A Versatile Decentralized 3D Volumetric Fusion for On-line Reconstruction

Asif Rajput
Department of Computer Science
Sukkur IBA University
Sukkur, Pakistan
asifali@iba-suk.edu.pk

Arif Hussain
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
arif.hussain@iba-suk.edu.pk

Faheem Akhtar
Department of Computer Science
Sukkur IBA University
Sukkur, Pakistan
fahim.akhtar@iba-suk.edu.pk

Zahid Hussain Khand
Department of Computer Science
Sukkur IBA University
Sukkur, Pakistan
zahid@iba-suk.edu.pk

Hina Magsi
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
hina.magsi.iba-suk.edu.pk

***Abstract*-Advancement in depth-sensing technology has allowed mobile robots to visualize the surrounding environment in 3D models. Regardless of the sensing technology (i.e. active, passive, or laser-based), a complete system that integrates recent depth data in previous 3D models in real-time is done by employing Simultaneous Localization And Mapping (SLAM) algorithms followed by a 3D reconstruction engine. Unfortunately, both the SLAM algorithm and the 3D reconstruction engine are usually executed on a single computing device, making the whole system exceptionally costly and heavy and restricting the robot's mobility. This paper proposes a decentralized, modular reconstruction system capable of employing various sensors to facilitate online 3D reconstruction from a resource-limited mobile robot.**

Keywords-3D reconstruction; visual SLAM; depth fusion

I. INTRODUCTION

The recent development of active depth-sensing technology in which a known pattern is projected onto a suspected surface to measure depth has attracted impressive research and development momentum. For instance, Google initiated the project Tango [1] where active depth perception capabilities have been integrated within a standard mobile device to facilitate depth perception, area learning, motion tracking, and Augmented Reality (AR). Similarly, plug and play devices such as Structure [2] allow consumers to develop a 3-dimensional map of surroundings to facilitate relatively accurate 3D measurements and 3D scanning. Applications of active depth sensors are more common in mobile robots due to their higher sensing rate (a.k.a. frame-rate), lightweight design, and publicly available development resources [3, 4]. Therefore, various applications exist in which semi-autonomous or autonomous mobile robots equipped with such sensors are deployed to perform mundane to specialized tasks. For instance, a mobile robot equipped with a depth sensor can scan the 3D geometry of a medium size room and store the output meshes in relatively good quality. Unfortunately, scaling the overall design to accommodate larger environments requires a complete 3D fusion-based reconstruction (such as InfiniTAM [5], FastFusion [6], and RFusion [7]) executed with higher computational and memory resource computing devices. Furthermore, their working principle limits active depth sensors to be bounded in an indoor environment, hence, scanning mechanisms should exhibit generic qualities to facilitate outdoor reconstruction. Therefore, a decentralized 3D reconstruction system in which both computing and scanning devices are connected through a well-known data sharing architecture to reconstruct large-scale environments is expected to efficiently handle such scenarios, as shown in Figure 1.

This paper proposes a novel decentralized approach to the 3D reconstruction framework, shown in Figure 2. The overall workload is divided, so that process of depth sensing and localization is performed in a scanning node, and every potential keyframe is integrated with a remote computing node. Simultaneously, both computers are connected with a well-known Robot Operating System (ROS) [8] publisher/subscriber architecture over TCP/IP. This allows flexible, scalable, and

Corresponding author: Faheem Akhtar

independent architecture, which can be adapted in potentially countless applications.

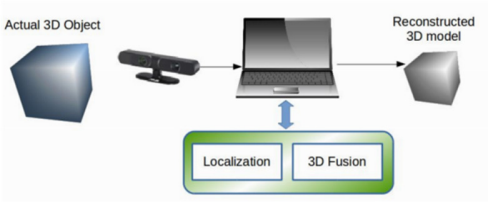


Fig. 1. A typical 3D fusion framework.

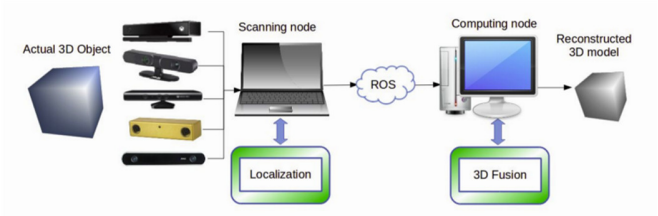


Fig. 2. A decentralized 3D fusion framework.

II. LITERATURE REVIEW

Many VisualSLAM algorithms exist that are explicitly designed to target particular sensor systems. For instance, Large Scale Dense (LSD) monocular slam [9] is a computationally lightweight algorithm that can directly be implemented in mobile devices such as cellphones and tablets. In principle, LSD-SLAM takes the live camera image stream and localizes camera position by tracking features within frames while estimating sensor ego-motion. Unfortunately, monocular SLAM algorithms are prone to accumulate estimation drift caused by lower potential features and depth ambiguity. To solve the estimation drift, algorithms such as RGBD-SLAM [10] and Stereo-LSD SLAM [11] counter the problem by introducing additional features such as the 3D geometry of the environment. In principle, 3D information acquired from a depth camera from either active depth camera or passive stereo estimation can be used to generate vital features with an improved overall tracking efficiency. Unfortunately, these algorithms are explicitly designed, which restricts the generic nature of overall framework design. In other cases, various implementations of VisualSLAM algorithms use predictive filtering (such as Kalman filters) to fuse tracking information with sensor ego-motion from the Inertial Measurement Unit (IMU). These fusion-based SLAM algorithms [12] provide further sensor localization accuracy at the expense of extended hardware and computational resources. Fortunately, ORB-SLAM2 [13] 's generic nature provides state-of-the-art tracking information without affecting the overall computational profile. It is therefore expected that employing ORB-SLAM2 in scanning systems will provide unrestricted sensor selection to accommodate both small- and large-scale 3D reconstruction.

Authors in [14] proposed the core concept of volumetric 3D integration in 1996. They proposed that range images may be represented and integrated by a voxel grid containing Signed Distance Function (SDF) from the expected surface. In principle, each incremental update of range image is represented in implicit form, and weighted addition is applied to update globally consistent 3D models. Unfortunately, the integration approach did not receive research interest due to its extensive memory and computational resources. In 2011, Microsoft released a Software Development Kit (SDK) with support for their Kinect depth sensor, which revived the 3D fusion and reconstruction concept. KinectFusion [15] was the first standalone reconstruction framework that performed localization and integration with GPU computational resources using an active depth camera. Initially, the scale of reconstruction supported by KinectFusion was bounded due to limited GPU resources. Authors in [17] extended the working of KinectFusion to accommodate large scale reconstruction. At present, various implementations of volumetric fusion [5, 6, 16, 17] utilize modern CPU and GPU architecture to facilitate virtually boundless reconstruction with the help of hashed voxel grids. Therefore, in the presented system, we have extended a regularized implementation of [7] that provides controlled regularization to facilitate the implicit representation smoothing.

In this paper, a novel 3D reconstruction framework is proposed to facilitate a decentralized, remote 3D reconstruction scenario. The proposed framework is targeted to utilize multiple computationally inexpensive computing nodes that efficiently distribute the visual slam algorithm and 3D reconstruction framework workload. Unfortunately, authors could not find an appropriate baseline framework for extensive comparison to view the presented work's novelty and the problem domain. However, our work will serve as the baseline for upcoming research in this application domain.

III. PROPOSED METHODOLOGY

A. Scanning Node

In order to maintain generality, it is presumed that a depth sensor of either active or passive depth sensing nature is attached to mobile computing devices such as laptops and streams of color and depth images are easily available. For further simplicity, it is presumed that both color and depth cameras are calibrated, and their respective intrinsic parameters such as focal lengths and central points are known. In general, all sensors can produce a steady image stream of 30 frames per second, a single registered time-stamped depth and color image that can be respectively denoted. In our implementation of the proposed system, the latest version of ORBSLAM2 is employed as a localization algorithm to track camera movements within the environment. Furthermore, the generic nature of ORBSLAM2 allows flexibility in sensor selection to accommodate both small- and large-scale 3D reconstruction. After the successful estimation of camera poses within the localization module, an instance is created containing respective depth and color image and camera pose information in standard notation, which contains rotation and translation information respectively. Therefore, each depth sample can be converted easily to a world coordinate system by using (1):

$$P_\omega = R_k \cdot \begin{bmatrix} (row - c_x) & \frac{D_k(row,col)}{f_x} \\ (col - c_y) & \frac{D_k(row,col)}{f_y} \\ D_k(row, col) & \end{bmatrix} + T_k \quad (1)$$

At this moment, I_k is broadcasted through ROS TCP/IP as a packet (referred to as topic in ROS terminology) which is received by a computing node to perform 3D integration and rendering to provide real-time 3D reconstruction.

B. Computing Node

The computing node accepts each keyframe and performs 3D fusion by employing recursive total variation based implicit regularization to reduce noise effects. The system first converts each valid depth sample to a list of the 3D point cloud (denoted compactly as in upcoming text) in the world coordinates using rotation and translation. The system then employs a least square-based regularized integration that inherently reduces noise effects and produces smoother iso-surfaces. The process of regularizing the implicit surface involves representing the obtained in the axis-aligned volumetric grid in which each cell (referred to as *voxel*) contains a projective signed distance from the surface. For each valid 3D point $p \in L_k$, the system extracts a set of voxels (referred to as SDF-signal in [7]) that lie along the ray from camera position and p. In least squares terminology, the extracted SDF-signal (denoted as y) is presumed to be the current measurement of the system which is prone to accumulate estimation noise. Assuming a linear relation between y and the unknown true state of system x with a system matrix A exists (see [18] for more details), an estimated state of the system $\hat{x}$ can be calculated by minimizing the following least squares system:

$$\hat{x} = arg\min_x\{\|Ax - y\|_2^2\} \quad (2)$$

However, in order to introduce a smoothing aspect in (2), an additional regularization term is added which acts as counter weight and influences the $\hat{x}$ to accommodate neighboring elements. Such minimization system can be written as:

$$\hat{x} = arg\min_x\{\|x - y\|_2^2 + \lambda\|g(x)\|_2^2\} \quad (3)$$

where λ is a regularization parameter that controls the influence of neighboring elements and $g(x)$ is a function which approximates the second order finite difference of x. In principle, the number of elements in x, y and $\hat{x}$ (referred to as support in [5]) directly affects the noise handling properties of the overall system. Therefore, longer streams are suitable to handle high degrees of depth noise, such as those acquired from passive depth sensors. Simultaneously, shorter support can easily accommodate relatively accurate depth information captured from an active depth camera. After solving the minimization task, the values of $\hat{x}$ are updated to particular voxel locations. The complete process of regularized integration ensures that the underlying implicit representation is smooth, and the resulting volumetric grid can be easily rendered using a standard marching cube algorithm [19].

IV. EVALUATION

Two distinct datasets were selected to evaluate the proposed decentralized system's generic nature containing a series of stereo image pairs and traditional RGB-D images from KITTI [20] and CoRBS [21] datasets respectively. The KITTI vision benchmark suite (written compactly as KITTI) contains a comprehensive set of visual and numerical data such as stereo image pairs, LiDAR point cloud, and vehicle GPS information. The versatile nature of the data present in the KITTI dataset allows researchers to quantitatively evaluate their research in the fields of stereo matching, scene flow, optical flow, depth estimation, visualSLAM, and object tracking (as explained in [20]). Similarly, the CoRBS dataset contains a set of RGB and depth images captured from the Kinect v2 camera. Simultaneously, the ground truth pose of the camera and 3D models are also provided to evaluate SLAM and 3D reconstruction algorithms' efficiency.

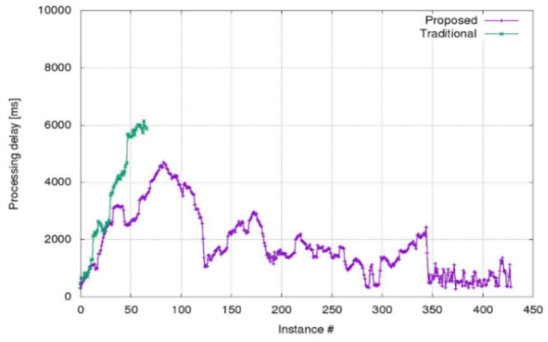
It is worth mentioning that the system's goal is to perform well in an online scenario where the localization algorithm (e.g. ORBSLAM2, etc.) estimates sensor position in real-time. The baseline for performance benchmark is calculated by employing localization and fusion modules on the same computing device (referred to as the traditional approach). Such baseline implementation highlights the strain of processing as well as throughput delays caused by severe data processing. To isolate and quantify throughput delays, a precise timing mechanism is applied. The localization module issues and attaches a timestamp to the processing instance (i.e. either RGB-D or stereo image pair). The fusion module is designed to execute in an independent processing thread, another timestamp is generated, and the difference is recorded as a processing delay. In the optimal scenario, the difference between timestamps is expected to contain a smaller value, while non-optimal systems are expected to produce higher values. Such high processing delays further degrade the system's performance as previous time differences can accumulate to hinder the reconstruction framework's real-time profile.

Figure 3 illustrates the problems caused by throughput delays in a traditional 3D fusion framework. The system's performance is decreased with every successive input keyframe, and the system crashes after processing approximately 60 and 25 keyframes due to memory overflow in both 06_KITTI and 07_KITTI trajectories, respectively. Contrarily, the proposed system continues to work with an average throughput delay of 500ms for all input instances. Similar observations have been observed during the experimentation process of the Desk1 trajectory from the CoRBS dataset (referred to as CoRBS_D1) and are shown in Figure 4. Absolute Trajectory Error (ATE) is an important measure that compares the estimated camera poses with ground truth trajectory. It calculates statistical error measures such as min, max, mean, median, standard deviation, and RMSE quantities to establish a fair comparison between the estimated trajectories between the traditional system and the proposed decentralized system. All camera poses estimated from the proposed system were truncated and then compared by a publicly available analysis tool [21].

Although no modifications were performed in the visualSLAM engine of the proposed method and the traditional system, it was observed that throughput delays indirectly affect

the estimation process of SLAM. This counter-intuitive finding is caused by the fact that traditional on-line visualSLAM modules are usually memory dependent. Once the memory buffer is filled with incoming successive keyframe images, the system either compensates with increasing the memory buffer area or drops incoming frames. We suspect that due to the dropping of keyframes, the visualSLAM module fails to register the camera movement successfully, and the final result is increased in overall absolute trajectory error. Figure 5 shows the reconstructed 3D models from 06_KITTI and 07_KITTI trajectories. It is worth mentioning that these models are the result of fusing 428 and 242 keyframes respectively. Since traditional on-line 3D fusion frameworks crashed at much smaller keyframe numbers, their partially reconstructed results are deliberately withheld.

(a)



(b)

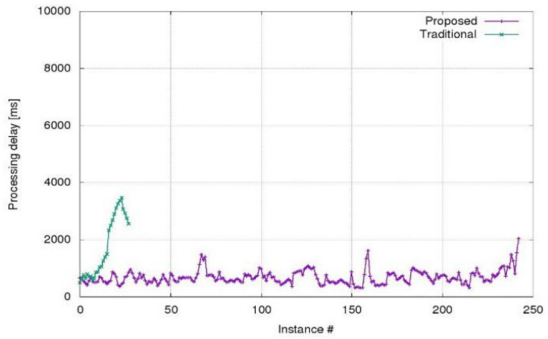


Fig. 3. Processing delays in traditional and proposed method for: (a) 06_KITTI and (b) 07_KITTI trajectory.

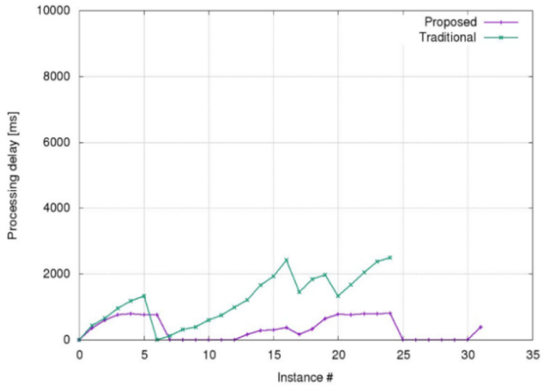


Fig. 4. Processing delays in traditional and proposed method for the Desk1 trajectory.

(a)

(b)

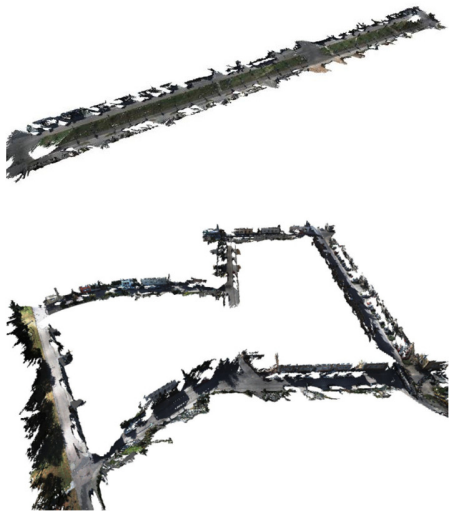


Fig. 5. Reconstructed 3D models from (a) 06_KITTI and (b) 07_KITTI trajectories.

V. CONCLUSION

This paper proposed a decentralized online 3D reconstruction framework capable of working with various available depth sensors. The experimental results show that a decentralized online 3D reconstruction framework is robust enough to handle traditional visualSLAM limitations, such as integrating fast incoming keyframes on its own. Furthermore, it was observed and shown with quantitative results that the proposed system also facilitates a modular approach, which, in result, can offer extendibility without changing the complete system. The communication between decentralized computing nodes is done with the state-of-the-art Robot Operating System, which is open source and performs with exceptional reliability. Finally, it is shown that a lightweight decentralized system outperforms a traditional 3D reconstruction system.

REFERENCES

[1] J. Kastrenakes, "Google's Project Tango is shutting down because ARCore is already here," *The Verge*, Dec. 15, 2017. https://www.theverge.com/2017/12/15/16782556/project-tango-google-shutting-down-arcore-augmented-reality (accessed Dec. 11, 2020).

[2] "Give Your iPad 3D Vision," *Structure by Occipital*. https://structure.io/ (accessed Dec. 11, 2020).

[3] M. K. Villareal and A. F. Tongco, "Multi-sensor Fusion Workflow for Accurate Classification and Mapping of Sugarcane Crops," *Engineering, Technology & Applied Science Research*, vol. 9, no. 3, pp. 4085–4091, Jun. 2019, https://doi.org/10.48084/etasr.2682.

[4] M. B. Ayed, L. Zouari, and M. Abid, "Software In the Loop Simulation for Robot Manipulators," *Engineering, Technology & Applied Science Research*, vol. 7, no. 5, pp. 2017–2021, Oct. 2017, https://doi.org/10.48084/etasr.1285.

[5] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray, "Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1241–1250, Nov. 2015, https://doi.org/10.1109/TVCG.2015.2459891.

[6] F. Steinbrücker, J. Sturm, and D. Cremers, "Volumetric 3D mapping in real-time on a CPU," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014, pp. 2021–2028, https://doi.org/10.1109/ICRA.2014.6907127.

[7] M. A. A. Rajput, E. Funk, A. Börner, and O. Hellwich, "Recursive Total Variation Filtering Based 3D Fusion," in *International Conference on Signal Processing and Multimedia Applications*, Lisbon, Portugal, Dec. 2020, vol. 5, pp. 72–80.

[8] M. Quigley *et al.*, "ROS: An Open-Source Robot Operating System," presented at the ICRA Workshop on Open Source Software, Jan. 2009, vol. 3.

[9] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *Computer Vision – ECCV 2014*, 2014, pp. 834–849, https://doi.org/10.1007/978-3-319-10605-2_54.

[10] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D Mapping With an RGB-D Camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, Feb. 2014, https://doi.org/10.1109/TRO.2013.2279412.

[11] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, Sep. 2015, pp. 1935–1942, https://doi.org/10.1109/IROS.2015.7353631.

[12] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, Mar. 2015, https://doi.org/10.1177/0278364914554813.

[13] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017, https://doi.org/10.1109/TRO.2017.2705103.

[14] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, New York, NY, USA, Aug. 1996, pp. 303–312, https://doi.org/10.1145/237170.237269.

[15] R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, Basel, Switzerland, Oct. 2011, pp. 127–136, https://doi.org/10.1109/ISMAR.2011.6092378.

[16] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially Extended KinectFusion," MIT, Technical Report MIT-CSAIL-TR-2012-020, Jul. 2012.

[17] M. A. A. Rajput, E. Funk, A. Börner, and O. Hellwich, "Boundless Reconstruction Using Regularized 3D Fusion," in *E-Business and Telecommunications*, 2017, pp. 359–378, https://doi.org/10.1007/978-3-319-67876-4_17.

[18] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: a factored solution to the simultaneous localization and mapping problem," in *Eighteenth national conference on Artificial intelligence*, USA, Jul. 2002, pp. 593–598.

[19] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 163–169, Aug. 1987, https://doi.org/10.1145/37402.37422.

[20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013, https://doi.org/10.1177/0278364913491297.

[21] O. Wasenmüller, M. Meyer, and D. Stricker, "CoRBS: Comprehensive RGB-D benchmark for SLAM using Kinect v2," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, Mar. 2016, pp. 1–7, https://doi.org/10.1109/WACV.2016.7477636.