Towards Automated Void Detection for Search and Rescue with 3D Perception

Ananya Bal[1] Ashutosh Gupta[2*] Pranav Goyal[2*] David Merrick[3] Robin Murphy[4] Howie Choset[1]

***Abstract*—In a structural collapse, debris piles up in a chaotic and unstable manner, creating pockets and void spaces that are difficult to see or access. Often, these regions have the highest chances of concealing a survivor and identifying such regions can increase the success of a search and rescue (SAR) operation while ensuring the safety of both survivors and rescue teams. In this paper, we present an approach for *ex post facto* void detection in rubble piles by using registered 3D point clouds reconstructed from aerial images captured at multiple times on the scene. We perform a temporal layering of these point clouds to capture the dynamic surface of the rubble pile from multiple days of the SAR operation and analyze this 3D structure to detect candidate regions corresponding to void spaces. The layering is achieved by a parallel 3D point cloud reconstruction of the scene using the COLMAP Structure from Motion pipeline. The void detection is achieved by applying multiple point filtering criteria in thin segments of the 3D point clouds of the rubble. We test our approach on aerial images collected from the Surfside Structural Collapse at Miami in June 2021. Our method achieves an improvement in registration compared to the use of standard point cloud registration methods on individual 3D reconstructions. Through our method, we see translation errors reduce by 82%. Additionally, our method detects 9 out of 10 void spaces that were observed by experts in the rubble.**

***Index Terms*—Search and Rescue, Point Clouds, Registration**

I. Introduction

Structural collapses lead to substantial loss of life and therefore, improving the response to structural collapse disasters is a major focus in search and rescue robotics research [1] [2]. Of the 98 fatalities in the Surfside Structural Collapse, 9 were deemed to not be caused by crush injuries, indicating that 9 victims might have been rescued if they had been quickly found and extricated from the rubble [3], [4]. For rescue workers, locating void spaces is paramount as these regions have the highest chance of containing trapped survivors. But identifying voids can be challenging as they are small, irregularly shaped, and obscured by rubble [5]. Aerial imagery collected from the disaster site with uncrewed aerial systems (UASs) provides an broad view of the disaster scene which is helpful locating multiple void regions at once. 3D reconstructions generated from aerial images capture essential information that can be used to identify and characterize void spaces [6] and analyze terrain mobility for robots. While the use of 3D reconstructions is not real time yet, data analyzed through this method can help predict the occurrence, location, appearance and likelihood of voids in future collapses.

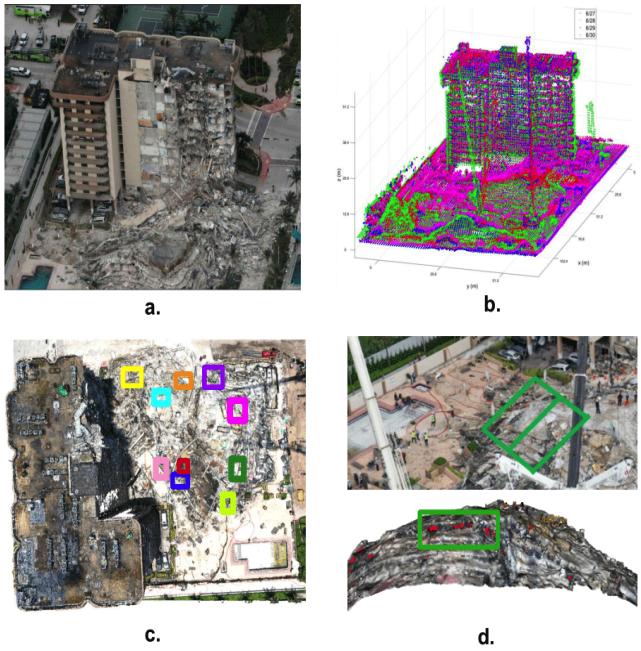

Fig. 1. Key contributions presented in our work - 1) The use of UAS images (Seen in a.) for generating a layered 3D reconstruction (Seen in b.) of the collapse scene. 2) Voids annotated by experts (Seen in c.) being identified by our method (Seen in d.).

To this end, we use 3D point cloud reconstructions of a disaster scene for void detection. A single 3D point cloud, generated from aerial images through Structure from Motion (SfM), is sufficient to characterize the surface of the rubble. However, for analyzing the internal structure of voids, it is essential to study the variations in geometry of the rubble as it is cleared over time. This warrants the use of multiple 3D reconstructions.

In this paper, we present a novel approach for semi-automated *ex post facto* detection of voids in the rubble pile of the Surfside Structural Collapse at Miami. Our pipeline utilizes the COLMAP [7] SfM workflow to reconstruct point clouds representing the dynamic SAR scene at different times. This reconstruction is done in parallel for images from multiple days in a single coordinate frame, resulting in registered point clouds. This provides us with lower translation errors as

* Authors of Equal Contribution
1 are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
2 are with Birla Institute of Technology and Science Pilani, Goa, India
3 is with Florida State University, Tallahassee, FL, USA
4 is with Texas AM University, College Station, TX, USA

compared to using standard point cloud registration methods. We use these stacked point clouds to determine regions that have changed over time. These regions are processed for detecting shape and color priors that we associate with void spaces. The layers in the stack helps us assess the internal dimensions of detected voids. The data we process and the outcomes from our contributions are presented in Fig 1.

Our method can be applied to similar previous data or data collected in the future to find regions presenting characteristics of void spaces. A collection of such regions could one day help build learning-based predictive systems for such disasters.

The rest of the paper is organized into the following sections - Section II discusses related work in the areas of UAS imagery, SfM for SAR operations, point cloud registration and void detection. Section III discusses our data collection and processing, followed by our methods in Section IV and results in V. Finally, we present conclusions, discussion and future work in Sections VI and VII.

II. Related Work

Aerial imagery, either from satellite cameras or through UAS flights on-site, can be used to assess damages to buildings after disasters [8]. Images captured from UASs are increasingly being leveraged for search and rescue operations like 3D reconstructions of buildings at disaster sites post earthquakes [9], floods [10], and fires [11], find traversable paths for robots [12], and perform surveillance of constantly evolving situations [13]. Temporal data capturing the evolution of the surface of rubble can help in quantifying the volumetric changes occurring during the clearing at a disaster site. Point-based monitoring techniques together with aerial photogrammetric surveys, are established techniques to derive surface displacements [14]. Registration of multi-temporal 3D surfaces is also used for quantifying changes in the natural environment [15]. However, a layering approach to rubble analysis is novel and our team is the first to apply this to a structural collapse.

3D point cloud reconstructions from a series of 2D images can easily be achiecved with Structure-from-Motion (SfM), a feature-based 3D scene reconstruction method in multi-view reconstruction in computer vision. It is a standard technique which uses 2D images to estimate the 3D structure of a scene and is available for use through multiple open-source and commercial software. While most studies conducting SfM on disaster data have used commercial photogrammetry software, a few recent studies [16] [17] show the use of open source software such as COLMAP. [16] proposes a new solution to autonomously map building damages with a commercial UAV in near real-time. [17] develops a framework for a COLMAP-based 3D reconstruction approach with a team of autonomous small UASs using a distributed behavior model. [18] is a comprehensive study outlining the performance of multiple open source and commercial SfM software on reconstructing 3D point clouds ffor search and rescue data. It shows that COLMAP is the fastest when it comes to performing feature matching and it generates high fidelity reconstructions with low errors. To the best of our knowledge, we are the first to apply COLMAP to search and rescue data for generating layered, registered reconstructions from multiple days.

While individual reconstructions from the scene done on images from different days can be registered using standard point cloud registration methods such as Iterative Closest Point (ICP) [19], Coherent Point Drift (CPD) [20] , and PointNetLK [21], these methods struggle to handle dynamic data. [22] tests multiple variants of ICP and outlines some of the challenges that cannot be handled by any of the variants. CPD is susceptible to noise and discontinuities [23] and PointNetLK has to be trained on search and rescue point cloud datasets to learn to handle changes in geometry [24].

There are several methods for void detection in point clouds. [25] presents a solution that has the user manually detect point cloud voids, and is then able to fill these voids in geometrically complex areas. This approach however cannot process both smooth and sharp (or low and high frequency geometry) voids at the same time. Other methods create polygon meshes from input point clouds using a Delaunay Triangulation algorithm, and then either perform a set of angle checks in order to determine which vertices are on the boundary of a void [26], or conduct a data search on the mesh to determine which vertices are present only in a single polygon, and thus lie on the void boundary [27]. [28] employs Ball Pivoting mesh generation to improve void detection. While these methods have shown results in automatically detecting holes in 3D point clouds, they do not perform well on highly unstructured data, and point clouds that are missing sections, such as that of a rubble pile. The approach presented in this work uses knowledge of geometric priors corresponding to voids to detect candidate regions and makes certain heuristic choices.

III. Data

The data used in this work was collected by flying a quadrotor UAS (DJI Mavic 2 Pro) over the Surfside disaster site in a grid-like flight pattern over multiple days of the SAR operation. Aerial top-down RGB and thermal images, as well as oblique images were collected during the flights. GPS locations were not captured and therefore, camera poses had to be estimated.

We use images captured over four days June 27 - June 30, 2021, from one mapping flight at roughly the same time each day: June 27 (Day 1), 13:30, June 28 (Day 2), 13:30, June 29 (Day 3), 13:00, and June 30 (Day 4), 11:00 local time. Each flight captured approximately 350 images, 75% of which had the collapse rubble and standing portion of the building. These four days were chosen because it captures the scene when majority the large debris on the top of the rubble was cleared. It is also a manageable set of data and with limited human activity on the rubble. The flight altitude during the days analysed in this work, was at an average $\sim$ 84 m.

IV. Methods

Our pipeline begins with a reconstruction of registered point clouds followed by point filtering for candidate region detection corresponding to void spaces. The components of the pipeline are visualized in Fig 2.

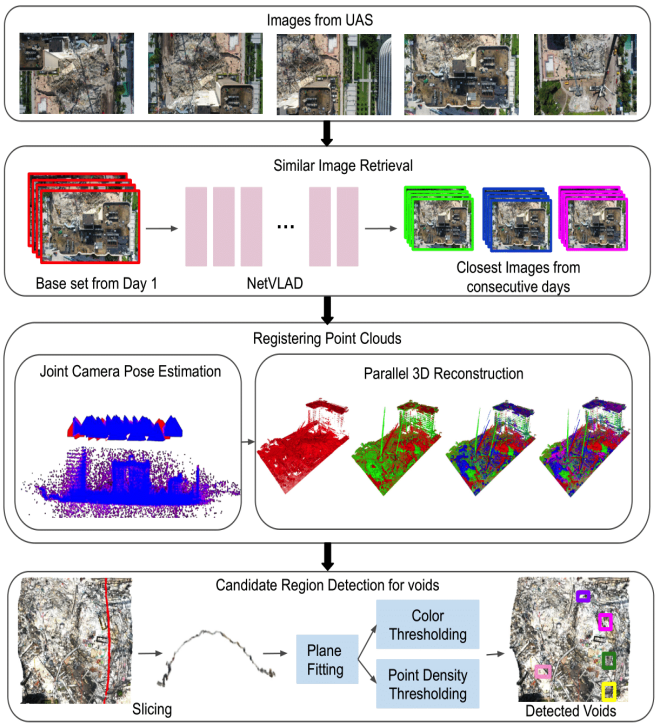

Fig. 2. Our pipeline with illustrations of results in intermediate steps.

A. Registering Point Clouds

Typically, all images processed through SfM result in the reconstruction of a single metric 3D point cloud. Processing all our images from four days together with SfM results in a single noisy point cloud/3D rubble surface as features on the dynamic rubble are not matched correctly over time. Since our approach relies on stacking multiple point clouds capturing the surface of the rubble from different days, images from different flights need to be processed with SfM separately. As SfM reconstructions are metric, the resulting point clouds will not be in the same coordinate frame and will differ by rotation and translation, requiring registration.

Early on, we processed our data to generate metric reconstructions from different days with Agisoft Metashape [29], assuming that standard point cloud registration methods would achieve reasonable results with low translation and rotation errors. However, a thorough comparison study of three standard point cloud registration methods [30] showed that these methods estimated registrations that at best had a translation error of 0.83 m for point clouds which that were initially 72 m apart. See Table I. Point clouds with such translation errors will not suffice for assessing accurate dimensions of void spaces which is necessary to determine envelopes for robot operation. The key takeaway was that standard point cloud registration methods failed for our highly dynamic scene where multiple distractors such as vehicles, people, collection containers etc., are present and reconstructed in the point clouds. Additionally, it was observed these metric reconstructions had small variations in scale.

This motivated us to pivot to registering the frames in which the point clouds are reconstructed. For this, we use a parallel data processing method which utilizes components of the standard Hierarchical Localization (hloc) + COLMAP pipeline [31] [32] [7].

TABLE I
LOWEST ROTATION AND TRANSLATION ERRORS OBTAINED BY POINT CLOUD REGISTRATION TECHNIQUES

Algorithm	Rotation Error (°)	Translation Error (m)
Point-to-Point ICP	1.0389	0.833
Rigid Coherent Point Drift	0.7402	0.889
PointNetLK	0.7438	1.062

As our flights collected images of buildings and roads surrounding the collapse site, processing all images would be overkill. To reduce our processing we retain only images pertaining to the collapse. This is also done in an semi-automated manner using learning-based image retrieval with NetVLAD [33]. First, we manually picked all images capturing the rubble and the adjoining standing portion of the building from the set of all images captured on Day 1. This formed our base set of images. Each image in this base set was then passed as a query to NetVLAD and the closest match from each of the consecutive days was stored in the retrieved set of images for the respective day.

All images from the base set and the three retrieved sets were then processed with COLMAP for a joint camera pose estimation for images from all days in a single coordinate frame. A reconstruction was consequently obtained through the COLMAP pipeline of feature matching, geometric verification, image registration, triangulation and bundle adjustment. This provides us with camera poses for images from each day, but is however, a noisy combined sparse reconstruction of the scene which fuses data from all days. To obtain clean registered dense point cloud reconstructions, we use estimated joint poses but process dense reconstruction in parallel for data from different days. This resulted in four registered dense point clouds, one from each day, representing stacked layers of the rubble's surface as it was cleared.

To refine this registration further for the rest of our pipeline, we apply ICP to determine translation along the depth axis (Z axis). This is done to minimize translation error to zero for obtaining accurate measurements for void dimensions. Once the translation error is negligible, we proceed with candidate void detection.

B. Detecting Candidate Regions for Void Spaces

In this work, based on a few previously detected voids in our data by SAR experts [34], we obtained a few appearance priors for void regions and tried to automate the detection of regions displaying such priors. We deemed the detected 3D point cloud regions corresponding to such priors as candidate void regions which were then compared to the regions detected by experts as voids in the rubble. The regions verified by experts as corresponding to voids (both naturally occurring,

as well as caused by excavation) are annotated over a point cloud reconstruction of the scene from day 1. See Fig 3.

Fig. 3. Annotated regions corresponding to voids located in the collapse scene by experts. (Naturally occurring voids shown in double boundaries and voids created by excavation shown in single boundaries)

As the rubble itself is highly dynamic due to the rescue operations, feature matching in images of the rubble could be sparse and erroneous. Therefore, images of static structures around the rubble is retained and used in point cloud reconstruction through the SfM pipeline. However, these static structures need to be eliminated before we analyze the point clouds for void detection. For this purpose, we manually define a bounding box to crop all four point clouds and retain only points corresponding to the rubble.

Once this is done, we proceed to slice the point clouds along the X and Y axes with a slicing width of 0.25 m. These stacked slices or 'cross-sections' are analyzed through plane fitting and color thresholding for detecting candidate regions. Voids in structural collapses tend to present under large debris such as concrete slabs, columns etc., [5] and are dark near the opening (if present) at the surface. These are the two characteristics we look for.

1) Plane fitting for sharp edge detection: To detect sharp edges, we analyze the angles between planes fit to the surface of the rubble. For fitting reasonably small planes to the surface, we need to divide the point cloud into point clusters which are small, having geometries that can be approximated by planes. We leverage octrees for this purpose and obtain such clusters from deep tree cells.

An octree was created for each individual sliced point cloud in the cross-section with a maximum depth of 8. We queried point clusters at the deepest level of the octree and found through multiple trials that this gave us adequately granular plane fitting. We fit a plane to each point cluster using the least-squares method and also find a projection of the centroid on the fit plane. Using these projected centroid points to find the nearest neighbours, we then calculated the angles of a given plane to its two closest neighboring planes. We chose two neighbours as each fit plane spans the entire width of the slice and the two nearest neighbours capture the angles of the current plane with its preceding and following planes. Any angle determined to be greater than 45° indicated a sharp edge and the corresponding point clusters resulting in this angle are retained as part of a candidate region.

2) Color thresholding for dark region detection: The retained points from the plane fitting procedure are then subjected to a color thresholding to detect dark regions [35]. All points with colors between $[0, 0, 0]$ and $[0.2, 0.2, 0.2]$ in the RGB space are retained with the others being discarded.

3) Point density as an indicator of void spaces: In our data, we find that the above two processes are sufficient to detect most candidate void regions sufficiently well. However, we also hypothesized and tested a filtering based on point density for detecting these candidate regions. Most void spaces with a sufficiently wide openings would present as a dark region in images. Standard feature detection methods will not detect feature matches in the pixel areas corresponding to these dark regions. Therefore, unless extensive hole filling methods are applied to make the point cloud reconstructions smooth, such dark regions could appear as holes or low point density patches in the reconstructed point cloud [36]. Looking for such regions can be an alternative to detecting dark regions, and in tandem with plane fitting, can help capture candidate regions when the color thresholding behaves unexpectedly due to change in lighting or needs fine-tuning.

In our implementation, for point clusters retained after plane fitting to slices, we checked for point density by defining a neighbourhood radius. All points having a number of neighbours in this radius lower than 0.75 times the average across the point cloud, were retained as part of identified candidate regions. We used this method after plane fitting as a parallel step to color thresholding and add newly detected regions to those identified by color thresholding.

After these methods, we apply a final noise filtering by using surface normals and DBScan Clustering with $eps = 0.2$ [37]. All point clusters which present surface normals pointing upwards are likely part of planar rubble and not voids. DBScan helps us remove very small point clusters which were erroneously picked up during plane fitting as it is susceptible to noise. It also helps us group sufficiently close point clusters as being a part of a larger void. Through this, we are able to retain larger detected point clusters corresponding features seen for voids.

We implement this detection method through two data processing schemes. In the first scheme, all candidate regions are found in the cropped point cloud from day 1 and no data from the consecutive point clouds is used. The main assumption here is that most regions having void-like characteristics would be visible before rubble clearing begins. However, taking into account that more void spaces can be uncovered as debris are cleared, we also use a second scheme, where only the point cloud regions with detected height changes as compared to the next point cloud, are processed through our detection pipeline.

TABLE II
ROTATION AND TRANSLATION ERRORS FOR REGISTRATION OBTAINED FROM OUR METHOD

Comparison Pair	Rotation Error (°)	Translation Error (m)
Day 2 to Day 1	0.0002	0.0744
Day 3 to Day 1	0.0006	0.1755
Day 4 to Day 1	0.0017	0.1929

C. Finding the maximum height of the detected voids

Once all candidate regions are located, we enlarge them by 3 m in both the X and Y axes to analyze a slightly larger region for obtaining the maximum height. For each point in the enlarged detected candidate region, by finding the corresponding points in the next layer, we compute point-wise distances. We provide the maximum distance as an upper bound on the height of the candidate void. In reality, given that voids are typically situated beneath large and thick debris, it is most likely possible that the true height of the void is lower. However, this value is hard to ascertain through imagery alone.

V. RESULTS

A. Registration

As we assigned images from day 1 to be the base set for our registration through SfM pipeline, we consider the point cloud reconstructed from images from day 1 to be the target, and the remaining point clouds to be sources. Rotation and translation errors from our method are presented in Table II.

It is observed that the rotation error is mostly negligible. The translation errors, with an average value of 0.1476 m, have drastically reduced in comparison to using standard point cloud registration methods for individual metric reconstructions. From a minimum translation error of 0.83 m [30], we are able to improve registration with our method by reducing the translation error by a further 0.68 m on an average. This is an 82% reduction in translation error.

Most of the translation offset in the reconstructions was along the Z (vertical) axis, perhaps due to depth ambiguity in the SfM arising from the dynamic nature of the collapse scene and a low ratio of oblique images.

B. 3D Reconstruction

COLMAP is an open source SfM solution that has not been applied widely for reconstructing search and rescue scenes, and has not been used to reconstruct multiple registered time-instance point clouds for a highly dynamic scene. In addition to quantifying how well it works for registration, we also quantify its performance in 3D reconstruction. This is done to provide a better understanding of how well COLMAP fares against reconstructions from commercial photogrammetry software like Agisoft Metashsape which are widely used by SAR experts.

We compare a point cloud reconstructed from our method against a reference reconstruction from the same data with Agisoft Metashape. We find that the average Root Mean Square Error is 0.115 m. The application of statistical outlier filtering shows that 1.7% of the points in our reconstruction are outliers. These values indicate that the COLMAP reconstruction has a high fidelity and is very close to the reconstruction from Agisoft Metashape [38]. Some regions with sparse correspondences lead to gaps in the reconstruction from COLMAP, however, the region corresponding to the rubble is as dense as the reconstruction from Agisoft Metashape.

TABLE III
PERFORMANCE OF OUR DETECTION METHOD FOR TWO PROCESSING SCHEMES

Point Cloud Processing Scheme	Naturally Occuring Voids Detected	Excavation Related Voids Detected	False Positives	Total Candidates
Point Cloud from Day 1	4/4	5/6	18	28
Height thresholded regions	4/4	5/6	6	16

C. Void Detection

We implement the candidate void detection pipeline - (1) for the entire cropped region of the point cloud pertaining to the rubble from day 1, and (2) for the points retained in the crop from day 1 after checking for height changes between subsequent layers. The detected regions are visually overlayed on the point cloud of the collapse site, shown in Fig 4. We compare the regions detected by our pipeline against the ones identified by experts as voids and present how many of the naturally occurring and excavation artifact voids were detected. We also present how many extra candidate regions (false positives) were captured in the point cloud which did not correspond to voids. See Table III. Locating survivors is most important, therefore false positive detections are less harmful than missed void detections.

Here, we observe that both point cloud processing schemes detect of 11 out of 12 void regions in the rubble pile, failing together only on a single region corresponding to an excavation artifact. One key result to note is that the second processing scheme, which uses only the point cloud regions that have changed in height over time, provides us with lesser false positive candidates, and is therefore, the preferred processing scheme. This shows that the layering method is better for candidate void detection.

Figs. 5 - 8 show the regions of the rubble corresponding to four detected void regions (two naturally occurring and two excavation artifacts) and their corresponding cross-sections showing maximum heights.

D. Heights of void regions

Table IV shows the cause of formation and measured maximum height found through our method for all 10 expert-identified voids. As the cyan void was not captured by our method, we do not present a measured height for it. We find that the average maximum height of candidate void regions is

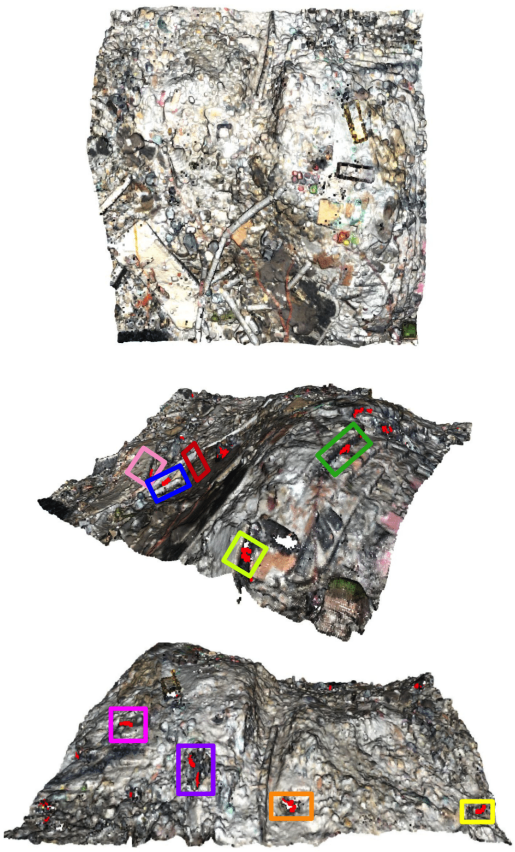

Fig. 4. Top: Point cloud from day 1 bounded by user-defined crop. Bottom: Detected candidate void regions (true and false positives) overlayed on the point cloud.

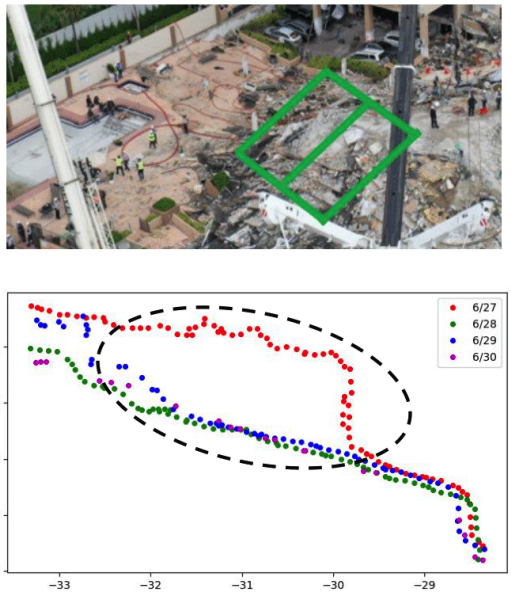

Fig. 5. The naturally occurring green void region and its corresponding cross section along the XZ axis. The removal of some material in the green void region is seen between day 1 and day 2.

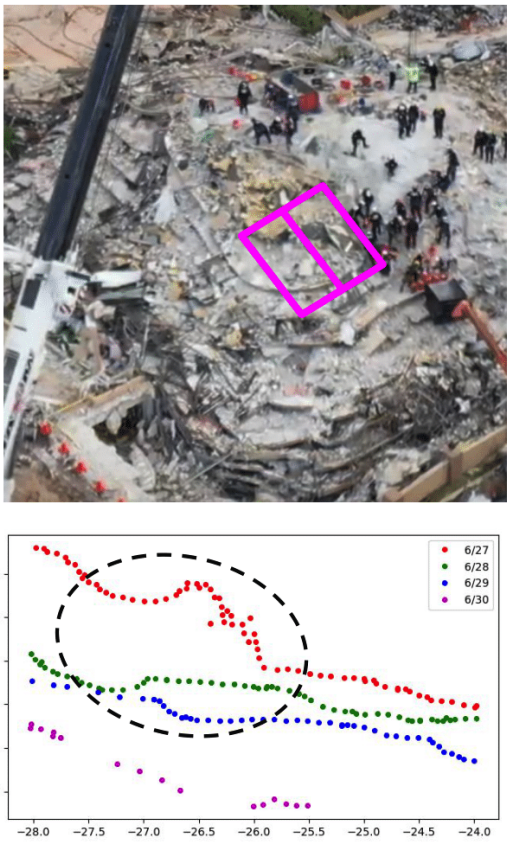

Fig. 6. The naturally occurring magenta void region and its corresponding cross section along the XZ axis. Features similar to the green void are observed in the cross section.

TABLE IV
DETECTED VOIDS AND THEIR MEASURED MAXIMUM HEIGHTS

Identified Void	**Cause**	**Max Height (m)**	**Identified Void**	**Cause**	**Max Height (m)**
Magenta	Natural	1.10	Yellow	Excavation	1.62
Green	Natural	0.97	Cyan	Excavation	-
Lime	Natural	2.29	Orange	Excavation	1.84
Pink	Natural	1.30	Purple	Excavation	1.94
Blue	Excavation	1.59	Maroon	Excavation	1.75

1.6 m. Most of these voids themselves are not large enough to be survivable. However, they could offer breathing space to survivors. These dimensions and the corresponding cross-sections show the types of void spaces that a robot would need to reach and inspect.

VI. CONCLUSION

To advance a crucial part of SAR operations, we present a novel method for generating a layered multi-day 3D point cloud reconstruction of a dynamic scene and apply this to *ex post facto* void detection in rubble with limited human input. We utilize image registration and joint camera pose estimation in SfM to reconstruct registered point clouds in parallel, becoming the first to do so for a highly dynamic

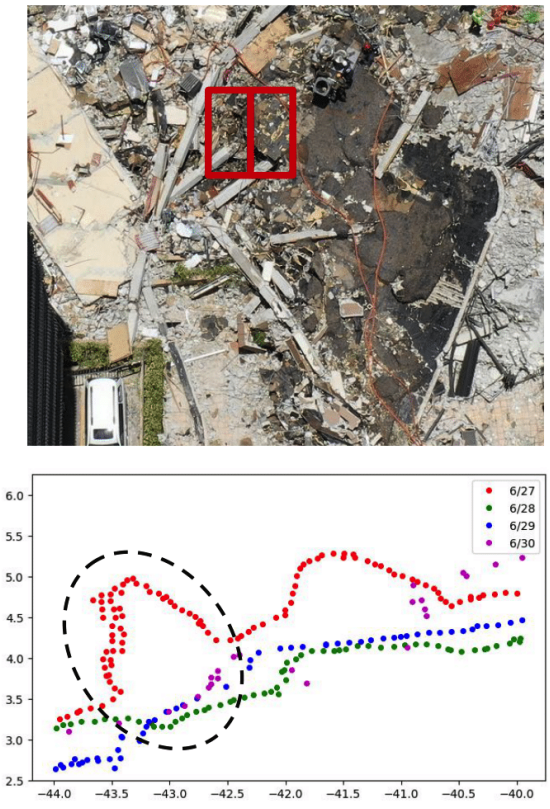

Fig. 7. The excavation artifact occurring for the blue void region and its corresponding cross section along the XZ axis. The presence of pillars shows two peaks in the cross section. Gaps were formed under the pillars as debris was cleared.

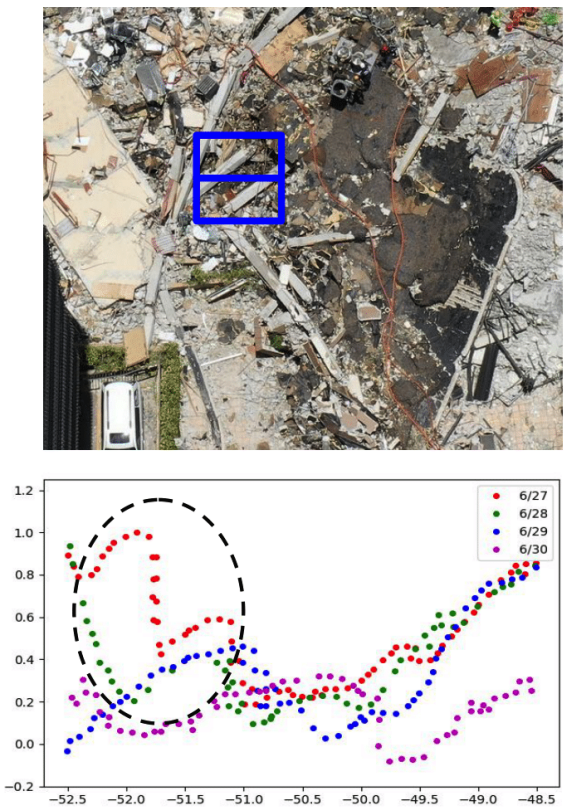

Fig. 8. The excavation artifact occurring for the maroon void region and its corresponding cross section along the YZ axis. There was a gap created under a ledge after rescue workers cleared some debris.

and unstructured SAR scene. We show the benefit of using this registered layering of point clouds not only to determine the heights of uncovered voids but also to detect candidate void regions with lower false positives. Our method for void detection is a systematic approach that starts with all points in the rubble region of the point cloud and filters out uninteresting regions through multiple priors such as checks on edges and color. The whole process helps us reduce errors in point cloud registration and make our height measurements between layers accurate. We also show that our candidate void detection method successfully locates all naturally occurring voids in the rubble and captures almost all excavation-generated void regions, with a reasonable number of false positives.

VII. Discussion and Future Work

We observe that the translation error from our registration method is majorly along the Z axis which is the depth axis in our SfM pipeline. This is caused due to depth ambiguity which can be lowered in future data collections by including more oblique images. Post our registered reconstruction, we use ICP for estimating transformations that minimize our translation errors to zero. However, we perform this registration only along the Z axis. This is done as we observed that the dynamic features in the point clouds were affecting the point correspondences detected in ICP and causing the algorithm to fall into local minima during the optimization process. Using the obtained transformations for registration in all three axes were causing translation errors to increase further. This fundamental problem motivates the development of a point correspondence finding method that can work well with dynamic point clouds.

Currently, we require a human to specify the point cloud region corresponding to the rubble. The layering approach can help us identify dynamic elements through height differences. While this should ideally only capture parts of the rubble that were cleared, it captures many distractors such as people, vehicles, vegetation etc., as well, making a human-defined crop necessary to retain purely the rubble for void detection. In the future, the application of point cloud segmentation can help segment these dynamic features and remove them from the scene.

Lastly, our candidate void detection method depends on certain heuristic values which we have fine-tuned for our dataset. While these can be tweaked to suit other datasets, efforts should be made to adjust the method to suit multiple SAR datasets. For this purpose, testing on other publicly available datasets is planned.

REFERENCES

[1] T. Linder, V. Tretyakov, S. Blumenthal, P. Molitor, D. Holz, R. Murphy, S. Tadokoro, and H. Surmann, "Rescue robots at the collapse of the municipal archive of cologne city: A field report," in *2010 ieee safety security and rescue robotics*. IEEE, 2010, pp. 1–6.

[2] R. R. Murphy and J. L. Burke, "Up from the rubble: Lessons learned about hri from search and rescue," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 3. SAGE Publications Sage CA: Los Angeles, CA, 2005, pp. 437–441.

[3] W. Rhodes and R. Ruiz-Goiriena, "At least 9 victims of 98 dead may have initially survived surfside condo collapse, but were not found by rescue teams, investigation shows," *USA Today*, Aug. 24, 2021; Mar 14, 2022, https://www.usatoday.com/story/news/nation/2021/08/24/surfside-condo-collapse-there-may-have-been-more-survivors-florida/8152538002/.

[4] S. Thaler, "Nine of the 98 surfside victims may have survived the initial condo collapse: Fire rescue logs show woman was alive in rubble for ten hours," *UK Daily Mail*, 2021, https://www.dailymail.co.uk/news/article-9924083/At-9-98-Surfside-victims-SURVIVED-initial-condo-collapse.html.

[5] D. Hu, S. Li, J. Chen, and V. R. Kamat, "Detecting, locating, and characterizing voids in disaster rubble for search and rescue," *Advanced Engineering Informatics*, vol. 42, p. 100974, 2019.

[6] R. R. Murphy, *Disaster robotics*. MIT press, 2014.

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

[8] R. Gupta, R. Hosfelt, S. Sajeev, N. Patel, B. Goodman, J. Doshi, E. Heim, H. Choset, and M. Gaston, "xbd: A dataset for assessing building damage from satellite imagery," 2019. [Online]. Available: https://arxiv.org/abs/1911.09296

[9] I. Kruijff-Korbayová, L. Freda, M. Gianni, V. Ntouskos, V. Hlaváč, V. Kubelka, E. Zimmermann, H. Surmann, K. Dulic, W. Rottner *et al.*, "Deployment of ground and aerial robots in earthquake-struck amatrice in italy (brief report)," in *2016 IEEE international symposium on safety, security, and rescue robotics (SSRR)*. IEEE, 2016, pp. 278–279.

[10] H. Surmann, D. Slomma, R. Grafe, and S. Grobelny, "Deployment of aerial robots during the flood disaster in erftstadt/blessem in july 2021," in *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, 2022, pp. 97–102.

[11] H. Surmann, D. Slomma, S. Grobelny, and R. Grafe, "Deployment of aerial robots after a major fire of an industrial hall with hazardous substances, a report," in *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2021, pp. 40–47.

[12] S. Zhang, X. Zhang, T. Li, J. Yuan, and Y. Fang, "Fast active aerial exploration for traversable path finding of ground robots in unknown environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[13] J. J. Roldán-Gómez, E. González-Gironda, and A. Barrientos, "A survey on robotic technologies for forest firefighting: Applying drone swarms to improve firefighters' efficiency and safety," *Applied Sciences*, vol. 11, no. 1, p. 363, 2021.

[14] M. V. Peppa, J. P. Mills, P. Moore, P. E. Miller, and J. E. Chambers, "Automated co-registration and calibration in sfm photogrammetry for landslide change detection," *Earth Surface Processes and Landforms*, vol. 44, no. 1, pp. 287–303, 2019.

[15] D. Wujanz, D. Krueger, and F. Neitzel, "Identification of stable areas in unreferenced laser scans for deformation measurement," *The Photogrammetric Record*, vol. 31, no. 155, pp. 261–280, 2016.

[16] F. Nex, D. Duarte, A. Steenbeek, and N. Kerle, "Towards real-time building damage mapping with low-cost uav solutions," *Remote sensing*, vol. 11, no. 3, p. 287, 2019.

[17] D. S. Smith Jr and H. E. Sevil, "Design of a rapid structure from motion (sfm) based 3d reconstruction framework using a team of autonomous small unmanned aerial systems (suas)," *Robotics*, vol. 11, no. 5, p. 89, 2022.

[18] A. Martell, H. A. Lauterbach, A. Nuchtcer *et al.*, "Benchmarking structure from motion algorithms of urban environments with applications to reconnaissance in search and rescue scenarios," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2018, pp. 1–7.

[19] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[20] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[21] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7163–7172.

[22] Z. Dong, F. Liang, B. Yang, Y. Xu, Y. Zang, J. Li, Y. Wang, W. Dai, H. Fan, J. Hyyppä *et al.*, "Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 163, pp. 327–342, 2020.

[23] Z. Zhou, J. Zheng, Y. Dai, Z. Zhou, and S. Chen, "Robust non-rigid point set registration using student's-t mixture model," *PloS one*, vol. 9, no. 3, p. e91381, 2014.

[24] X. Huang, Y. Wang, S. Li, G. Mei, Z. Xu, Y. Wang, J. Zhang, and M. Bennamoun, "Robust real-world point cloud registration by inlier detection," *Computer Vision and Image Understanding*, vol. 224, p. 103556, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314222001345

[25] "Shape-controllable geometry completion for point cloud models," *The Visual Computer*, vol. 33, no. 3.

[26] T. K. Dey and J. Giesen, "Detecting undersampling in surface reconstruction," *Discrete and Computational Geometry*, vol. 15, pp. 329–345, 2003.

[27] Y. Jun, "A piecewise hole filling algorithm in reverse engineering," *Computer-Aided Design*, vol. 37, no. 2, pp. 263–270, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010448504001320

[28] B. Bird, B. Lennox, S. Watson, and T. Wright, "Autonomous void detection and characterisation in point clouds and triangular meshes," *International Journal of Computational Vision and Robotics*, vol. 9, no. 4, pp. 368 – 386, Aug. 2019.

[29] L. Agisoft, "Agisoft metashape user manual," *Release*, vol. 1, no. 0, pp. 1–199, 2020.

[30] A. Bal, R. Ladig, P. Goyal, J. Galeotti, H. Choset, D. Merrick, and R. Murphy, "A comparison of point cloud registration techniques for on-site disaster data from the surfside structural collapse," 2022.

[31] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019.

[32] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *CVPR*, 2020.

[33] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[34] A. Rao, R. Murphy, D. Merrick, and H. Choset, "Analysis of interior rubble void spaces at champlain towers south collapse," in *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2022, pp. 379–384.

[35] J. Guo, Z. Zhang, Y. Mao, S. Liu, W. Zhu, and T. Yang, "Automatic extraction of discontinuity traces from 3d rock mass point clouds considering the influence of light shadows and color change," *Remote Sensing*, vol. 14, no. 21, p. 5314, 2022.

[36] X. Guo, J. Xiao, and Y. Wang, "A survey on algorithms of hole filling in 3d surface reconstruction," *The Visual Computer*, vol. 34, pp. 93–103, 2018.

[37] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[38] F. Remondino, L. Morelli, E. Stathopoulou, M. Elhashash, and R. Qin, "Aerial triangulation with learning-based tie points," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, no. B2-2022, pp. 77–84, 2022.