A Comparison of Point Cloud Registration Techniques for On-site Disaster Data from the Surfside Structural Collapse

Ananya Bal[1], Robert Ladig[2], Pranav Goyal[3], John Galeotti[1], Howie Choset[1], David Merrick[4], Robin Murphy[5]

***Abstract*— 3D representations of geographical surfaces in the form of dense point clouds can be a valuable tool for documenting and reconstructing a structural collapse, such as the 2021 Champlain Towers Condominium collapse in Surfside, Florida. Point cloud data reconstructed from aerial footage taken by uncrewed aerial systems at frequent intervals from a dynamic search and rescue scene poses significant challenges. Properly aligning large point clouds in this context, or registering them, poses noteworthy issues as they capture multiple regions whose geometries change over time. These regions denote dynamic features such as excavation machinery, cones marking boundaries and the structural collapse rubble itself. In this paper, the performances of commonly used point cloud registration methods for dynamic scenes present in the raw data are studied. The use of Iterative Closest Point (ICP), Rigid - Coherent Point Drift (CPD) and PointNetLK for registering dense point clouds, reconstructed sequentially over a time-frame of five days, is studied and evaluated. All methods are compared by error in performance, execution time, and robustness with a concluding analysis and a judgement of the preeminent method for the specific data at hand is provided.**

I. INTRODUCTION

Structural collapses worldwide regularly lead to substantial loss of life and financial damages and hence have long been a big focus of the search and rescue (SAR) research community [1][2][3][4]. In this work, the structural collapse in focus is the Champlain Towers South Condominium collapse at Surfside, Florida on June 2021. It is the third largest structure collapse in the United States, resulting in 98 deaths, 9 of which could have been avoided if the victims had been quickly found and extricated from the rubble [5]. The prospect of being able to save more lives is a significant research driver towards automated 3D perception of rubble piles. 3D perception of rubble from a structural collapse becomes more and more paramount for SAR operations, to understand terrain mobility for robot deployments or potentially locating and characterizing void spaces for extricating possible survivors. With the increased adoption and lowering cost of LIDAR and RGB-D sensing solutions and computationally cheap Structure from Motion (SfM) methods, point clouds are gradually becoming the favoured modality to represent collapse sites in 3D [6], [7].

Point clouds are unordered sets of points in 3D space, typically defining the surface of an object or a scene. The lack of order in point clouds renders processing them more complex than processing images as they require geometric processing or non-convolutional feature-learning methods. Whether a point cloud is captured or reconstructed is dependant on the modality of sensing. LIDAR and RGB-D solutions are able to capture point clouds whereas 2D imaging feeds into SfM methods to reconstruct point clouds. In the case of the Condominium disaster at Surfside, Floarida, uncrewed aerial systems (UAS) were deployed to collect aerial images, which were then fed post-deployment into SfM software to reconstruct dense point clouds of the site at regular intervals as the rubble was being cleared.

The resulting point clouds represent a highly dynamic scene of the collapse site and potentially could be analyzed for a variety of important SAR indicators, such as void spaces or volume differences in the rubble when aligned correctly in 3D space. However, data captured on-site as well as the reconstructed data can be noisy and/or inaccurate due to a number of reasons, which we discuss in Section III. This results in incorrectly aligned or unregistered point clouds in 3D space. Point cloud registration plays a critical role in numerous computer vision and perception applications like 3D reconstruction. For the purposes mentioned above, it is imperative that the reconstructed point clouds are as well aligned as possible. Manually aligning the point clouds can be an arduous task and could take long while being prone to human errors. Therefore, automated alignment of these point clouds in 3D space is necessary and can be achieved through point cloud registration methods.

There are numerous different types of registration methods: optimisation-based methods, feature learning-based methods, heuristic approaches, etc. In this paper, three widely-adopted point cloud registration techniques are compared and assessed on real-world data, specifically the data collected from the Champlain Towers South Condominium collapse at Surfside (from here on out shortened to only "Surfside" for convenience). GPU implementations of optimization-based point-to-point ICP and Rigid CPD, and end-to-end learning-based PointNetLK (discussed in Section IV) on point clouds representing the dynamic scene from Surfside are tested and evaluated. Insights into the performances in terms of quantitative errors, execution times and qualitative visualizations of the alignments are provided

*Portions of this work were supported by NSF Award CMMI 2140528. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[1]A. Bal J. Galeotti and H. Choset are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA abal@andrew.cmu.edu

[2]R. Ladig is with the Department of Science and Engineering, Ritsumeikan University, Kusatsu, JAPAN

[3]P. Goyal is with the Birla Institute of Technology & Science - Pilani, Goa, INDIA

[4]D. Merrick is with the College of Social Science and Public Policy, Florida State University, FL, USA

[5]R. Murphy is Raytheon Professor of Computer Science and Engineering, Texas A&M University, TX, USA

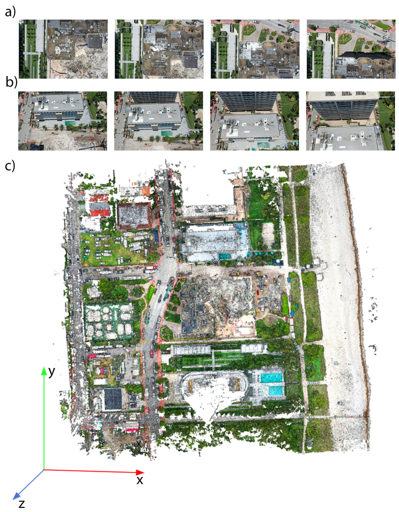


Fig. 1: a) Top-down pictures captured during the deployment b) Oblique pictures captured during the deployment c) Dense point cloud generated post-deployment ($\approx 10^8$ points)

in Section VI. We discuss our findings, conclusions and the future work in Sections VII, VIII and IX.

II. RELATED WORK

Aerial imagery (either via satellite imagery or data collection on-site) combined with computer vision algorithms can be used to asses damages to buildings after disasters [8]. Images captured from UAS are increasingly being utilized in search and rescue operations like reconstructing 3D models of buildings at disaster sites post earthquakes [9], floods [6], and fires [7], find traversable paths for robots [10], and perform surveillance of constantly evolving situations[11].

Time-series surface observations can aid quantification of the volumetric changes occurring at a disaster site. Point-based monitoring techniques, based on global navigation satellite systems, together with aerial photogrammetric surveys, are established techniques to derive surface displacements [12]. Registration of multi-temporal 3D surfaces is also used for quantifying deformations in the natural environment [13].

There are various methods for point cloud registration. Iterative Closest Point (ICP) algorithm, introduced by Besl and McKay [14] and Zhang [15], is one of the the most popular methods for rigid point set registration due to its simplicity. Over time, many variants of ICP have been proposed that modify different phases of the algorithm from the selection and matching of points to the minimization strategy [16], [17]. ICP requires that the initial position of the two point sets be adequately close [18].

To overcome the limitations of ICP, probabilistic methods were developed [19], [20]. These methods establish correspondences between all combinations of points according to some probability, which generalizes the binary correspondence assignment in ICP. Among these methods are Robust Point Matching (RPM) algorithm introduced by Gold et al. [21], and its later variants [19], [22], [18]. In [23], it was shown that in RPM, alternating soft-assignment of correspondences and transformation is equivalent to the Expectation Maximization (EM) algorithm for Gaussian Mixture Models (GMM), where one point set is treated as GMM centroids with equal isotropic covariances and the other point set is treated as data points. [18] was the first work to propose a closed form solution to the maximization step (M-step) of the EM algorithm for a general multidimensional case, giving us Coherent Point Drift.

Inspired by feature-based inlier estimation techniques, [24] introduced PointNet, a learning-based approach to extract task-specific features from point clouds. PointNet helps process unordered point clouds in a learning paradigm. PointNetLK [25] takes inspiration from [26] and utilizes the classical Lucas & Kanade (LK) algorithm for point cloud registration in conjunction with learned features from PointNet.

III. DATA

On June 24th, 2021 at approximately 1:22 am local time, the Champlain Towers Condominium suffered a partial collapse, killing 98 people. The first sortie using UAS to collect aerial footage was conducted on June 25th at 16:30 local time. Aerial top-down RGB and thermal images, as well as oblique images angled by 80° to the ground were collected in a grid-like, sequential pattern using a commercially available of-the-shelf quadrotor UAS (DJI Mavic 2 Pro). The UAS used for the data in this work is equipped with a camera with an approximately 77° field of view and a 1 inch CMOS image sensor. Image resolution of the acquired top-down pictures was 5472×3648 pixels. To keep an easy overview of the data, the focus in this work will be on the collected data of 5 separate days at a similar time of day. Those are the sorties at June 27th 13:30, June 28th 13:30, June 29th 13:00, June 30th 11:00 and July 1st 09:00. The flight altitude during the days analysed in this work, was at an average of 83.94m.

The images were fed into a commonly used photogrammetry software, Agisoft Metashape (formerly Photoscan) [27], where the images were processed into a dense 3D point cloud using a 12th Gen Intel i9, NVIDIA GeForce RTX 3090 and 128 GB memory at a average of 54 minutes per generated point cloud. All data discussed in this work was processed post-deployment as on-site processing was constrained by limited computational capability. The software estimated arithmetic average euclidean distance error over all dimensions is 2.208 m. Examples of the images captured on one of the 5 days, as well as the resulting dense point cloud can be seen in Fig. 1.

A. Challenges in Data Preparation

From the real-world Surfside data collection, it was observed that the SfM reconstruction could be noisy, full of inaccuracies such as inconsistent shifting of static landmarks or have missing features within the dense point clouds. It was concluded that the following reasons contributed to an imperfect image data collection that led to problems during post-deployment data processing:

Fig. 2: Cropped and Normalized Sample Point Cloud from 28th June seen from the X, Y and Z axes respectively from left to right ($\approx$ 2x10^6 points)

- Approximations for camera poses during bundle adjustment in the creation of the SfM process. Until the June 30th data, camera poses are exclusively calculated by the SfM software, since neither real time kinematics (RTK), nor ground control points (GCP) were available.
- The lack of or inconsistency in ground control points (GCP). Rescue workers would tend not not prioritize placing GCP markers which would give lower estimation error of the camera positions for SfM reconstruction. For the Surfside deployment, painted GCP markers were properly established only after 30th June, 5 days after the first sortie. Even after markers were placed, they were easily occluded by dust, equipment or shifting rubble and hence did not stay consistent during the whole time of deployment.
- Sub-optimal altitude. Since the top-down and oblique images were collected using a scripted flight path, manually avoiding obstacles was not possible. To clear the cranes used to remove rubble at Surfside, as well as clear the height of the surrounding buildings, the flight altitude was chosen higher then would be optimal for the used UAS.
- Imprecise camera calibration. It was found that some pre-flight calibrations during the deployment produced entirely wrong calibration data, resulting in unusable dense point clouds reconstructions. This was due to the inability to quickly check the robustness of the camera calibration on-site.
- Shifting illumination during different day-times. While the attempt was made to keep flight times in between days as consistent as possible, events out of out control such as strong winds, rain, or presidential no fly zones keeps the UAS grounded for some sorties, introducing gaps in the data.
- Presence of occlusions like smoke from small fires during the first days of deployment.
- Motion blur during sorties during the evening due to low exposure times.

B. Unique Challenges in Registration for Surfside Data

As discussed earlier, during a real life SAR deployment, numerous factors contribute to imperfect data collection. This introduces unwanted rotation and translation to the reconstructed point clouds time slices (3D point clouds reconstructed from 2D images captured at certain different points in time). These factors could also introduce subtle variations in the scale of the point clouds, although no notable difference in scale was observed and therefore a rigid transformation solution was assumed for registration.

Additionally, the geographic areas represented in the point clouds vary with sortie paths and are not necessarily the same. While the central rubble is always imaged and present in the point clouds, the extent of capture for static and dynamic features around the site, like adjacent buildings, roads, cars etc., varies. This variation, if not handled, turns the alignment into to a partial-to-partial or partial-to-complete point cloud registration which requires more complex methods that involve mask prediction [28] or correspondence mapping [29]. In order to keep the amount of points to analyse at a reasonable number while still being able to obtain the proper alignment, processing for all time slices is limited to the same area, the main condominium block of Surfside (rubble pile with the partial standing building structure - See Fig. 2) .

In addition to requiring heavy computational power due to the high number of points in the dense slices, registering the slices is more complex than registering point clouds of static objects as our data represents a dynamic scene where:

- Geometry of certain features, namely the rubble pile, changes with time as the rescue workers clear debris.
- Objects can move around the scene without changing shape, namely cars, excavation vehicles etc.
- The same objects can disappear from the scene entirely.

These features can affect how registration algorithms perform. For example, changing shapes can affect point correspondences in ICP.

C. Data for Registration Experiments

For our experiments, we have considered five point clouds reconstructed from the mid-day sortie on each day from 27th June, 2021 to 1st July, 2021 (see Section III). Each of these point clouds was cropped to limit the region of interest to the rubble pile, the remaining part of the standing Condominium structure and the dumping area for the cleared rubble See Fig 2. Additionally, these point clouds were normalized to have all points be within a unit cube. We have limited our experiments to these days as most of the initial rubble clearance was done during these days. We have tested the registration methods for each source and template combination possible from the five slices.

TABLE I: ROTATION & TRANSLATION ERRORS AND EXECUTION TIMES FOR 20 PAIR-WISE POINT CLOUD REGISTRATIONS OVER 3 CATEGORIES

Algorithm	10000 points				15000 points				20000 points			
	Rot. Error (°)	Norm. Trans. Error	Trans. Error (m)	Time (s)	Rot. Error (°)	Norm. Trans. Error	Trans. Error (m)	Time (s)	Rot. Error (°)	Norm. Trans. Error	Trans. Error (m)	Time (s)
ICP : O(N^2)	1.0175	0.00658	0.842	0.048	1.0389	0.00651	0.833	0.064	1.036	0.0066	0.844	0.079
CPD : O(N^2)	0.7402	0.00695	0.889	5.39	0.8736	0.00695	0.889	9.255	0.5661	0.0079	1.011	13.919
PNLK : O(N)	0.7647	0.00844	1.08	0.067	0.7633	0.0086	1.1	0.092	0.7438	0.0083	1.062	0.103

D. Ground Truth

The performances of the registration methods in this study are quantified by comparing obtained transformations from the registration techniques to manually defined transformations. We have used the open source solution CloudCompare [30] for manually aligning point clouds and obtaining the ground truth baseline for registration.

IV. METHODS

This section explains the normalization method we have applied to our data, as well as the registration techniques we are comparing.

A. Importance of Unit Cube Normalization

ICP implementations usually require an initial transformation initialization. This initial estimation is supposed to roughly align the source point cloud (to be registered) to the template point cloud. Estimating this initialization requires additional computation through global registration methods. Since our data has reconstruction errors around 2.2 m, which leads to a maximum of bidirectional translation error of 4.4 m in the point clouds, finding or estimating an initialization for the unnormalized point clouds would require additional computation or guess-work. Normalizing the source and template point clouds eliminates the need to find an initialization for ICP as it scales down the translation difference between two point clouds, making an identity matrix suitable to be provided as an initialization. For unit cube normalization, the highest length of the point cloud along the X, Y or Z axis is used as the scaling factor. The (X,Y,Z) coordinate of each 3D point is divided by this value, which in our case was 128.

B. Point-to-point ICP

Point-to-point ICP is an optimization-based registration method that iteratively estimates point correspondences and performs a least squares optimization using the point-to-point distance metric, till the error is below a defined threshold or does not reduce further. As mentioned in [31], the mathematical theory behind the algorithm guarantees a convergence and the algorithm generalizes to different kinds of geometries without requiring any training data. However, the algorithm struggles to overcome variations like noise, outliers, density variations and partial overlap. Many sophisticated strategies with high computational costs are required to help make point-to-point ICP robust to these variations [31] . Although there are ICP implementations with k-d trees which have a reduced time complexity of $O(N \log N)$, we use the original iterative implementation with time complexity $O(N^2)$, where N is the number of points in the source or template point cloud (assuming the number of points in the source and template point clouds are comparable or same). We have used a Pytorch3D GPU implementation of point-to-point ICP, with an 4x4 identity matrix at the initialization. We do not estimate a separate initialization.

C. Coherent Point Drift

Coherent Point Drift is a Gaussian Mixture Model-based optimization method for point cloud registration. The algorithm formulates the distance estimation into a likelihood maximization problem. GMM centroids calculated from the source point cloud are forced to move coherently towards the template. CPD can be applied to non-rigid registration as well but we apply it to rigid registration. In the rigid variant, the centroids' coherence constraint is re-parameterized with rigid parameters of GMM centroid locations. We have used a GPU implementation of this variant. CPD is more robust to outliers than ICP [31]. However, the algorithm is lengthy and requires multiple computations. The time complexity of the algorithm is $O(N^2)$, where N is the number of points in the source or template point cloud (assuming the number of points in the both point clouds are comparable or same).

D. PointNetLK

PointNetLK is a feature learning-based end-to-end registration method which fuses the PointNet feature encoder with an Inverse Compositional (IC) formulation of the Lucas Kanade image registration algorithm. The time complexity of the algorithm is $O(N)$ where N is the number of points in the source. As PointNetLK samples the same number of points from both the source and template point clouds, the number of points in template is the same as in the source. This lower time complexity makes PointNetLK scale better to processing more points in large point clouds. Our implementation of PointNetLK runs on the GPU and uses pre-trained weights from training on the ModelNet40 dataset. We use these weights instead of training on domain-specific data as our paper focusses on read-to-run methods and training on SAR data would consume further time. Using network weights from a different domain may not give the best performance possible, however, the added option to optimize the performance of this method by training from scratch is an advantage.

TABLE II: BREAKDOWN VALUES FOR ROTATION AND TRANSLATION PERTURBATIONS

Algorithm	20000 points								
	Rot. along X axis (°)	Rot. along Y axis (°)	Rot. along Z axis (°)	Norm. Trans. along X axis	Trans. along X axis (m)	Norm. Trans. along Y axis	Trans. along Y axis (m)	Norm. Trans. along Z axis	Trans. along Z axis (m)
ICP	3.5	2.5	3.5	0.012	1.536	0.014	1.792	0.016	2.048
CPD	5	3	3	0.007	0.896	0.009	1.152	0.012	1.536
PNLK	88	67	75	0.04	5.12	0.02	2.56	0.018	2.304

All these methods were run on an NVIDIA Titan RTX 2080 Ti GPU with 24GB memory.

V. EXPERIMENTS

Our first set of experiments is to compare three registration methods on 20 pair-wise registrations for 10000, 15000, 20000 sampled points in the source and template point clouds.

The second set of experiments was performed to test the robustness of the registration methods to initial perturbations. By adding an initial perturbation to the source point cloud, in the form of increasing rotation and translation (individually along each axis) in small intervals, we check at what perturbation, the algorithms break down to produce unsuccessful registrations. A successful registration is one where the rotation error is smaller then 5° and the normalized translation error is smaller then 0.01 [25].

VI. RESULTS

The results from the first set of experiments are presented in Table I. A scaling factor of 128 is used to obtain the unnormalized tranlsation errors in meters as it's the averaged scaling factor for all point clouds considered. The averages of rotation error in degrees, normalized translation error, translation error in meters and the execution time for all three methods are provided across three categories. We observe that ICP gives us the lowest translation error with 15000 points and that Rigid CPD gives us the lowest rotation error with 20000 points over the 20 registrations. ICP and PointNetLK execute very fast compared to CPD.

Another observation is that the rotation and translation errors do not decrease significantly with an increase in the number of points in the point cloud. This can be seen clearly in Fig. 3. This signifies that subsampling our point clouds to retain 10000-20000 points is enough to represent all the features well enough for registration.

The results from the second set of experiments are presented in Table II, where we can observe the breakdown perturbations, both in rotation and translation, at which the registration algorithms stop producing successful registrations. We see that PointNetLK is the most robust algorithm as it can register point clouds even when they are heavily misaligned (rotated by up to 85°and translated by up to 5 m). ICP and CPD are much less robust and can withstand misalignment in the point clouds only up to a few degrees in rotation and a meter in translation on an average. We see different breakdown values over the axes as the number of points in our data along any given axis varies. Therefore, the breakdown value along any particular axis is dependent on the geometry of the point clouds.

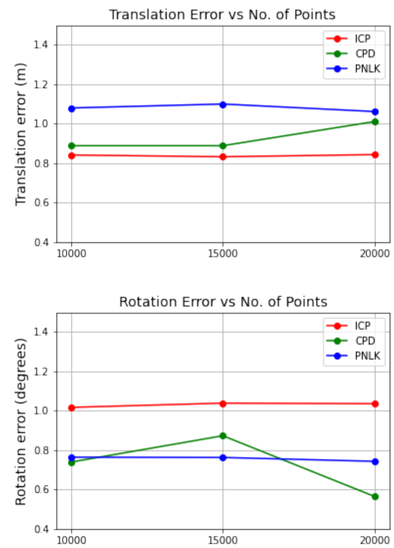
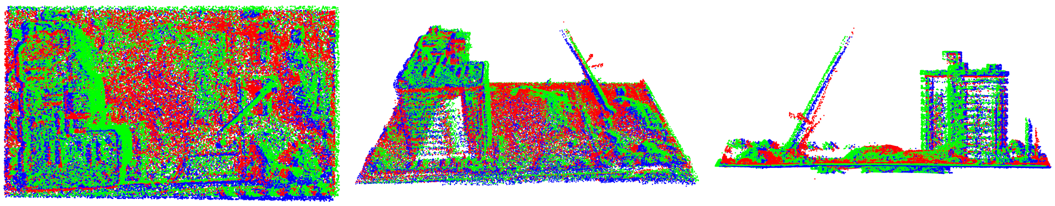
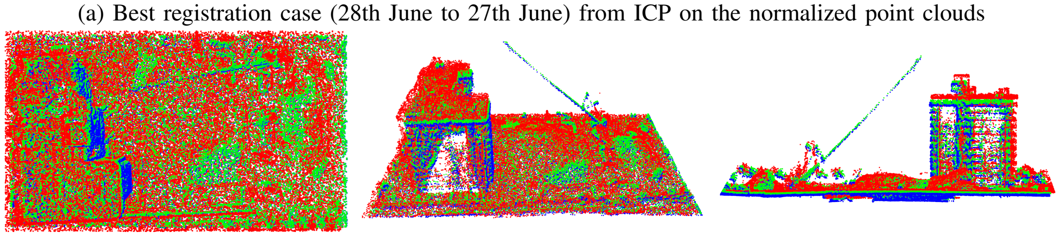


Fig. 3: Rotation and Translation Errors in regards to No. of points in the source/template point cloud

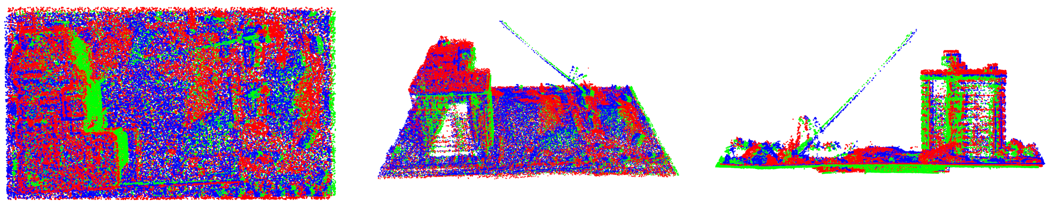
We provide qualitative results through images of the best case performance of each of the algorithms (See Fig. 4). ICP and PointNetLK seem to perform better than CPD as the registered point cloud (seen in blue) is closer to the template (seen in red) which is especially visible in the registration visualization around the standing portion of the building and the outer edges of the point clouds. We also visualize the cross-sectional view of a region of static feature (swimming pool) of the first point cloud from 27th June as the template to which all the other point clouds are registered (See Fig. 5).This view is provided for registrations from ICP, CPD and PointNetLK. We see that slices from 28th June to 1st July are better aligned to the template (27th June) in the y direction for ICP and PointNetLK. To give an additional numerical metric to interpret the visualizations, we compute the Chamfer Distance for each of the registered source point clouds after applying transformations obtained from the three algorithms. We compute an average of the chamfer distances over the 4 time slices (28th June to 1st July) seen in Table III. We observe that ICP gave the lowest average chamfer distance, followed by PointNetLK.



(a) Best registration case (28th June to 27th June) from ICP on the normalized point clouds



(b) Best registration case (29th June to 30th June) from Rigid CPD on the normalized point clouds



(c) Best registration case (29th June to 30th June) from PointNetLK on the normalized point clouds

Fig. 4: Point clouds of the best registration results for each of the methods - visualized with 50000 points and 3 views. Source - green, Template - Red and Registered point cloud - blue

TABLE III: MEAN CHAMFER DISTANCE OF THE SOURCE POINT CLOUDS TO THE TEMPLATE POINT CLOUD

Algorithm	Mean Chamfer Distance of templates from source
ICP	1.409388
CPD	1.954433
PointNet LK	1.479197

VII. DISCUSSION

Reduced translation error helps to accurately track changes on the surface of the rubble by eliminating artifacts caused by offsets. Therefore, for the purposes of post-deployment analysis like volumetric reconstruction or void detection, minimizing translation error takes precedence over minimizing rotation error. Hence, point-to-point ICP applied post unit cube normalization, with its low translation error and execution time is the preferred method for registering dense point clouds representing a dynamic SAR scene. This could be attributed to its ability to find good point correspondences despite changing geometry. However, its success is dependent on the error in the collected data being below a defined threshold. In our case, this error was 2.2 m. ICP works well when the errors in the reconstructed data are close to this value or lesser. Additionally, the chamfer distance for a static feature post registration with ICP was the lowest.

VIII. CONCLUSION

Data collection from UAS sorties during structural collapses can be deficient and cause various errors in the

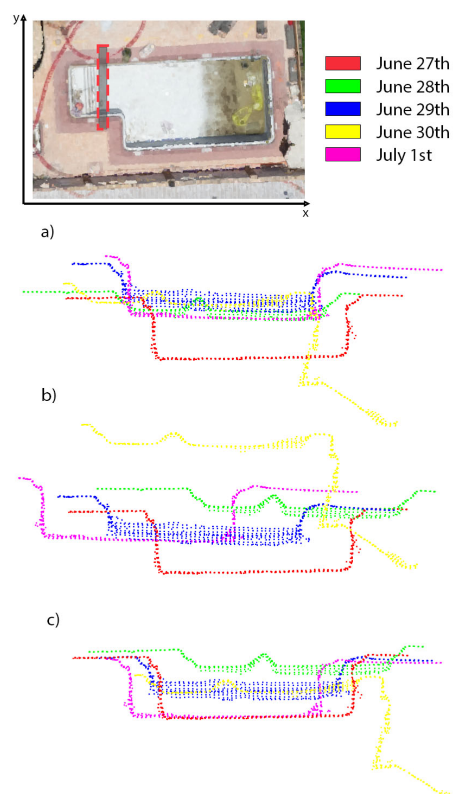


Fig. 5: Cross sections of a static feature within the point clouds (swimming pool). Top: Cross section area highlighted in red. Bottom: Cross section registration results for (a) ICP, (b) CPD and (c) PointNetLK along the y-axis.

reconstruction of 3D data. The point clouds reconstructed from the Surfside data have significant reconstruction errors due to numerous factors we have touched upon in this work, causing problems especially when trying to register those point clouds. Three registration methods were compared, and based on experiments conducted, point-to-point ICP, applied after unit cube normalization, performed best among the given methods for registering the dynamic data for this SAR operation. This holds for our data given the inherent offsets are below a certain threshold.

IX. FUTURE WORK

Even though the methods we have compared in this paper produce successful registrations according to commonly accepted thresholds, more work needs to be done to improve these methods for dynamic scenes such as the one we capture. This is because translation errors $\sim$0.8 m can affect volumetric analysis or void detection greatly as the 3D cross-sectional shapes indicating volume shifts or void spaces often have dimensions $\sim$1 m [32]. Keeping this in mind, if the performance of ICP has to be improved to reduce rotation error, CPD can be used in combination with ICP, perhaps applied to the subsampled source and template point clouds to estimate a better initialization for ICP. We intend to test this combination for our data in the future.

Given that PointNetLK has room for performance improvement if trained from scratch on domain-specific disaster data, doing so could prove critical in reducing translation errors further. We plan on testing PointNetLK's registration performance post training on more relevant data, especially dense point clouds representing urban landscapes. This will be very helpful when the reconstructed point clouds have more rotational and translational misalignment.

REFERENCES

[1] Smith, Erling A., and Howard I. Epstein. "Hartford Coliseum roof collapse: structural collapse sequence and lessons learned." Civil Engineering—ASCE 50, no. 4 (1980): 59-62

[2] Barbera, Joseph A., and Anthony Macintyre. "Urban search and rescue." Emergency Medicine Clinics 14, no. 2 (1996): 399-412.

[3] Linder, Thorsten, Viatcheslav Tretyakov, Sebastian Blumenthal, Peter Molitor, Dirk Holz, Robin Murphy, Satoshi Tadokoro, and Hartmut Surmann. "Rescue robots at the collapse of the municipal archive of cologne city: A field report." In 2010 ieee safety security and rescue robotics, pp. 1-6. IEEE, 2010.

[4] Murphy, Robin R., and Jennifer 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, pp. 437-441. Sage CA: Los Angeles, CA: SAGE Publications, 2005.

[5] K. Barnett, "Surfside condo collapse is third largest building failure in country's history," CBS Miami, 2021.

[6] Surmann, Hartmut, et al. "Deployment of Aerial Robots During the Flood Disaster in Erftstadt/Blessem in July 2021." 2022 8th International Conference on Automation, Robotics and Applications (ICARA). IEEE, 2022.

[7] Surmann, Hartmut, et al. "Deployment of Aerial Robots after a major fire of an industrial hall with hazardous substances, a report." 2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2021.

[8] Gupta, Ritwik, et al. "xbd: A dataset for assessing building damage from satellite imagery." arXiv preprint arXiv:1911.09296 (2019).

[9] Kruijff-Korbayová, Ivana, et al. "Deployment of ground and aerial robots in earthquake-struck amatrice in italy (brief report)." 2016 IEEE international symposium on safety, security, and rescue robotics (SSRR). IEEE, 2016.

[10] Zhang, Shiyong, et al. "Fast Active Aerial Exploration for Traversable Path Finding of Ground Robots in Unknown Environments." IEEE Transactions on Instrumentation and Measurement 71 (2022): 1-13.

[11] Roldán-Gómez, Juan Jesús, Eduardo González-Gironda, and Antonio Barrientos. "A survey on robotic technologies for forest firefighting: Applying drone swarms to improve firefighters' efficiency and safety." Applied Sciences 11.1 (2021): 363.

[12] Peppa, Maria V., et al. "Automated co-registration and calibration in SfM photogrammetry for landslide change detection." Earth Surface Processes and Landforms 44.1 (2019): 287-303.

[13] Wujanz, Daniel, Daniel Krueger, and Frank Neitzel. "Identification of stable areas in unreferenced laser scans for deformation measurement." The Photogrammetric Record 31.155 (2016): 261-280.

[14] Besl, Paul J., and Neil D. McKay. "Method for registration of 3-D shapes." Sensor fusion IV: control paradigms and data structures. Vol. 1611. Spie, 1992.

[15] Zhang, Zhengyou. "Iterative point matching for registration of free-form curves and surfaces." International journal of computer vision 13.2 (1994): 119-152.

[16] Fitzgibbon, Andrew W. "Robust registration of 2D and 3D point sets." Image and vision computing 21.13-14 (2003): 1145-1153.

[17] Rusinkiewicz, Szymon, and Marc Levoy. "Efficient variants of the ICP algorithm." Proceedings third international conference on 3-D digital imaging and modeling. IEEE, 2001.

[18] Myronenko, Andriy, and Xubo Song. "Point set registration: Coherent point drift." IEEE transactions on pattern analysis and machine intelligence 32.12 (2010): 2262-2275.

[19] Rangarajan1997, Anand, et al. "A robust point-matching algorithm for autoradiograph alignment." Medical image analysis 1.4 (1997): 379-398.

[20] Luo, Bin, and Edwin R. Hancock. "Structural graph matching using the EM algorithm and singular value decomposition." IEEE Transactions on Pattern Analysis and Machine Intelligence 23.10 (2001): 1120-1136.

[21] Gold, Steven, et al. "New algorithms for 2D and 3D point matching: pose estimation and correspondence." Pattern recognition 31.8 (1998): 1019-1031.

[22] Chui, Haili, and Anand Rangarajan. "A new point matching algorithm for non-rigid registration." Computer Vision and Image Understanding 89.2-3 (2003): 114-141.

[23] Chui, Haili, and Anand Rangarajan. "A feature registration framework using mixture models." Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. MMBIA-2000 (Cat. No. PR00737). IEEE, 2000.

[24] Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[25] Aoki, Yasuhiro, et al. "Pointnetlk: Robust & efficient point cloud registration using pointnet." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[26] Wang, Chaoyang, et al. "Deep-lk for efficient adaptive object tracking." 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018.

[27] Barbasiewicz, Adrianna, Tadeusz Widerski, and Karol Daliga. "The analysis of the accuracy of spatial models using photogrammetric software: Agisoft Photoscan and Pix4D." In E3S Web of Conferences, vol. 26, p. 00012. EDP Sciences, 2018.

[28] Sarode, Vinit, et al. "MaskNet: A fully-convolutional network to estimate inlier points." 2020 International Conference on 3D Vision (3DV). IEEE, 2020.

[29] Pan, Liang, Zhongang Cai, and Ziwei Liu. "Robust partial-to-partial point cloud registration in a full range." arXiv preprint arXiv: 2111.15606 (2021).

[30] Girardeau-Montaut, Daniel. "CloudCompare." France: EDF R&D Telecom ParisTech 11 (2016).

[31] Huang, Xiaoshui, et al. "A comprehensive survey on point cloud registration." arXiv preprint arXiv:2103.02690 (2021).

[32] Rao, Ananya, et al. "Analysis of Interior Rubble Void Spaces at Champlain Towers South Collapse." 2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2022.