptop. This places constraints on the selection of the segmentation neural networks architecture. Acceptable execution speed can be achieved with the the Deeplab V3+ model [5] with a ResNet-50 [**?**] backbone, using images downscaled to 853 × 480 pixels and speed-up by TensorRT and fp-16 execution. Initial experiments were performed with the ResNet [9] backbone and at higher resolutions, but the accuracy was similar, while being significantly slower.

To improve the accuracy of the semantic segmentation network, we have re-trained it on a combination of several public datasets [12, 13, 14, 15, 16]. Public datasets all use different classes and different data formats, which makes merging a non-trivial task. In order to create a usable dataset, seven classes have been defined that are considered to be of interest to the project. ‘Wall’, ‘floor’ and ‘ceiling’ can be used by the floor-plan generation algorithms to make them more accurate. ‘Window’, ‘door’ and ‘stairs’ are special objects of interest that should be marked on the floor plan, because they are highly relevant for the commander. Finally, all other objects are marked as ‘other’, to ensure that the algorithm is not forced to assign any of the first six classes to irrelevant objects such as chairs. The mapping of classes from public datasets to our own classes of interest is shown in Table 1. The ”stairs” class in Stanford2D3DS and 3DFacilities is not annotated, but is now a part of the datesets’ ”unknown” class. To prevent network confusion during training, the ”unknown” class from these two datasets is ignored entirely and instead, it

Classes	**ADE20k**	**SUNRGBD**	**Stanford2D3DS**	**3DFacilities**	**COCO-stuff**	**Ours**	
wall	✓	✓+ whiteboard	✓+ column + beam	✓+ column + beam	✓	✓	
floor	✓	✓	✓	✓	✓+ carpet + rug	✓	
ceiling	✓	✓	✓	✓	✓	✓	
window	✓+ blind + screen	✓+ blinds	✓	✓	✓	✓	
door	✓+ screen door	✓	✓	✓	✓	✓	
stairs	✓+ step + escalator	*x*	*x*	✓	✓	✓	**Total**
#images	22,210	10,335	70,496	25,586	123,287	393	251,914
#images	11,692	9,710	69,993	10,143	19,822	393	**121,360**

Table 1. Summary of combined datasets for semantic segmentation. All sets are publicly available, except ”test set”, which consists of our own annotated recordings of four different university buildings and four buildings on another terrain.

is mapped to our ”other” class. Furthermore, the images are filtered on scene type (only indoor scenes are selected) and class coverage (at least 30% of the pixels in an image should be within the classes of interest).

The deep learning network is evaluated on a small, self-annotated dataset gathered in buildings on the other terrain, containing 258 annotated images. To be able to train the network with images taken by the ZEDm camera used in this work, without compromising the validity of the test set, a separate set of 135 images from three university buildings is annotated and added to the training set. Finally, this results in a train set of 121,102 images for training and 258 for testing.

D. Column filtering

The 3D reconstruction example in the left part of Fig. 3 contains noise caused by incorrect depth estimations. This noise can be regarded as outliers in reconstructed building structures. Thus, we filter the reconstructed point-cloud by removing points that are detected as outliers. Our filtering technique assumes the following steps. 1) The baseline output contains a dense reconstruction of the main structures, such as walls, doors and cupboards (i.e. main structures should contain more points than the noisy point-cloud areas). 2) All the structures of interest (walls, doors, windows) are perpendicular to the ground plane. 3) The buildings are single-story or multi-story with aligned walls (the structure of each floor is the same). This study does not target reconstruction of ceilings and floors, since they occlude the inside structures and prevent viewing them. Hence, we employ a ’3D column filter’ to remove points not belonging to the structures of interest.

Fig. 3 depicts an example of the filtering procedure. We first separate the horizontal plane into a 10×10-cm grid map of 2D cells, since we found that the scanned mean thickness of the walls are close to 10 cm. Each grid cell aggregates corresponding points in a vertical direction. The boundaries of the cell extruded in the vertical direction define the vertical 3D columns. We calculate the amount of points that belong to each column as the column value. Each column can be seen as a square pillar on the top-view grid map. Computation of the column value converts the pillar to one value assigned to the corresponding grid cell. We then apply a sliding 3×3 mask window to the grid map to check the neighbors of each column. We calculate the mean value under each mask and refer to it as ‘local threshold’. We have also experimented with median and standard deviation operation for the ’local threshold’ calculation. The median operation provides a grid map with many scattered grids at noisy areas (more noise components are preserved) compared to the other two operations. Considering the computational cost, we have adopted the low-cost mean operation, since its performance is similar to using the standard deviation. We accept a column value if that value is higher than the corresponding local threshold.

After filtering by the local threshold, we calculate the mean value of all surviving columns to define a ‘global threshold’. If the column value is higher than the global value, then this column is seen as valid and the points located inside are kept as a part of the building structure. Otherwise, the points are considered outliers and removed from the point-cloud.

E. Integration of semantic segmentation

We use the depth image combined with the semantic segmentation image to create a “semantic point-cloud” by aligning the 2D semantic information with the 3D point-cloud. Each voxel represents a cube of optimal size $20 \times 20 \times 20$ cm. The class label of each voxel is defined by the most frequently occurring semantic class (e.g. door) of all 3D points in that voxel, combined with the class labels of the neighboring voxels.

Experiments and results

The first part of the section shows the quantitative semantic segmentation results on our test set. This is followed by qualitative results for the column-filtered point-cloud, and finally, the semantic 3D integration model is discussed.

Dataset creation is based on using the ZED Mini (ZEDm) stereo camera. All results depend on this dataset. Four buildings with different properties were scanned and reconstructed for evaluation. Table 2 shows important properties of these four buildings.

Building #	**B1**	**B2**	**B3**	**B4**
Light	Off	On	On	Off
Window	Open	Close	Close	Half closed
Roof type	Tilted	Titled	Flat	Flat
Floors	2	1	3	1

Table 3. Properties of the four different scanned buildings, referred to as B1 through B4, in our testing dataset.

A. Semantic segmentation results

To evaluate the semantic segmentation network, we compare both the accuracy and mean Intersection over Union (mIoU) on our annotated other terrain testset. The results are shown in Table 3. As a sanity check, first the scores for trivial segmentations consisting entirely of the most common class (wall) were computed to derive a reference point, giving a 55% score, which is not plotted in the table. Then we use the 8 classes for performance testing and increase the dataset sizes, as shown in the table. To verify that the network has sufficient learning capacity, we overtrain on the test set, which indeed reaches near-100% perfor-

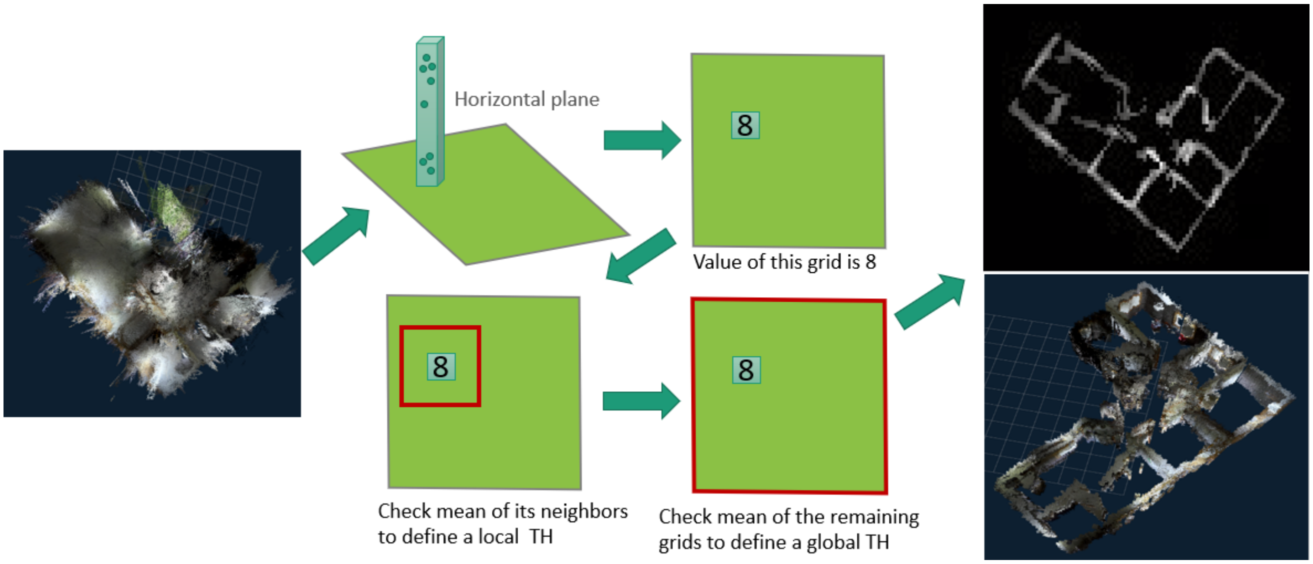

Horizontal plane
8
Value of this grid is 8
8
Check mean of its neighbors to define a local TH
8
Check mean of the remaining grids to define a global TH

Figure 3. *Procedure of the column filter. Input is the row point-cloud (left image), the output is a clean point-cloud (bottom right) and 2D floor plan (top right).*

Model: 8 classes	**Accuracy**	**mIoU**
Trained on ADE20k	71.40%	40.32%
Trained on public datasets	70.93%	44.54%
Trained on public + own data	71.70%	46.58%
Overtrained (upper bound)	97.69%	94.17%

Table 2. Impact of combining public datasets and adding a small amount of same-camera data on accuracy and mIoU.

mance. Next, we evaluate the performance of the network trained only on the commonly used ADE20k dataset as a realistic baseline. This already results in reasonable performance, but a higher mIoU is desirable to make the final semantic floor plan more accurate. Training with the filtered combination of public datasets improves the mIoU by over 4 percentage points. Finally, adding the 135 university-building images (each image 100 times for balancing reasons) improves the mIoU by another 2 percent points. This means that having images taken with the same camera that is used for testing is still valuable, even when the scenes are completely different and the number of training images with the same camera is small.

B. Column-filtering results

Fig. 4 shows the experimental results. The top subfigures depict the original 3D point-clouds, while the bottom subfigures show the filtered clean 3D maps of buildings in our dataset. It can be observed that most radial noise points are removed by the column filtering. It can be noticed that there are empty areas in the reconstructed wall structures. The reasons for these empty areas are twofold. 1) The empty areas could not be scanned during dataset creation (e.g. the scanning route was blocked by furniture). 2) The empty areas can be windows, with the state of these windows being open (e.g. building B1). Only the frames of the window are left in this area and there are not enough points to complete the inner part of the window. Thus, these areas are seen as invalid and the points inside are removed. In future work, we will aggregate semantic information to the column-filtering procedure to prevent removing points that belong to an opened window. It can be observed that the internal structures of building B2, B3 and B4 are visible. The filtered building B1 is not as clean as other buildings, since it has a tilted roof. Our column filter is currently limited in filtering these kinds of non-vertical structures, since it is beyond the scope of our building assumptions.

C. Integration of semantic segmentation

As explained in the method section, we align the 2D semantic information with the 3D point-cloud based on a voxel-grid approach. The resulting semantic 3D reconstruction is shown in Fig. 5. Colors present different classes. Walls, windows and doors are colored as red, yellow and blue, respectively. Currently, the semantic reconstruction is not very accurate and needs to be improved in future work. However, it is a meaningful map for a commander to use in practice, since it offers clean building structures and provides direct clues on the structural elements of the rooms.

Conclusion

In this paper, we have introduced a ROS-based SLAM system for a passive stereo camera, which also creates a depth signal besides RGB data. The proposed system generates a clean indoor 3D reconstruction, even with inaccurate depth information from the small baseline stereo camera. The key contribution to our system performance is as follows. We deploy a point-cloud "column-filtering" approach to remove undesired data points, while retraining a DeepLab V3+ model with a filtered combination of public datasets to extract semantic information. Then we utilize a voxel-grid approach to integrate the 2D semantic information with the clean (filtered) 3D reconstruction. The important objects such as walls, doors and windows are labeled in the clean 3D reconstruction. Experimental results show that for our own created dataset, we are able to remove most of undesired points in the 3D reconstruction of buildings with different properties. The semantic reconstruction provides important clues on the inside structure of an unseen building which can be used for navigation. For future work, a more accurate semantic 3D map is pursued.

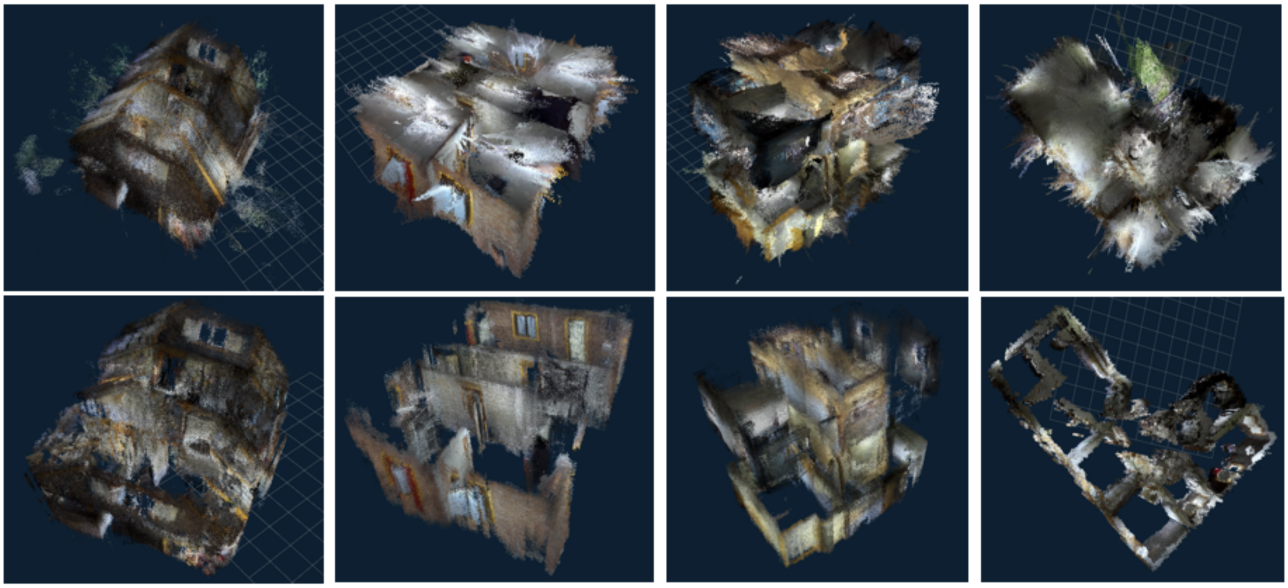


Figure 4. *Comparison between SLAM baseline output and column-filtering output. 3D Reconstruction of four different buildings are listed from left to right. Images in the top line are results from RTAB-Map, Images in the bottom line are column-filtered results.*

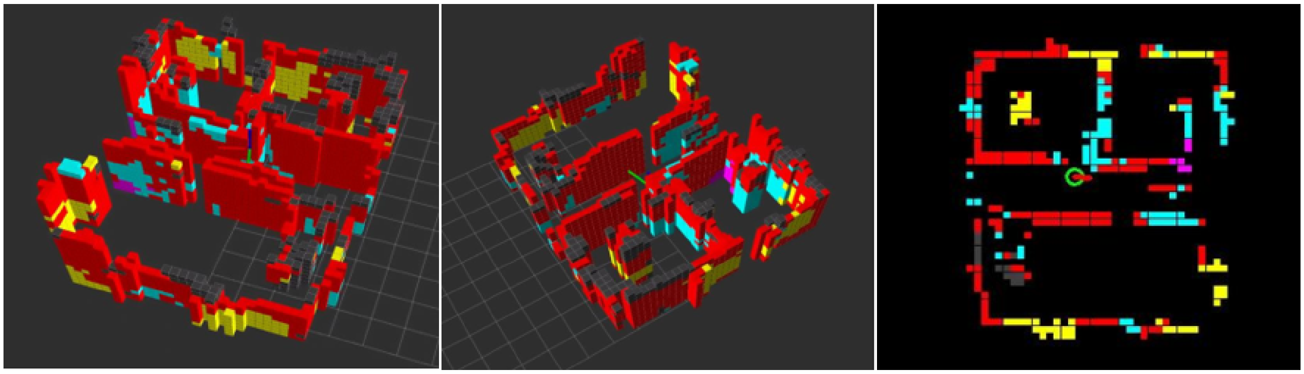


Figure 5. *3D Reconstruction with semantic information. Left and middle subfigures portray 3D views, the right figure is a top view. Different colors present different classes. Red: wall. Blue: door. Yellow: window. Black: others.*

References

[1] W. Zhang, Q. Zhang, K. Sun and S. Guo, “A laser-slam algorithm for indoor mobile mapping,” Remote Sensing and Spatial Information Sciences, pp. 351-355, July 2016.

[2] R. Mur-Artal and J. D. Tardos, “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras,” CoRR, vol. abs/1610.06475, 2016.

[3] M. Labbe and F. Michaud, “RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” Journal of Field Robotics , pp. 416-446, 2018.

[4] S. Yang and S. Scherer, “Monocular Object and Plane SLAM in Structured Environments,” IEEE Robotics and Automation Letters, pp. 3145-3152, 2019.

[5] L. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, ”Encoder-decoder with atrous separable convolution for semantic image segmentation”, ECCV, 2018.

[6] STEREOLABS, “ZED SDK,” STEREOLABS, 2020. [Online]. Available: https://www.stereolabs.com/developers/release/.

[7] J. Engel, T. Schöps and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM”, in ECCV, 2014.

[8] K. He, X. Zhang, S. Ren, J. Sun, ”Deep Residual Learning for Image Recognition”, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[9] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Muller, R. Manmatha, M. Li, A. Smola, ”ResNeSt: Split-Attention Networks”, in arXiv preprint arXiv:2004.08955, 2020.

[10] OpenCV, “Depth Map from Stereo Images,” OpenCV, 2020. [Online]. Available: `https://docs.opencv.org/3.4/dd/d53/tutorial_py_depthmap.html`

[11] R. A. Newcombe et al., ”KinectFusion: Real-time dense surface mapping and tracking,” 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, 2011, pp. 127-136, doi: 10.1109/ISMAR.2011.6092378.

[12] S. Song, S. P. Lichtenberg and J. Xiao, ”SUN RGB-D: A RGB-D scene understanding benchmark suite,” 2015 IEEE Conference

on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 567-576, doi: 10.1109/CVPR.2015.7298655.

[13] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso and A. Torralba, ”Scene Parsing through ADE20K Dataset,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 5122-5130, doi: 10.1109/CVPR.2017.544.

[14] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese.Joint2d-3d-semantic data for indoor scene understanding.arXiv preprint-arXiv:1702.01105, 2017.

[15] Thomas Czerniawski and Fernanda Leite.3DFacilities: Annotated 3D Re-constructions of Building Facilities, pages 186–200. 01 2018.

[16] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing andstuff classes in context. InComputer vision and pattern recognition (CVPR),2018 IEEE conference on. IEEE, 2018.

Author Biography

Xin Liu received her MSc degree in Electrical Engineering in 2019 from the Eindhoven University of Technology. She is currently a PhD candidate at the same university. Her research interests include computer vision and simultaneous localization and mapping

Egor Bondarev obtained his PhD degree in the Computer Science Department at TU/e, in research on performance predictions of real-time component-based systems on multiprocessor architectures. He is an Assistant Professor at the Video Coding and Architectures group, TU/e, focusing on sensor fusion, smart surveillance and 3D reconstruction. He has written and co-authored over 50 publications on real-time computer vision and image/3D processing algorithms. He is involved in large international surveillance projects like APPS and PS-CRIMSON.

Sander Klomp received both his BSc and MSc degrees from the Eindhoven University of Technology (2016,2018) with the designation Cum Laude. He is now pursuing a PhD degree at TU/e in collaboration with ViNotion, with a focus on efficient deep learning algorithms.

Joury Zimmermann received his bachelor’s degree in Software Engineer from Fontys Hogescholen Eindhoven (2013, 2016). Currently he is employed as a software developer at ViNotion.

Peter H.N. de With is Full Professor of the Video Coding and Architectures group in the Department of Electrical Engineering at Eindhoven University of Technology. He worked at various companies and was active as senior system architect, VP video technology, and business consultant. He is an IEEE Fellow and member of the Royal Holland Society of Sciences, has (co-)authored over 600 papers on video coding, analysis, architectures, and 3D processing and has received multiple papers awards. He has been a program committee member of several IEEE conferences and holds some 30 patents.

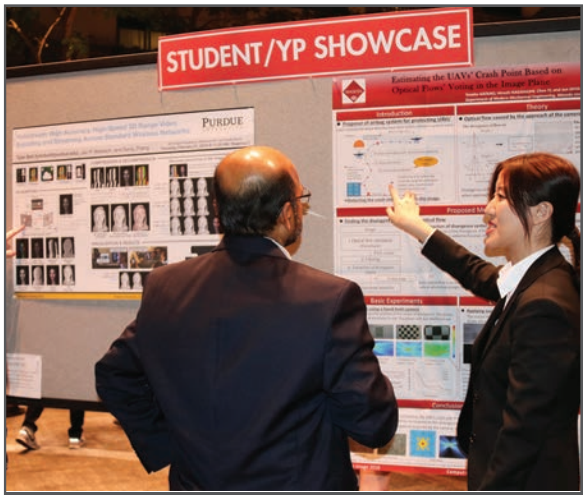
JOIN US AT THE NEXT EI!

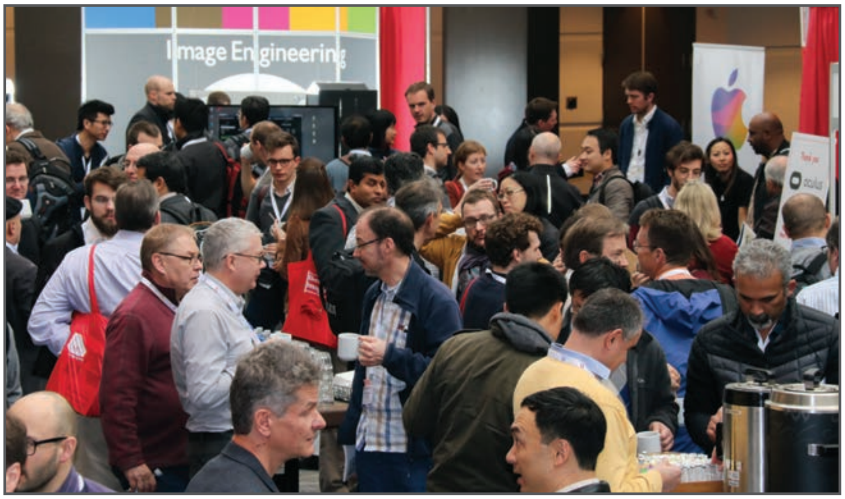
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!


STUDENT/YP SHOWCASE
Estimating the UAVs' Crash Point Based on Optical Flows' Voting in the Image Plane
PURDUE



AR is a fundamental and transformative interface and platform.
20+ years from my initial working demo to today's AR displays
Optimistic that it won't take another 20 to reach ubiquitous consumer AR!
My first working AR prototype
Today's platforms
Ubiquitous consumer AR

• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org