Advanced Topics in Software Engineering: Augmented Reality - Anchoring, Scanning, and World Understanding

Sonal Lakhotia
MatrNr: 14913835
Supervisor: Dr. Philip Makedonski

March 17, 2023

Contents

1 Introduction

Technological advancements have changed how we see, feel, choose, and perceive things. Sci-fi movies like "The Matrix" no longer seem fictional. Innovative solutions have led to an amalgamation of the natural and virtual worlds. One such technology is *Augmented Reality* (AR). AR integrates the user's natural environment with computer-generated digital content that enhances the perception of the real world. The physical and virtual worlds work in coordination and create a new and improved natural environment where the virtual content enhances the user's understanding of the real world. AR does not replace natural environments like *Virtual Reality* (VR). It either adds virtual information to physical space or masks the natural environment. As shown in Figure 1, an image on a textbook is augmented into a Three-Dimensional (3D) model to provide a realistic view and greater understanding to the users. Figure 2 illustrates usage of AR while navigating through the streets. It superimposes information and details on the display as the user walks.

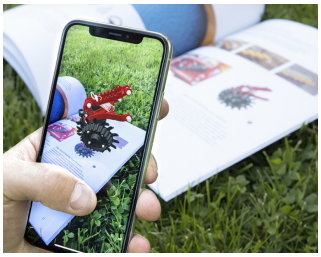


Figure 1: 3D model superimposed on the image [1]

Figure 2: Enhanced world understanding in AR [1]

AR provides an interwoven real and virtual experience by accurately registering the objects in virtual and physical worlds in 3D and placing them in the 3D real world in a realistic way [2]. Almost all sectors use AR for training and collaboration tasks as it provides better perception and retention of the concepts being discussed. Some notable industries using AR are movies, gaming, healthcare, education, military, and construction [3]. Applications like Snapchat[1], Google Lens[2], Pokémon GO[3], and IKEA Place[4] use AR. Novel advancements in AR have enabled applications like Snapfeet[5] and Qlone[6]. Snapfeet provides a real-time 3D reconstruction of a user's feet with precision, that enables them to select footwear online with ease. As shown in Figure 3, a user scans a foot, a 3D reconstruction of the foot is created accurately which is then exported to the application, and an AR footwear is selected according to the size of the user's feet. Qlone supports 3D scanning and reconstruction of models to be used in AR for superimposing visual information or anchoring it in the natural environment.

These AR applications are Simultaneous Localization and Mapping (SLAM) enabled. SLAM allows dynamic determination of the user's position and orientation with scene entities

[1] *Snapchat*-`https://www.snapchat.com/`. last retrieved: 22.11.2022
[2] *Google Lens*-`https://lens.google/`. last retrieved :22.11.2022
[3] *Pokémon GO*-`https://pokemongolive.com/en/`. last retrieved: 22.11.2022
[4] *IKEA Place*-`https://apps.apple.com/us/app/ikea-place/id1279244498`. last retrieved: 22.11.2022
[5] *Snapfeet*-`https://snapfeet.io/en/`, last retrieved: 22.11.2022
[6] *Qlone*-`https://www.qlone.pro/`, last retrieved: 22.11.2022

and precise anchoring of the virtual entities in the real world [4]. As shown in Figure 4, the 3D virtual character is accurately anchored with respect to its surrounding environment. SLAM finds use cases in domains of robotics, autonomous driving, computer vision, AR and VR. SLAM approaches utilized expensive laser scanners to develop autonomous applications. The emergence of robust, dense real-time mapping and consumer-grade Red Green Blue-Depth (RGB-D) cameras led to the development of several SLAM-enabled AR consumer products [5]. RGB-D cameras are advantageous as they provide the Red Green Blue (RGB) as well as the depth information, which is useful in estimating the full 6 degrees of freedom of position in a cost effective way [6]. This paper discusses a real-time dynamic, semantic, and an instance-aware RGB-D SLAM system, MaskFusion [7]. It can recognize, track, detect, and reconstruct multiple moving rigid objects and assign semantic labels to precisely segmented object instance. MaskFusion does not require knowing the objects' models in advance and works well with numerous independent motions of the entities in the scene. Instance-level semantic segmentation enables MaskFusion to fuse semantic labels into an object-aware map.

Figure 3: Snapfeet [8]

Figure 4: SLAM enabled anchoring in AR [9]

In the next sections we examine the progress in real-time SLAM systems and provide an overview of the MaskFusion system. Rest of the paper is organized according to the following structure: Section 2, describes the development of various real-time SLAM systems and segmentation. Section 3, provides the details about the principles and the working of the RGB-D dynamic SLAM system. In Section 4, we discuss how the system tackles the problems not solved by traditional SLAM systems and we conclude the paper and discuss the future scope in section 5.

2 Background

The confluence of SLAM and AR led to advancements in AR applications [4]. Despite advances, SLAM and its application to AR remain in their infancy in two areas. First, SLAM methods assume that the environment is static, and the moving entities are considered outliers and removed. Second, SLAM methods purely output a geometric map of the environment. Visual SLAM is offering solutions to the challenges of jointly tracking the positions and orientations of a moving camera while reconstructing a map of the environment. It is the camera-only variant of SLAM that required expensive lasers and Intertial Measurement Units (IMUs). The emergence and availability of consumer-grade RGB-D cameras, such as Microsoft Kinect, have stimulated dense real-time reconstruction methods [7].

1. **Dense RGB-D SLAM:** RGB-DSLAM enables quick acquisition of colorful 3D models of the scenes and the objects with kinect-style depth cameras [10]. These methods accurately map indoor environments and gained popularity in robotics and augmented reality. KinectFusion [11] established that Truncated Signed Distance Function (TSDF) based map representation can attain efficient mapping and tracking in small environments because it represents the scene volumetrically and each location stores the distance to the nearest surface.

   Surface elements (surfels) are of great importance in computer vision and computer graphics. It is a powerful paradigm that renders complex geometric objects at interactive frame rate very efficiently [12]. A map of surface elements differs from point cloud in the aspect of encoding local surface properties. These properties include radius, normal and location. Surfel clouds representations were introduced to RGB-D SLAM and proved that they are more memory efficient than TSDF-based maps [7].

2. **Scene Segmentation:** Object and scene segmentation data applied to visual tracking and mapping systems enhance their functionality. For instance, it enables robots to detect objects. There exist methods to segment RGB-D data based on geometric properties of surface normals [13–16] which produce perfect object boundaries but tend to over-segment and convey no semantic information.

3. **Semantic Scene Segmentation:** Related work aims at semantically segmenting 3D scenes using Markov Random Fields (MRFs). These methods use 3D labeled data which is not readily available like the Two-Dimensional (2D) image data, and the training data needs manual annotation as seen in [17]. Datasets like NYUv2 [18] that contain isolated RGB-D frames are not applicable here. It is also seen in [19] that building reconstructed datasets for segmentation require lots of efforts.

4. **Semantic SLAM:** Semantic SLAM systems are motivated by the integration of deep neural networks with real-time SLAM systems [20, 21]. 2D information is responsible for making inferences avoiding the need for 3D annotated data. The system offers strategies to fuse labeled data into segmented 3D maps. The system does not consider object instances and hence does not attain tracking multiple models independently.

5. **Dynamic SLAM:** Dynamic SLAM involves two scenarios. First, a non-rigid surface reconstruction assumes a deformable world and performs a rigid registration. Second, a multibody formulation of independently moving objects wherein the instances are identified and tracked sparsely or densely. These methods employ template-based techniques [22–24] that require observing the object of interest in advance.

   MaskFusion is a novel dynamic real-time SLAM system that reconstructs an entire scene, segments object instances, and adds semantic labels to the objects. Figure 5 provides a comparative overview of the real-time SLAM system with respect to MaskFusion under five significant properties. Static-Fusion [25] segments and ignores the dynamic parts of a scene and Co-Fusion [26] does not offer real-time tracking, segmenting and reconstructing objects based on semantic labels. In contrast to the related semantic system systems Slam++ [23], CNN-SLAM [20], 2.5D is not enough [22], and Semantic Fusion [21], MaskFusion reconstructs objects even when their motion differs from

the camera. It also segments object instances. Unlike dense non-rigid reconstruction systems Non-Rigid RGBD [24], Fusion4D [27], and Dynamic-Fusion [28], MaskFusion reconstructs the entire scene and adds semantic labels to different objects.

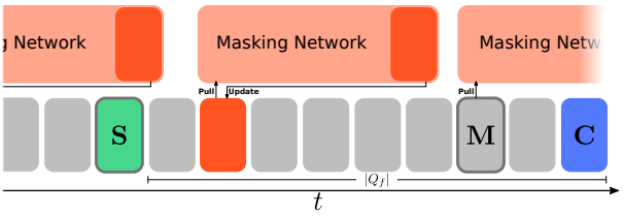
Important properties considered for the related systems					
Exisitng related SLAM systems	Model-free	Scene Segmen-tation	Semantics	Multiple moving objects	Non-Rigid
Static-Fusion	✓	✓			
2.5D is not enough		✓	✓		
Slam++		✓	✓		
CNN-SLAM	✓	✓	✓		
Semantic-Fusion	✓	✓	✓		
Non-Rigid RGBD					✓
Dynamic-Fusion	✓				✓
Fusion4D	✓				✓
Co-Fusion	✓	✓		✓	
Mask-Fusion	✓	✓	✓	✓	

Figure 5: Comparing MaskFusion to other related real-time SLAM systems [7]

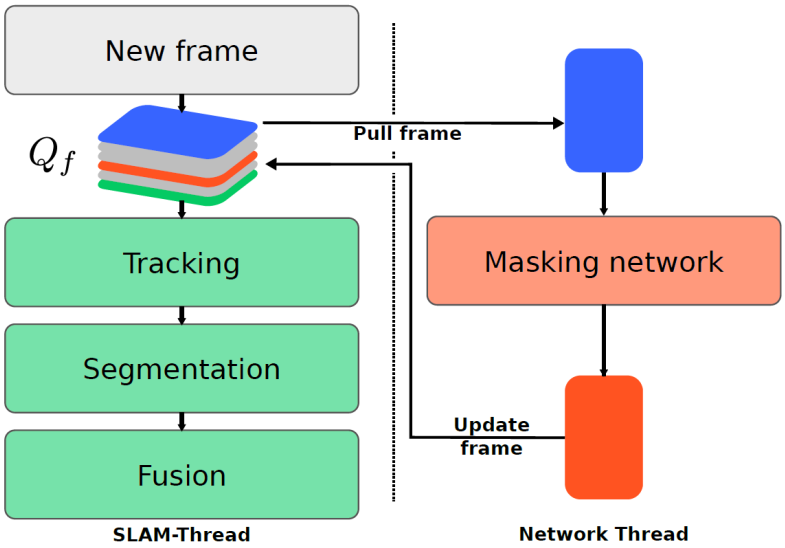
3 System Overview

MaskFusion represents dynamic scenes at the level of objects and assigns semantic labels to them. It uses *Mask-Region Based Convolutional Neural Network* (Mask-RCNN)[29], an image-based instance-level segmentation algorithm that can predict object labels for up to 80 object classes, and a geometry-based segmentation algorithm that increases the accuracy of the object boundaries in object masks. Dynamic SLAM frameworks use the object masks as an input to track and fuse moving objects and propagate the semantic image labels into the 3D map labels. The system uses instance-level semantic segmentation over pixel-level semantic segmentation to avoid having different entities that belong to the same object category be considered as a single blob. For each object tracked in the scene, MaskFusion maintains a 3D representation. Every model is tracked and fused separately. With each new frame acquired by the camera, SLAM and masking deep neural networks interact as shown in Figure 6

Tracking, Segmentation and Fusion are performed whenever a new frame is acquired by the camera. They are discussed in the following sections.



(a) In this timeline, frame S and frame M are highlighted with thick borders, as the SLAM and masking threads are working on them respectively. C, the current frame is shown in blue, the head of the queue is shaded in green, and frames with available object masks are marked orange [7].



(b) Data flow in MaskFusion: Q_f indicates a fixed-length queue to which the camera frames are added. The SLAM system which is represented in green in operates on the queue's head. The semantic masking deep neural network pulls the frames from the queue and adds the updated frame back to the queue when the semantic masks are available [7].

Figure 6: An overview of SLAM back-end, masking network and their interaction [7].

3.0.1 Tracking

A set of surfels represent the 3D geometry of each object.The tracking thread minimizes the photometric cost. It looks for photometric consistency between the surface of the objects in sight, the 3D model, and the observed image. Iterative closest point geometric cost minimizes the distance between the model and the newly constructed surfels. Tracking jointly optimizes photometric and geometric costs. A rigid transformation enables inferring the current position of the camera and segmented objects. Non-static objects are tracked separately to lower the computational demand and enhance robustness. Strategies to check for a static object include checking for motion inconsistency as in [26] and human-interaction with the object. If an object is touched by a human, it is considered non-static [7].

3.0.2 Segmentation

MaskFusion combines semantic and geometric segmentation. Mask-RCNN is used to obtain object masks with semantic labels. Although this algorithm provides good object masks, it has two disadvantages. First, the algorithm does not operate in real-time and can only work at a maximum of 5Hz. Second, imperfect object boundaries as they tend to leak into the background. A geometric segmentation algorithm, based on an analysis of depth discontinuities and surface normals, overcomes these limitations. The geometric segmentation runs in real time and produces very accurate object boundaries in contrast to the semantic instance segmentation as seen in Figure 7. Although, semantic labelling provides smooth boundaries it fails to accommodate important details from the segment of interest.

(a) Segment of interest (b) Semantic only (c) With geometric

Figure 7: Illustration of boundaries produced by semantic labelling and the merged geometric and semantic labelling [7]

MaskFusion tracks and reconstructs multiple objects simultaneously, and maintains separate models. It is essential to associate new data with the correct model before fusion. Segmentation takes place in 2D instead of associating data in 3D inspired by Co-Fusion [26]. As per this mechanism, new frames are masked and subsets of data are fused with existing models. Masking is based on semantic instance segmentation labels in synchronicity with geometric segmentation proposed by DNN [29], which improves the quality of object boundaries. MaskFusion's semantic segmentation pipeline provides masks at about 30Hz.

The pipeline is designed based on the following observations: (i) Current instance-level semantic segmentation methods detect objects but fail to produce precise object boundaries. (ii) Mask-RCNN [15] cannot be executed at frame rate. (iii) The details included in RGBD frames allows over-segmentation of the image.

The second observance directly suggests that to achieve an overall real-time performance, instance-level semantic segmentation must be performed in a parallel thread along with the tracking and fusion threads. The concurrent execution of two programs at different frequencies requires a synchronization strategy. As shown in 6(a), new frames are buffered in a queue Qf and direct the SLAM system to the queue's head, while the semantic segmentation operates on the queue's rear. The execution of the SLAM pipeline is delayed by the worst-case processing time of semantic segmentation. Due to the lower execution frequency of the masking component semantic segmentation is not available for most frames. Each frame requires labeling to fuse new data, associating regions of mask-less frames with existing models provides a solution to this issue. The pipeline shown in 6 is executed for each frame that

is processed by the SLAM-Thread shaded in green in 6. While geometric segmentation is performed for each frame, the mapping between geometric labels and semantic masks depends on the availability of the semantic masks. If they do not exist, the geometric labels are directly associated with the associated models as shown in 8

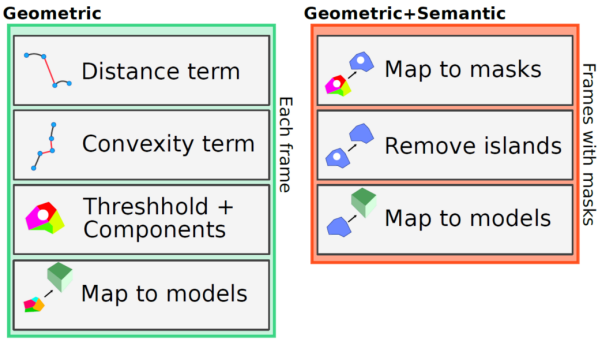


Figure 8: An overview of merged segmentation in MaskFusion [7]

Mapping geometric labels to masks Input frames are segmented geometrically and the components obtained as a result are mapped to masks by determining the one with the maximum overlap. If the overlap is greater than the threshold a mapping is assigned. Multiple components are capable of mapping to the same mask but only one mask can be linked to a component. If an assignment is made, an updated label is computed and it replaces the component with mask IDs [7].

Mapping masks to models An overlap between grouped components and the projected object labels takes place. The projected object labels are generated by rendering all the models using an OpenGL pipeline. It is affirmed that the mask IDs of the model and mask should coincide. If the components are not assigned to the model, it is considered that they would be assigned directly. Mask-RCNN may fail to recognize objects and frames may not show any masks. An overlap between the remaining components and labels is evaluated again and the final segmentation contains the object IDs of the models and the relevant components. Areas to be ignored during fusion are explicitly identified to prevent their reconstruction.

3.0.3 Fusion

Each object's geometry is fused using object labels to associate the surfels to the correct model following the same strategy used in [30] and [31].

4 System Evaluation

MaskFusion is evaluated based on its abilities to solve challenging problems. We discuss both quantitative and qualitative results of the system.

4.1 Quantitative Results

A quantitative evaluation of the reconstruction and segmentation is carried out.

4.1.1 Reconstruction

The authors in [7] conducted a quantitative evaluation of the quality of the 3D reconstruction of an object from the YCB Object and Model Set [32]. A ground truth model was chosen from the dataset as in Figure 9. A dynamic sequence was acquired to evaluate errors in the 3D reconstruction. Figure 9 depicts an image of the object (Real object), the ground truth model (3D model), and the 3D reconstruction (Ours) using MaskFusion. A heatmap shows the errors in 3D reconstruction per surfel. The average 3D error for the object was 7.0mm with a standard deviation of 5.8mm. Images (e) and (f) depict the dynamic sequence used to determine the errors in 3D reconstruction in a quantitative manner.

4.1.2 Segmentation

The authors in [7] acquired a 600-frame-long sequence and provided ground truth 2D annotations for the masks of one of the objects to assess the quality of the segmentation quantitatively. Figure 10 shows the Intersection Over Union (IoU) graphs for three distinct runs which establish that combination of geometric and semantic cues results in accurate segmentation. The IoU of the per-frame segmentation masks obtained with Mask-RCNN is shown in red, and Mask-RCNN combined with the geometric segmentation is shown in blue. The blue curve (ours) shows the IoU obtained using methods used by Authors in [7], where the object masks are acquired by re-projecting the reconstructed 3D model.

Figure 9: Reconstructing an object from the YCB dataset [7]

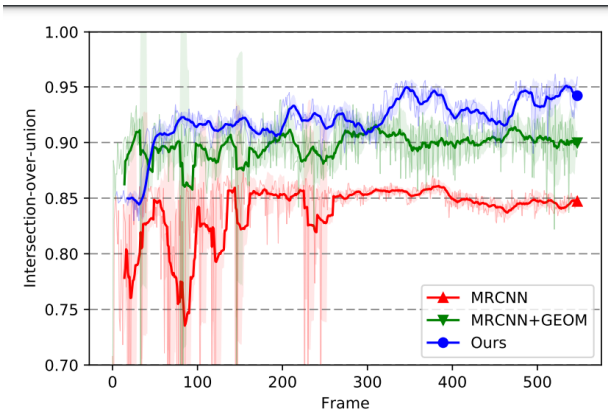


Figure 10: Comparison of labelling performance over time [7].

4.2 Qualitative Results

MaskFusion is tested on various dynamic sequences which indicate that it is an effective toolbox for various use cases.

4.2.1 Grasping

Grasping objects is a challenging task in robotics as a robot is required to identify the correct object as well as grasping points on the object. MaskFusion is suited in this use case as it detects and reconstructs the objects densely. Furthermore, it continues to track during interaction with the object. This is best illustrated in Figure 11 where a series of 6 frames are shown depicting the efficient recognition, tracking and mapping capability of MaskFusion. A vase (pink), spray-bottle (orange), a keyboard (grey) and a teddy bear (white) are detected from the starting time frame. Between frame 300 and 600 a ball (blue) appeared. Figure 11 (d) and (h) illustrates the reconstruction and estimated normals. Between frames 600 and 1000, the spray bottle (orange) was moved by a person and the person-related geometry was explicitly avoided by MaskFusion while continuously tracking the object during the interaction.

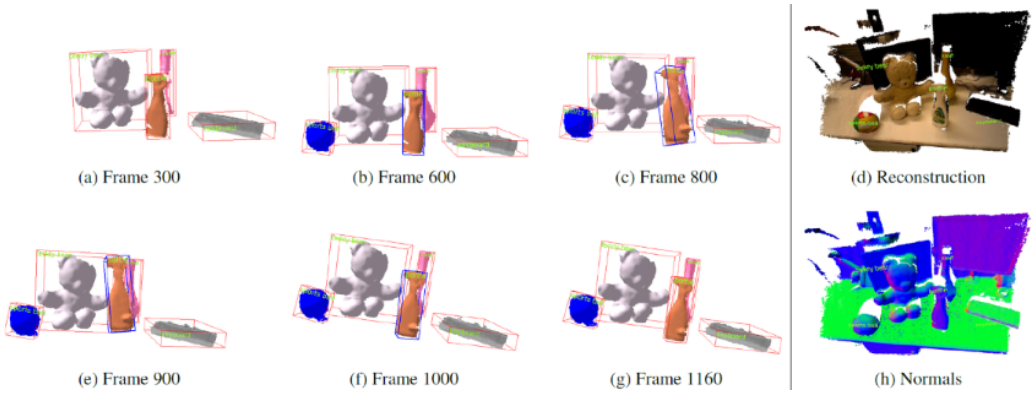


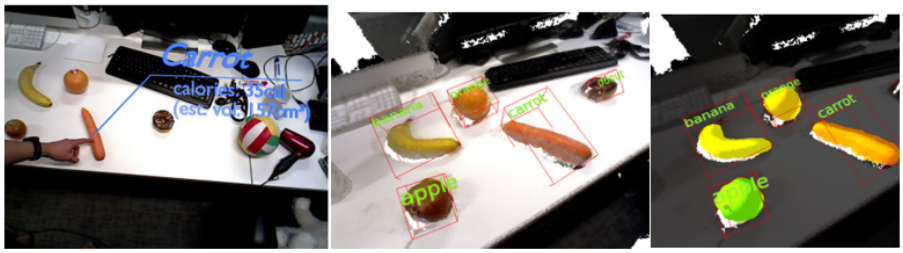
Figure 11: Recognition, Tracking and Mapping in MaskFusion [7]

4.2.2 Augmented Reality

Many AR applications use Visual SLAM as their building block. The addition of semantic information can lead to development of new kinds of AR applications. The demos below illustrate that MaskFusion can be used for augmented reality applications. These demos rely on the semantic as well as the geometric data in dynamic scenes.

Calories demo This AR prototype aims to estimate the calories of an object based on its shape and class. Body-volumes are estimated using primitive fitting and a database is provided with calories per volume for different classes of objects. The footage is augmented with desired information. Figure 12 depicts an AR prototype that depicts the calories in various grocery items.

Skateboard demo Another AR prototype presents a virtual character that reacts actively to its environment. As per the experiments carried out by the authors in [7] the character jumps on the skateboard as soon as it appears in the scene and it remains on it even after a person kicks it and sets it to motion. This is possible through accurate tracking of the skateboard and camera simultaneously as seen through Figure 13.



Estimated calories for a carrot | 3D reconstruction | Object labels in 3D

Figure 12: Calorie estimation of different groceries [7]

(a) Semantic reaction | (b) Object interaction | (c) Obeying dynamics

Figure 13: Virtual character interacting with a scene in an AR application [7]

5 Conclusion and Future Scope

Augmented reality assists in understanding and reconstructing a model of the surrounding world to facilitate precise positioning and interaction of virtual entities within the surrounding world. In this report a real-time dynamic RGB-D SLAM system is discussed that enables accurate anchoring and mapping of the real world can be used to develop novel augmented reality applications. Additional information or virtual entities could be added to or removed from dynamic scenes with many moving entities, which was challenging before the development of SLAM systems. Unique, instance-aware, semantic and dynamic AR use cases can be possible with MaskFusion.

Although MaskFusion makes progress in achieving an accurate and robust geometric and semantic SLAM system it has limitations in all the three areas it addresses - Recognition, Reconstruction and Tracking. Currently, MaskFusion can only recognize objects from classes in which Mask-RCNN has been trained. There are only 80 classes of the MS-COCO dataset [33]. It does not account for the miss-classification of the instance labels and it limited to rigid objects. When no 3D models are available, tracking small objects with less geometric information results in errors. Solving these limitations pave a way for future scope of work in this domain.

Abbreviations and Acronyms

AR *Augmented Reality*

VR *Virtual Reality*

3D Three-Dimensional

2D Two-Dimensional

SLAM Simultaneous Localization and Mapping

RGB Red Green Blue

RGB-D Red Green Blue-Depth

IMUs Intertial Measurement Units

TSDF Truncated Signed Distance Function

MRFs Markov Random Fields

Mask-RCNN *Mask-Region Based Convolutional Neural Network*

IoU Intersection Over Union

List of Figures

References

[1] Vaishali Sonik. Augmented reality apps | what is ar? what is it's impact over different industries?, Published On: September 25, 2019 Last Updated: November 26, 2019. https://www.apptunix.com/blog/augmented-reality-apps-industries-uses-ar/,Last accessed on 21-11-2022.

[2] Jule M Krüger, Kevin Palzer, and Daniel Bodemer. Learning with augmented reality: Impact of dimensionality and spatial abilities. *Computers and Education Open*, 3:100065, 2022.

[3] Taemin Lee, Changhun Jung, Kyungtaek Lee, and Sanghyun Seo. A study on recognizing multi-real world object and estimating 3d position in augmented reality. *The Journal of Supercomputing*, 78(5):7509–7528, 2022.

[4] Charalambos Theodorou, Vladan Velisavljevic, Vladimir Dyo, and Fredi Nonyelu. Visual slam algorithms and their application for ar, mapping, localization and wayfinding. *Array*, 15:100222, 2022.

[5] Yi-Chin Wu, Liwei Chan, and Wen-Chieh Lin. Tangible and visible 3d object reconstruction in augmented reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 26–36. IEEE, 2019.

[6] Chung Kuo Hao and N Michael Mayer. Real-time slam using an rgb-d camera for mobile robots. In *2013 CACS International Automatic Control Conference (CACS)*, pages 356–361. IEEE, 2013.

[7] Martin Runz, Maud Buffier, and Lourdes Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20. IEEE, 2018.

[8] Snapfeet. https://snapfeet.io/en/, Last accessed on 22-11-2022.

[9] Vitaliy Goncharuk. Pokemon go: How to make it truly augmented reality game? slam sdk!, 2016. https://augmentedpixels.com/pokemon-go-make-truly-augmented-reality-game-slam-sdk/,Last accessed on 22-11-2022.

[10] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The international journal of Robotics Research*, 31(5):647–663, 2012.

[11] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

[12] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342, 2000.

[13] Alvaro Collet, Siddhartha S Srinivasay, and Martial Hebert. Structure discovery in multi-modal data: a region-based approach. In *2011 IEEE International Conference on Robotics and Automation*, pages 5695–5702. IEEE, 2011.

[14] Ross Finman, Thomas Whelan, Michael Kaess, and John J Leonard. Efficient incremental map segmentation in dense rgb-d maps. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5488–5494. IEEE, 2014.

[15] Andrej Karpathy, Stephen Miller, and Li Fei-Fei. Object discovery in 3d scenes via shape analysis. In *2013 IEEE international conference on robotics and automation*, pages 2088–2095. IEEE, 2013.

[16] Keisuke Tateno, Federico Tombari, and Nassir Navab. Real-time and scalable incremental segmentation on dense slam. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4465–4472. IEEE, 2015.

[17] Dragomir Anguelov, B Taskarf, Vassil Chatalbashev, Daphne Koller, Dinkar Gupta, Geremy Heitz, and Andrew Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 169–176. IEEE, 2005.

[18] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[19] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[20] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6243–6252, 2017.

[21] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635. IEEE, 2017.

[22] Keisuke Tateno, Federico Tombari, and Nassir Navab. When 2.5 d is not enough: Simultaneous reconstruction, segmentation and recognition on dense slam. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 2295–2302. IEEE, 2016.

[23] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013.

[24] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (ToG)*, 33(4):1–12, 2014.

[25] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon, and Daniel Cremers. Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3849–3856. IEEE, 2018.

[26] Martin Rünz and Lourdes Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4478. IEEE, 2017.

[27] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016.

[28] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[29] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[30] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 1–8. IEEE, 2013.

[31] Thomas Whelan, Stefan Leutenegger, Renato Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems, 2015.

[32] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, 36(3):261–268, 2017.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.