Time-of-Flight Cameras in Space: Pose Estimation with Deep Learning Methodologies

Time-of-Flight Cameras in Space: Pose Estimation with Deep Learning Methodologies

Alkis Koudounas
Politecnico di Torino
Turin, Italy
alkis.koudounas@polito.it

Flavio Giobergia
Politecnico di Torino
Turin, Italy
flavio.giobergia@polito.it

Elena Baralis
Politecnico di Torino
Turin, Italy
elena.baralis@polito.it

***Abstract*—Recently introduced 3D Time-of-Flight (ToF) cameras have shown a huge potential for mobile robotic applications, proposing a smart and fast technology that outputs 3D point clouds, lacking however in measurement precision and robustness. With the development of this low-cost sensing hardware, 3D perception gathers more and more importance in robotics as well as in many other fields, and object registration continues to gain momentum. Registration is a transformation estimation problem between a source and a target point clouds, seeking to find the transformation that best aligns them. This work aims at building a full pipeline, from data acquisition to transformation identification, to robustly detect known objects observed by a ToF camera within a short range, estimating their 6 degrees of freedom position. We focus this work to demonstrating the capability of detecting a part of a satellite floating in space, to support in-orbit servicing missions (e.g. for space debris removal). Experiments reveal that deep learning techniques can obtain higher accuracy and robustness w.r.t. classical methods, handling significant amount of noise while still keeping real-time performance and low complexity of the models themselves.**

***Index Terms*—tof cameras, point cloud registration, deep learning**

I. Introduction

Time-of-Flight (ToF) cameras can produce 3D images by using the ToF principle: the distance between each point in the image and the camera is computed measuring the phase shift that occurs between an emitted signal and the signal that returns after it bounces on the target object (Figure 1 provides a visual aid for the ToF principle). The output of these cameras are clouds of points: these are a representation of the scene, as captured by the camera.

ToF cameras are cheap and are characterized by a high frame rate, a low weight and a small size: all characteristics that make them suitable for autonomous robotics tasks [1]. One advantage of their usage is the complete removal of the typical stereo vision pipeline since they output 3D point clouds. However, these cameras are extremely vulnerable to changes in lighting conditions: this inevitably results in inaccurate data and erroneous raw measurements [2]. In this work, we focus on the point cloud registration problem, the task of identifying a transformation that aligns a source point cloud (e.g. the acquisition of a ToF camera) to a target (e.g. a known orientation of the object being observed). This kind of task is particularly useful when an object of interest is placed in an unknown orientation. We conducted this work in collaboration with a large European aerospace company. As such, our case study focuses on the identification of the orientation of a satellite floating in space: in the context of a space debris [3] removal scenario, knowing the orientation of the pieces to be extracted is fundamental. Given the noisy nature of ToF camera images, we additionally covered the denoising of the images themselves. We first explore the state-of-the-art techniques available for both denoising and point cloud registration and we compare them according to six indicators. We identify the most promising approaches (both traditional and learning-based) and we build an end-to-end pipeline, from data acquisition to transformation estimation. Extensive experiments show that the proposed framework offers, along with increasingly precise ToF cameras, an interesting new range of possibilities for robotic sensing and, in particular, in aerospace.

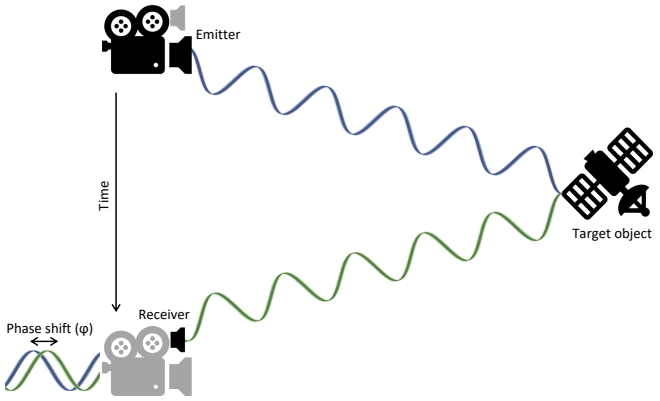

Fig. 1. ToF principle for measuring the distance of a target object: for a signal with modulation frequency f_m, the distance can be computed as $\frac{c\varphi}{4\pi f_m}$, c being the speed of light.

II. Related Works

As the authors of [2] illustrate, the potential of ToF cameras is limited by several errors:

- *Systematic Errors* (distance-related errors, amplitude-related errors, fixed pattern phase noise), which are predictable and manageable by calibration.
- *Non-Systematic Errors* (bad signal-to-noise ratio, Multiple Path Interference (MPI), light scattering), that are in-

stead inherently dependent on the measurement principle, thus they are much more difficult to correct.

To address the systematic errors, a calibration of the camera is needed. Specifically, a *lateral calibration* approach that computes the internal and the external parameters for the camera taken into account and a *depth calibration* technique to also calibrate the information concerning the distance.

Regarding the non-systematic errors, they have been historically tackled in a pipeline architecture where each step disentangles an individual sub-problem alone, resulting in cumulative error and information loss [4]. This is the reason why, recently, instead of building a reconstruction pipeline or depending on auxiliary hardware, new data-driven approaches were introduced [4]–[13]. These models leverage neural networks to generate a point cloud directly from the raw modulated exposures of the ToF camera.

Point clouds have recently become more and more employed as a way of representing the 3D world, as thoroughly explained in [14]. This is particularly true if one considers the accelerated growth of high precision sensors, i.e., LiDAR, Kinect and ToF cameras. The main limitation of these devices is the narrow sight range at which they can observe a scene. This is the reason why registration schemes are becoming of paramount importance to build and extract wider 3D real world sections. The point cloud registration problem requires estimating the transformation matrix between two point clouds. Through this transformation matrix, it is possible to obtain a full 3D point cloud by incorporating several incomplete scans of the same scene. Point cloud registration has been a contributing factor in various computer vision tasks, such as 3D reconstruction, 3D localization and pose estimation. According to the comprehensive survey in [14], these methods can be divided into three main categories:

- *optimization-based* methods are based on optimization policies to search for the correspondences and estimating the transformation, e.g. ICP [15] and derived [16] [17] and, more recently, FGR [18]
- *feature-learning* methods use learning methodologies to learn feature representations, that are then used for the estimation of the transformation matrix, as shown in PPF-FoldNet [19], IDAM [20], DCP [21] and FRR (FPFH [22] + RANSAC [23])
- *end-to-end learning-based* methods instead embed the transformation problem into the neural network model and already provide it as the output, instead of producing features to be used in another way. Examples are PointNetLK [24], PointVoteNet [25], DGR [26], 3DRegNet [27] and FMR [28]

We evaluated these techniques, based on the claims made in the respective papers, according to six indicators we identified. In particular, we considered the models' *accuracy* (i.e. quality of the results), *model size*, *robustness* to input noise, *training cost*, *latency* of predictions (given that they should work in real time) and *range of applications* (i.e. how well the model generalizes to new types of data). We evaluate each algorithm with a ranking from "A" (best) to "D" (worst). We report the results in Table I.

TABLE I
COMPARISON OF SEVERAL MODELS FOR TOF DENOISING AND POINT CLOUD REGISTRATION, ACCORDING TO SIX DIFFERENT INDICATORS. IN BOLD ARE THE METHODS THAT WE CONSIDER TO BE MORE PROMISING, AMONG THE RESPECTIVE GROUPS.

	Method	Accuracy	Size of Model	Robustness	Time Cost	Latency	Range of Application
ToF Data Denoising	ToFNet [4]	D	C	C	D	C	C
	MOM+MRN DNN [6]	B	C	B	C	B	D
	DeepToF [5]	C	D	C	B	A	B
	Gupta and Xu, 2019 [7]	C	A	C	A	A	C
	Coarse-Fine CNN [8]	**B**	**A**	**B**	**A**	**A**	**B**
	Son et al., 2016 [9]	C	A	C	A	A	C
	Chen et al., 2020 [10]	B	C	B	A	A	C
	Buratto et al., 2021 [11]	B	A	B	C	B	B
	SHARP-Net [12]	A	B	B	B	B	A
	DA-F [13]	**A**	**B**	**A**	**B**	**B**	**A**
Point Cloud Registration	**Optimization-Based Methods**						
	ICP [15]	D	A	C	A	B	C
	Go-ICP [16]	C	B	B	B	D	B
	LM-ICP [17]	C	B	B	A	A	B
	FGR [18]	**B**	**B**	**A**	**A**	**A**	**B**
	Feature-Learning Methods						
	PPF-FoldNet [19]	B	C	A	B	B	A
	IDAM [20]	B	C	B	C	A	B
	DCP [21]	C	D	B	B	A	B
	FRR [22] [23]	**B**	**A**	**B**	**B**	**B**	**A**
	End-to-End Learning-Based Methods						
	PointNetLK [24]	D	B	A	C	A	B
	PointNetLK + Awe-Net [29]	C	C	A	B	B	A
	PointVoteNet [25]	B	C	B	C	C	B
	DGR [26]	A	D	B	C	A	B
	3DRegNet [27]	A	B	A	C	A	A
	FMR [28]	**A**	**B**	**A**	**B**	**A**	**A**

III. DATA ACQUISITION

We are interested in simulating a plausible situation where a robotic arm is used for the collection of debris in space: because of this, for the data collection, we used as the target object a 3D printed model of a satellite. The model has been printed, sanded and painted white to reduce its reflectivity, since all ToF sensors behave poorly with highly reflective objects. For the data acquisition, the camera and the model have been placed in a dark room. The model was hung from the ceiling, at a nominal distance of 1 m from the sensor.

We started with the acquisition of data from a ToF camera. We recorded the amplitude (grayscale) image and the point cloud (depth image) of each scene, collected at the same sampling frequency. Specifically, the operating modes provided by the camera are 5, 10, 15, 25, 35 and 45 frames per second (fps), respectively. A higher fps implies a faster data acquisition, which comes at the cost of a higher noise. For a rate > 25 fps we qualitatively observed that the satellite model could not be recognized. Because of this, in this work we only considered input data with a low frame rate (up to 25 fps).

IV. METHODOLOGY

Figure 2 summarizes the pipeline that has been applied in this work. Each of the blocks that characterizes the pipeline is described in further detail in this section.

A. Raw data denoising

The comparison in Table I shows that Coarse-Fine CNN approaches [8] [13] are the most promising. However, those approaches require acquiring multiple simultaneous modulation frequencies. We are instead interested in a single modulation

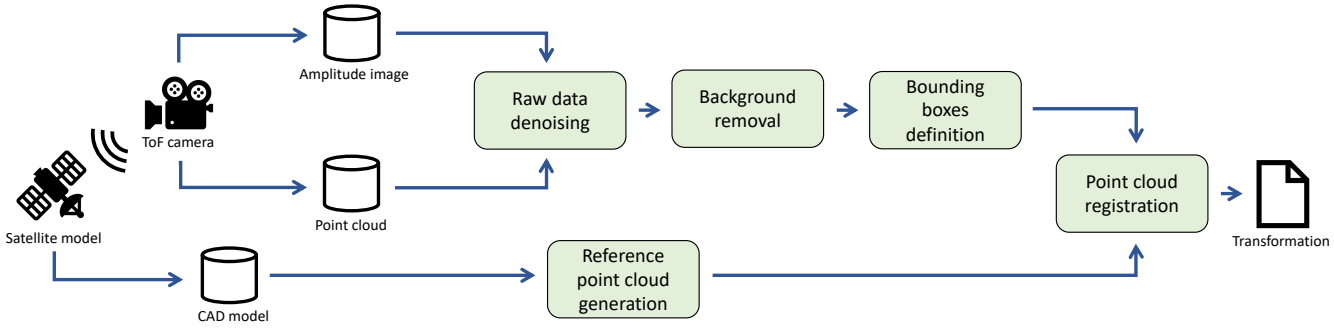

Fig. 2. Pipeline adopted for this work, from data acquisition to estimation of the transformation

frequency approach, because of its lower cost. Based on this requirement, we instead identify as the best candidate SHARP-Net (Spatial Hierarchy Aware Residual Pyramid Network) [12], a model that uses 3 blocks: the Residual Regression Module, the Residual Fusion Module and the Depth Refinement Module.

We study two additional methodologies for comparison, ToF-KPN [30] (a U-Net-based approach) and a version of SHARP-Net without the Residual Fusion Module and Depth Refinement (WOFusRef in [12]), with some hyperparameter tuning.

B. Reference point cloud generation

Since the point cloud registration algorithms require a reference point cloud to be able to perform the alignment, we extracted a point cloud from the satellite model's CAD file through Yuksel's procedure [31]. This approximately guarantees the same distance between a point and each of its neighbors. As detailed in the experimental section, we studied scenarios with a varying number of target points.

C. Background removal

The images acquired during the data collection phase contain the background information of the room that has hosted the experiment. This is a limitation that comes with not running the data collection in space. To remove the background, we used a functionality offered by the Open3D library [32] that detects the object of interest in a point cloud and cuts off all the background points accordingly.

D. Bounding boxes definition

Before the point cloud registration, we applied object detection on the processed point clouds to identify the bounding boxes that contain the satellite model. A first bounding box whose edges are parallel to the axes is identified. From this, we extract an oriented bounding box.

E. Point cloud registration

Based on the considerations summed up in Table I, we identified the best-performing methodology for each of the three categories of approaches for point cloud registration. In particular, we selected Fast Global Registration [18] (FGR), FRR [22] [23] and Feature-Metric Registration [28] (FMR).

- FGR [18] uses FPFH (Fast Point Feature Histogram) [22] as a feature extraction step, to then compute (only once) the correspondences between source and target and optimize a robust objective to align the two point clouds.
- FRR leverages FPFH [22] features to then apply the RANSAC [23] global algorithm based on feature matching to align the source to the target. A final "Point-to-Point" ICP (Iterative Closest Point) [15] is optionally employed to refine the estimated transformation.
- FMR [28] is based on a simple autoencoder, trained in either semi-supervised or unsupervised manner, in which the encoder extracts rotation-attentive features while the decoder tracks down the original 3D point clouds. The registration problem is tackled by using the inverse compositional (IC) algorithm to minimize a feature-metric projection error and predict the final transformation.

V. Experimental results

In this section we compare the performance of the various approaches under study. In particular, Subsection V-A refers to the denoising approaches, whereas Subsection V-B focuses on the point cloud registration frameworks. The source code with the details about the configurations and the experiments run is available on the online repository[1].

A. Raw data denoising results

We compare the performance of all 3 denoising algorithms across 4 different frame rates (5, 10, 15, 25 fps).

We evaluate the performance by computing the percentage reduction in the number of outliers of the denoised point cloud with respect to the input one. Inliers are all the points of the point cloud belonging to the satellite model, while outliers refer to all the noisy points, either belonging to the background or to objects other than the satellite.

We introduce the Relative Outliers (RO) index to measure the fraction of outliers for the p^{th} input point cloud as follows:

$$RO_p^{(in)} = \frac{N_{OL}}{N_{IL} + N_{OL}} \tag{1}$$

[1]https://github.com/koudounasalkis/Time-of-Flight-Cameras-in-Space

TABLE II
PERFORMANCE EVALUATION OF DENOISING APPROACHES IN TERMS OF $M\Delta_{RO}$ (HIGHER IS BETTER). THE VALUES ARE COMPUTED AS THE AVERAGE ACROSS $N = 5$ DIFFERENT INPUTS ACQUIRED FOR EACH FRAME RATE.

Model	Frame Rate for Depth Input			
	5 fps	10 fps	15 fps	25 fps
ToF-KPN	-0.4832	-0.4485	-0.2246	-0.1387
WOFusRef	**0.0208**	0.0283	0.007	**0.0476**
SHARP-Net	0.0194	**0.0331**	**0.0656**	0.0354

TABLE III
EXECUTION TIME OF DENOISING APPROACHES. THE VALUES ARE COMPUTED AS THE AVERAGE ACROSS $N = 5$ DIFFERENT INPUTS ACQUIRED FOR EACH FRAME RATE.

Model	Frame Rate for Depth Input			
	5 fps	10 fps	15 fps	25 fps
ToF-KPN	**0.375 s**	**0.380 s**	**0.374 s**	**0.369 s**
WOFusRef	0.570 s	0.579 s	0.590 s	0.581 s
SHARP-Net	0.915 s	0.918 s	0.922 s	0.919 s

where N_{IL} refers to the number of inliers points belonging to the volume of the detected satellite, while N_{OL} refers to the number of outliers. We can similarly define $RO_p^{(out)}$, the Relative Outliers for the output (i.e., denoised) point cloud.

For a collection of N point clouds, we define the $MRO^{(in)}$ and the $MRO^{(out)}$ respectively as the mean $RO^{(in)}$ and $RO^{(out)}$ over the entire collection.

To evaluate the quality of a denoising process, we consider the mean decrease in Relative Outliers of the output, w.r.t. the input:

$$M\Delta_{RO} = \frac{1}{N}\sum_{i=1}^{N}(RO_i^{(in)} - RO_i^{(out)}) \tag{2}$$

In this way, each point cloud contributes positively to $M\Delta_{RO}$ if its input (i.e., the noisy version) contains more outliers than the output (i.e., the denoised version). It contributes negatively otherwise. Based on the provided definitions, it follows that:

$$M\Delta_{RO} = MRO^{(in)} - MRO^{(out)} \tag{3}$$

Table II presents the performance achieved by each model in terms of $M\Delta_{RO}$. SHARP-Net and its variation WOFusRef have similar performance. Instead, ToF-KPN shows a significantly less performing behavior, removing both inliers and outliers, compromising the shape of the satellite itself. We attribute this to two factors: first, ToF-KPN is based on a simpler model w.r.t. SHARP-Net and second, the distribution from which the training set has been drawn to train ToF-KPN differs from the distribution from which we sampled the data with the ToF camera.

The similar performance obtained by the WOFusRef version of SHARP-Net may be explained by the selected camera model. The ToF camera we used is not affected by the wiggling phenomenon that is common to many other ToF sensors. As such, it is more resistant to shot and especially MPI noise. The Depth Refinement module of SHARP-Net may therefore be removed with no significant drop in performance.

Instead, Table III shows the execution times of the various algorithms. ToF-KPN is able to achieve the fastest execution, while SHARP-Net is the slowest among the the three models. This is in accordance with the expectations, given the previous discussion on the complexity of the three models.

B. Point cloud registration results

In this subsection, we compare the 3 point cloud registration frameworks identified, FGR, FRR with and without ICP (FRRwICP and FRRwoICP, respectively) and FMR. We use 4 different target point cloud dimensions (20k, 30k, 50k and 100k points respectively), while the input is always captured at 5 fps, since this frame rate produces the cleanest acquisitions (higher sampling frequencies produce noisier results).

We consider three different starting poses, or "alignments": the first one consists of two $90°$ rotations along two different axes, while the second and the third alignments are instead smaller rotations along all three axes. We intuitively expect the first alignment to be an easier task w.r.t. the others, since it only requires performing rotations along two axes, while the third one is left unchanged. Figure 3 qualitatively shows some alignments performed by the considered methods. When considering a simple scenario (Figure 3 (a)-(d)), all the approaches are able to correctly align the two point clouds, both for a low and high number of points in the target point cloud.

We explore two other starting poses in Figures 3 ((q)-(t) and (G)-(J)). In these cases, FMR still qualitatively succeeds in all alignment tasks. On the other hand, both FGR and FRRwICP struggle with some of the reconstructions. We can once again observe that there is no discernible trend with the size of the target point cloud.

On top of the qualitative results, we also perform a quantitative comparison of the models' performance. Inspired by the indicators already used in [32], we use the Hit-Rate (HR) and the RMSE as evaluation metrics.

The Hit-Rate represents the fraction of correctly aligned points over all points in the source point cloud:

$$HR = \frac{|\{p : p \in P^{(src)} \wedge f(p) \in P^{(tgt)}\}|}{|P^{(src)}|} \tag{4}$$

Where $P^{(src)}$ and $P^{(tgt)}$ represent the source and target point clouds respectively, while f is the learned transformation.

The RMSE instead represents the root mean squared error of all the retrieved inlier correspondences and is defined as:

$$RMSE = \sqrt{\frac{1}{|\mathcal{C}|}\sum_{p,q\in\mathcal{C}} ||f(p) - q||_2^2} \tag{5}$$

where $\mathcal{C}$ is the set of pairs of corresponding points.

Table IV summarizes the results in terms of these metrics for FGR, FRR and FMR. The results reported here depend both on the size of the target point cloud and on the initial

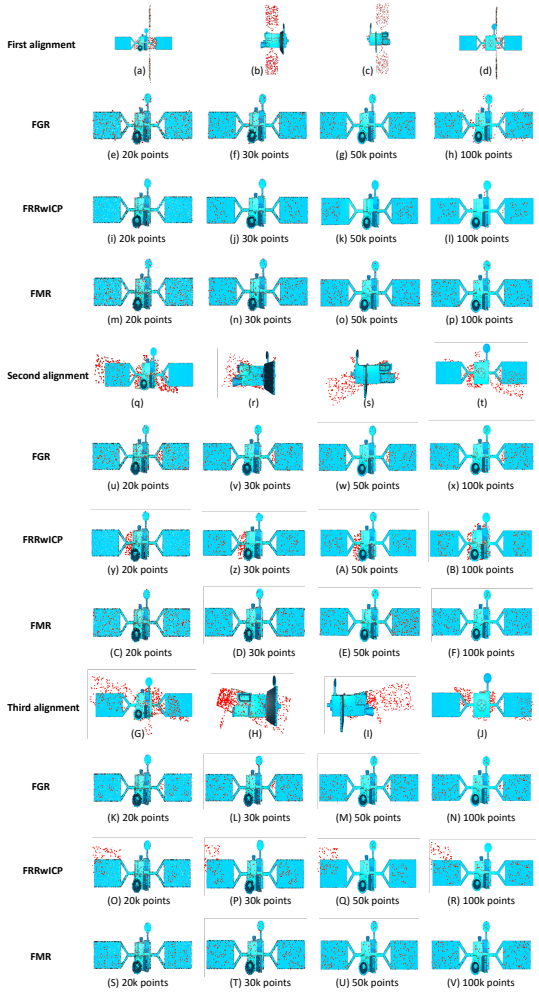

Fig. 3. Point clouds alignment of the approaches under study, considering different source poses and target sizes. In blue is the target object, in red is the source point cloud. Figures (a)-(d), (q)-(t) and (G)-(J) show the three alignments, from different angles. The other figures represent the alignments obtained with the various approaches, for different target point cloud sizes.

TABLE IV

PERFORMANCE EVALUATION OF PCR APPROACHES, IN TERMS OF HIT-RATE (HIGHER IS BETTER) AND RMSE (LOWER IS BETTER) METRICS. THE VALUES ARE COMPUTED AS THE AVERAGE ACROSS $N = 5$ DIFFERENT INPUTS ACQUIRED FOR EACH TARGET POINT CLOUD SIZE.

Performance Metrics			No. of points in target point cloud			
			20k	30k	50k	100k
First Alignment	HR	FGR	0.9621	0.9630	0.9539	0.9534
		FRRwoICP	0.9214	0.9231	0.9220	0.9295
		FRRwICP	0.9531	0.9534	0.9634	0.9644
		FMR	**0.9705**	**0.9706**	**0.9708**	**0.9708**
	RMSE	FGR	0.0106	0.0106	0.0106	0.0105
		FRRwoICP	0.0153	0.0168	0.0159	0.0151
		FRRwICP	0.0144	0.0155	0.0151	0.0142
		FMR	**0.0101**	**0.0100**	**0.0100**	**0.0099**
Second Alignment	HR	FGR	0.9244	0.9212	0.9239	0.9221
		FRRwoICP	0.7679	0.7601	0.7690	0.7681
		FRRwICP	0.8111	0.8107	0.8103	0.8109
		FMR	**0.9704**	**0.9705**	**0.9705**	**0.9703**
	RMSE	FGR	0.0203	0.0204	0.0204	0.0204
		FRRwoICP	0.0391	0.0390	0.0394	0.0389
		FRRwICP	0.0281	0.0286	0.0283	0.0284
		FMR	**0.0102**	**0.0101**	**0.0102**	**0.0102**
Third Alignment	HR	FGR	0.9312	0.9308	0.9310	0.9311
		FRRwoICP	0.8361	0.8314	0.8344	0.8359
		FRRwICP	0.8801	0.8814	0.8810	0.8799
		FMR	**0.9721**	**0.9722**	**0.9722**	**0.9722**
	RMSE	FGR	0.0182	0.0185	0.0184	0.0184
		FRRwoICP	0.0272	0.0281	0.0279	0.0277
		FRRwICP	0.0244	0.0251	0.0239	0.0257
		FMR	**0.0098**	**0.0098**	**0.0099**	**0.0097**

TABLE V

EXECUTION TIME OF THE POINT CLOUD REGISTRATION APPROACHES UNDER STUDY. THE VALUES ARE COMPUTED AS THE AVERAGE ACROSS $N = 5$ DIFFERENT INPUTS ACQUIRED FOR EACH TARGET POINT CLOUD SIZE.

Model	No. of points in target point cloud			
	20k	30k	50k	100k
FGR	0.431 s	0.748 s	**1.997 s**	5.441 s
FRRwoICP	16.840 s	21.121 s	24.968 s	29.914 s
FRRwICP	17.012 s	21.511 s	25.098 s	30.115 s
FMR	**0.392 s**	**0.694 s**	2.130 s	**5.012 s**

alignment between the source and the target. In particular, we consider 4 possible target sizes as previously mentioned, as well as 3 starting poses for the source point cloud's initial position, as shown in Figure 3.

We observe that FMR consistently outperforms the other approaches. Additionally, all approaches are not significantly affected by the change in target point cloud dimension.

Interestingly, we note that FMR is also the most consistent of the 4 studied approaches: the performance across the 3 alignments is approximately the same. For the other methodologies, instead, there is a high variability. For example, the other models perform better on the first alignment (a simpler task) than on the second and third ones.

Additionally, Table V shows the execution times for the various algorithms. FMR is faster in terms of execution w.r.t. the other algorithms in most situations. FGR closely follows, with times that are comparable and – in one case – better. FRR with and without ICP is almost two orders of magnitude slower.

In conclusion, FMR is not only the fastest of the three algorithms, it is also the one to achieve the best performance. As authors of [28] suggest, this could be explained by considering the strength of the unsupervised part of the framework that offers a feature extraction network which is truly capable of embedding peculiar features in order to intrinsically understand the point cloud geometry. Thus, the FMR approach provides an effective and efficient way to tackle the 3D point cloud registration problem, with an overall limited complexity and, most importantly, the capability of working in (near-)real-time, especially when the point clouds have a limited amount of points, as is the case with the ones produced by the ToF camera we used.

VI. CONCLUSIONS

In this work we addressed the problem of Object Detection and 6 DoF pose estimation with input data acquired with a

ToF camera through the usage of deep learning and traditional approaches. We compared 10 methods for the ToF denoising task and 14 for the point cloud registration problem based on six different indicators relevant for robotics-based applications.

We have built an end-to-end framework which takes as input the raw noisy data captured from a ToF camera, denoises it through the usage of SHARP-Net (and variants), and finally returns as output, by applying the learning-based FMR approach (or one of the mentioned alternatives), the rigid transformation that is able to best align the source denoised point cloud to a target reference.

We have worked with a relatively new ToF sensor, fully exploiting its potential and being, to the best of our knowledge, the first to apply this kind of device to a point cloud registration task for a space-based application.

We conducted extensive experiments by taking into account different frame rates for the acquisition, different starting poses of the source point cloud and different dimensionalities of the target one. For the cloud point registration problem, we have shown that the deep learning approaches not only outperform classical methods in terms of performance, but also in terms of execution time.

The main focus of the future works concerns the acquisition of additional data: all models adopted so far have been pre-trained on separate datasets and only used for inference. Despite the satisfactory results, we expect that the availability of a larger dataset for training (or fine-tuning) will yield even better results.

VII. Acknowledgements

This work has been partially supported by the Smart-Data@PoliTO center on Big Data and Data Science.

References

[1] S. May, B. Werner, H. Surmann, and K. Pervolz, "3d time-of-flight cameras for mobile robotics," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 790–795.

[2] S. May, D. Droeschel, S. Fuchs, D. Holz, and A. Nüchter, "Robust 3d-mapping with time-of-flight cameras," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1673–1678.

[3] H. Klinkrad, *Space debris: models and risk analysis*. Berlin, Germany: Springer Science & Business Media, 2006.

[4] S. Su, F. Heide, G. Wetzstein, and W. Heidrich, "Deep end-to-end time-of-flight imaging," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6383–6392.

[5] J. Marco, Q. Hernandez, A. Munoz, Y. Dong, A. Jarabo, M. H. Kim, X. Tong, and D. Gutierrez, "Deeptof: off-the-shelf real-time correction of multipath interference in time-of-flight imaging," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 6, pp. 1–12, 2017.

[6] Q. Guo, I. Frosio, O. Gallo, T. Zickler, and J. Kautz, "Tackling 3d tof artifacts through learning and the flat dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 368–383.

[7] K. Gupta and Y. Xu, "Denoising 3d time-of-flight data," *Accessed: Jun*, vol. 15, p. 2021, 2017.

[8] G. Agresti and P. Zanuttigh, "Deep learning for multi-path error removal in tof sensors," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds. Cham: Springer International Publishing, 2019, pp. 410–426.

[9] K. Son, M.-Y. Liu, and Y. Taguchi, "Learning to remove multipath distortions in time-of-flight range images for a robotic arm setup," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3390–3397.

[10] Y. Chen, J. Ren, X. Cheng, K. Qian, L. Wang, and J. Gu, "Very power efficient neural time-of-flight," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 2246–2255.

[11] E. Buratto, A. Simonetto, G. Agresti, H. Schäfer, and P. Zanuttigh, "Deep learning for transient image reconstruction from tof data," *Sensors*, vol. 21, no. 6, p. 1962, 2021.

[12] G. Dong, Y. Zhang, and Z. Xiong, "Spatial hierarchy aware residual pyramid network for time-of-flight depth denoising," in *European Conference on Computer Vision*. Springer, 2020, pp. 35–50.

[13] G. Agresti, H. Schaefer, P. Sartor, and P. Zanuttigh, "Unsupervised domain adaptation for tof data denoising with adversarial learning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5579–5586.

[14] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," *arXiv preprint arXiv:2103.02690*, 2021.

[15] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239–256, 1992.

[16] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2241–2254, 2015.

[17] A. W. Fitzgibbon, "Robust registration of 2d and 3d point sets," *Image and vision computing*, vol. 21, no. 13-14, pp. 1145–1153, 2003.

[18] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European conference on computer vision*. Springer, 2016, pp. 766–782.

[19] H. Deng, T. Birdal, and S. Ilic, "Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 620–638.

[20] J. Li, C. Zhang, Z. Xu, H. Zhou, and C. Zhang, "Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration," in *European conference on computer vision*. Springer, 2020, pp. 378–394.

[21] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3523–3532.

[22] R. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," *2009 IEEE International Conference on Robotics and Automation*, pp. 3212–3217, 2009.

[23] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, 1981.

[24] 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.

[25] F. Hagelskjær and A. G. Buch, "Pointvotenet: Accurate object detection and 6 dof pose estimation in point clouds," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 2641–2645.

[26] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2514–2523.

[27] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3dregnet: A deep neural network for 3d point registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7193–7203.

[28] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 366–11 374.

[29] A. E. Khazari, Y. Que, T. L. Sung, and H. J. Lee, "Deep global features for point cloud alignment," *Sensors*, vol. 20, no. 14, p. 4032, 2020.

[30] D. Qiu, J. Pang, W. Sun, and C. Yang, "Deep end-to-end alignment and refinement for time-of-flight rgb-d module," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9994–10 003.

[31] C. Yuksel, "Sample elimination for generating poisson disk sample sets," in *Computer Graphics Forum*, vol. 34, no. 2. Wiley Online Library, 2015, pp. 25–32.

[32] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.