

Crossmodal Point Cloud Registration in the Hough Space for Mobile Laser Scanning Data

Bence Gálai¹, Balázs Nagy^{1,2} and Csaba Benedek^{1,2}

¹Distributed Events Analysis Research Laboratory, Institute for Computer Science and Control
H-1111 Kende u. 13-17 Budapest, Hungary, Email: lastname.firstname@sztaki.mta.hu

²Péter Pázmány Catholic University, H-1083, Práter utca 50/A, Budapest, Hungary

Abstract—In this paper we propose a general approach for registration of point clouds obtained by various mobile laser scanning technologies. Our method is able to robustly match measurements with significantly different density characteristic including the sparse and inhomogeneous instant 3D (I3D) data taken by self-driving cars, and the dense and regular point clouds captured by mobile mapping systems (MMS) for virtual city generation. The core steps of the algorithm are robust scan segmentation, abstract street object extraction, object based coarse transformation estimation in the Hough accumulator space, and point-level registration refinement. Experimental results are provided using three different sensors: Velodyne HDL64 and VLP16 I3D scanners, and a Riegl VMX450 MMS. Application examples are shown regarding self localization of autonomous cars through crossmodal I3D and MMS frame registration, IMU-less SLAM and change detection based on I3D data.

I. INTRODUCTION

Mobile vehicle mountable *Lidar* laser scanners are gaining ground in various application fields, since they are able to provide accurate 3D measurements from the surrounding environment with a very high acquisition speed. Automatic point cloud registration is a key step in many applications such as simultaneous localization and mapping (SLAM) and mobile surveillance, especially if precise position information of the acquisition platform is not available due to lack of accurate navigation signals. However answering different functional requirements and due to the manufacturer's unique innovative approaches the available sensors may provide point clouds with very different density characteristics [1], limiting the general usability of standard point cloud registration techniques [2], [3], or methods developed for specific sensors [4].

Autonomous vehicles (AV) demand instant 3D (I3D) data acquisition and processing techniques operating onboard on mobile platforms. Since the decision of the navigation or control unit of a AV must be instantly made depending on the actually available environmental information actual full-view point cloud frames must be generated with 10-15fps, and the size of the transferable data should also be limited for enabling real time processing. As a consequence the measurements have usually a low spatial density, which quickly decreases as a function of the distance from the sensor, and the point clouds may exhibit particular patterns typical to sensor characteristic, such as the ring patterns of the Velodyne sensor (see Fig. 1(a) and 1(b)). Although the 3D measurements are quite accurate (up to few *cms*) in the sensor's local coordinate system, the

global positioning error of the vehicles may reach several meters in city regions with poor GPS signal coverage.

New generation Geo-Information Systems (GIS) used by city administrations maintain extremely detailed 3D point cloud maps of the cities for road network management, surveillance and urban planning applications. Recent Mobile Mapping Systems (MMS), such as the Riegl VMX450 (Fig. 1(c)) are able to provide dense and accurate point clouds from the environment with homogeneous scanning of the surfaces and a nearly linear increase of points as a function of the distance [1]. The point density of MMS point clouds is with 2-3 orders of magnitude higher than the density of I3D scans.

In this paper, we propose a new general point cloud registration method for mobile laser scanning applications, focusing on challenging scenarios where one or both point clouds are notably sparse having specific or irregular density characteristics, or they may originate from different I3D or MMS sensors. In particular, our algorithm will give solutions for the following three problems:

- 1) *Crossmodal I3D and MMS point cloud registration*: AVs could fully take the advantage of using a global and detailed 3D map of the city for accurate self localization, route planning and high level scene understanding. This process needs the quick and precise registration of the sparse I3D data to the dense MMS point clouds.
- 2) *IMU-free SLAM based on sparse I3D data*: Without reliable assistance of external navigation sensors such as Global Positioning Systems (GPS) or Inertial Measurement Units (IMU), SLAM based on I3D Lidars suffers from challenges of automated matching inhomogeneous and low density point clouds.
- 3) *Registration of clouds from different I3D sources*: Registering scans from different AVs which may carry different types of I3D Lidar sensors opens prospects in change detection, dynamic environment surveillance and co-localization of communicating vehicles. However, this process needs matching scans which are both sparse and also exhibit different distance-density characteristic.

For experimental evaluation of the algorithms, we used I3D data by our institute's Velodyne HDL64E (Fig. 1(a)) and VLP16 (Fig. 1(b)) rotating multi-beam Lidar sensors, and MMS point clouds captured with a Riegl VMX450 system (Fig. 1(c)) in urban roads under ordinary traffic. As shown in

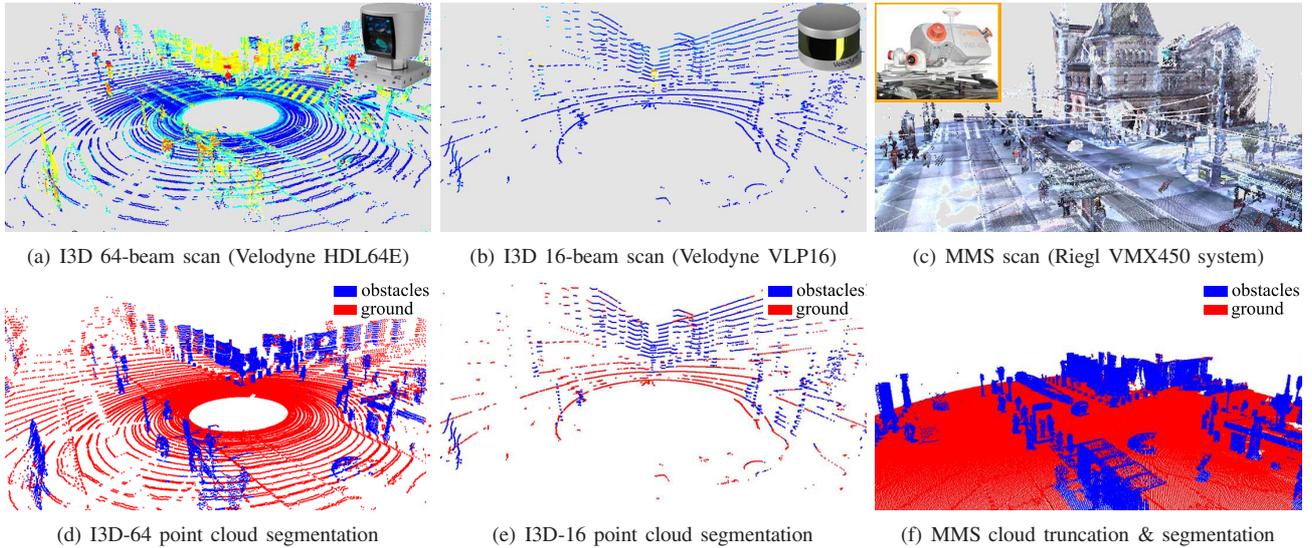


Fig. 1. *Top row*: Point clouds of three different vehicle mounted Lidar systems, captured from the same scene at Fővám tér, Budapest. *Bottom row*: segmentation results for each cloud by our proposed method presented in Sec. III-A

Fig 1(a) and (b), HDL64E and VLP16 records similar “ring patterned” instant 3D data, however, VLP16 - working with 16 laser beams in contrast to the HDL’s 64 scanning units - provides four times less points in a full rotation, and the two sensors also have different firing angles.

II. PREVIOUS WORK

Although various established techniques do exist for point cloud registration, such as Iterative Closest Point (ICP) [2] and Normal Distribution Transform (NDT) [3], these methods may fail if the two point clouds are not pre-aligned somehow, and particular density characteristics may also mislead the matching step [4]. Various improvements have been published for the ICP algorithm, such considering color information [5], or geometrical features from the point neighborhoods [6]. However, these approaches still have large computational cost. Other techniques do not improve the core of the ICP algorithm itself, but use trajectory information for correcting ICP errors [7]. An enhanced ICP algorithm is introduced in [8], which uses a point cloud segmentation step [9], then the nearest neighbor search of ICP is done only in associated segments across scans. Non-ICP-based approaches have also been proposed e.g. [10] which exploits the nature of a rotating multi-beam Lidar (such as the Velodyne sensors) for plane detection, and applies real-time registration of the extracted planes. Although this method could lead to real-time SLAM, we must note that in many real world scenarios the plane detection step may mean a significant bottleneck of the process.

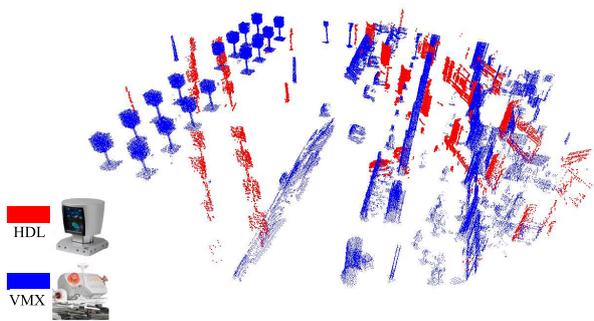
III. PROPOSED POINT CLOUD REGISTRATION ALGORITHM

In this section we introduce a new registration algorithm for point cloud pairs captured with possibly different I3D and MMS sensors. Let us assume that the translational component of the frame displacement between the two clouds is

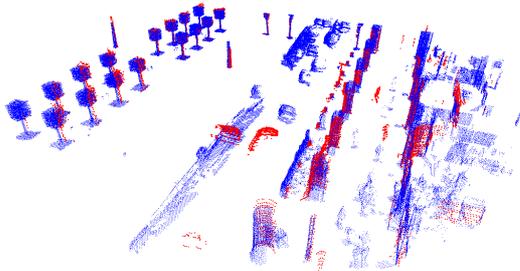
below 10m due to the availability of coarse self localization information, but the orientation difference may be arbitrarily large, as demonstrated in Fig. 2. Our algorithm consists of four major steps: point cloud segmentation, abstract field object extraction, object based coarse transformation estimation, and accurate registration refinement.

A. Point cloud segmentation

Point clouds captured by Rotating Multi-beam Lidar sensors contain characteristic ring patterns on the ground (see Fig. 1(a) and 1(b)), which is largely disadvantageous for standard point cloud registration algorithms. Our experiences [4] show that applying ICP or NDT on the raw Velodyne frames often yield erroneous registration where the concentric rings are aligned to each other, instead of finding the match between the structural scene elements. On the other hand, by aligning Velodyne to MMS scans the poor matching rate between the terrain regions can also mislead the registration process. For these reason, we start the procedure with ground removal: we apply a locally adaptive terrain modeling approach similarly to [4], which is able to accurately extract the road regions, even if their surfaces are not perfectly planar. First we fit a regular 2D grid with fixed rectangle side length onto the horizontal $P_{z=0}$ plane, using the Lidar sensor’s vertical axis as the z direction. We assign each p point of the point cloud to the corresponding cell, which contains the projection of p to $P_{z=0}$. We use point height information for assigning each grid cell to the corresponding cell class. Before that, we detect and remove grid cells that belong to irrelevant clutter regions, so that we neglect each cell from further calculation, which contains less points than a predefined sensor-specific threshold. After clutter removal, all the points in a cell are classified as ground, if the difference of the minimal and maximal point elevations in the cell is smaller than an elevation threshold (used 25cm),



(a) Extracted HDL and VMX obstacle clouds in initial position



(b) Crossmodal registration result

Fig. 2. Velodyne HDL64E to Riegl VMX450 point cloud registration results using our proposed method (Deák Ferenc tér, Budapest).

moreover the average of the elevations in neighboring cells does not exceeds an allowed height range. The result of ground segmentation is shown in Fig. 1(d)-(f), which confirms that our technique handles robustly the various I3D and MMS Lidar point cloud types. Since with the considered I3D sensors we can only see the bottom parts of the building facades, we truncate the MMS scans at the elevation of 4 metres above the ground level (Fig. 1(f)).

B. Abstract field object extraction

Expecting that significant translation and arbitrarily large orientation difference must be compensated by the proposed registration technique, we estimate first the transform between the point cloud frames at object level. Following the removal of clutter and ground points, our next task is to find distinct groups of points in the remaining *obstacles* cloud (marked by blue point in Fig. 1(d)-(f)), which might belong to different urban objects in the scene. For fast object separation we use a grid-based approach with efficient dynamic processing techniques [11]. In simpler scenes, disjoint field objects can be separated with floodfill propagation on a standard grid lattice, starting from a random seed cell, which step should be repeated until every *obstacle* cell receives a unique object label. For handling more difficult scenarios with several nearby adjacent objects, we also adopted our earlier proposed hierarchical 2-level model [11]. This later method separates first large objects or object groups at a coarse grid level with large cells, then in the refinement it can efficiently separate the

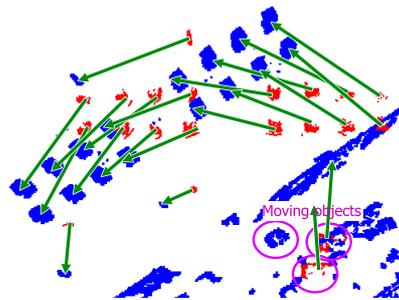


Fig. 3. Demonstration of the optimal object based scan match after top-view projection of the point clouds from Fig. 2, following Algorithm 1.

individual objects within each group, while the computational requirement of the whole algorithm remains two order of magnitude lower than kd-tree based 3D propagation methods.

C. Object based transformation estimation

Assume that in the previous step we have extracted two sets of object centers $C1$ and $C2$ from the two point cloud frames, respectively. Following a similar approach to fingerprint minutia matching of [12], we attempt to find the best transformation in the sense that when applying the transformation to the objects of the set $C1$ as many of these points as possible overlap with the objects from the set $C2$. Since moving, displaced or misdetectd field objects may occur in both frames, there may object center points in either set that do not match with any point in the other set. Similarly to [12] we estimate the transformation parameters using the generalized Hough transform. We discretize the set of all allowed transformations, then for each transformation we calculate a matching *fitness* score. Finally the transformation with the highest score is taken as result.

Since the Lidar point clouds reflect the true object distances from the 3D world, we can consider the transformation as a composition of translation and rotation only. Note as well that since the vehicles carrying the sensors are moving on urban roads, which rarely contain sudden steep slopes, orientation difference is mainly expected around the vertical z axis of the captured point cloud's local coordinate system, while translation in the x and y direction, along the $P_{z=0}$ horizontal plane. Exploiting that this object level step only aims to find an approximate solution for the matching, we project the point clouds to their $P_{z=0}$ plane, and estimate the 2D translation and scalar rotation in this image plane, as demonstrated in Fig. 3. In this way, the searched transformation takes the following form:

$$T_{dx,dy,\alpha} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

The space of transformation consists of triplets (dx, dy, α) , where each parameter is discretized into a finite set of values.

Fitness scores for the transformation candidates are collected in the accumulator array A , where the $A[dx, dy, \alpha]$

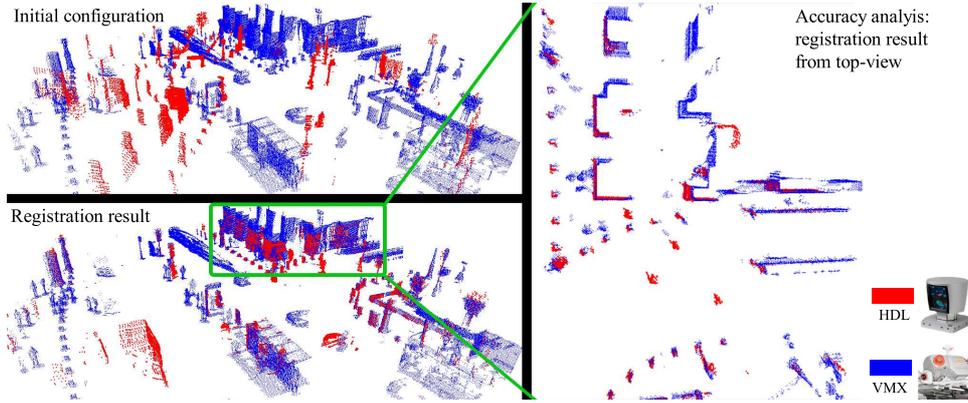


Fig. 4. Crossmodal Velodyne HDL64 to Riegl VMX450 registration results using our proposed method (Fővám tér, Budapest).

element counts the evidence for the concerning $T_{dx,dy,\alpha}$ transformation. The A array can be filled in an iterative way. For each object pair $(o1, o2)$ where $o1 = (x_1, y_1)$ is a point in the set $C1$ and $o2$ is a point in the set $C2$ we determine all possible $T_{dx,dy,\alpha}$ transformations that map $o1$ to $o2$ and we increment the evidence for these transformations in the array. Here we exploit that for every possible rotation value α there is a unique translation vector $[dx, dy]^T$ such that $T_{dx,dy,\alpha}(o1) = o2$, and it can be calculated as:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = o2 - \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} o1$$

The obtained dx and dy values need to be quantized to the nearest bins for appointing the actually increaseable element of the A array. The complete pseudo code of the scan alignment method is shown in Algorithm 1.

Algorithm 1 The cloud alignment algorithm. Takes two clouds as inputs and calculates the transformation between them. $Rot(\alpha)$ represents the rotational matrix along z axis.

```

1: procedure SCANALIGNMENT( $F1, F2, T$ )
2:    $C1 \leftarrow ObjectDetect(F1)$ 
3:    $C2 \leftarrow ObjectDetect(F2)$ 
4:   Initialize 3D accumulator  $A$ 
5:   for all  $o1 \in C1$  do
6:     for all  $o2 \in C2$  do
7:       for  $\alpha \in [0, 359]$  do
8:          $o1' \leftarrow Rot(\alpha) * o1$ 
9:          $(dx, dy) \leftarrow o2 - o1'$ 
10:         $A[dx, dy, \alpha] \leftarrow A[dx, dy, \alpha] + 1$ 
11:      end for
12:    end for
13:  end for
14:   $\alpha, dx, dy \leftarrow FindMaximum(A)$ 
15:   $F1, T1 \leftarrow TransformCloud(F1, \alpha, dx, dy)$ 
16:   $F1, T2 \leftarrow NDT(F1, F2)$ 
17:   $T \leftarrow T2 * T1$ 
18: end procedure

```

D. Point level refinement of registration

Although the above object based scan matching process proved to be largely robust for the considered urban point cloud scenes, its accuracy is limited by the considered planar translation and rotation transformation constraints, and the inaccuracies of object center estimation from the different point clouds. As detailed in [4], due to the special data acquisition technology used in mobile laser scanning, the ground-less *obstacles* cloud can be efficiently used for automated scene matching with the Normal Distribution Transform (NDT) [3] in case of a high quality initial transformation estimation, which is available in our case by taking the output of the object-level step. Therefore in the proposed registration approach, we transform first the *obstacles* cloud according to the optimal $T_{dx,dy,\alpha}$ obtained in Sec. III-C, thereafter we apply NDT for the resulting clouds (see line 16 of Algorithm 1).

IV. EXPERIMENTS

We have evaluated the proposed registration algorithm in the context of the three applications introduced in Sec. I.

A. Crossmodal I3D and MMS point cloud registration

We obtained MMS scans from a Riegl VMX450 system containing eight different urban areas shown in Fig. 5, and subsequently recorded I3D measurement sequences from the same regions with our Velodyne HDL64 and VLP scanners. Table 5 gives an overview on the scene types, and lists the global initial offset and rotation values between the raw I3D and MMS scans. We evaluated the proposed registration process with matching the measurements of both Velodyne sensors to the MMS point clouds. Qualitative results regarding the Deák and Fővám scenes are shown in Fig. 2 and 4 respectively, and the quantitative analysis results of the matching process is given in Table 5. Since *ground truth* transformation was not available, we calculated first the asymmetric Modified Hausdorff Distance (MHD) between the \mathcal{P}_{I3D} Velodyne and \mathcal{P}_{MMS} MMS clouds:

$$MHD(\mathcal{P}_{I3D}, \mathcal{P}_{MMS}) = \frac{1}{\#\mathcal{P}_{I3D}} \sum_{p \in \mathcal{P}_{I3D}} \min_{q \in \mathcal{P}_{MMS}} dist(p, q)$$

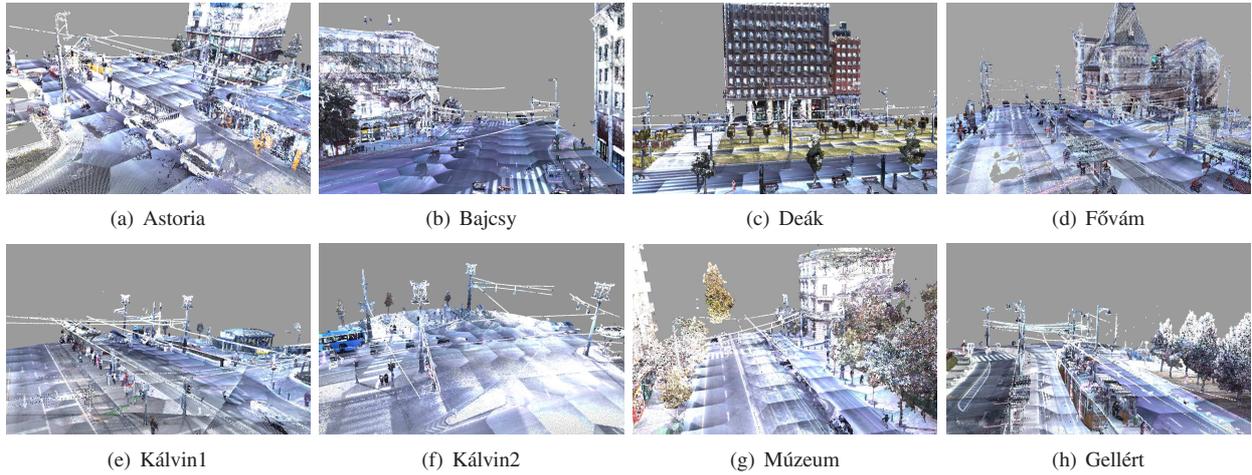


Fig. 5. Test scenes for crossmodal I3D and MMS point cloud registration. MMS point clouds were provided by Budapest Közút Zrt.

where $\#\mathcal{P}$ denotes set cardinality. Columns 5-7 of Table 5 contain the obtained MHD values initially, after the object based Hough matching step, and in the final stage following NDT-based registration refinement. We can observe that both steps significantly decrease the distances between the scans in almost all data sets. However, the absolute MHD values do not reflect properly the accuracy of the algorithm, since the presence of several moving objects (especially large trams or tracks) irreverently increase the calculated average distances. For this reason, we also used a modified error metrics called Median Point Distance (MPD), where we sort the points in \mathcal{P}_{I3D} from the lowest to the highest value of $\min_q \text{dist}(p, q)$, and take the median of the distances among all $p \in \mathcal{P}_{\text{I3D}}$. As shown in the 8-10th columns of Table 5 the MPD values are also significantly decreased during the registration process, and in seven out of the eight scenes the resulting MPD errors are below 3cm, which fact was also confirmed by visual verification. Only the test scene Bajcsy yielded erroneous registration result both by visual and quantitative (MHD, MPD) analysis. In this sample both I3D point clouds contained several moving vehicles, including large buses which occluded various relevant scene structure elements. As future plan, we aim at developing an efficient frame dropping algorithm to filter out such situations, which mislead the registration. The 11th (last) column of Table 5 lists for each scene the computational time of the complete matching process (varying between 0.3 and 2.2 seconds), which confirms that the approach is close to online usability.

B. IMU-free SLAM based on sparse I3D data

Using the registration algorithm on selected consecutive frames of a single Lidar sensor, an accurate 3D map of the urban environment can be constructed without the help of any external or internal navigation sensors such as GPS or IMU. Assuming that the speed of the vehicle carrying the Lidar can be constrained by the admitted range in urban traffic, we used a compact A accumulator size ($50 \times 50 \times 360$) covering an

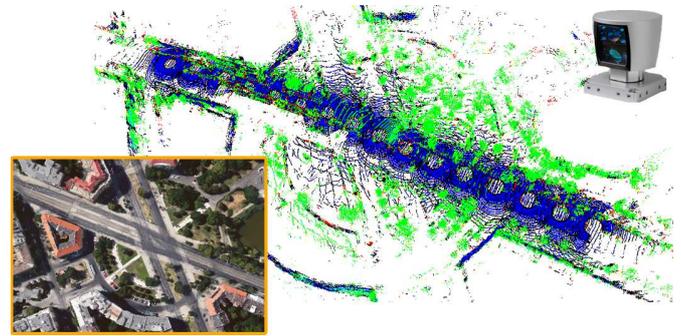


Fig. 6. SLAM results with Velodyne HDL64 in Kosztolányi tér, Budapest (1.2M points from 80 frames captured at 3fps from a moving vehicle)

offset range $[-25m, 25m]$ between two consecutive frames. This choice ensured very low computational cost up to 1m registration error accuracy, which can already be handled by the NDT step of the process. Fig. 6 and 7 show efficient registration results using Velodyne HDL64 and VLP16 sensors, respectively. Future work will concern the quantitative evaluation of the SLAM usage of the proposed algorithm.

C. Registration of clouds from different I3D sources

We examined at a proof-of-concept level, that the proposed approach is highly appropriate for matching the Velodyne HDL64 and VLP16 sensor measurements from the same region (see a sample frame in Fig. 8), which fact opens prospects for various sorts of information exchange and fusion between the concurrently active autonomous car community.

V. CONCLUSIONS AND ACKNOWLEDGEMENT

We introduced a new automated registration algorithm for mobile laser scanning data, which can work efficiently with different kind of sensor characteristics through the combination of segmentation, object level and point level matching steps. The authors thank Budapest Közút Zrt (Road Management Department) for the provision of the Riegl VMX

TABLE I
RESULTS OF CROSSMODAL I3D AND MMS POINT CLOUD REGISTRATION (VELODYNE HDL64/VLP16 TO RIEGL VMX SCAN MATCHING)

Name	Scene type	Sensor	initial offset, rotation	MHD (m)			MPD (m)			Comput time
				Init	Hough	Final [†]	Init	Hough	Final	
Astoria	traffic hub	HDL	2.2m, 62°	3.641	0.773	0.415	1.587	0.511	0.022	1.923
		VLP	2.2m, 99°	5.045	0.582	0.221	3.623	0.231	0.008	0.665
Bajcsy	main road	HDL	2.0m, 92°	5.657	11.441	10.105	1.177	2.702	4.539	0.992
		VLP	10.3m, 72°	6.971	20.115	17.796	4.179	17.319	14.341	0.329
Deák	road & square	HDL	1.4m, 32°	3.638	0.717	0.338	1.516	0.345	0.004	1.960
		VLP	3.6m, 127°	7.348	0.870	0.911	5.502	0.143	0.101	0.769
Fővám	square	HDL	2.0m, 134°	8.404	3.494	2.870	6.143	1.339	0.008	3.796
		VLP	0.1m, 20°	5.143	1.849	1.431	3.393	0.216	0.010	1.182
Kálvin part 1	road & square	HDL	1.4m, 118°	9.891	0.774	0.205	5.808	0.469	0.005	1.159
		VLP	2.0m, 42°	11.427	7.016	8.178	5.007	0.752	0.014	0.573
Kálvin part 2	road & square	HDL	5.8m, 104°	19.445	2.252	2.002	4.968	0.437	0.023	0.288
		VLP	6.1m, 56°	19.663	2.901	5.909	16.826	0.817	0.065	0.221
Múzeum	boulevard	HDL	2.2m, 70°	14.911	3.358	1.373	12.354	1.315	0.009	2.574
		VLP	5.0m, 91°	6.970	2.489	3.412	1.477	0.312	0.018	1.403
Gellért	square	HDL	1.0m, 125°	3.180	0.949	1.046	1.238	0.224	0.014	1.045
		VLP	0.0m, 34°	5.241	2.438	1.574	4.037	1.173	0.029	0.852
Average values [‡]		HDL	2.3m, 92°	9.016	1.760	1.178	4.802	0.663	0.012	1.821
		VLP	3.7m, 68°	8.691	2.592	3.091	5.695	0.521	0.035	0.809

Error measures: MHD: Modified Hausdorff distance, MPD: median point distance.

[†]Final result refers to the Hough+NDT cascade, [‡]Bajcsy was excluded from averaging, due to unsuccessful registration



Fig. 7. SLAM results with Velodyne VLP16 in Bartók Béla út, Budapest (0.3M points from 200 frames captured at 5fps from a moving vehicle).

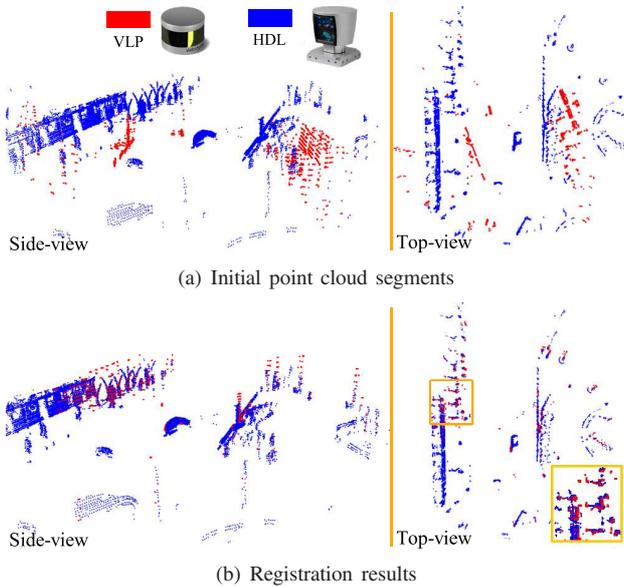


Fig. 8. Velodyne VLP16 and HDL64 point cloud registration results.

MLS test data. The work was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, and by the National Research, Development and Innovation Fund (NKFI).

REFERENCES

- [1] J. Behley, V. Steinhage, and A. Cremers, "Performance of histogram descriptors for the classification of 3D laser range data in urban environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, 2012, pp. 4391–4398.
- [2] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.
- [3] M. Magnusson, "The three-dimensional normal-distributions transform – an efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro University, December 2009.
- [4] O. Józsa, A. Börcs, and C. Benedek, "Towards 4D virtual city reconstruction from Lidar point cloud sequences," in *ISPRS Workshop on 3D Virtual City Modeling*, ser. ISPRS Annals Photogram. Rem. Sens. and Spat. Inf. Sci., Regina, Canada, 2013, vol. II-3/W1, pp. 15–20.
- [5] H. Men, B. Gebre, and K. Pochiraju, "Color point cloud registration with 4D ICP algorithm," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 1511–1516.
- [6] A. Gressin, C. Mallet, and N. David, "Improving 3D LIDAR Point Cloud Registration Using Optimal Neighborhood Knowledge," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 111–116, Jul. 2012.
- [7] A. Gressin, B. Cannelle, C. Mallet, and J.-P. Papellard, "Trajectory-Based Registration of 3D LIDAR Point Clouds Acquired with a Mobile Mapping System," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 117–122, Jul. 2012.
- [8] B. Douillard, A. Quadros, P. Morton, J. P. Underwood, M. D. Deuge, S. Hugosson, M. Hallström, and T. Bailey, "Scan segments matching for pairwise 3d alignment," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 3033–3040.
- [9] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3D LIDAR point clouds," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 2798–2805.
- [10] W. S. Grant, R. C. Voorhies, and L. Itti, "Finding planes in LiDAR point clouds for real-time registration," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 4347–4354.
- [11] A. Börcs, B. Nagy, and C. Benedek, "Fast 3-D urban object detection on streaming point clouds," in *Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving at ECCV'14*, ser. LNCS. Zürich, Switzerland: Springer, 2015, vol. 8926, pp. 628–639.
- [12] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain, "A real-time matching system for large fingerprint databases," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 799–813, 1996.