

Smoky Maps: Radar-Camera Fusion for Navigation In Smoky Environments

Trevor Sherrard*, Justin Smith†, Zhuoyun Zhong‡

Worcester Polytechnic Institute (WPI)

Dept. of Robotics Engineering

Worcester MA, USA

Email: *twsherrard@wpi.edu, †jrsmith4@wpi.edu, ‡zzhong3@wpi.edu

Abstract—Firefighters and other first responders working in hazardous and low visibility environments are subject to countless bodily perils due to lack of visual feedback. As such, localization and navigation aids can be seen to be helpful in reducing accidents, and improving the overall operational confidence and effectiveness of firefighting teams. This work proposes a system that utilizes these sparse 2D point clouds generated from the radar returns, along with pose estimation and visual-based mapping extracted from a camera, to generate a smoke robust localization and mapping system. The static map generation step makes use of a least square homography estimation and trust index based technique to combine the map generated by the visual SLAM process and the map generated from up-sampled radar scans. It was found that the result of map fusion had a lower error index in reference to the ground truth as compared to the individual maps used in the fusion operation. This held up in the presence of three different noise levels for each sensor’s generated map.

Index Terms—Millimeter Wave Radar, First Responder Localization, Monocular SLAM, Disaster Response

I. INTRODUCTION

Fire fighting personal typically operate in very low visibility environments. This leads to the need to "feel" their way through an environment. This can lead to several risks for said personnel, including fall hazards and loss of orientation within a burning building. As such, it is desired to create a way for these first responders to "see through smoke". One such sensor within the suite of options for doing this is a frequency modulated continuous wave (FMCW) millimeter wave radar sensor. A picture of the Texas Instruments IWR1443 radar module and an associated camera used in this study can be seen in figure 1 in this section.

Like traditional radar systems, radars employing the millimeter wave radar architecture emit RF signals that reflect off of objects. These radar returns can then be compared to the emitted signals, and the difference in data in said signals can be used to determine the distance to an object, the angle of the object with respect to the radar, and the velocity of said object with respect to the radar [1]. The block diagram of said system can be seen in figure 2 in this section.

One particular characteristic of note pertaining to the millimeter wave radar is the operating wavelength of the emitted radar signals. Given that the wavelength is typically within the millimeter range, they are long enough to penetrate particulates in the air (rain, snow, fog, smoke, etc), while still being short



Fig. 1. Texas Instruments IWR1443 Radar Module And Camera

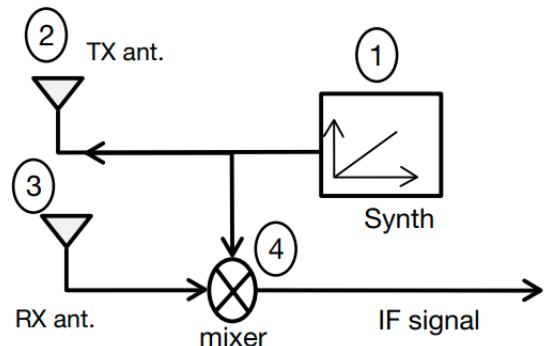


Fig. 2. Texas Instruments IWR1443 Radar Module Block Diagram [1]

enough to reflect off of solid objects like walls [1]. This characteristic is what makes this sensor so desirable for use in self-driving cars that need to operate in all weather conditions. This also allows these sensors to be used in localization and mapping applications in smoke filled environments (i.e. a burning building).

The millimeter wave radar itself produces a sparse 2D point-cloud per radar data frame, in which each point contains range, angle and Doppler velocity information. This point cloud is unfortunately too sparse to be used as a single data source for localization and mapping purposes. It can also contain erroneous data from multi-path returns and other such issues.

As such, machine learning method as generative adversarial network (GAN) can be applied to overcome the sparsity and noise issues. Furthermore, additional sensors can be used to account for said errors. In this study, that secondary sensor source is a monocular camera (as seen in figure 1). Monocular SLAM can be used to generate a 3D point cloud representing the estimated state of the environment, while localizing within said environment. The environmental understanding generated through Monocular SLAM can be combined with the results of the map constructed using millimeter wave radar scans to generate a more complete model of the environment. The next section will discuss some points of related work, and the section proceeding will outline the problem examined here in more technical detail.

II. RELATED WORK

A. Radar Mapping

Because of the price advantage and adaptability in various weather conditions, Radar is widely used sensor in robots and autonomous vehicles. Marck, et al. proposed to use Kalman filter for noisy Radar point cloud and constructed an indoor grid map [2]. Radar image could also be adapted to visual graph-based SLAM. Schuster, et al. and Hong et al. tried to extract features from obtained Radar image and performed graph SLAM method for the application of vehicle navigation [3], [4]. However, as these methods are using raw Radar data, the resulted maps were relatively sparse and noisy.

The work of Lu, et al., examines the use of generating occupancy grid maps from sparse radar point clouds through the use of a GAN [5]. Individual radar data frames were overlaid onto a global map as patches. The position and orientation of these patches within the global map was obtained from the odometry of the robot base that the millimeter wave radar sensor is attached to. In the training phases, this map was then used along with LiDAR map, which serves as ground truth, to train GAN to produce maps with high accuracy and density, even in the presence of signal loss from smoke. The generated map, along with LiDAR map and the original radar patches for different levels of smoke can be seen in figure 3.

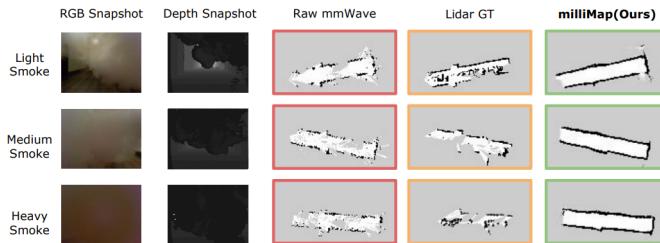


Fig. 3. Generated Occupancy Grid Maps In Different Smoke Levels [5]

The authors also implement a semantic mapping pipeline that can detect what type of object the radar returns are coming from. This method makes use of a convolutional neural network (CNN) that take sections of the radar return intensity function at given distances as input. Both the GAN

based mapping and semantic mapping pipelines can be seen graphically in figure 4.

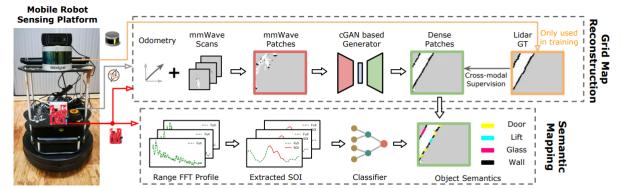


Figure 1: System overview of milliMap, comprising of (1) mobile robotic sensing (2) map reconstruction (3) semantic mapping.

Fig. 4. GAN Based Mapping and Semantic Mapping Pipelines [5]

B. Visual SLAM

The work Andrew,et al [7]. reviews an early implementation of monocular SLAM (monoSLAM). With monoSLAM, a frame is initialized by observing key features on a known object and then estimating camera motion and performs environmental reconstruction through Kalman filtering. Through out this process, a state-space is continuously updated making this implementation computationally expensive.

The work of Raúl et al. [6], introduces ORB-SLAM2, a method that eliminates the shortcomings associated with traditional monoSLAM and is able to track relatively aggressive camera trajectories while reconstructing 3D point clouds online with low computational cost. As depicted in figure 5, the higher level breakdown consists of three simultaneously-operating threads: tracking, local mapping, and loop closing. The tracking thread is dedicated to localizing the camera and generating new key frames. It does this by detecting features in successive image frames and relating these matches to a local map. Motion-only bundle adjustment is then employed to estimate the pose of the camera and outputs key frames respectively. The local mapping thread processes each new key frame. After this, camera poses are then optimized through bundle adjustment. The loop closing thread searches for duplicate data with the introduction each new key frame. If similarities are detected to be out of a defined constraint, points are fused and point drift is corrected for a more accurate global point cloud.

C. Radar-camera Fusion

Due to the differences of acquiring data from real world, Radar-camera system usually requires calibration to match features before usage [8]. Natour et al. proposed a simple method to use calibrated Radar data to remap image slices back to 3D space [9]. Most of the works on Radar-camera fusion is on target detection or object recognition [10]. Nabati and Qi proposed a Radar and camera system to improve the recognition accuracy of vehicles and pedestrian [11]. The main advantage of using Radar for detection is its ability of capturing radial velocities of the surrounding dynamic objects.

However, to the best of our knowledge, Radar-camera is never an attractive solution to map construction or SLAM, as traditional Radar data could not provide enough information to

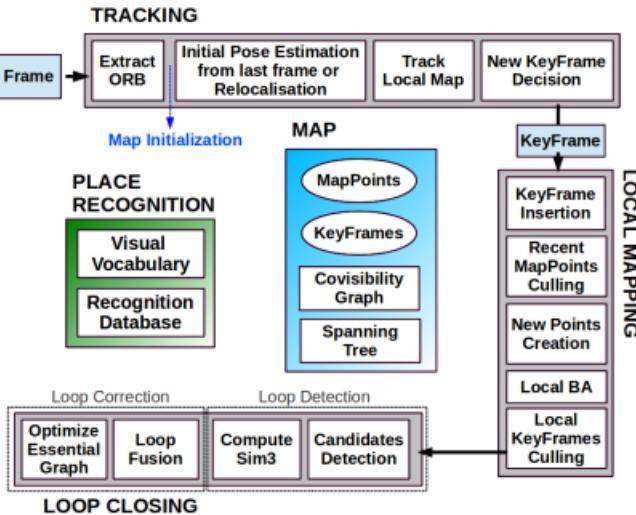


Fig. 5. ORB-SLAM System Breakdown [6]

assist visual SLAM. On the other hand, LiDAR could provide dense 3D information of the environment, which is widely used to fuse with camera in SLAM application [12].

III. PROPOSED METHOD

As previously mentioned, the goal of this endeavour is to design a system that can generate an occupancy map of a smoky environment that can be used for later navigation. This occupancy grid map will be constructed by fusing information obtained from the TI IWR1443 module and the monocular camera both seen in figure 1.

A. Generation Of Radar Occupancy Grid Map

The first step of this process is to generate a local occupancy grid map from a given radar data frame. This local occupancy grid map is generated on the decimeter scale, and is generated in the radar's coordinate frame (i.e. without reference to the global map frame). One of these local occupancy grid maps generated from a radar data frame can be seen in figure 6 in this section.

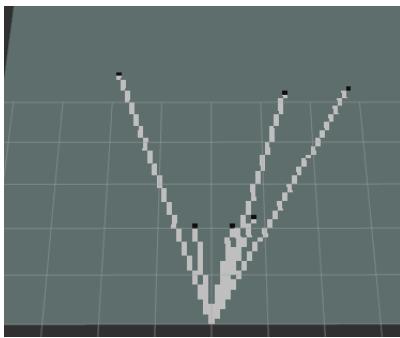


Fig. 6. Local Occupancy Grid Map From Radar Data Frame

From here, the local occupancy grid maps need to be moved into a global grid map, in which the 2D position and rotation

of the origin of the local occupancy grid map will be accounted for. These position translations and rotations will be generated by the estimated pose of the system obtained from monocular SLAM. The global map will only be updated once the pose of the system changes by a large enough threshold. This will reduce the overlap of the radar scans within the global map. Once a sparse Radar global map is generated, it will be segmented and passed through the aforementioned millimap generative adversarial network (GAN) to fill in the gaps within the map, and correct for other abnormalities. The resulting map can then be fused with the 2D occupancy grid map generated from the aforementioned monocular SLAM process.

B. Generation Of Monocular SLAM Occupancy Grid Map

As previously mentioned, a monocular SLAM algorithm will be running on images obtained from the camera seen in figure 1. The pose of the system will be obtained at a given timestep from the localization within the map generated in said process. The map produced by the monocular SLAM algorithm (in this case, ORB-SLAM2) is represented as three dimensional points in the world coordinate frame. A local gridmap generated by ORB-SLAM can be seen in figure 7.

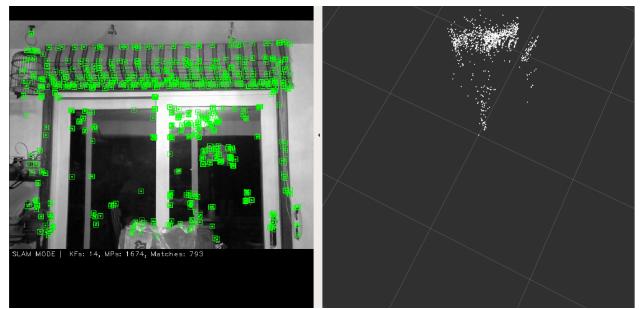


Fig. 7. Local Occupancy Grid Map From ORB SLAM

These points will need to be "flattened" into a 2D occupancy grid map that can be fused with the occupancy grid generated from the radar. There may be scaling differences between the map generated from the output of the millimap model and the map generated from the flattening of the ORB-SLAM world points. As a result, some manual change of scales might be required.

C. Fusing ORB-SLAM and Millimap Data

At this point, two separate occupancy grid maps have been generated. One from the flattened ORB-SLAM occupancy grid, and one from the up-scaled radar frame mapping process. These two maps will likely be of slightly different scales and will likely contain different features due to the different sensor modalities. As such, a map registration will need to be used that can account for differences in scale, rotation and perspective changes.

As such, a homography can be computed to move one map into the frame of another. As the map generated by the monocular SLAM method will be used as the ground truth for the placement of the radar scan patches, the homography

will be calculated with the flattened ORB SLAM grid map as the "destination" frame, and the up-scaled radar map as the "source" frame. This homography will be computed using a least square error. Once the homography is calculated, it will be applied to the up-scaled radar map, and the warped radar map and original flattened ORB-SLAM map will be superimposed using a algorithm based on trust index, which will be defined based on the smoke noise in an image.

Once this map is constructed from both the data sources, it can be saved off to hard disk storage for later use. This static map can be used for localization purposes using the camera rig seen in figure 1. This localization process will make use of a particle filter that will attempt to generate the pose of the camera rig based off the current radar data frame, and the current camera frame. This method will not account for dynamic obstacles or a dynamic environment.

IV. IMPLEMENTATION

A. Construction Of Radar Occupancy Grid Map

As previously indicated, the first step of generating a radar occupancy grid map is to generate a local radar occupancy grid map for a single radar return. This was done by making use of the Bresenham's line drawing algorithm to determine the unoccupied grid spaces between the observation point (radar) and the detected obstacles within the radar's field of view. An example of one of these local radar occupancy grid maps can be seen in figure 6. Note that for this implementation, a occupancy grid resolution of **0.1m** was chosen. This allowed the retention of high map resolution, without increasing the space complexity to a point that would be computationally prohibitive.

As the radar moved through the environment, the collocated monocular camera was being used to generate an occupancy point cloud and an estimation of the pose of the camera-radar rig within the environment through the use of ORB-SLAM2 [6]. Once the estimated camera pose changed by 0.2 meters in the *X* or *Y* direction, or rotates more than $\frac{\pi}{8}$ radians about the *z*-axis, the local map is placed within the global radar occupancy grid map. During this placement process, the local occupancy grid map is first rotated about its origin by the newly received camera pose rotation angle around the *z*-axis. This is done by rotating each individual map element using the expression seen in equation 1 in this section.

$$\begin{aligned} col_{new} &= \frac{num_cols}{2} + \cos(\theta) * (col_{old} - \frac{num_cols}{2}) \\ &\quad - \sin(\theta) * (row_{old} - \frac{num_rows}{2}) \\ row_{new} &= \frac{num_rows}{2} + \sin(\theta) * (col_{old} - \frac{num_cols}{2}) \\ &\quad + \cos(\theta) * (row_{old} - \frac{num_rows}{2}) \end{aligned} \quad (1)$$

Once the local occupancy grid map was rotated, it was simply inserted into the global radar occupancy grid map by placing the origin of the local occupancy grid map at the *X* and

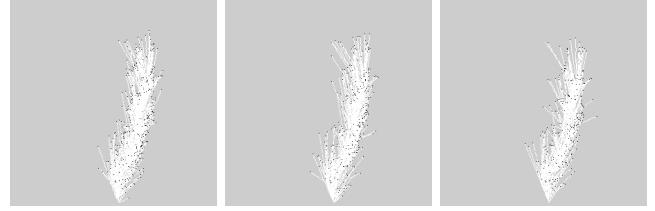


Fig. 8. Sparse Representation Of Global Radar Occupancy Grid Maps with noise ratio values of 0,0.1 and 0.2 introduced

Y locations of the global occupancy grid map corresponding to the current camera pose estimation retrieved from ORB-SLAM. After a sufficient amount of these local occupancy grid maps were inserted into the global occupancy grid map, the map began to take shape. This sparse global radar occupancy grid map can be seen in figure 8 in this section.

It can be seen in figure 8 that the map is much to sparse to use for navigation purposes in the current representation. The map will need to be "up-sampled" to be of use in such a technical endeavour. As such, the aforementioned milliMap eGAN model was utilized in the up-sampling process [5]. This model was trained to produce a map representation based on LiDAR seen in the training process, and current environmental observations taken in the form of radar scans [5]. Due to some technical issues, the map up-sampling had to happen after the completion of the sparse map construction process (i.e. offline). As a result, a ROS service was implemented to allow for an external command to be issued to save the generated sparse map representation to disk.



Fig. 9. Dense Representation of Global Radar Occupancy Grid Map After milliMap eGAN up-sampling with noise ratio values of 0,0.1,0.2 introduced

The map seen in figure 8 was passed through the millimap eGAN model [5], and the map seen in figure 9 was generated as an output. It can be seen in figure 8 that the generated map is still not perfect, and has some issues at the seams of the individual map tiles. With some additional work, this map could look much closer to what would be expected from raw LiDAR data. All that being said, it can clearly be seen that the map in figure 9 is much more usable for navigation purposes as compared to the map seen in figure 8.

B. Construction Of ORB-SLAM2 Occupancy Grid Map

As mentioned previously, the 3D point could generated by ORB-SLAM needs to be transformed into to 2D in order to be fused with the radar occupancy grid map. To do this, 3D point could data is extracted from a pointcloud2 ROS message and the *z* component is removed. From here, the point cloud data is

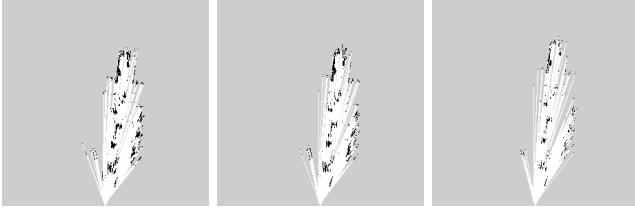


Fig. 10. Occupancy Grid Map Obtained From ORB-SLAM With Noise Ratio Values of 0, 0.2, and 0.4 introduced

scaled as to match the decimeter scale of the radar occupancy grid map. Similarly, the camera localization extracted from a ROS Pose message and converted to decimeters, and the z axis is removed. The associated data was then rounded to the nearest decimeter and supplied to Bresenham's line algorithm to determine the unoccupied space between the camera position and cloud points. Upon initial implementation, the end result of this process was less than desirable as reflective surfaces presented a significant amount of noise in the output. This made the surfaces of obstacles not well defined. Additionally, Bresenham's algorithm is required to run recursively, making it computationally expensive when point cloud data sets are too large. To alleviate this setback, an intermittent algorithm counts the number of points that occupy a grid and filters out grids that do not contain an adequate number of points to be considered occupied. For our implementation, grids that contained fewer than 3 points were rejected. This number was determined through incremental adjustments until a desirable output was achieved. The output of this filtering also eliminates duplicate entries, preventing data from bottle-necking with Bresenham's line algorithm and allows for the occupancy grid map to be constructed online.

C. Fusion of Both Maps

As mentioned before, a homography transform needs to be applied to the ORB-SLAM occupancy grid map. A least square error is defined.

$$E = \frac{1}{n} \sum (Truth[i, j] - Image[i, j])^2 \quad (2)$$

The homography is computed by brute force that minimizes the least square error. After it is found, we applied it to the SLAM occupancy map and start the fusing process.

The fusion of these two maps are based on a trust index t , which is defined based on the smoke noise existing in an image. The higher smoke noise will indicate the less usability of visual feedback, which will increase the trust of the radar occupancy map. In our implementation, trust index is hardcoded, but for real implementation, some noise detection algorithms or segmentation algorithm can be used to detect noise and can be defined empirically.

The fusing result of the two maps is given by a linear combination of the two maps for the known area.

$$Fusion_Map = t * Radar_Map + (1-t) * SLAM_Map \quad (3)$$

If an area is known in one map and unknown in the other, we decide if we could keep the known value by judging if the trust is high enough to believe based on one source.

V. SYSTEM TESTING AND EVALUATION

As mentioned, this system is designed to work in smoky environments. In lieu of generating physical smoke to evaluate the performance of the system, the effects of different smoke levels are to be emulated in software. For the radar, this testing is accomplished by removing a ratio of returns from given radar data frames. This ratio is increased based on the level of "smoke" the system is operating in. Specific radar return points are removed from the scan until the ratio of loss has been achieved. The generated local radar occupancy grid map with no noise, and with a noise ratio of 0.75 can be seen side by side in figure 11.

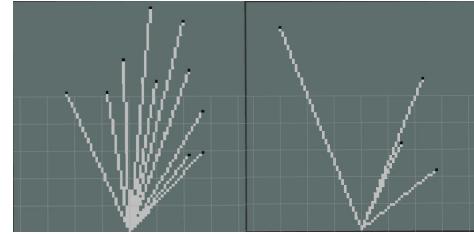


Fig. 11. Local Radar Occupancy Grid Map with noise ratio of zero (left) and 0.75 (right)

For the camera and associated monocular SLAM algorithm, so called "salt-and-pepper" noise will be applied to each image in the data stream. Salt-and-Pepper noise randomly selects a set of pixels to set to $(0, 0, 0)$ (pepper) and randomly selects a set of pixels to set to $(255, 255, 255)$ (salt). The number of pixels to be selected is determined by a "noise factor" set from 0 to 1. Where 0 is no noise, and 1 is all pixels are noise. This will allow the visual loss from smoke to be represented in the testing process. As the movement of smoke can be seen as a stochastic process, the noise application in a simulated environment should be as well. As such, this aforementioned noise application process suits in testing the final product. A single image with a noise factor of 0.2 can be seen in figure 12 in this section.



Fig. 12. Image With a Salt-and-Pepper Noise Factor of 0.2

In any case, the accuracy of the fused maps must be tested against the ground truth representation of the environment. This ground truth representation was constructed by generating a simulation environment that was to scale of the actual environment. From here, a virtual agent with a LiDAR module was allowed to roam the environment, generating a ground truth map. This map can be seen in figure 13 in this section.

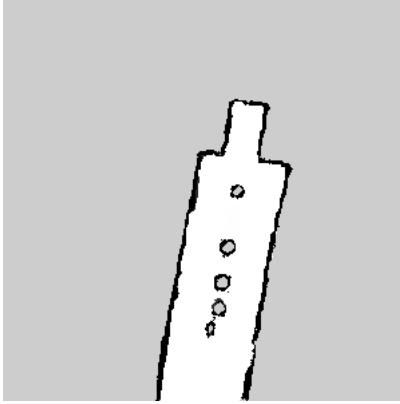


Fig. 13. Ground Truth Occupancy Grid Map

VI. RESULTS

Using the aforementioned map fusion method, each of the up-sampled global radar occupancy grid maps with different loss levels (seen in figure 9) was fused with the each of the flattened ORB-SLAM2 occupancy grid maps (seen in figure 10). The resulting fused maps can be seen in figures 14 through 16 in this section.



Fig. 14. Map Fusion Results For Radar Occupancy Grid Map (0% loss) and Flattened ORB-SLAM2 Occupancy Grid Map (0% loss)

These aforementioned fused maps could be compared with the ground truth map representation seen in figure 13. This comparison was done by evaluating equation 2 on both map image representations. The resulting error index could then be seen as an indicator of how accurate a given map was to the ground truth representation of the operating environment. A summation of all the various maps' error index values as they

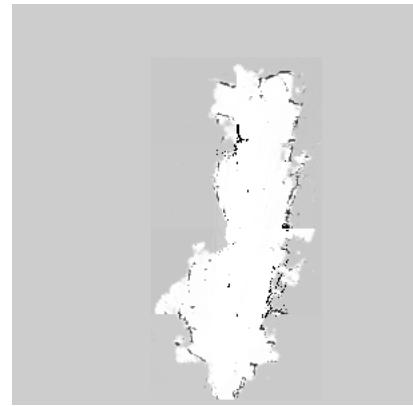


Fig. 15. Map Fusion Results For Radar Occupancy Grid Map (10% loss) and Flattened ORB-SLAM2 Occupancy Grid Map (20% loss)

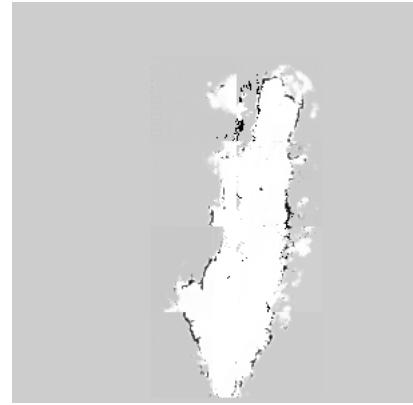


Fig. 16. Map Fusion Results For Radar Occupancy Grid Map (20% loss) and Flattened ORB-SLAM2 Occupancy Grid Map (40% loss)

compare to the ground truth map representation can be seen in table I in this section.

It can be seen in table I that as the loss factor of both the global radar occupancy grid map increases, the overall error index also increases. That pattern also follows for the flattened ORB-SLAM2 occupancy grid map. This makes sense as additional information loss would lead to a less accurate representation of the environment. The important thing to see is that the results of each of the fusion operations result in a **lower** error index as compared to the error index of each of the maps being fused. As such, it can be seen that the fusion of each of the maps resulted in a better representation of the environment than each of the independent maps.

VII. DISCUSSION AND CONCLUSIONS

As previously mentioned, the goal of this work was to create a software stack that would allow for the generation of a map based on data obtained from a Texas Instruments IWR1443 radar module, and from a monocular camera seen in the constructed rig in figure 1. Ideally, this map could be constructed in environments that had heavy smoke cover, such that the system could serve as a navigational aid for firefighters operating within an active structure fire.

Map Name	Error Index	Map Index
Sparse Radar (0% loss)	0.476	1
Sparse Radar (10% loss)	0.497	2
Sparse Radar (20% loss)	0.543	3
Up-sampled Radar (0% loss)	0.188	4
Up-sampled Radar (10% loss)	0.189	5
Up-sampled Radar (20% loss)	0.207	6
V-SLAM (0% loss)	0.339	7
V-SLAM (20% loss)	0.364	8
V-SLAM (40% loss)	0.419	9
Fusion Applied Between Maps 4 & 7	0.105	10
Fusion Applied Between Maps 5 & 8	0.151	11
Fusion Applied Between Maps 6 & 9	0.171	12

TABLE I
ERROR INDICES CALCULATED BETWEEN GROUND TRUTH AND VARIOUS MAP REPRESENTATIONS

It was found that using the individual radar data frames, local occupancy grid maps could be constructed for each data frame. From here, the pose estimation of the camera-radar rig was used to generate a sparse global radar occupancy grid map by placing local radar occupancy grid maps within said larger grid map. This resultant map can be seen in figure 8. From here, the map was successfully up-sampled using the milliMap eGAN model, resulting in the dense radar map seen in figure 9. Independently of this process, the pointcloud2 message generated by the ORB-SLAM2 process was flattened and scaled to generate the occupancy grid map seen in figure 10. From here, both of these maps were fused by making use of a homography calculated between the two map representations. During the fusion of the now aligned maps, a trust index was taken into account to give either more or less preference to the map obtained through monocular SLAM as compared to the map obtained from radar data. This preferential weighting can be seen explicitly defined in equation 3.

At this point, three global radar occupancy grid maps were generated in the presence of different noise levels (0% loss, 10% loss and 20% loss). Three flattened ORB-SLAM2 maps were also generated in the presence of different noise levels (0% loss, 20% loss and 40% loss). The error index between each of these six maps and the ground truth map seen in figure 13 were computed using equation 2. The aforementioned fusion process was then applied between each of these maps at the different noise levels, resulting in three fused maps seen in figures 14 through 16. It was found that each of these fusion results had a lower error index as compared to either of the maps fused to get said result. Thus, it can be seen that the fusion process did indeed result in a more accurate representation of the environment as compared to a single sensor, even in the presence of heavy noise levels.

The radar module was able to successfully capture and generate occupancy grid values when adding noise independently from the camera. In contrast, ORB-SLAM suffers heavily upon introducing low-level noise ratios, only partially functioning at a noise level of about 0.0015. Visual data that ORB-SLAM requires to detect corners is severely obstructed for both near and far objects. This is particularly problematic as radar-camera rig requires ORB-SLAM for odometry data limiting

the system to the ability of the camera. As such, the system cannot be used as originally intended. Future work will be required to select and evaluate odometry methods that are robust to sensor noise induced by smoke and other particulates.

VIII. IMPROVEMENTS AND FUTURE WORK

This project has several points of possible improvement. These range from large scale technical and architecture changes, to smaller details relating to the implementation details about this specific work. Thus, the discussion of these improvements will be split up as such. Please see below for improvements to the projects structure/architecture.

- Currently, the pose of the camera rig is determined from ORB-SLAM2 running exclusively off of images obtained from the monocular camera in the camera-radar rig. As such, this camera can be seen as a linchpin to the whole system. If the pose generated from ORB-SLAM is not available or is incorrect, then the placements of the local radar occupancy grid maps in the global radar occupancy grid map will fail. As such, the fusion of the global radar occupancy grid map and flatten ORB-SLAM2 map will also fail. Thus, the localization of the camera rig should take into account data from both the camera and the radar during the SLAM process.
- Currently, the map generation process is split into an on-line and off-line construction phase. The online construction phase updates the flattened ORB-SLAM2 map and sparse global radar occupancy grid map as new data is received. Once the whole environment is traversed, the maps are saved to disk. From here, the sparse global radar occupancy grid map is up-sampled using the eGAN model. The two models can then be fused into one map. This map can then be re-loaded and used for localization using data from the camera and radar scans. It would be desirable to amend/extend this work to perform all of these aforementioned actions "on-line" such that this system could be used for localization and mapping within completely unknown environments.

Please see below for technical improvements that could be made to the project in its current form:

- Currently, the flattened ORB-SLAM2 occupancy grid map must be scaled manually to roughly align with the size of the global radar occupancy grid map. If this process could be performed automatically, that would be a desirable point of improvement to make the system even more usable in a full-fledged product.
- As previously mentioned, the milliMap eGAN model relies on a torch library version that requires python 3, whereas all ROS nodes generated in this project make use of python 2.7. As a result, this project could be re-implemented using ROS2 to allow for these nodes to be re-written in python 3. This would allow for the milliMap up-sampling to be performed "on-line".

In addition to improvements that could be made to the work itself, there were several points of future work generated in the

process of completing this project. Please see a list of a few of these points below:

- As previously mentioned, the final generated map created from fusing the flattened ORB-SLAM2 map and dense radar occupancy grid map is saved to disk. This map can be used for localization purposes. Future works could attempt to implement said localization using data streams that are augmented with noise to simulate smoke. Thus, allowing an evaluation to take place of the potential performance of the system in a smoky environment.
- Another potential point of future work is the examination of other sensor sources to add to the sensor fusion pipeline. For example, the addition of thermal cameras in place of standard RGB cameras may provide additional information to help perform SLAM within a building full of smoke. Additionally, the use of ultrasonic sensors in the platform may be useful, as the sound they emit for distance measurement purposes may be resistant to particulate in the air from smoke. Potentially adding an Inertial Measurement Unit (IMU) might also serve as a useful data source to extract odometry estimation of the rig itself. This will be especially important once the system will need to consider movement in the vertical direction (e.g. going up stairs)

REFERENCES

- [1] C. Lovescu and S. Rao, "The fundamentals of millimeter wave radar sensors," Jul. 2020.
- [2] J. W. Marck, A. Mohamoud, E. vd Houwen and R. van Heijster, "Indoor radar SLAM A radar application for vision and GPS denied environments," 2013 European Microwave Conference, 2013, pp. 1783-1786, doi: 10.23919/EuMC.2013.6687024.
- [3] F. Schuster, C. G. Keller, M. Rapp, M. Haueis and C. Curio, "Landmark based radar SLAM using graph optimization," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 2559-2564, doi: 10.1109/ITSC.2016.7795967.
- [4] Z. Hong, Y. Petillot and S. Wang, "RadarSLAM: Radar based Large-Scale SLAM in All Weathers," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 5164-5170, doi: 10.1109/IROS45743.2020.9341287.
- [5] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni, and A. Markham, "See through smoke," Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services, 2020.
- [6] Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós, ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015.
- [7] A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052-1067, June 2007, doi: 10.1109/TPAMI.2007.1049.
- [8] J. Oh, K. Kim, M. Park and S. Kim, "A Comparative Study on Camera-Radar Calibration Methods," 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2018, pp. 1057-1062, doi: 10.1109/ICARCV.2018.8581329.
- [9] G. El Natour, O. Ait Aider, R. Rouveure, F. Berry and P. Faure, "Radar and vision sensors calibration for outdoor 3D reconstruction," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 2084-2089, doi: 10.1109/ICRA.2015.7139473.
- [10] Z. Wang, Y. Wu and Q. Niu, "Multi-Sensor Fusion in Automated Driving: A Survey," in *IEEE Access*, vol. 8, pp. 2847-2868, 2020, doi: 10.1109/ACCESS.2019.2962554.
- [11] R. Nabati and H. Qi, "CenterFusion: Center-based Radar and Camera Fusion for 3D Object Detection," 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), 2021, pp. 1526-1535, doi: 10.1109/WACV48630.2021.00157.
- [12] C. Debeunne and D. Vivet, "A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping," *Sensors*, vol. 20, no. 7, p. 2068, Apr. 2020.