Capturing Uncertainty in Monocular Depth Estimation: Towards Fuzzy Voxel Maps

Andrew R. Buck*, Derek T. Anderson*, Raub Camaioni†, Jack Akers*,
Robert H. Luke III†, and James M. Keller*
*Electrical Engineering and Computer Science (EECS) Department, University of Missouri, Columbia, MO, USA
†US Army DEVCOM C5ISR Center, Fort Belvoir, VA, USA
Email: {buckar, andersondt, kellerj, jdapm8}@missouri.edu, {robert.h.luke2.civ, raub.j.camaioni.civ}@army.mil

***Abstract*—Monocular depth estimation methods using structure from motion (SfM) have become increasingly capable of reconstructing 3D representations from a sequence of 2D images. In the context of unmanned aerial vehicles (UAVs), these techniques can be used to create an occupancy map of an environment, which is useful for planning and navigation. OctoMap and a recent improvement, UFOMap, are commonly used hierarchical representations that represent the value of a voxel cell as the probability of being occupied. Although this captures some uncertainty in the map and allows for dynamic updates, it does not fully utilize the known characteristics of the sensor and SfM algorithm, and it can lead to unnecessarily noisy results. In this paper, we propose an approach to assign a weight to each point in a point cloud update based on camera extrinsics and SfM confidence. The weighted points are then added to the voxel map in a way that more closely resembles a degree of confidence rather than a probability. In this way, we take the first steps toward designing a fuzzy voxel map that is more robust in noisy situations and captures useful uncertainty to help with UAV applications. We demonstrate our approach on simulated scenarios using Unreal Engine and AirSim.**

***Index Terms*—fuzzy voxel map, structure from motion, monocular depth estimation**

I. Introduction

The representation of a 3D environment is an important problem for many applications. In mobile robotics, a common representation is a probabilistic occupancy grid. This approach has been adopted and standardized by a large portion of the community, with easy to stand up approaches such as OctoMap [1] and UFOMap [2] capturing a significant amount of interest. These solutions adopt a Bayesian view of the world, assuming that observations come from range sensors in the form of point clouds, and that updates to the map are done in a probabilistic respect. Although these methods have proven to be very successful, they do not capture all types of uncertainty that may be present.

One type of uncertainty that is not typically modeled with existing methods is the imprecision of range (depth) that grows at farther distances. This is particularly important for sensors that have to estimate depth without measuring it explicitly, such as when using structure from motion (SfM) techniques from a single camera. These algorithms can often produce a self-assessment of confidence when generating new predictions. This confidence is rarely utilized, however, which can lead to errors in the reconstructed map.

In this article, we demonstrate a technique to incorporate sensor confidence into existing voxel occupancy map software packages such as UFOMap. We use a monocular depth estimation algorithm called EpiDepth [3], which is optimized for the conditions found when operating a small, single-camera UAV. The expected quality of the depth predictions is dependent on the relative camera poses, and the sensitivity of the method can be determined by modifying the algorithm parameters. This results in a per-pixel confidence value that can be used to weight the observations as they are added to the map.

The basic approach of assigning weights to the point cloud measurements is only a partial solution to what we envision as a complete fuzzy voxel map implementation. By using UFOMap as the mapping framework, we are restricted in how new points are added to the map. Each new point updates a single voxel as occupied, and free space is determined by extending a ray from the camera focal point. Ideally, a fuzzy voxel map could update multiple voxels with each point update (i.e. using a frustum rather than a ray), expanding the spatial uncertainty and decreasing the confidence for points farther from the camera. The meaning of the value stored for each pixel might also have different interpretations, such as set membership rather than probability of occupancy. For the current work, we only modify the point weights and do not model spatial uncertainty, saving this extension for future work.

The remainder of the article is as follows: Sections II and III describe the related work, including EpiDepth and UFOMap. Section IV describes our method for computing confidence and inserting points into the map. We demonstrate the approach with selected experiments in Section V and give our conclusions in Section VI.

II. Closely Related Work

The current article has related work in the following areas. While we focus on a 3D mapping algorithm called EpiDepth [3], related state-of-the-art methods include multi-view stereo (MVS) and structure from motion (SfM) [4], simultaneous localization and mapping (SLAM) [5]–[7], etc.), deep learning (DL) for single image depth estimation (SIDE) [8]–[10]), and DL-based optical flow [11]. While we present specific ways to extract measures of uncertainty from EpiDepth, this philosophy can be extended and tailored to other algorithms.

Next, a world model can be organized in many ways. For example, most mapping algorithms yield 3D point clouds in a local camera coordinate system. Methods like SfM and SLAM combine observations across looks (images or image pairs) to produce a relative, or when external position information is present, global 3D point cloud. Frequently, such data is quantized into a more efficient discrete data structure like an octree. In areas like robotics, many simplify the world further into an occupancy grid map (OGM). In an OGM, each discrete unit typically has a number (e.g., probability) that is ultimately used to determine its state, e.g., unknown, free, or occupied (UFO). In recent years, more advanced UFO probabilistic structures such as OctoMap [1] and UFOMap [2] have been proposed for ground robotics and drones. In [12], O'Meadhra et al. introduced a compressed, generative, and more efficient variable resolution structure based on probabilistic occupancy mapping via Gaussian mixture models for autonomous cave surveying [13] and reactive collision avoidance [14].

In [15], Oriolo et al. introduced the most directly related work to our current article. Specifically, they proposed a concept called a fuzzy map that is simply an OGM derived from an ultrasonic range sensor. At each time step, they calculate a membership degree with respect to empty and occupied based on range, angular position, and errors originating from multi-reflections. Next, a simple aggregation operation based on t-norms and t-conorms is presented, which is used to update the map. Last, a simple scheme based on A* is put forth for path planning on this fuzzy map. In our final conclusions, we highlight and compare key differences between our article an Oriolo-style fuzzy map.

III. Directly Related Work

A. EpiDepth

Many small UAVs are equipped with a monocular image sensor and a GPS/IMU/magnetometer module that provide a color image and camera extrinsics. This is sufficient for SfM techniques to estimate 3D depth. The EpiDepth algorithm [3] takes a stream of world-aligned image frames as input and produces (for acceptable image pairs) a dense, per-pixel depth estimation. EpiDepth can run in real-time on embedded hardware with parameters to adjust the scale and quality.

Frames are selected sequentially based solely on their relative camera extrinsics. In order to be considered, a pair of frames must have a temporal and spatial displacement within a defined range, and also be oriented in the same general direction. From a valid frame pair, the color images are warped and aligned so that optimized block matching techniques can be applied to produce disparity and depth images. By perturbing the pixel disparity estimations of the algorithm, multiple depth maps are produced. An estimate of error sensitivity and uncertainty can be computed from the difference in resulting depth maps. The camera-space pixels are then projected into a 3D point cloud and mapped into world coordinates from the known camera extrinsics. In this study, we do not perform any refinement of the camera position (i.e. using SLAM methods), and rely on the accuracy of the UAV position and orientation sensors to place the points into a common world-space coordinate frame.

B. UFOMap

A voxel grid is a commonly used way to represent occupied and free space for robotics applications. However, a naive implementation is too inefficient to use in practice. OctoMap [1] is a hierarchical solution based on octrees that stores the probability of occupancy for each voxel. UFOMap [2] is a more recent improvement of OctoMap that explicitly stores unknown space and can be extended to export dense maps with a uniform voxel size for additional analysis.

In both methods, point cloud observations are added to the map incrementally, using the camera location to determine free space. For each pixel in an image, a ray is cast from the camera's focal point through the pixel and ending at a distance corresponding to the estimated depth of that pixel. The voxel corresponding to the endpoint of the ray is updated as an occupied voxel and all other voxels intersected along the ray are updated as free space.

Voxel occupancy probabilities are stored in a log-odds format, such that an "occupied" observation is equivalent to adding some fixed amount, l_{occ}, to the voxel value, and a "free" observation is equivalent to subtracting a fixed amount, l_{free}. This provides an efficient representation that allows for fast updates, and the values can be clamped to prevent saturation and maintain responsiveness to dynamic environments.

Notably, these methods use a Bayesian approach for modeling sensor uncertainty in the form of the update values l_{occ} and l_{free}. These correspond to probabilities p_{occ} and p_{free} such that

$$l = \log\left(\frac{p}{1-p}\right) \tag{1}$$

and

$$p = \frac{1}{1+\exp(-l)}. \tag{2}$$

However, spatial uncertainty is not explicitly modeled. Along a given ray, only a single voxel is marked as occupied, and only voxels intersecting the ray are marked as free. In reality, as the distance from the sensor grows, so does the uncertainty. A model that accounts for this might use a frustum rather than a ray to determine which voxels to update.

IV. Method

A. Extrinsic Quality Metric

One of the greatest factors in determining the expected quality of an EpiDepth prediction is the relative poses of the two cameras. In order to achieve a good reconstruction, the two cameras must be looking in generally the same direction with a baseline perpendicular to the viewing direction. Consider the examples in Fig. 1, which show several different camera configurations (limited to a 2D plane for visualization). In each, the vectors A and B represent the look directions of the two cameras, which are separated by a baseline D. Our extrinsic quality metric seeks to capture the intuitive quality of these examples.

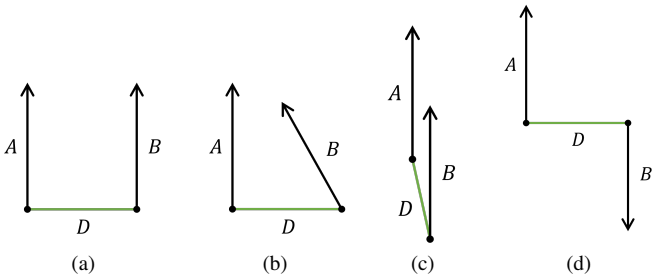


Fig. 1. Extrinsic quality examples. (a) Excellent: Frames are separated perpendicular to the look direction and aligned. (b) Good: Frames are separated and mostly aligned. (c) Poor: Frames are aligned, but separated in the look direction. (d) Bad: Frames are not aligned.

We use the following two heuristics when creating an extrinsic quality metric.

- The angle $\angle AB$ should be *small*.
- Angles $\angle AD$ and $\angle BD$ should be *close* to 90°.

These are not exhaustive, and additional factors could be included for specific use cases (i.e. favor nadir or slanted look angles instead of those level with the horizon), but these two heuristics serve to demonstrate the effectiveness of utilizing a metric based on camera extrinsics.

To capture the first heuristic, we use the cosine similarity between A and B, clipped to only positive values,

$$S_{AB} = \cos(\angle AB) = \frac{A \cdot B}{\|A\|\|B\|}, \tag{3}$$

$$H_{AB} = \max(S_{AB}, 0). \tag{4}$$

To capture the second heuristic, we again use cosine similarity,

$$S_{AD} = \cos(\angle AD) = \frac{A \cdot D}{\|A\|\|D\|}, \tag{5}$$

$$S_{BD} = \cos(\angle BD) = \frac{B \cdot D}{\|B\|\|D\|}. \tag{6}$$

However, we would like the measure to be less sensitive to small changes when the look vector and baseline are close to perpendicular (e.g. Fig. 1b). Because the cosine function has maximum slope at 90°, we flip and shift the values so that the resulting measure is close to 1 when the vectors are perpendicular and stays relatively high until the vectors are nearly parallel.

$$R_{AD} = \sqrt{1 - S_{AD}^2} \tag{7}$$

$$R_{BD} = \sqrt{1 - S_{BD}^2} \tag{8}$$

The combined metric uses the minimum of these three values to give the overall extrinsic quality.

$$Q_{ABD} = \min(H_{AB}, R_{AD}, R_{BD}) \tag{9}$$

This metric will be used in conjunction with the per-pixel confidence values to assign weights to the updates in UFOMap.

Fig. 2 shows several examples of our extrinsic quality metric on simulated image frames. The first two rows demonstrate ideal conditions: moving forward while looking nadir and performing a strafe maneuver while looking slightly downward. The next two rows show common scenarios at a slant angle, moving up or forward, with a slightly reduced score. The bottom two rows are poor conditions, moving directly into or out of the camera frame, while looking forward or down.

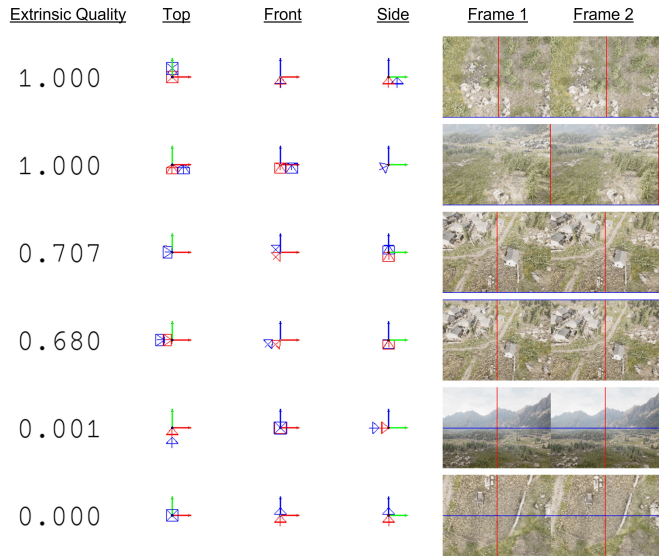


Fig. 2. Extrinsic quality examples. The left column gives the extrinsic quality metric as computed by our method. The next three columns give a top, front and side view of the two camera poses in 3D. The last two columns show the corresponding image frames. A red and blue line is plotted on each image to indicate the location of the epipole.

B. EpiDepth Prediction Confidence

For each selected frame pair, EpiDepth generates a predicted depth image, P_0. Adjusting the pixel disparity estimations of the algorithm gives two additional depth predictions, P_- and P_+. The three predictions indicate the uncertainty in depth at different locations in the image. Due to the way these images are generated, not every pixel will be assigned a depth value in all three images. When a pixel p does have a value in one or more depth images, we assume that $P_-(p) \leq P_0(p) \leq P_+(p)$. Fig. 3 shows an example image frame and the corresponding ground truth depth alongside the three EpiDepth predictions. Pixels with no predicted depth are shown in gray.

In the standard EpiDepth approach, only the P_0 image is used to project 3D points, with no indication of the prediction confidence. A comparison of this image with the ground truth depth D_{GT} can be seen in Fig. 4a. This image shows the difference as a blue-white-red color map, with accurate predictions shown in white and under/over predictions shown in red/blue respectively using a log scale. An image that is mostly white indicates a qualitatively accurate depth prediction by EpiDepth.

The additional prediction images P_- and P_+ show the sensitivity of the EpiDepth algorithm to different parameter settings. The absolute difference between these images is a measure of confidence in the predicted depth value. Smaller

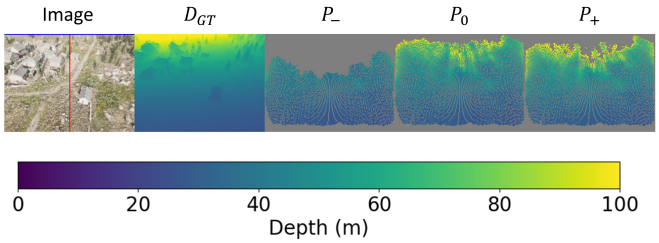
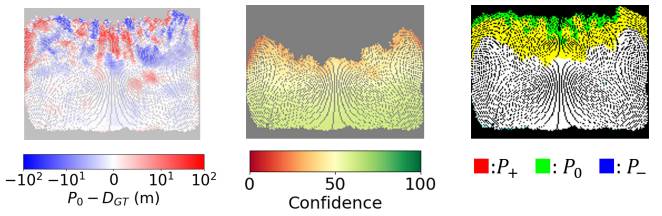


Fig. 3. EpiDepth predictions for an example frame pair. Three predictions are made: P_-, P_0, and P_+. These give a measure of sensitivity and can be compared to the ground truth depth, D_{GT}.



(a) (b) (c)

Fig. 4. EpiDepth analysis images. (a) Difference between predicted depth, P_0, and the ground truth depth, D_{GT}. (b) Calculated confidence image, C. (c) Overlap image showing each EpiDepth prediction P_+, P_0, and P_- on a different color channel.

differences indicate greater confidence and larger differences indicate lower confidence. We design a per-pixel confidence measure based on this difference and scale to the (arbitrary) range $[0, 100]$. The scores for all pixels in the image are then multiplied by the extrinsic quality metric, Q_{ABD}, to generate the confidence image,

$$C(p) = \max\Big(100 - a \cdot \big(P_+(p) - P_-(p)\big), 0\Big) \cdot Q_{ABD}. \quad (10)$$

We use a value of $a = 5$ in our experiments. An example of the confidence image is shown in Fig. 4b. Only pixels with a value in all three depth prediction images are assigned a confidence, since P_- and P_+ are needed for the calculation, and P_0 is the actual depth that will be projected (for consistency with the standard approach). The overlap between these prediction images can be seen in Fig. 4c, where each one has been assigned a different color channel.

C. Voxel Map Updates

In the standard EpiDepth approach, each selected frame pair produces a single depth image that is inserted once into UFOMap using the known camera pose. For each pixel p in P_0, a ray is projected from the camera's focal point through p for a distance given by $P_0(p)$. The value of the voxel at the endpoint of the ray is incremented by l_{occ} and all voxels that the the ray intersects are decremented by l_{free}. Because these values are stored in log-odds notation, this updates the probability that each of these voxels represents occupied space.

The default version of UFOMap assumes that l_{occ} and l_{free} are constants, which are applied to all inserted points equally. An updated version of UFOMap could allow these to be different for each inserted point, but we instead use a workaround for the current study. By partitioning the points in an update based on confidence, we can insert the same points multiple times, which has the same effect as scaling l_{occ} and l_{free}, albeit with some inefficiencies. We refer to this as version of the algorithm as UFOMap$^\alpha$ because the confidence thresholds can be understood as α-cuts.

To achieve the desired result of updating the map with individual α-cuts, we define the number of unique confidence thresholds as N_t. An insertion of the same point N_t times into UFOMap$^\alpha$ should be equivalent to a single insertion into the standard UFOMap. Therefore, $l_{occ} = N_t \cdot l^\alpha_{occ}$ and $l_{free} = N_t \cdot l^\alpha_{free}$, where l^α_{occ} and l^α_{free} are the constant update values used for UFOMap$^\alpha$.

Since the updates are performed in the log-odds space, the corresponding probabilities (needed by UFOMap) can be calculated as

$$p = \frac{1}{1 + \exp\big(-\frac{1}{N_t}\log(\frac{l}{1-l})\big)}. \quad (11)$$

For our experiments, we use $N_t = 10$. The default update probabilities of $p_{occ} = 0.7$ and $p_{free} = 0.4$, which correspond to log-odds values of $l_{occ} = 0.847$ and $l_{free} = -0.405$, are then adjusted for the fuzzy approach to be $p^\alpha_{occ} = 0.521$ and $p^\alpha_{free} = 0.490$.

V. Experiments

Our vision for a complete fuzzy voxel map solution will require significant modifications to the tools that are currently available. In the meantime, we demonstrate the viability of our ideas on some example scenarios using the aforementioned fuzzy methods described above. These experiments are performed on an outdoor scene modeled in Unreal Engine using the AirSim plugin [16]. We use lightly modified versions of EpiDepth [3] and UFOMap [2] to support our analysis. The components are connected via ROS and can support both real-time operation and offline playback.

A. Wall Reconstruction

To demonstrate the advantage of using a fuzzy voxel map that is confidence-aware, we examine a wall reconstruction scenario in detail. For this experiment, a large wall was placed in the environment to provide a fixed reference object at a known distance. We first perform a strafe maneuver, moving from right to left along the wall, which gives a very good initial reconstruction. We then move backwards while looking at the wall with a pitch angle that is nearly level with the horizon. This results in a poor depth estimation by EpiDepth, as would be expected from the extrinsic configuration.

Some analysis images from these two movements are shown in Fig. 5. The first two rows are from the initial strafe maneuver and we observe that the predicted depth P_0 closely matches the ground truth depth D_{GT}, particularly on the wall where the difference image shows mostly white. The bottom two rows are from when the camera is looking at the wall

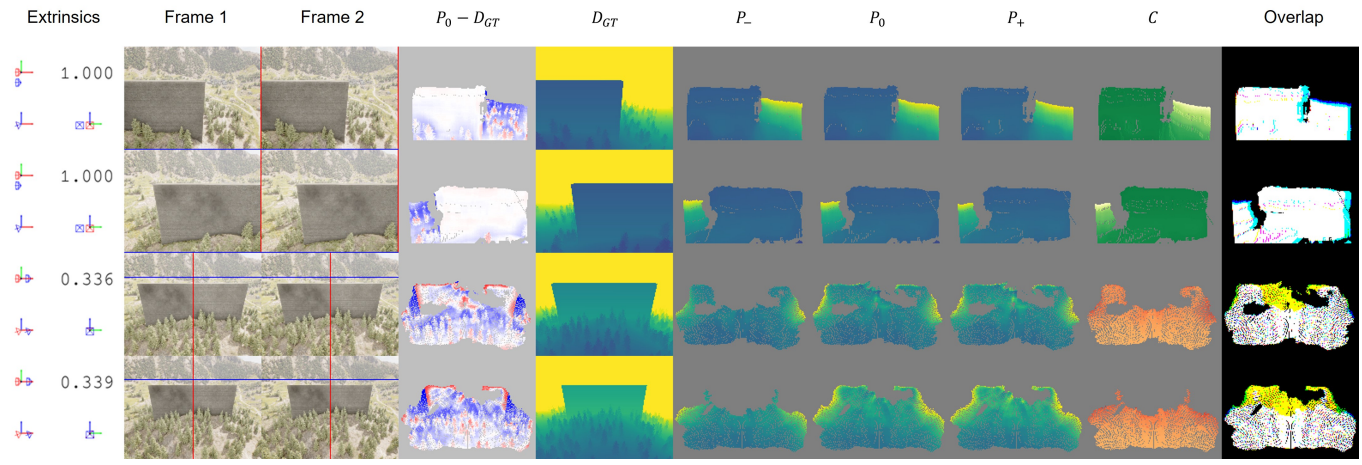


Fig. 5. Fuzzy wall scenario evaluation images. The first two rows are from the initial strafe maneuver, which results in a very good initial reconstruction. The bottom two rows are from the receding away movement, which results in a poor depth estimation due to the camera extrinsics and increasing distance from the wall. Each row shows a pair of image frames and the resulting depth prediction from EpiDepth. From left to right, the images show the camera extrinsics and quality metric, the two image frames, the difference between the predicted depth and the ground truth, the ground truth, the three EpiDepth predictions, the confidence image, and the overlap between the predictions.

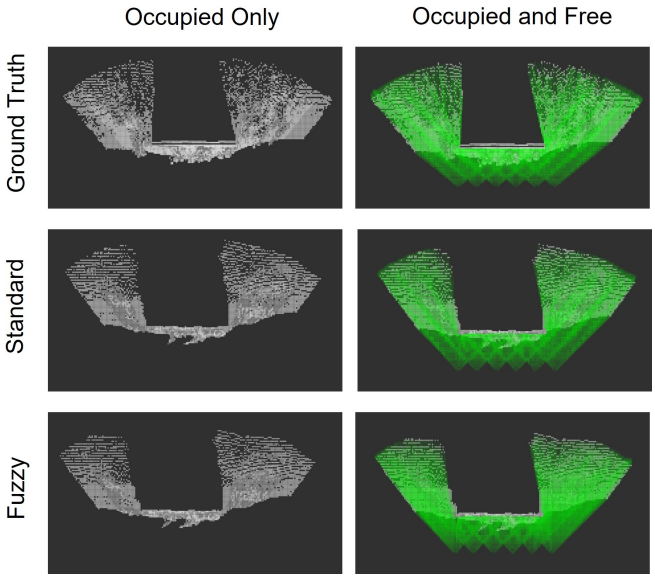


Fig. 6. Top-down views of the voxel maps after six initial high-quality observations moving sideways along a wall. Both the standard and fuzzy versions closely match the ground truth with very good initial reconstructions. The left column shows only the voxels marked as occupied, and the right column also shows the voxels marked as free space in green.

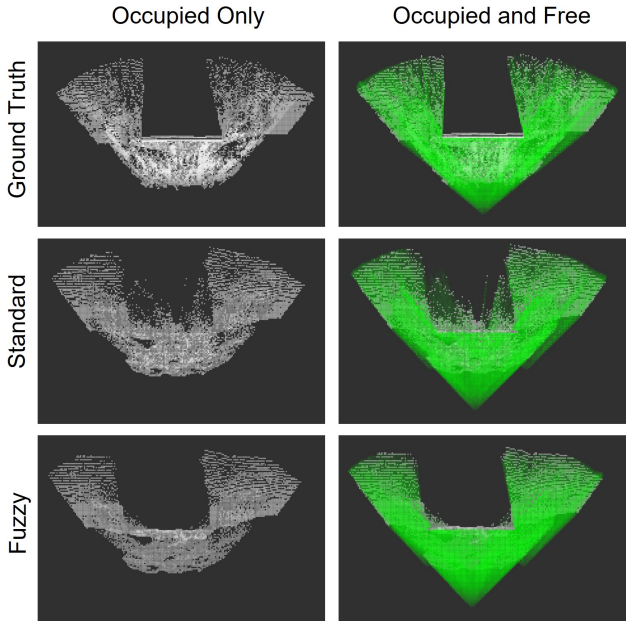


Fig. 7. Top-down views of the voxel maps after six additional observations with low confidence, moving backwards (down) away from the wall. The left column shows only occupied voxels, and the right column shows free space in green. Here, the standard method incorrectly marks voxels in the unknown region behind the wall, whereas the fuzzy method is able to more accurately capture the ground truth.

while moving away. Because the epipole is contained within the image, the extrinsic quality measure is much lower than during the sideways movement. The depth prediction is also much less accurate, both over and underestimating the depth. The confidence images from the first movement show very high confidence on the wall, whereas the confidence is much lower in images from the second movement.

When we insert the observations into UFOMap, the advantage of using the fuzzy UFOMap$^{\alpha}$ becomes clear. We start by inserting six EpiDepth observations from the initial pass into the map using both the fuzzy and standard methods. Fig. 6 shows top down views of the resulting voxel maps. For both methods, the initial reconstruction matches the ground truth very closely. Next, we update the map with six additional observations from the backwards movement. These new observations are much less confident than the first six, and the fuzzy method's ability to handle this uncertainty results in a more accurate reconstruction. Fig. 7 shows top down views of the

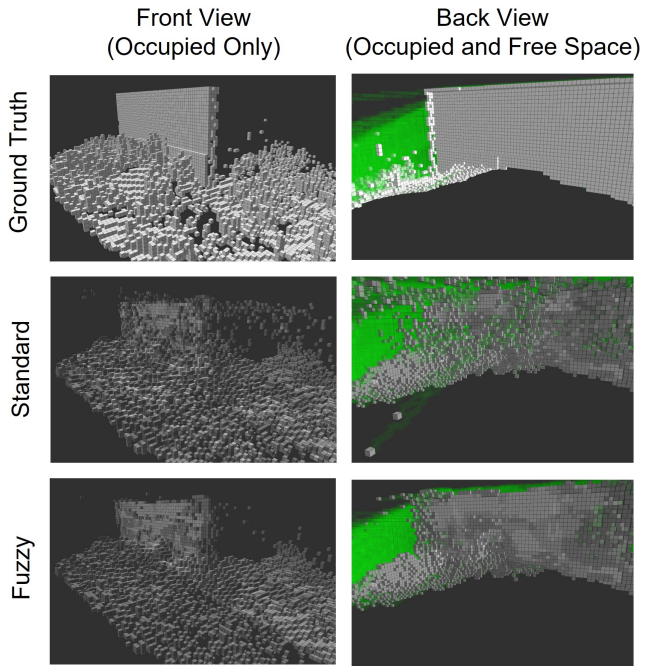
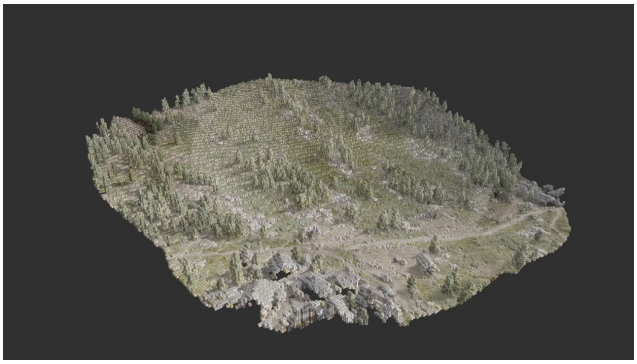


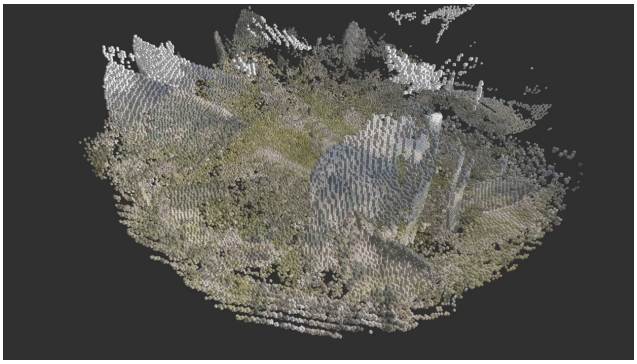
Fig. 8. Additional 3D views of the wall reconstruction scenario after the low-confidence update. The left column shows the front side of the wall (occupied voxels only), and the right column shows the unobserved (uncertain) back side of the wall with observed free space shown in green. Significant holes can be observed in the back side using the standard method.

voxel maps after the update. We see that the standard method has eroded away parts of the wall as some points are projected behind the existing structure. As these points are added, the free space updates along the rays that intersect the wall reduce the probability of occupancy for these voxels. This results in some holes and spurious occupied voxels that are added in what should be unknown space behind the wall. The fuzzy method has significantly fewer reconstruction errors, mainly because the low confidence of the observations resulted in fewer actual updates to the map. Some additional 3D views of the voxel maps after the second set of observations are shown in Fig. 8. These views show how noisy the standard method can be, and how the fuzzy method is able to create cleaner reconstructions by not committing observations with low confidence.

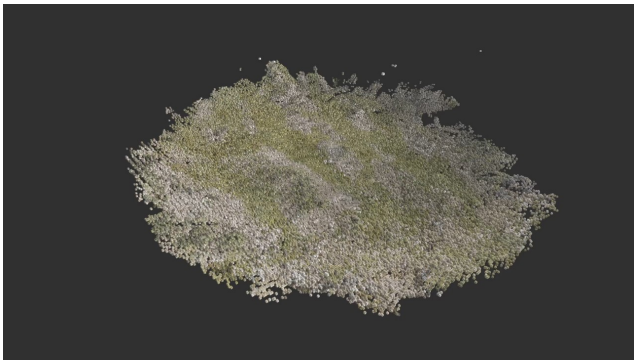
As an aside, we note that the map produced as ground truth also contains some reconstruction errors, particularly on the edges of objects with large differences in depth. This is an artifact of using a simulated depth image, where a pixel might represent multiple objects at different depths in the scene. The rendering algorithm may choose to average these depths or apply some additional processing that is not suitable for generating accurate reconstructions. This is an issue we intend to address in a future article, but for now it may be better to refer to this reference as the "gold standard" vs ground truth.



(a)



(b)



(c)

Fig. 9. Voxel maps constructed by the random movement scenario. (a) Ground truth. (b) Standard UFOMap approach. (c) Fuzzy UFOMap$^{\alpha}$ approach.

B. Random Movement

Our second experiment demonstrates how UFOMap$^{\alpha}$ compares to the standard approach in a real-time setting using random movements. In this scenario, the drone is commanded to fly continuously to random waypoints within a fixed region, each time with a different random pose. After over an hour of flying and more than 700 EpiDepth frame pairs selected, the resulting voxel maps can be compared.

Fig. 9 shows the voxel maps constructed by the standard and fuzzy methods, as well as the ground truth map (gold standard) observed during the experiment. The map produced by the standard approach is clearly much noisier than the one produced by UFOMap$^{\alpha}$. Many voxels have been added in the

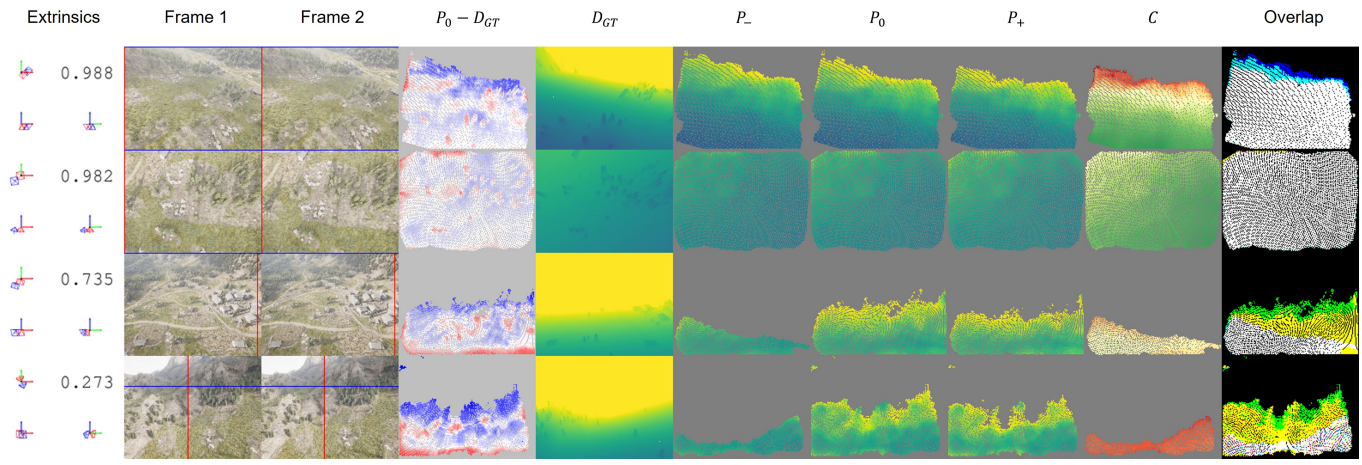


Fig. 10. Example frame pairs and evaluation images from the random movement scenario. Each row shows a pair of image frames and the resulting depth prediction from EpiDepth. From left to right, the images show the camera extrinsics and quality metric, the two image frames, the difference between the predicted depth and the ground truth, the ground truth, the three EpiDepth predictions, the confidence image, and the overlap between the predictions.

sky, where the EpiDepth algorithm failed to detect and clip out the invalid regions. The map produced by UFOMap$^{\alpha}$ is much cleaner, although it lacks many of the fine details present in the ground truth. These details are also missing from the standard UFOMap reconstruction, indicating that some improvements could still be made to the depth prediction algorithm.

Some example frame pairs selected by EpiDepth from this scenario are shown in Fig. 10. We see a variety of pose configurations that lead to different extrinsic quality scores. Frame pairs with low extrinsic scores result in lower confidence images and less weight applied to the map update. Frame pairs with higher extrinsic scores are added to the map with more confidence and weight. We notice that the prediction accuracy when compared to ground truth is generally good for broad terrain features, but not necessarily small details. Accuracy also decreases at farther distances and occasionally around the edge of the image. Many of these features such as trees and buildings are not currently detected properly by EpiDepth, which limits how well they can be reconstructed in the voxel map. These are known limitations of EpiDepth which can be improved with better image quality and parameter selection. This is part of the motivation to be able to better utilize the confidence of the depth predictions when building the voxel map.

VI. Conclusions and Future Work

The standard version of UFOMap and most voxel-based 3D mapping solutions do not fully capture the variable uncertainty that may be present when adding new sensor measurements. Small UAVs with image sensors can provide real-time depth estimates with SfM techniques such as EpiDepth, but these methods are fundamentally different than range based approaches like LiDAR that measure depth directly. The known algorithmic limitations of SfM lead to depth prediction confidences that are lost with the standard approach. Our fuzzy method, UFOMap$^{\alpha}$, is able to use a measure of prediction confidence to build cleaner and more accurate 3D maps.

The ideas presented in this article are intended to highlight the importance of capturing all types of uncertainty present when building 3D voxel maps of an environment. These maps ultimately represent some notion of the occupancy of a region of space (voxel). In a Bayesian setting, this might be in the form of a belief or probability that a voxel is occupied. A fuzzy interpretation might use separate maps to represent membership in the sets for occupied and free space as in [15]. In both cases, space is represented in a way that handles uncertainty and the unknown, but by not being limited to probabilities, additional logic and reasoning can be applied to build more useful maps. For instance, by using the agreement or contrast between the occupied and free sets, one can compute a measure of ambiguity or indeterminacy. This can lead to a more formal definition of which voxels are safe or unsafe to visit.

In future work, we intend to develop and make publicly available a complete fuzzy voxel mapping solution that extends the method presented in the present article. The fixed confidence thresholds of our approach can be made continuous, and the update procedure can be optimized. The region of space that is updated with some confidence of occupancy for each point can be made to extend beyond a single voxel, using a frustum that captures spatial uncertainty as well as confidence. We will also be investigating quantitative evaluation metrics to score and compare the different methods. These improvements will bring us closer to realizing a practical implementation of a full fuzzy voxel map.

References

[1] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

[2] D. Duberg and P. Jensfelt, "UFOMap: An Efficient Probabilistic 3D Mapping Framework That Embraces the Unknown," *arXiv:2003.04749 [cs]*, Mar. 2020.

[3] R. Camaioni, R. H. Luke, A. Buck, and D. T. Anderson, "EpiDepth: A real-time monocular dense-depth estimation pipeline using generic

image rectification," in *Geospatial Informatics XII*, vol. 12099. SPIE, May 2022, pp. 101–114.

[4] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[5] D. Kiss-Illés, C. Barrado, and E. Salamí, "GPS-SLAM: An augmentation of the ORB-SLAM algorithm," *Sensors*, vol. 19, no. 22, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/22/4973

[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[7] R. Mur-Artal and J. D. Tardos, "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.

[8] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[9] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova, "Unsupervised monocular depth learning in dynamic scenes," 2020. [Online]. Available: https://arxiv.org/abs/2010.16404

[10] M. Goldman, T. Hassner, and S. Avidan, "Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[11] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 8981–8989.

[12] C. O'Meadhra, W. Tabib, and N. Michael, "Variable resolution occupancy mapping using gaussian mixture models," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2015–2022, 2019.

[13] W. Tabib, K. Goel, J. Yao, C. Boirum, and N. Michael, "Autonomous cave surveying with an aerial robot," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1016–1032, 2022.

[14] A. Dhawale, X. Yang, and N. Michael, "Reactive collision avoidance using real-time local gaussian mixture model maps," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3545–3550.

[15] G. Oriolo, G. Ulivi, and M. Vendittelli, "Fuzzy maps: A new tool for mobile robot perception and planning," *Journal of Robotic Systems*, vol. 14, no. 3, pp. 179–197, 1997.

[16] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065