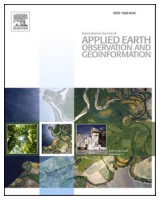
International Journal of Applied Earth Observations and Geoinformation 110 (2022) 102767

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag



Point cloud registration and change detection in urban environment using an onboard Lidar sensor and MLS reference data

Örkény Zováthi [a,b,*], Balázs Nagy [a], Csaba Benedek [a,b]

[a] Machine Perception Research Laboratory, Institute for Computer Science and Control, Budapest, Hungary
[b] Péter Pázmány Catholic University, Faculty of Information Technology and Bionics, Budapest, Hungary

ARTICLE INFO

Keywords:
multi-beam lidar
mobile laser scanning
point cloud registration
change detection

ABSTRACT

This paper presents a new method for urban scene analysis, which comprises 3D point cloud registration and change detection through fusing Lidar point clouds with significantly different density characteristics. The introduced method is able to extract dynamic scene segments (traffic participants or urban renewal regions) and seasonal changes (vegetation regions) from instant 3D (i3D) measurements captured by a Rotating Multi-beam (RMB) Lidar sensor mounted onto the top of a moving vehicle. As reference data, we rely on a dense point cloud-based environment model provided by Mobile Laser Scanning (MLS) systems. The proposed approach is composed of new solutions for two main subtasks. First, a novel multimodal point cloud registration algorithm is introduced, which can improve the alignment of the sparse i3D measurements to the dense MLS data, where conventional point level registration or keypoint/segment matching strategies fail. Second, an efficient Markov Random Field-based change extraction step is implemented between the registered point clouds, which exploits that due to geometric considerations of mapping with the given sensor configuration, the essence of the problem can be solved quickly in the 2D range image domain without information loss. Experimental evaluation is conducted on a new Benchmark set that contains three different heavy traffic road sections in city center areas covering in total nearly 1 km long pathway sections. Test data consists of relevant industrial measurements provided by a state-of-the-art RMB scanner (with a point density of around 50 −500 points/m^2) and an up-to-date MLS system (more than 5000 points/m^2). The clear advantages of the new method are quantitatively demonstrated against various reference techniques. In comparison to six different point cloud registration methods, the median value of point level distances is decreased by 1–2 orders of magnitude by the proposed approach. Regarding change detection, the new method outperforms the reference models either in F1-scores (by around 10–25%) or in computational complexity (10–1000 times faster).

1. Introduction

The utilization of new sensor technologies for three-dimensional (3D) data acquisition is a key step for deeply understanding and widely exploiting spatial information in our environment. Widespread sensor platforms are mapping systems based on Lidar (Light Detection and Ranging) technology, as they provide accurate 3D measurement flows with high acquisition speed (Benedek et al., 2021). However, due to a trade-off between the available laser scanners' temporal and spatial resolution, the provided point clouds show significantly different quality and density characteristics (Behley et al., 2012).

This paper focuses on the fusion of different Lidar-based data modalities for advanced scene analysis in urban environments. A long-term vision of the conducted research work is to facilitate the joint exploitation of the measurements from the cars' instant sensing platforms and offline spatial database content of the newest Geo-Information System (GIS) solutions. On one hand, the proposed new algorithms may help autonomous vehicles (AVs) to obtain relevant spatial 3D map information for decision support in real-time. On the other hand, they also may provide opportunities for extending and updating the GIS databases based on the sensor measurements of the vehicles in everyday traffic. The present paper deals with two specific aspects of this application context: multimodal point cloud registration and change detection between different sensor measurements.

First, we utilize instant 3D (i3D) measurements taken by a Rotating Multi-beam (RMB) Lidar sensor (Börcs et al., 2017; Douillard et al.,

* Corresponding author at: Institute for Computer Science and Control (SZTAKI), Kende street 13-17, H-1111 Budapest, Hungary.
E-mail address: zovathi.orkeny@sztaki.hu (Ö. Zováthi).

https://doi.org/10.1016/j.jag.2022.102767
Received 11 October 2021; Received in revised form 12 March 2022; Accepted 30 March 2022
Available online 10 May 2022

2012) operating onboard on AVs (Bayat et al., 2017). RMB Lidars provide sequences of point clouds about their environment in real-time, with a capturing frequency of 5-20 fps, so that dynamic scene objects can be detected and tracked across the time frames. However, the captured data within a single time frame is notably sparse and non-uniformly distributed: the point clouds have low vertical resolutions, while their densities decline rapidly if the objects are located further from the sensor (see Fig. 1(a)). Beyond the RMB Lidar sensor, we also assume that the vehicle carries a Global Positioning System (GPS) receiver, however, we expect that in various dense city regions the GPS measurements might be inaccurate, providing position errors up to several meters.

Second, we obtain detailed segmented 3D environment models of dense urban areas to augment the AVs' limited i3D measurements with prior beliefs. In industrial practice, vector-based high definition (HD) maps are often adopted for this purpose (Mi et al., 2021), as they store precise and high-quality metadata about the static parts of the environment (Ma et al., 2019). Nevertheless, the creation and maintenance of these HD maps need continuous attendance of human operators (Zhao et al., 2019), moreover, the vectorization process may also drop out relevant point level details. Following a different approach, we utilize offline integrated Mobile Laser Scanning (MLS) measurements that provide accurate and geo-referenced point clouds (Zheng et al., 2017; Guan et al., 2015) with fairly homogeneous point distribution in local regions, while their point density remains very high (above 5000 points/m^2) over the whole scene regions of interest, which are relevant for the vehicle's environment perception modules (Fig. 1(b)). Without any vectorization, as preprocessing we segment the raw and noisy MLS data using an automatic C^2CNN (Nagy and Benedek, 2019) technique and remove all regions which contain *ground* areas or dynamic objects (e.g. *parking* or *moving cars*). Thereafter, we extract static objects from the remaining MLS classes (*pillar-like*, *street furniture*, *vegetation*, *facade*) by 3D Euclidean clustering (Rusu and Cousins, 2011), and consider these object point clouds as highly detailed reference landmarks for the AVs' onboard i3D measurements.

As the output of the proposed approach, we aim to achieve accurate (up to cms) global localization of the AVs and to separate changed dynamic (including traffic participants, urban renewal areas) or seasonally varying (vegetation areas, tree crowns, bushes) regions, and unchanged static (among others street furniture, traffic lights, signs) areas from the AVs' i3D measurements exploiting the prior information obtained from the segmented MLS data.

To the best of our knowledge, this is the first work that addresses a joint multi-sensory point cloud registration and change detection problem by fusing sparse i3D point clouds and preliminary recorded MLS data, where, as a key challenge, the MLS point clouds are in several regions 100–1000 times denser than the corresponding i3D measurement segments. For the above reason, a newly constructed dataset is presented, called the *SZTAKIBudapest* Benchmark, which contains real RMB Lidar measurements and industrial MLS data, enabling the research of the concerning problem and validation of the proposed approach.

The introduced method is composed of various steps demonstrated in Fig. 2, where we propose two significant contributions that improve the state-of-the-art:

1. A novel efficient algorithm is introduced for the *registration* of sparse i3D and dense MLS point clouds with a relatively poor initial alignment with position errors up to 12 meters, and orientation errors up to ± 60 degrees. In practice, this method can compensate for the global positioning error of the AVs in dense urban areas where high-quality GPS signals cannot be received.
2. We propose a new Markov Random Field-based approach (RangeMRF) for multi-class *change extraction* and classification (dynamic, seasonal, or no change) between the registered point clouds in a *lossless* way by 2D range image representations.

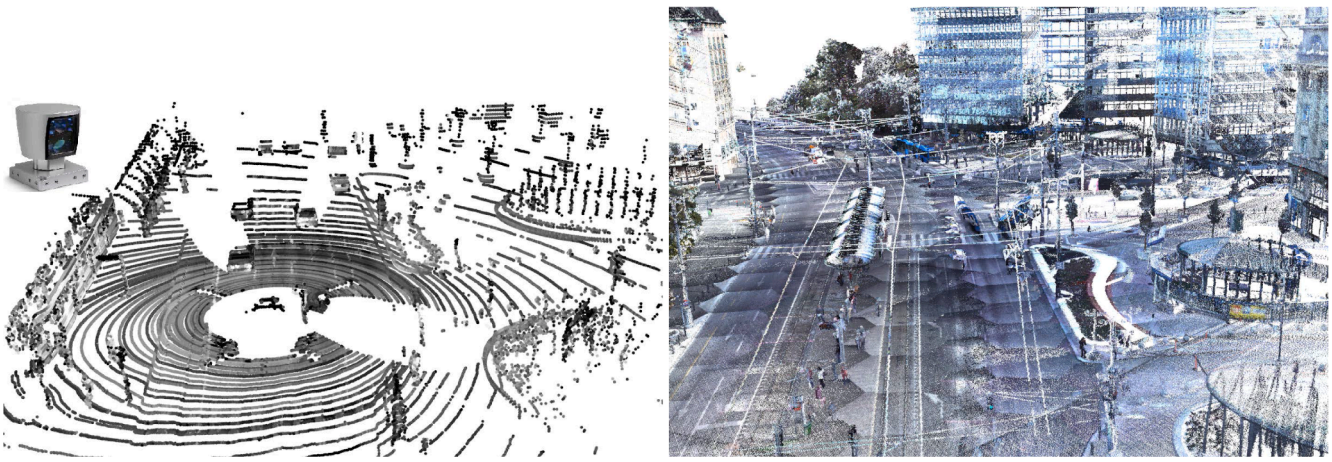
Using the *SZTAKIBudapest* Benchmark it is experimentally shown in the paper (with extensive quantitative analysis), that earlier proposed point cloud registration or change detection methods from the literature have limited efficiency in the above application context, and the advantages of the new approach versus earlier methods will be demonstrated.

2. Related work

In this section, we present a state-of-the-art overview of point cloud registration and change detection techniques.

2.1. Point cloud registration

In the field of point cloud registration (Arun et al., 1987), coarse-to-fine alignment strategies are considered as standard approaches (Cheng et al., 2018). Hereby the coarse-alignment process is usually based on matching corresponding 3D feature points or characteristic primitives extracted from the point clouds (Ge et al., 2019), while the fine-alignment step aims to minimize distances between points within the overlapping point cloud regions by an iterative process. As discussed in (Ge et al., 2019), general coarse-alignment methods consist of two main steps:



(a) i3D measurement frame by an RMB Lidar with 64 laser beams (b) Point cloud segment recorded by an up-to-date MLS platform

Fig. 1. Comparison of (a) sparse RMB Lidar-based and (b) dense MLS point clouds captured in the same inner-city area.

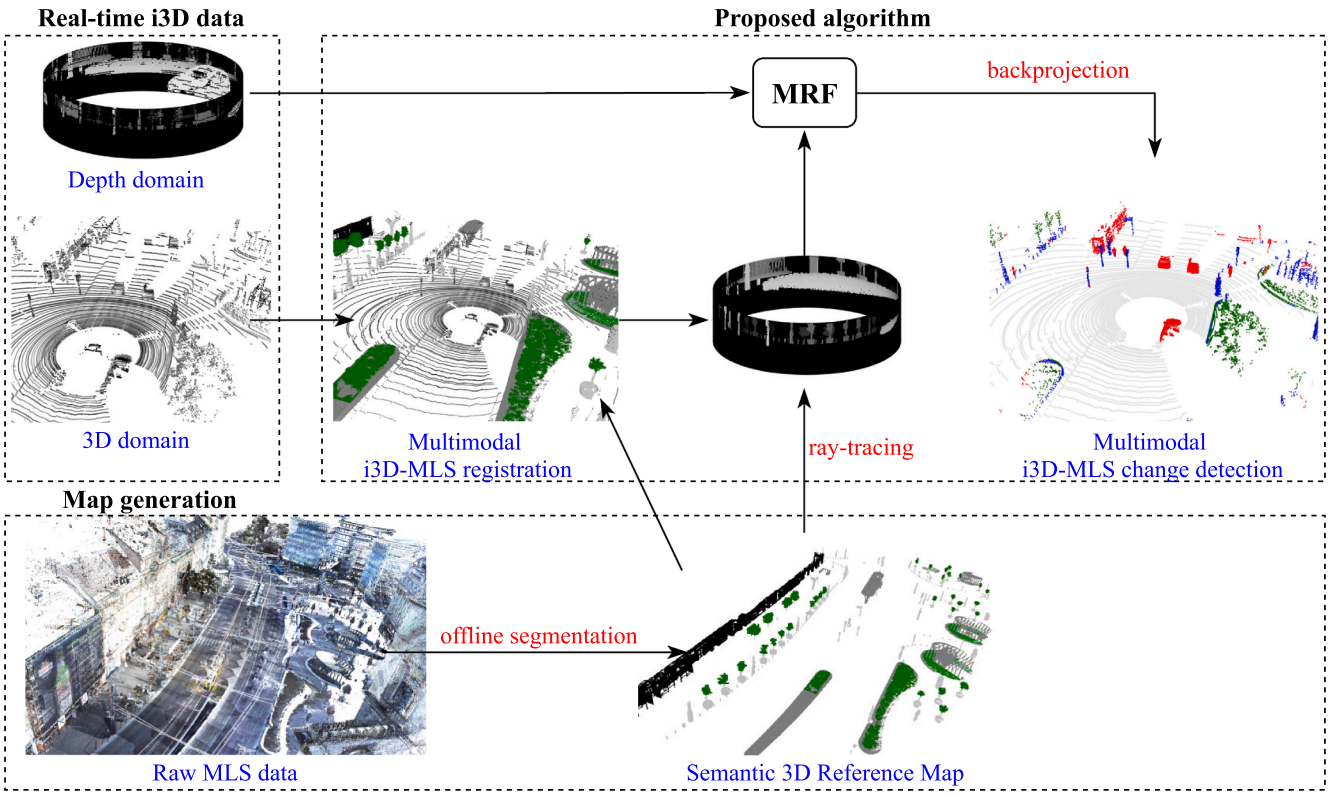


Fig. 2. The workflow of the proposed method.

1. efficient keypoint detection (Hänsch et al., 2014), followed by meaningful and discriminative keypoint description (Rusu et al., 2009),
2. executing an iterative process for finding correspondences on a solving platform.

For the dense MLS data, general handcrafted (Hänsch et al., 2014; Sipiran and Bustos, 2011; Zhong, 2009) and learning-based (Li and Lee, 2019; Lu et al., 2020) detectors can be directly used to extract keypoints (Theiler et al., 2014). However, the adaption of such methods often fails for sparse, inhomogeneous and noisy i3D data (Li and Olson, 2010). The usage of popular feature descriptors based on normal vectors, curvature, density, and pairwise point-to-point distances in local regions (Rusu et al., 2009; Dong et al., 2017) cannot enrich the information in cross-modal scenarios, since such parameters are significantly different regarding the i3D and MLS point clouds, respectively. Therefore, the lack of pairable feature points in the two compared point clouds results that correspondence-based iterative solving platforms like classical RANSAC (Choi et al., 2015) approaches or even the very recent TEASER++2020 (Yang et al., 2021) cannot provide accurate solutions.

Alternatively to relying on feature points, further methods (Douillard et al., 2012; Dubé et al., 2018) focus on region-based point cloud alignment. (Douillard et al., 2012) solves data mapping by matching segments instead of points across different i3D scans. This approach can efficiently match sparse and small i3D segments with similar point characteristics, but it induces significant computational complexity: around five to fifteen seconds is needed even for such small i3D data pairs, while the running time grows rapidly by using larger point clouds. The SegMap2018 (Dubé et al., 2018) approach matches segments from a few consecutive i3D scans (local map) to a global 3D map with high accuracy. This technique builds first a dynamic voxel grid from the point clouds, then segments the voxels and extracts geometry-based or data-driven features from both segmented clouds. Finally, it finds correspondences in the feature space. However, matching these features correctly is extremely sensitive to the differences in the point cloud characteristics. While according to the experiments in (Dubé et al., 2018), the method works robustly when the global map is generated from the same sensor data as the aligned i3D measurements (e.g. in Simultaneous Localization and Mapping (SLAM) tasks), the method's extension to different sensor modalities is yet to be solved (Dubé et al., 2018). Using dense MLS point clouds and sparse i3D data, we cannot extract corresponding segments based on similar feature contexts, which fact we experimentally demonstrate in Sec. 5.

There are methods addressing directly the problem of registering cross-source point clouds (Huang et al., 2021), which is more challenging due to varying noise, the large number of outliers and the significantly different density characteristics (Huang et al., 2017). To improve on point-level correspondences, the GCTR2019 approach (Huang et al., 2019) uses triplet point similarities, and solves the optimization problem in the tensor space by an iterative process. However, its computation time remains around 1-2 minutes for matching simple indoor scan-pairs, making it unfeasible for real-time applications. Instead of relying on point-point correspondences, the FMR2020 method (Huang et al., 2020) solves the cross-source registration problem by minimizing a feature-metric projection error, however, the density differences of its aligned point clouds are less significant than in our i3D-MLS scenario.

For point level registration, which is usually applied in a refinement step after the coarse alignment, the Iterative Closest Point (ICP) (Zhang, 1994) is a frequently adopted algorithm with several improvements (Gressin et al., 2012b; Gressin et al., 2012a). However, all of these variants perform local error minimization, thus they demand a high-quality initial estimation for the alignment. In practice, a position error of several meters does not satisfy this requirement, moreover, the computational time is considerably increased for scans with poor initial alignments.

2.2. Change detection

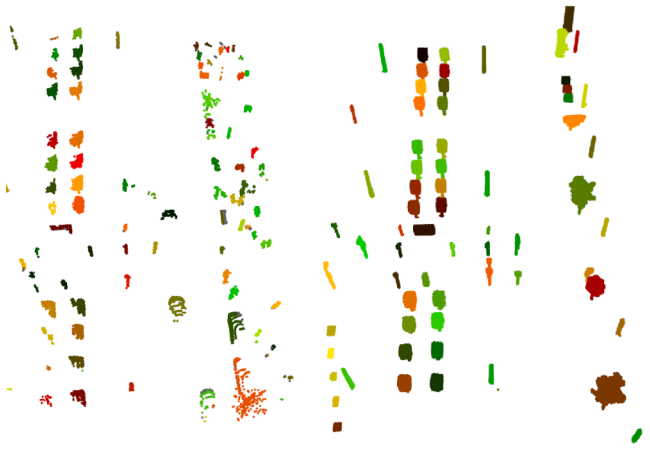
Next, we continue the discussion with point cloud-based change detection techniques. The majority of these methods can be adopted for MLS-MLS data comparison tasks (Ku et al., 2021; Xiao et al., 2015), where the two point clouds are captured with the same laser scanner (Ku et al., 2021) showing similar and locally homogeneous density characteristics. In practice, two different scannings of the same scene never detect the same surface points of the objects, thus there are point level deflections between the scanned models, even if the surfaces are perfectly aligned. However, using the same MLS sensor, change detection can be approximated by pointwise comparison steps in the 3D space, performing a preliminary uniform voxel-based downsampling (Aljumaily et al., 2017), or applying locally adaptive (Liu et al., 2021) or parametric (Girardeau-Montaut et al., 2005) radius thresholds. Nevertheless, the usability of any point level distances strongly depends on the density of the point clouds (Xiao et al., 2015), and it can be largely misleading during the analysis of multimodal data, like in an i3D-MLS scenario. Alternative methods (Schlichting and Brenner, 2016) use segment level comparison, where an object is marked as change if a given percentage of its segment points have no neighbors. (Voelsen et al., 2021) combines this approach further with segment classification to integrate semantic information in the change detection process. These methods demand accurate object extraction (and detailed classification) from the point sets, which is challenging in sparse i3D data (Börcs et al., 2017). To handle irregular point density, (Xiao et al., 2015) combines point-to-triangle distance calculation, ray-tracing, and occupancy grid generation. Although this method can effectively deal with occlusions and penetrable MLS objects, the triangulation step may mean a bottleneck in terms of computational speed and robustness, especially in noisy point cloud segments. Instead of using rays, (Liu et al., 2016) perform an occupancy technique along a regular voxel grid, and identify the changed regions based on the inconsistent voxels. As the main drawback, the above approaches work similarly for all point segments: On one hand, with parameter settings yielding high sensitivity, they produce many false-positive detections for sparse and noisy segments of the i3D scans. On the other hand, with low sensitivity, they might ignore crucial regions containing only a few points (e.g. pedestrian with only 10–40 points), which issue will be demonstrated in Sec. 5.

3. Proposed method

The consecutive steps of the proposed method are briefly summarized in Fig. 2. It starts with accurately registering the actual i3D measurement to the segmented MLS reference model. Hereby we propose a new object-based coarse-to-fine alignment algorithm, which significantly speeds up the process while keeping the registration accuracy high. Then, we perform change detection between the registered i3D and MLS point clouds by a Markov Random Field-based new classification model.

3.1. Registering the i3D data to the semantic MLS model

We developed a multimodal i3D to MLS point cloud registration technique to align the i3D point scans to the reference MLS model, which can compensate initial position errors up to several meters, expecting that in dense urban environments with poor GPS coverage, the initial position estimation of an AV might be notably inaccurate. Since we intend to use the i3D point cloud stream recorded by a moving vehicle for real-time decision support, the RMB Lidar point cloud should be fully automatically processed and matched to the MLS model. On the other hand, we can exploit here that in the preliminary segmented MLS point clouds, the ground regions are separated, and various abstract landmark objects (e.g. pillar-like entities or short street furniture instances) are extracted and labeled in advance (Fig. 3(b)). The steps of the algorithm are discussed in the next parts of this section.



(a) Separated objects in an i3D scan (b) Static objects in the MLS data

Fig. 3. Possible landmark objects in the same scene extracted from the i3D (a) and MLS (b) point clouds for the transformation estimation. Each object candidate is displayed with a different color.

3.1.1. Ground removal and object separation in the i3D point cloud

The initial step of the registration process is object segmentation in the RMB Lidar frames. Due to the limited resolution and inhomogeneous density of the i3D point clouds which can mislead even state-of-the-art object detectors (Lang et al., 2019), we use here a geometry-based approach for point cloud segmentation (Börcs et al., 2017). First, we apply fast 2D grid-based *ground-obstacle* separation in the i3D input point cloud, by classifying each cell on an estimated ground surface based on local point density and point elevation difference features. Then obstacle points are clustered to form separate abstract object candidates (Fig. 3(a)) using a 2D hierarchical grid-based region growing algorithm, which approach is around a hundred times faster than conventional 3D clustering methods, while it can efficiently separate even nearby objects (Börcs et al., 2017; Rusu and Cousins, 2011). For ensuring fast processing and robustness, this step does not perform any attempt on object classification: as a result, a set of 3D blobs is extracted, that may represent various moving or static obstacles in the urban environment.

3.1.2. Transformation estimation

Next, we estimate an optimal transformation to align the sparse i3D scan to the MLS reference model. As a key idea, instead of aligning the raw point clouds, we aim to register the frames via an object level voting algorithm – detailed in subsection c) –, which matches the previously extracted *landmark objects* of the MLS data and the *separated objects* from the i3D scan (Fig. 4). Here a major challenge is that in the automatically segmented RMB Lidar frames one should expect plenty of spurious objects which cannot be matched to the MLS landmarks: the i3D frames may include many dynamic objects, and further artifacts can be caused by partially extracted entities due to occlusions. For quickly matching the two object sets which typically contain many objects (often a few dozen) with possibly a large ratio (up to 80%) of outliers, we turned to a robust generalized Hough transform-based technique, that proved earlier efficient for different sorts of complex assignment problems such as fingerprint minutiae matching (Ratha et al., 1996).

First, we rely on the available GPS signals to initially align the actual i3D point cloud frame to the reference coordinate system of the MLS data. This initial alignment is usually notably inaccurate, which we enhance first at a coarse level, by searching for an optimized rigid transformation with a 3D translation and a rotation component between the point clouds. The translation component $(\Delta x, \Delta y, \Delta z)$ compensates for the GPS-based offset error, while – based on experiments – the rotation component can be fairly modeled by a single rotation value (θ)

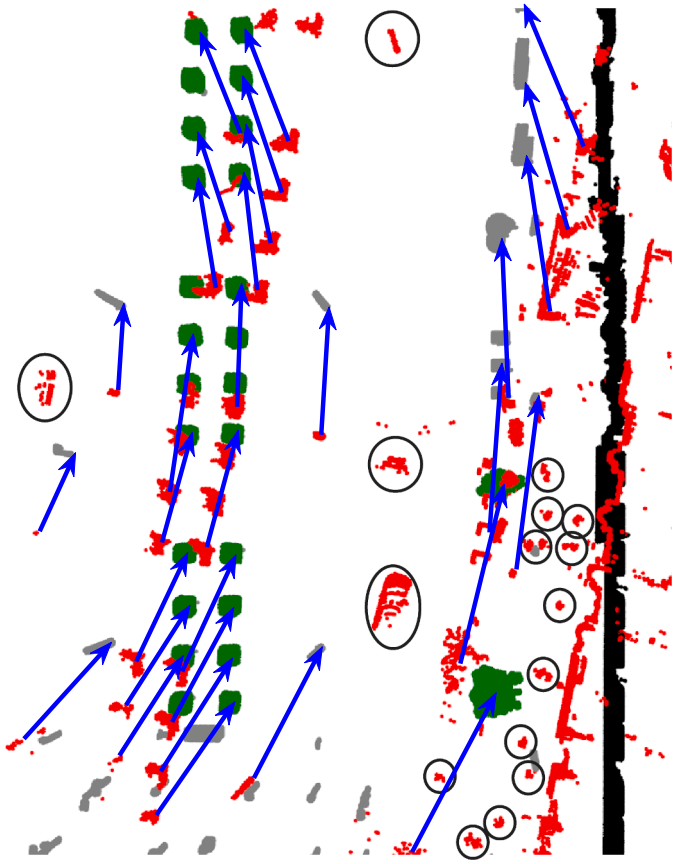


Fig. 4. Displaying the corresponding i3D-MLS object pairs estimated by the proposed method, based on the generalized Hough transform-based schema (Ratha et al., 1996). i3D points are displayed with red, object pairs are marked by **blue arrows**, some outlier objects are circled by **black**. (For clear visualization, a scene sample with relatively few moving objects has been chosen here).

around the upright axis of the vehicle. As a consequence, we model the optimal coarse transform by a 4×4 homogeneous matrix as shown in Eq. (1):

$$\mathbf{T}_{\Delta x,\Delta y,\Delta z,\theta} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & \Delta x \\ -\sin\theta & \cos\theta & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Next, we find the optimal parameters of this $\mathbf{T}_{\Delta x,\Delta y,\Delta z,\theta}$ transformation in three steps, presented in the following subsections.

3.1.2.1. a) Semantic compatibility constraints. As described earlier, the MLS-based 3D reference map contains separated tall *pillar-like* and shorter *street furniture* object point clouds, which can be used as landmark objects by the AVs. For these landmarks, we prescribe the following compatibility constraints:

- a detected i3D object (see Sec. 3.1.1) is compatible with a *pillar-like* MLS object if the vertical side length of its bounding box is more than twice of its width and depth parameters,
- an i3D object is compatible with a *street furniture* object if the ratio of their bounding volumes is between $[0.75, 1.25]$.

In the upcoming transformation estimation step, we will only consider the votes of compatible object pairs, to decrease the effects of outliers and speed up the process. Note that the impacts of false matches generated by dynamic objects of the i3D point clouds will be eliminated later through global parameter optimization.

3.1.2.2. b) Keypoint selection. Since the detected objects are *not pointwise*, appropriate keypoint extraction is a critical step which process should remain robust in the considered crossmodal i3D-MLS scenario, where several objects are only partially scanned, the object appearances in the i3D frames are different from the corresponding MLS objects, and they may also vary scan by scan. We have investigated various keypoint selection strategies and experienced that an 8-keypoint strategy (Nagy and Benedek, 2019) provided the best result, where, as keypoints, we choose the eight corners of the objects' 3D bounding boxes.

3.1.2.3. c) Optimal parameter estimation. We approximate the transformation which aligns the i3D and MLS point clouds via compatible keypoint pairs. First, we collect the extracted *i3D* and *MLS* objects into two object sets marked by $\mathscr{O}_{\text{i3D}}$ and $\mathscr{O}_{\text{MLS}}$, where each object is described by 8 keypoints (i.e. bounding box corners). Then the generalized 4D Hough transform (Ratha et al., 1996) is adopted to obtain an optimal transformation between the *i3D* and *MLS*-based keypoint sets, by a voting algorithm (Fig. 4).

To restrict the search space of transformations, we allow maximum offsets of $\pm 60^\circ$ for rotation (θ), ± 12 meters for planar translation (Δx and Δy), and ± 2 meters for vertical translation (Δz). As required by the Hough schema (Ratha et al., 1996), the parameter space is discretized with step sizes of 0.2 meters for translation and 0.25° for rotation, which choice enables both reasonable coarse alignment and quick computation. The votes of the possible parameter quartets are accumulated in a 4D array $\Phi[\Delta x, \Delta y, \Delta z, \theta]$, which is initialized with zero values.

Algorithm 1. The proposed coarse registration method.

procedure CoarseAlignment($\mathscr{O}_{\text{i3D}}, \mathscr{O}_{\text{MLS}}$)
  Reset the 4D accumulator array Φ
  for all $o_{\text{i3D}}, o_{\text{MLS}} \in \mathscr{O}_{\text{i3D}} \times \mathscr{O}_{\text{MLS}}$ **do**
    if compatible($o_{\text{i3D}}, o_{\text{MLS}}$) **then**
      for $i = 1 : 8$ **do**
        $k^i_{\text{i3D}} \leftarrow o_{\text{i3D}}$
        $k^i_{\text{MLS}} \leftarrow o_{\text{MLS}}$
        for all $\theta \in [-60^\circ, 60^\circ]$ **do**
          $k^{i*}_{\text{i3D}} \leftarrow Rot\theta \cdot k^i_{\text{i3D}}$
          $[\Delta x, \Delta y, \Delta z] \leftarrow k^i_{\text{MLS}} - k^{i*}_{\text{i3D}}$
          $\Phi[\Delta x, \Delta y, \Delta z, \theta] \leftarrow \Phi[\Delta x, \Delta y, \Delta z, \theta] + 1$
  $\Delta x^*, \Delta y^*, \Delta z^*, \theta^* \leftarrow FindMaximum(\Phi)$
  $T \leftarrow \Delta x^*, \Delta y^*, \Delta z^*, \theta^*$
return T

The optimization step searches for possible correspondences between the keypoints of all *compatible* object pairs $(o_{\text{i3D}}, o_{\text{MLS}}) \in \mathscr{O}_{\text{i3D}} \times \mathscr{O}_{\text{MLS}}$. For a given keypoint couple $k_{\text{i3D}}, k_{\text{MLS}}$ we increase the evidence of all $\mathbf{T}_{\Delta x,\Delta y,\Delta z,\theta}$ mappings, which move k_{i3D} to k_{MLS}. More specifically, for every possible $\theta' \in [-60^\circ, +60^\circ]$ value, we rotate k_{i3D} by θ' first, then the following $[\Delta x', \Delta y', \Delta z']^T$ offset is computed:

$$\begin{bmatrix} \Delta x' \\ \Delta y' \\ \Delta z' \end{bmatrix} = k_{\text{MLS}} - \begin{bmatrix} \cos\theta' & \sin\theta' & 0 \\ -\sin\theta' & \cos\theta' & 0 \\ 0 & 0 & 1 \end{bmatrix} k_{\text{i3D}} \tag{2}$$

Next, we vote for the calculated $T_{\Delta x',\Delta y',\Delta z',\theta'}$ transform so that we increase the $\Phi[\Delta x', \Delta y', \Delta z', \theta']$ element of the accumulator array by one (see Algorithm 1).

After iterating through the whole parameter space, the optimal $T_{\Delta x^*,\Delta y^*,\Delta z^*,\theta^*}$ transform is defined as follows:

$$(\Delta x^*, \Delta y^*, \Delta z^*, \theta^*) = \underset{\Delta x,\Delta y,\Delta z,\theta}{\text{argmax}} \Phi[\Delta x, \Delta y, \Delta z, \theta] \tag{3}$$

3.1.3. Registration refinement

While acknowledging its robustness, the precision of the object-based registration technique is affected by the discretization step of

the translation and rotation parameters, constraints on the modeled rigid transformation, and by issues of object bounding box fitting on partially detected objects (Fig. 5(c)). However, this coarse alignment step can provide an efficient initialization for a point level refinement algorithm, such as the ICP (Zhang, 1994), which we execute for point cloud segments corresponding to aligned object pairs only. The final transformation is taken as:

$$\mathbf{T}_{final} = \mathbf{T}_{\text{ICP}} \cdot \mathbf{T}_{\Delta x^*, \Delta y^*, \Delta z^*, \vartheta^*}. \tag{4}$$

Although the application of the ICP algorithm induces additional computation, by reducing the number of alignable points through object compatibility criteria checking, and by ensuring low initial alignment error by the proposed coarse registration algorithm, the whole process needs remarkably – with 1-2 orders of magnitude – less computational time compared to matching raw complete point cloud scans, while keeping the registration accuracy high (Fig. 5(b), (d)).

3.2. RangeMRF: range image-based change detection

After accurately registering the i3D-MLS measurements, the proposed method, called RangeMRF, detects changes in 2D range images derived from the point clouds. As key advantages of using a compact range image representation, the proposed method is notably quick, meanwhile, it can robustly handle the significantly different characteristics of the two point sets so that we define the i3D data-based (I_{i3D}) and the MLS range images (I_{MLS}) over the same discrete pixel lattice R. Since we search for changes caused by moving or static scene objects, before range image mapping, we remove ground points, yielding two obstacle clouds denoted by $\mathscr{P}_{\text{i3D}}$ and $\mathscr{P}_{\text{MLS}}$, which we wish to compare.

Algorithm 2. The main steps of the proposed RangeMRF change detection algorithm, using a Markov Random Field model in the range image domain.

1: **procedure** RangeMRF($\mathscr{P}_{\text{i3D}}, \mathscr{P}_{\text{MLS}}$)
2: $I_{\text{i3D}} \leftarrow \mathscr{P}_{\text{i3D}}$
3: $I_{\text{MLS}}, L_{\text{MLS}} \overset{ray-tracing}{\leftarrow} \mathscr{P}_{\text{MLS}}$
4: $d, \delta \leftarrow (I_{\text{i3D}}, I_{\text{MLS}}, L_{\text{MLS}})$
5: $L_{\text{i3D}} = MRF(d, \delta)$
6: **return** L_{i3D}

Exploiting the principle of operation of RMB Lidar sensors, the obstacle cloud $\mathscr{P}_{\text{i3D}}$ of a considered i3D frame can be represented as a range image I_{i3D} in a straightforward way. Here the laser emitters and sensors are vertically arranged, and each sensor scans the environment along a circular trajectory of 360° (see Fig. 1). Within a time frame, the consecutive range measurements of the ith sensor are stored in the ith row of the I_{i3D} image. This transform is geometrically equivalent to converting the representation of the point cloud from the 3D Descartes to a spherical polar coordinate system, where the polar direction and azimuth angles correspond to the horizontal and vertical pixel coordinates, and the distance is encoded in the corresponding pixel value. As a result of discretization, the range values of some image pixels might be undefined, which issue is handled by interpolation (Fig. 6(a), (b)). Apart from these issues, with using an appropriate, sensor-specific image resolution (used 1042×64 in the paper), the conversion of the point clouds to 2D range images is reversible, without causing information loss.

We derive the I_{MLS} background range map from the 3D MLS measurement ($\mathscr{P}_{\text{MLS}}$) with a ray-tracing approach. Relying on the previously introduced point cloud registration step, we emit simulated rays into the MLS point cloud from the estimated actual center position of the AV mounted RMB Lidar, and project the point distances to a spherical surface, on which we stretch an image lattice R having the same size and resolution parameters as the I_{i3D} range image. Hereby, utilizing that the reference MLS cloud is semantically segmented, we assign to each pixel $s \in R$ besides its calculated range value $I_{\text{MLS}}(s)$ a binary vegetation indicator label $L_{\text{MLS}}(s) \in \{V, \neg V\}$ based on the semantic class of the corresponding projected MLS point (Fig. 6(c), (d)).

In the next step, we construct a Markov Random Field (MRF)-based model for efficient estimation of the changes between the RMB Lidar's I_{i3D} and the MLS-based I_{MLS} range images, with also considering the L_{MLS} vegetation indicator map (Algorithm 2). The output change map of the MRF model is denoted by L_{i3D}. By assuming the presence of purely solid-shaped objects in the scene (such as vehicles, pedestrians, traffic signs), change detection could be considered as a binary classification problem with *foreground* (i.e. changes) and *background* (unchanged regions) classes, with applying background subtraction for foreground extraction. It is essential here to recognize even small changes between the inputs, such as pedestrians close to a bus station, therefore, this detection process must work with high sensitivity. However, we experienced that in this approach several false/irrelevant change predictions may occur, especially in vegetation regions, whose appearance widely varies in time, across different seasons. To overcome this artifact, we introduced a third class, called *seasonal change*. In summary, in the proposed change detection step, we distinguish three classes: (a) seasonal changes (S) in vegetation regions, (b) foreground changes (F) caused by moving objects or changed/re-located static street furniture elements, and (c) unchanged background (B). By handling the vegetation areas in a specific manner with reduced sensitivity, we may lose some information in extreme situations such as pedestrians hiding in trees or bushes, however, these cases are very rare and less relevant for analyzing traffic scenarios. Furthermore, the proposed method will be able to sharply recognize pedestrians standing near stations or facades, while also eliminating several false hits in vegetation areas.

Formally, the goal is to perform the following mapping:

$$L_{\text{i3D}} = \text{MRF}(I_{\text{i3D}}, I_{\text{MLS}}, L_{\text{MLS}}), \tag{5}$$

where $L_{\text{i3D}}(s) \in \{F, S, B\}$ for all $s \in R$.

To set up the MRF energy function, we define two distance values first which can give us reliable information about separating the different classes:

- The *geometric distance* $d(s, s')$ represents the depth difference between the corresponding range values of a given pixel s in the I_{i3D} and s' in the I_{MLS} images, respectively, which is calculated as follows:

$$d(s, s') = |I_{\text{i3D}}(s) - I_{\text{MLS}}(s')|. \tag{6}$$

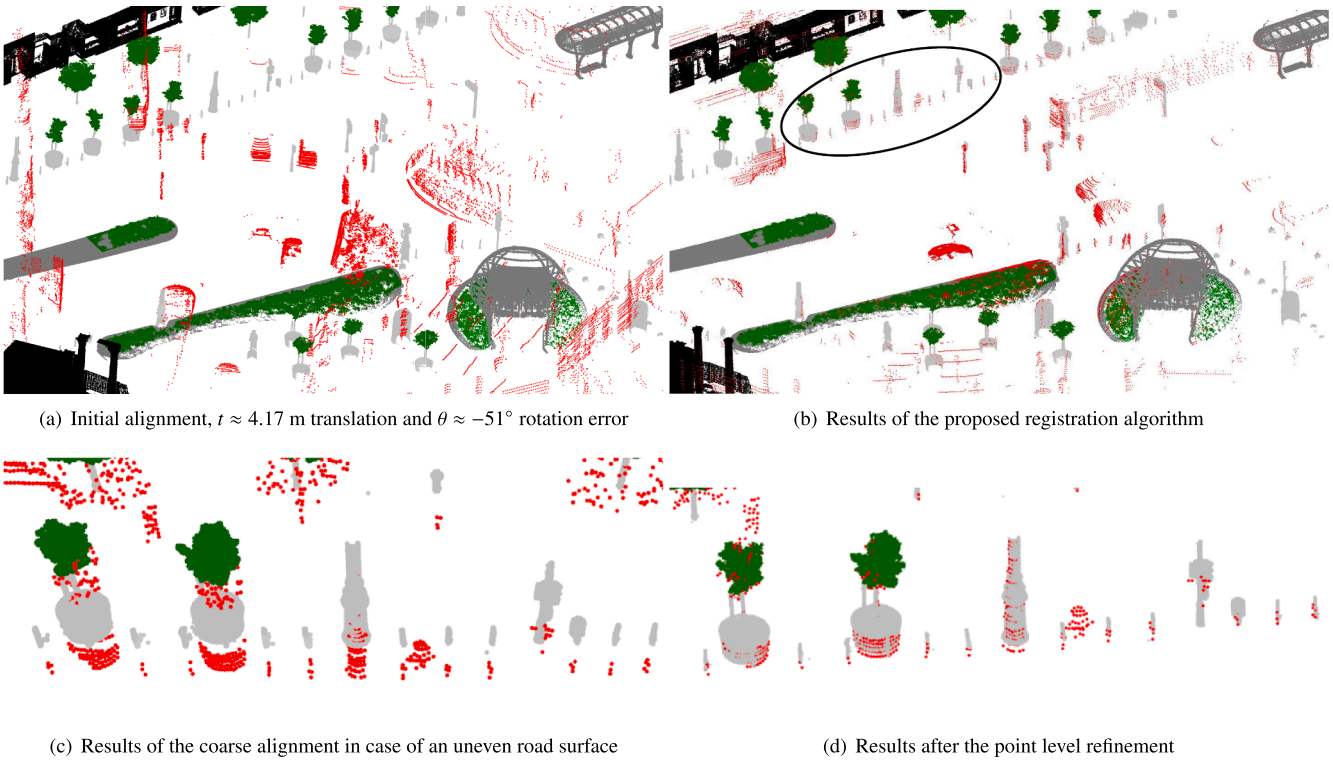
- The *vegetation distance* $\delta(s)$ informs us if the current pixel s is likely in vegetation regions. This parameter is calculated as the L2 distance between the locations of a given s pixel in I_{i3D} and the nearest pixel s' in I_{MLS} which has a vegetation label:

$$\delta(s) = \sqrt{(s_x - s'_x)^2 + (s_y - s'_y)^2}, \tag{7}$$

where pixel $s'(s'_x, s'_y)$ fulfills that $L_{\text{MLS}}(s') = V$, and the L2 distance $\sqrt{(s_x - s'_x)^2 + (s_y - s'_y)^2}$ between (s, s') is minimal among all nearby pixels.

Next, for every pixel in the range image lattice R, we define fitness scores for each class as functions of the above-described d geometric and δ vegetation distance values, in order to measure how a given pixel s fits the different classes using expert knowledge-based soft constraints. The types of the functions were chosen by experiments, by investigating the empirical distributions of the distance values within different regions of the training data, which is demonstrated in Fig. 8.

Dynamic foreground regions (F) should typically have increasing fitness scores with growing geometric distances ($d(s, s)$) between the corresponding pixel range values in the I_{i3D} and I_{MLS} maps. This soft constraint can be modeled by a *logistic* function with zero midpoint ($d_0 =$



(a) Initial alignment, $t \approx 4.17$ m translation and $\theta \approx -51°$ rotation error

(b) Results of the proposed registration algorithm

(c) Results of the coarse alignment in case of an uneven road surface

(d) Results after the point level refinement

Fig. 5. Results of the proposed point cloud registration algorithm. Subfigures (c) and (d) refer to the same area circled by black in subfigure (b). Color codes: **i3D points** are shown with red, the segmented MLS regions are marked by various colors depending on their semantic classes (**facade**, **vegetation**, **street furniture**, pillar-like column). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

0) and two parameters: maximum value (L) and steepness (k):

$$\mathscr{F}_F(s) = \frac{L}{1 + e^{-k(d(s,s)-d_0)}}. \tag{8}$$

On the contrary, static background (B) pixels have high fitness for small geometric distances, which allows us to use the same *logistic* function with a negative steepness parameter:

$$\mathscr{F}_B(s) = \frac{L}{1 + e^{k(d(s,s)-d_0)}}. \tag{9}$$

Seasonal change regions (S) have high fitness scores if and only if both the geometric distance term and the spatial distance to the closest vegetation pixel are near to zero. Formally, this constraint can be described by a *2D Gaussian* function with zero means ($\mu_d, \mu_\delta = 0$) and predefined small standard deviation (σ_d, σ_δ) parameters (Fig. 8):

$$\mathscr{F}_S(s) = \frac{1}{2\pi\sigma_d\sigma_\delta} e^{-\left[\left(\frac{d(s,s')-\mu_d}{2\sigma_d}\right)^2 + \left(\frac{\delta(s)-\mu_\delta}{2\sigma_\delta}\right)^2\right]}, \tag{10}$$

where pixel s' denotes the nearest vegetation pixel to a given pixel s. The chosen fitness functions and parameters are also summarized in Table 1.

The proposed change detection algorithm assigns a unique label $l_s \in \{F, S, B\}$ to every $s \in R$ pixel of the lattice, which minimize a Potts-like energy function:

$$E = \sum_{s \in R} -\log(\mathscr{F}_{l_s}(s)) + \sum_{s \in R} \sum_{s^* \in N_s} \beta \cdot 1\{l_s \neq l_{s^*}\}, \tag{11}$$

where $\beta > 0$ is responsible for obtaining smooth, connected regions in the segmented image, and N_s denotes the eight-neighborhood of pixel s. The MRF's output 2D change map is taken as $L_{\text{i3D}}(s) = l_s \ \forall s \in R$.

For the minimization of the MRF energy function E, a quick graph-cut based optimization method has been adopted (Boykov and Kolmogorov, 2004), which provides a high-quality three-class change map ($\{F, S, B\}$) in real-time as demonstrated in Fig. 6(e). As the last step, the labels available in the range image domain should be projected back to the corresponding points of the RMB Lidar point cloud (Fig. 7). Fig. 6e) and Fig. 7 show results for the same scene in the 2D and 3D domains, respectively.

4. Evaluation benchmark

Although several benchmarks (Geiger et al., 2012; Caesar et al., 2019) are available regarding 3D urban scene analysis based on onboard AV measurements (Lidar, camera, IMU, GPS), none of them contains dense 3D reference maps. Although the nuScenes dataset (Caesar et al., 2019) has a map expansion module, the map thereby refers to a top-view projection of the scene with semantic layer information. For joint utilization of both i3D Lidar measurements and MLS data, we evaluated the proposed algorithms on a new *SZTAKIBudapest* Benchmark that contains i3D point cloud streams captured by a Velodyne HDL64E 64-beam RMB Lidar sensor from different downtown areas of Budapest, where high-density, geo-referred point cloud maps are also recorded by a Riegl VMX450 MLS scanner. The Benchmark contains three different road scenarios, each one covering a path of around 300 meters with segmented reference MLS data (Nagy and Benedek, 2019). For quantitative evaluation, Ground Truth (GT) annotations verified by operators are available. The dataset with a detailed description is publicly available at the following link[1].

5. Evaluation

This section presents various numerical and qualitative results on the above-described *SZTAKIBudapest* Benchmark, which demonstrate the efficiency of the proposed approach and its superiority versus the state-

[1] https://github.com/sztaki-geocomp/Lidar-SCU#Dataset

(a) Raw range image generated from an i3D point cloud

(b) Interpolated range image (I_{i3D}) from subfigure (a)

(c) Reference range image derived by the MLS data via ray-tracing (I_{MLS})

(d) Vegetation label propagated from the semantic MLS data (L_{MLS})

(e) Change map: output of the MRF (L_{i3D})

Fig. 6. Change segmentation in the range image domain. Subfigures (a)-(c) are depth images where brighter pixels denote closer distance, and black pixels contain no measurements. Subfigure (d) displays semantic labels of the MLS data, where vegetation is marked by green. Subfigure (e) shows the segmentation output of the i3D data, where the pixels for static background are displayed by blue, for dynamic change by red, and for seasonal change by green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Fitness functions and their parameters.

	Class		
Param.	**F**	**B**	**S**
Geometric dist.	$d(s,s) \gg 0$	$d(s,s) \to 0$	$d(s,s') \approx 0$
Vegetation dist.	-	-	$\delta(s,s') \approx 0$
Fitness function	logistic	logistic	2D Gaussian
Parameters	L, k	$L, -k$	σ_d, σ_δ
Parameter dim.	2	0*	2

* The same L, k parameters are used for the **F** and **B** classes.

of-the-art reference techniques.

5.1. Multimodal point cloud registration

We begin the experiments with the evaluation of the proposed multimodal point cloud registration technique presented in Sec. 3.1. First, to justify the need for developing a new registration algorithm for the crossmodal i3D and MLS data alignment task, we demonstrate the limitations of methods based on existing keypoint selection strategies (discussed in Sec. 2.1) adopted for the sparse RMB Lidar and dense MLS point clouds, respectively. As shown in Table 2, the numbers of keypoints extracted from given object clusters (such as short and tall pillars, tree trunks, etc.) are in different orders of magnitude in point clouds captured by the different sensors, while Fig. 9 (a)-(d) show visually that we are unable to detect the same or even similar keypoints from the

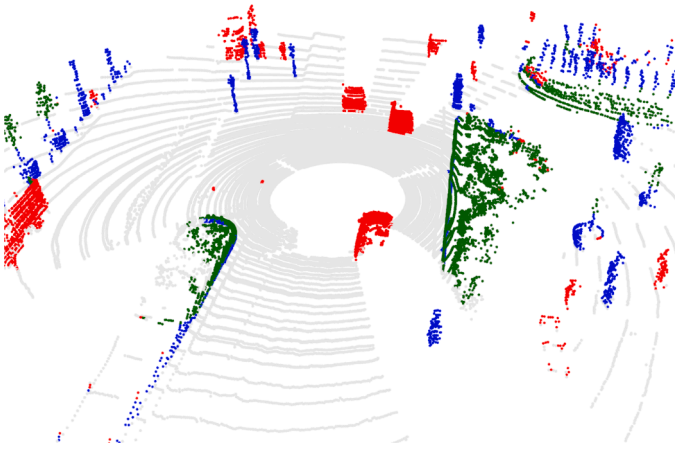
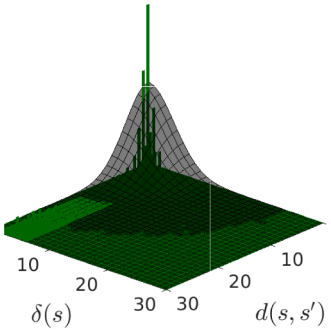
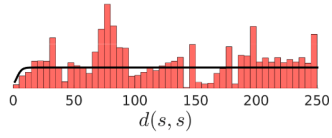


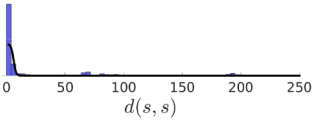
Fig. 7. Result of the change detection process in the 3D space about the same area as in Fig. 6. Color codes: static background, dynamic foreground, seasonal change, ground points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Seasonal change (S)



(b) Foreground change (F)



(c) Unchanged background (B)

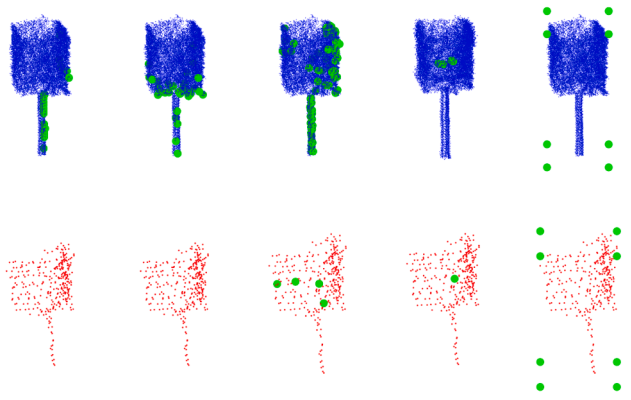
Fig. 8. The distribution of distance values (d, δ) and the applied fitness functions (marked with **black**) for each change class.

Table 2
Typical numbers of the extracted keypoints by different handcrafted and learning-based methods in MLS and i3D point clouds

Method	Extracted keypoints per cluster	
	MLS cloud	i3D cloud
Harris 3D	10–20	-
SIFT 3D	30–50	-
ISS 3D	100+	1–10
RSKDD-Net2020	5–20	0–5

sparse i3D and dense MLS scans of a 3D street object. The RSKDD-Net (Lu et al., 2020) strategy extracts the less different keypoints from the two point clouds, however, their similarity is still not enough for proper registration (see also Table 4). Among the region-based methods, we also tested the SegMap (Dubé et al., 2018) approach adopted to the i3D and MLS data (Fig. 10). Although the method extracts correctly the point cloud segments of most standalone static landmark objects in the i3D data (red points), it fails to find correct correspondences between segments of the i3D and MLS point clouds (black arrows) due to their local contextual differences (especially in point density).

We continue the discussion with the evaluation of the proposed Hough space-based registration technique. The efficiency of the algorithm can be demonstrated by qualitative examples, even in cases of very large initial alignment errors (see Fig. 5). For relevant quantitative



(a) Harris 3D (b) SIFT 3D (c) ISS 3D (d) RSKDD-Net2020 (e) Proposed

Fig. 9. 3D keypoint selection strategies for registration on the same object stored in the MLS (top) and i3D point. cloud (bottom).

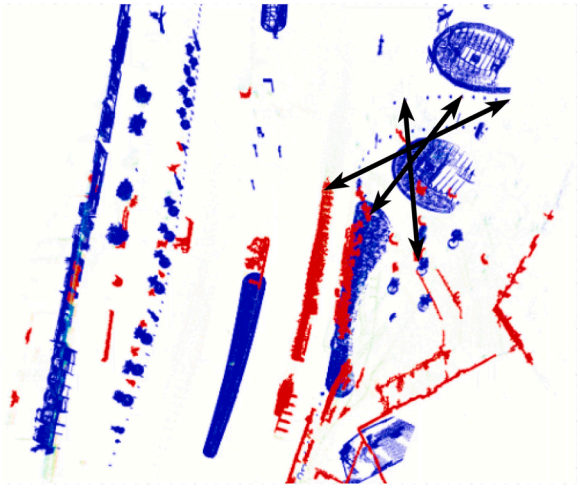


Fig. 10. False correspondences (**black arrows**) using the SegMap2018 approach (Dubé et al., 2018) between the sparse i3D local map and dense MLS global map.

analysis, the *SZTAKIBudapest* Benchmark contains Ground Truth transformation matrices derived through manual point cloud alignment, which enables direct comparison in the rigid transformation's independent parameter space (three rotation and three translation components) (Park et al., 2019). However, this manner of comparison has some limitations, as in several cases the error of only a single parameter can significantly distort the final transformation. Moreover, the manual registration process definitely includes some uncertainties (up to 10–20 cm translation, 0.5-1° rotation). For this reason, we calculated two further point-to-point distance metrics (Lehtola et al., 2017) to measure how well the given i3D frames can fit the global MLS segments. First, we considered the *Modified Hausdorff Distance (MHD)* between the $\mathscr{P}_{\text{i3D}}$ and $\mathscr{P}_{\text{MLS}}$ obstacle clouds:

$$Q_{\text{MHD}}(\mathscr{P}_{\text{i3D}}, \mathscr{P}_{\text{MLS}}) = \frac{1}{\#\mathscr{P}_{\text{i3D}}} \sum_{p \in \mathscr{P}_{\text{i3D}}} \min_{q \in \mathscr{P}_{\text{MLS}}} ||p - q||,$$

where $\#\mathscr{P}$ marks the number of points in set $\mathscr{P}$. However, for some scenes with many dynamic objects (including large vehicles), the Q_{MHD} metrics proved to be less relevant regarding the evaluation of the results, due to many outlier values which were accumulated. Therefore, we also turned to an alternative measure referred to as *Median Point Distance (MPD)*, which ranks the points in $\mathscr{P}_{\text{i3D}}$ by their $\min_q ||p-q||$ values and the median distance over all $p \in \mathscr{P}_{\text{i3D}}$ is calculated:

$$Q_{\text{MPD}}(P_{\text{i3D}}, P_{\text{MLS}}) = \underset{p \in P_{\text{i3D}}}{\text{Med}} \min_{q \in P_{\text{MLS}}} ||p - q||.$$

Note that the calculation processes of both MHD and MPD are unsupervised, thus they can be considered as relative quality measures rather than metric error values.

Table 3 displays the MHD and MPD rates for the three different scenes of the *SZTAKIBudapest* Benchmark, calculated at different stages of the proposed two-step registration algorithm. We can see that both quality values are significantly reduced by the process, and in all test scenarios, the final observed MPD error values are around 1-3 cm. Note that high registration accuracy can also be verified by qualitative analysis (see Fig. 5).

Table 4 presents a detailed quantitative comparison between the proposed method and different reference approaches for point cloud registration using the *SZTAKIBudapest* Benchmark. Quality measures provided here comprise both the differences in the transformation's independent parameters and the point distance-based MHD and MPD error rates. For each method, the GPS-based position was taken as an initial alignment. Based on the experiments, the one-step Least-Square (Arun et al., 1987) optimization can decrease the overall point distances but completely fails to find the optimal position and orientation when the density characteristics of the two point clouds significantly differ. Similarly, the RSKDD-Net (Lu et al., 2020) method cannot improve the original alignment due to the lack of extractable keypoint correspondences (see Fig. 9(d)). The high rotation errors of these methods (Arun et al., 1987; Lu et al., 2020) along the x (Δr_x) and z-axes (Δr_z) in Table 4 correspond to the large percent of test cases where the alignable point clouds were falsely flipped horizontally or vertically, respectively. The FMR2020 (Huang et al., 2020) method found false feature-matches derived from the local point-distribution differences of the point sets, resulting in higher translation errors along the x and y-axes. The ICP algorithm (Zhang, 1994) – which aims to minimize locally the errors – failed to converge from the poor GPS-based position in around 75% of the test frames, and for the remaining cases, the accurate (< 10 cm) registration needed 3-4 minutes computation time for a single frame. The further global registration techniques, RANSAC (Choi et al., 2015) and TEASER++ (Yang et al., 2021) have been applied as follows: first, the two point clouds were uniformly downsampled, then the FPFH (Rusu et al., 2009) feature descriptors were calculated to extract the local geometric properties of the point neighborhoods. Although these approaches could reduce both the position and orientation errors, they proved to be less accurate than the proposed method due to many outliers, which were eliminated by the proposed approach with the voting scheme. Note that the RANSAC-based registration also rotated the point clouds upside down in around 10% of the test frames, resulting in relatively higher rotation error along the x-axis (Δr_x).

Regarding the computation time, the coarse alignment step of the proposed algorithm can work with around 10 fps on a desktop environment, while the fine alignment steps need currently 2-4 seconds on sequential CPU implementation (see the last column of Table 4). However, recent studies (Qiu et al., 2009) show that using parallel implementation of the correspondence search in ICP, one can reduce its computational time with one order of magnitude on multi-core CPU (OpenMP) and with two orders of magnitude on GPU (CUDA). These hardware-accelerated implementations can make the proposed algorithm eligible for real-time applications.

5.2. Change detection

Next, we evaluate the change detection step of the proposed algorithm, which takes as input registered i3D and MLS point clouds, as a result of the previously discussed point set alignment. Since the *SZTAKIBudapest* Benchmark contains GT labels for the changed regions, we

Table 3
Quantitative results of the proposed registration algorithm.

Scenario	Metrics								
	Initial avg. offset [m]	Initial avg. rotation [°]	Q_MHD [m]			Q_MPD [m]			Computational time [s] (sequential CPU impl.)
			initial	coarse	fine	initial	coarse	fine	
Deák square	1.4	−54.98	5.5581	0.6806	0.6306	3.1395	0.0589	0.0185	1.58384
Fővám square	2.2	53.04	4.8530	1.0918	0.9561	2.1473	0.1612	0.0186	2.31684
Kálvin square	3.6	38.53	5.5040	0.8719	0.7454	3.3839	0.2853	0.0380	3.27187
Average values	2.4	48.85	5.2982	0.8818	0.7785	2.8732	0.1644	0.0246	2.59418

Table 4
Comparative evaluation of various point cloud registration methods and the proposed approach

Method	Absolute error compared to manual Ground Truth transformation						Distance-based evaluation		Computation time [s]
	Δr_x [°]	Δr_y [°]	Δr_z [°]	Δt_x [m]	Δt_y [m]	Δt_z [m]	Q_MHD [m]	Q_MPD [m]	
GPS baseline	1.831	1.492	48.784	1.485	5.999	1.103	5.298	2.873	-
SegMap2018	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]
RSKDD-Net2020	94.037	6.670	76.047	9.466	15.065	1.241	2.968	2.789	**0.048**[b]
Least-Square	89.335	9.117	85.655	5.516	5.493	0.627	2.510	2.176	0.364
ICP	1.951	2.170	30.778	2.023	5.411	0.608	1.483	0.700	239.768[c]
RANSAC	19.921	2.434	7.263	3.518	7.422	0.853	2.318	1.877	0.793
TEASER++2020	5.522	1.664	12.350	1.613	5.477	1.162	1.818	1.134	3.849
FMR2020	9.062	4.544	19.105	5.137	5.234	1.180	3.392	3.387	25.596[c]
Proposed coarse	1.831	1.492	**0.812**	**0.199**	**0.273**	**0.333**	**0.882**	**0.164**	**0.102**
Proposed with ICP	**0.598**	**0.959**	**0.333**	**0.069**	**0.154**	**0.183**	**0.779**	**0.025**	2.594[c]

[a] failed to find correspondences
[b] GPU-accelerated impl. (on CPU: 1.0844 [s])
[c] Sequential CPU impl. (GPU-acceleratable)

have also performed here quantitative evaluation at point level, so that we compared the labeled output i3D data to the GT annotation of the point cloud.

5.2.1. MRF parameter settings

During the evaluation, first, we compared more than 30000 parameter combinations in a multi-level five-dimensional grid search in order to optimize the parameters $(L, k, \sigma_d, \sigma_\delta, \beta)$ of the MRF model. The ROC curve of the results of different setups is summarized in Fig. 11. We achieved the best performance using the MRF smoothness parameter

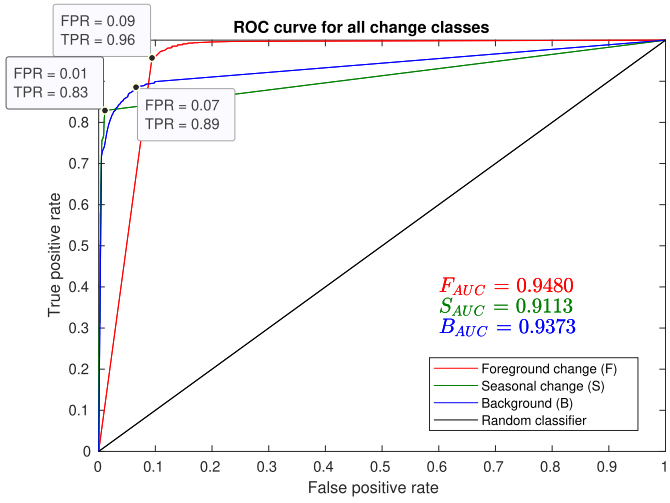


Fig. 11. The behavior of the proposed method using multiple parameter combinations $(L, k, \sigma\_d, \sigma\_\delta, \beta)$.

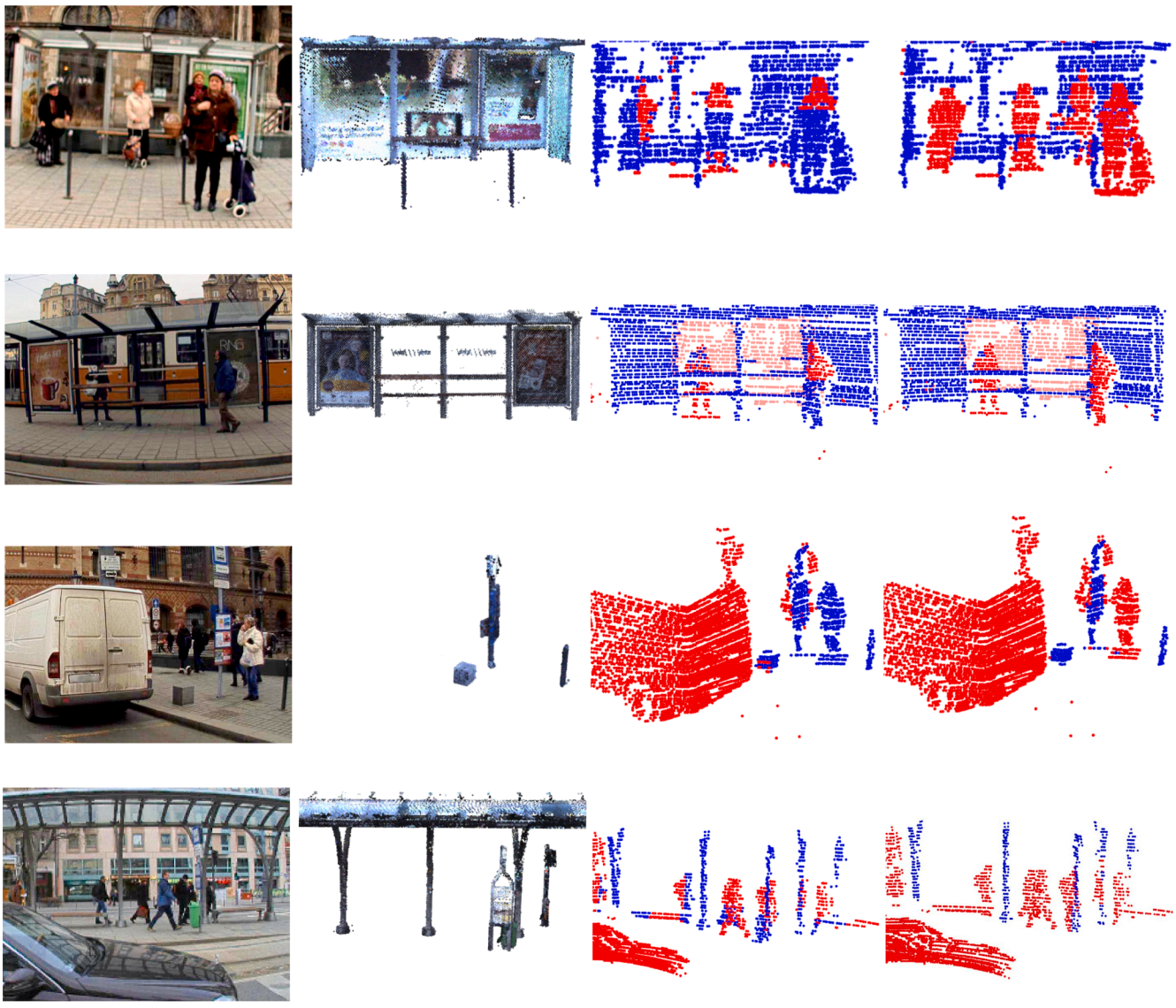
$\beta = 0.5$, the logistic function with $L = 0.01, k = 2.0$ and the Gaussian function with deviation parameters $\sigma_d = 1.4$ and $\sigma_\delta = 2.5$.

5.2.2. Reference methods

Since to our best knowledge the proposed RangeMRF method is the first approach dedicated to the i3D-MLS crossmodal change detection task, we selected and adopted reference techniques from the methods described in Sec. 2.2, which were proposed earlier for comparing registered MLS point clouds.

First, we implemented a fixed radius ($r = 15$ cm based on (Schlichting and Brenner, 2016)) nearest neighbor (NN) search (Schlichting and Brenner, 2016) between segments of the two point clouds. We labeled each segment of the i3D data as *foreground (F)* if 25% of its points have no neighbors, otherwise, we propagated the label of the neighboring MLS points to the corresponding segment (*background (B)* or *vegetation (V)*). Next, we implemented the radius search at point level which is used by (Liu et al., 2021; Girardeau-Montaut et al., 2005): Each point of the i3D is labeled as *F* if the distance to the nearest MLS point is higher than *r*, otherwise the label of the neighboring point is propagated. We tested this method with multiple *r* radius lengths (see Fig. 13). At small radius, the method failed to compensate the measurement noise and the variance of vegetation areas in the i3D data, while with higher radius values it produces a significantly increasing number of false-negative points. For numerical comparison, we applied an optimally balanced solution for both classes ($r = 30$ cm). To overcome the above-mentioned trade-off, we also implemented a point-to-triangle search based on (Xiao et al., 2015): for each point in the i3D data, we constructed a triangle surface from its 10 nearest points and calculated the distance ($d = 30$ cm applied based on (Xiao et al., 2015)) of the given point to the nearest triangle.

Finally, we also fitted a 3D voxel array (Liu et al., 2016) to the



(a) Camera images (for visual check) (b) Reference area in the MLS map (c) Voxel-based change detection (Liu et al., 2016) (d) Proposed change detection

Fig. 12. Change detection results in crowded sidewalk areas, with the presence of many static and dynamic objects. Camera images of column (a) were taken synchronously with the RMB Lidar points clouds, but they are only used for visual verification. Purely Lidar-based detection results are shown in column (c) for a reference method, and in column (d) for the proposed method: red points correspond to dynamic change (vehicle, tram, pedestrians, further objects missing from the map: ticket station, benches), blue points present static environment parts. (Note: shading the red color in the second row is only applied for better visualization.).

aligned point clouds and classified the given i3D points as *foreground* (*F*) if its voxel was empty in the MLS point cloud. Otherwise, the label of the MLS point was propagated. We evaluated this technique with various w voxel size parameters (Fig. 14) and and experienced that for the i3D-MLS data samples it provides the most efficient results with $w = 100$ cm settings, which also gave the best trade-off between the accuracy and computational time.

5.2.3. Comparative results

While we have observed that in point cloud regions with several large, non-contacting vehicle objects, the reference techniques have similarly high accuracy to the proposed method, we have experienced notable advantages of the RangeMRF model on challenging situations captured in cluttered sidewalk regions, containing multiple pedestrians and a wide variation of static objects, such as benches, boxes, columns (see Fig. 12). For this reason, we have focused the quantitative comparison of the methods on the cluttered street regions. Results are summarized in Table 5 displaying precision, recall, and F1-score metrics calculated for all approaches.

In these experiments, we have measured the lowest F1-scores with the segment level NN technique (Schlichting and Brenner, 2016). As main drawback here, the extracted segments often merge small dynamic traffic participants and large static environment parts. Therefore, such mixed regions were predicted falsely as unchanged regions, resulting in low recall values during the evaluation.

Performing the radius search at point level (Liu et al., 2021; Girardeau-Montaut et al., 2005) rather than across segments produced better results, however, this point level approach could not compensate for the irregular density characteristics of the i3D data. The voxel-based approach (Liu et al., 2016) outperformed both techniques regarding the F1-score metrics. However, all of them are not or just partially able to find pedestrians staying near to the static object (with a distance under voxel size, or under radius length, respectively), yielding many false-negative points. Conversely, the proposed RangeMRF model is able to

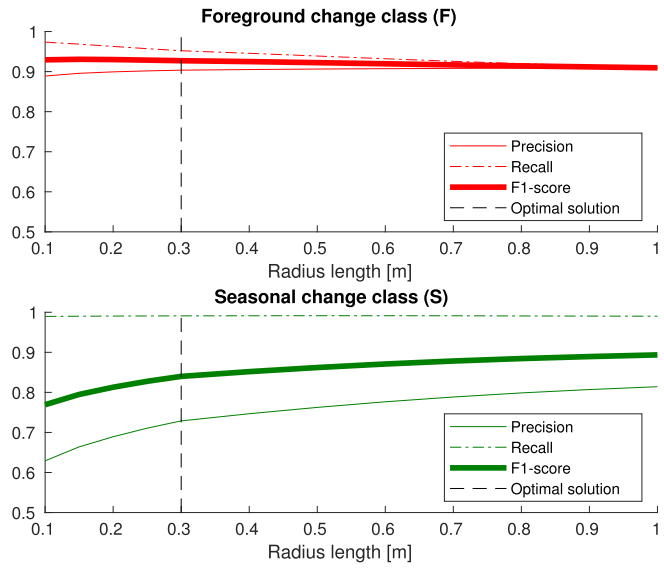


Fig. 13. The behavior of the point level radius nearest neighbor (NN) search (Liu et al., 2021; Girardeau-Montaut et al., 2005) using multiple r parameters. As a balanced solution, $r = 30$ cm was chosen.

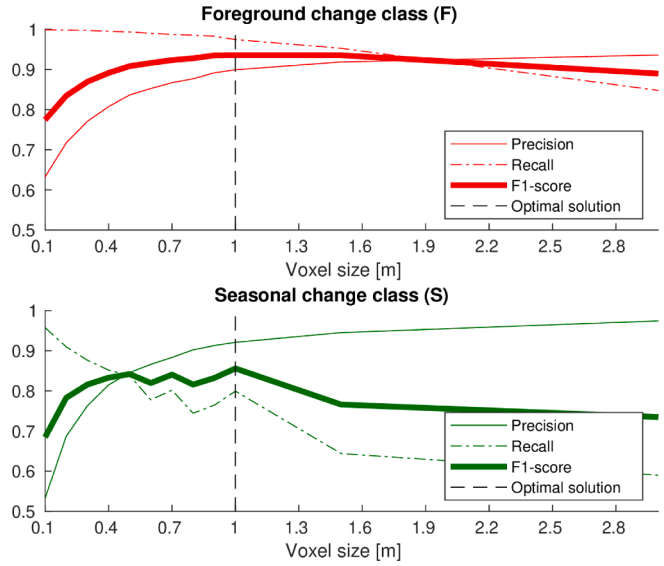


Fig. 14. The behavior of the voxel-based reference method (Liu et al., 2016) using multiple w parameters. The optimal solution for both classes is wit.h $w =$ 100 cm.

Table 5
Comparative evaluation of various change detection methods and the proposed RangeMRF approach in cluttered areas.

Method	Foreground change class (F)			Comp.
	Precision	Recall	F1-score	time [s]
Segment-NN	0.8549	0.4834	0.6176	0.5649
Point-NN	**0.9763**	0.5982	0.7419	1.2934
Point-Triangle	0.9011	0.7993	0.8472	215.78
VOXEL	0.8017	0.7902	0.7959	2.5176
RangeMRF	0.8695	**0.8769**	**0.8732**	**0.1426**

distinguish the static and dynamic point regions, which difference is demonstrated by a few qualitative samples in Fig. 12 as well. By applying the method of (Xiao et al., 2015), which compares points to nearest triangles, we could remarkably reduce the number of false-negative hits. However, building a surface from the neighbors of each point is a time-consuming step, the computation of one measurement frame takes here around 3-4 minutes using a desktop computer. The proposed MRF model's segmentation step, which works in the range image domain of each frame, needs around 145 ms, while the voxel-based approach takes around 2 (respectively 7) seconds for a given frame with parameter setting $w = 100$ cm (respectively $w = 10$ cm) using the same computer setup.

In summary, Fig. 7, Fig. 12, and Table 5 confirm the efficiency of the proposed RangeMRF method in real-world scenarios. We have quantitatively demonstrated its advantage versus the reference models, which produced either at least 7% lower F1-score rates on cluttered point clouds regions (Liu et al., 2021; Girardeau-Montaut et al., 2005; Schlichting and Brenner, 2016; Liu et al., 2016), or they needed a significantly longer computational time (Xiao et al., 2015).

6. Conclusion

This paper presented a novel method for multimodal registration and change detection between point clouds with significantly different density characteristics. The proposed algorithm can be used to localize sparse and instantly sensed point cloud data captured by autonomous vehicles in a high-density 3D point cloud map with an accuracy of up to a few centimeters in Median Point Distance, and based on the registered point clouds, it can separate different sources of changes with an overall F1-score of 92% for complete i3D point cloud frames and 87% for complex sidewalk areas. We estimate that with a parallel implementation of many substeps, the whole workflow can run with around 5–10 fps. We proposed significant novel contributions in terms of robust alignment and efficient change detection between multimodal 3D data. We also provided several experiments to show the advantages of the new methods versus the state-of-the-art.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially funded by the National Research, Development and Innovation (NRDI) Office within the frameworks of the Autonomous Systems National Laboratory, and the Artificial Intelligence National Laboratory programs, and the TKP2021-NVA-01 project. The work of Ö. Zováthi was partially funded by the KDP-2020 Cooperative Doctoral Program (KDP-977852) and the ÚNKP-21–3 New National Excellence Program (ÚNKP-21–3-I-PPKE-14) from the source of the NRDI Fund.

Appendix A. Supplementary attachments

The used dataset and the source code of the proposed algorithm is publicly available at:
https://github.com/sztaki-geocomp/Lidar-SCU.

References

Aljumaily, H., Laefer, D.F., Cuadra, D., 2017. Urban point cloud mining based on density clustering and mapreduce. J. Comput. Civ. Eng. 31, 04017021. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000674.

Arun, K.S., Huang, T.S., Blostein, S.D., 1987. Least-squares fitting of two 3-D point sets. IEEE Trans. Pattern Anal. Mach. Intell. 9, 698–700. https://doi.org/10.1109/TPAMI.1987.4767965.

Bayat, B., Crasta, N., Crespi, A., Pascoal, A.M., Ijspeert, A., 2017. Environmental monitoring using autonomous vehicles: a survey of recent searching techniques. Curr. Opin. Biotechnol. 45, 76–84. https://doi.org/10.1016/j.copbio.2017.01.009.

Behley, J., Steinhage, V., Cremers, A.B., 2012. Performance of histogram descriptors for the classification of 3D laser range data in urban environments. In: IEEE Int. Conf. Robot. Autom. St. Paul, MN, USA, pp. 4391–4398. doi:10.1109/ICRA.2012.6225003.

Benedek, C., Majdik, A., Nagy, B., Rózsa, Z., Szirányi, T., 2021. Positioning and perception in LIDAR point clouds. Digit. Signal Process. 119, 103193. https://doi.org/10.1016/j.dsp.2021.103193.

Börcs, A., Nagy, B., Benedek, C., 2017. Instant object detection in lidar point clouds. IEEE Geosci. Remote Sens. Lett. 14, 992–996. https://doi.org/10.1109/LGRS.2017.2674799.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. 26, 1124–1137. https://doi.org/10.1109/TPAMI.2004.60.

Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O., 2019. NuScenes: a multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027.

Cheng, L., Chen, S., Liu, X., Xu, H., Wu, Y., Li, M., Chen, Y., 2018. Registration of laser scanning point clouds: A review. MDPI Sens. 18, 1641. https://doi.org/10.3390/s18051641.

Choi, S., Zhou, Q., Koltun, V., 2015. Robust reconstruction of indoor scenes. In: IEEE Conf. Comput. Vis. Pattern Recognit. Boston, MA, USA, pp. 5556–5565. doi:10.1109/CVPR.2015.7299195.

Dong, Z., Yang, B., Liu, Y., Liang, F., Li, B., Zang, Y., 2017. A novel binary shape context for 3D local surface description. ISPRS J. Photogramm. Remote Sens. 130, 431–452. https://doi.org/10.1016/j.isprsjprs.2017.06.012.

Douillard, B., Quadros, A., Morton, P., Underwood, J.P., Deuge, M.D., Hugosson, S., Hallström, M., Bailey, T., 2012. Scan segments matching for pairwise 3D alignment. In: IEEE Int. Conf. Robot. Autom. St. Paul, MN, USA, pp. 3033–3040. doi:10.1109/ICRA.2012.6224788.

Dubé, R., Cramariuc, A., Dugas, D., Nieto, J., Siegwart, R., Cadena, C., 2018. SegMap: 3D segment mapping using data-driven descriptors. In: Robot.: Sci. Syst.. doi:10.15607/RSS.2018.XIV.003.

Ge, X., Hu, H., Wu, B., 2019. Image-guided registration of unordered terrestrial laser scanning point clouds for urban scenes. IEEE Trans. Geosci. Remote Sens. 57, 9264–9276. https://doi.org/10.1109/TGRS.2019.2925805.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: IEEE Conf. Comput. Vis. Pattern Recognit. Providence, RI, USA, pp. 3354–3361. doi:10.1109/CVPR.2012.6248074.

Girardeau-Montaut, D., Roux, M., Marc, R., Thibault, G., 2005. Change detection on point cloud data acquired with a ground laser scanner. ISPRS Arch. Photogramm. Remote Sens. Spatial Inf. Sci., 36.

Gressin, A., Cannelle, B., Mallet, C., Papelard, J.-P., 2012a. Trajectory-Based Registration of 3D LIDAR Point Clouds Acquired with a Mobile Mapping System. ISPRS Ann. Photogramm. Remote Sens. Spatial. Inf. Sci. I-3, 117–122. https://doi.org/10.5194/isprsannals-I-3-117-2012.

Gressin, A., Mallet, C., David, N., 2012b. Improving 3D LIDAR Point Cloud Registration Using Optimal Neighborhood Knowledge. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. I-3, 111–116. https://doi.org/10.5194/isprsannals-I-3-111-2012.

Guan, H., Li, J.Z., Yu, Y., Chapman, M.A., Wang, C., 2015. Automated road information extraction from mobile laser scanning data. IEEE Trans. Intell. Transp. Syst. 16, 194–205. https://doi.org/10.1109/TITS.2014.2328589.

Hänsch, R., Weber, T., Hellwich, O., 2014. Comparison of 3D interest point detectors and descriptors for point cloud fusion. ISPRS Ann. Photogramm. Remote Sens. Spatial. Inf. Sci. II-3, 57–64. https://doi.org/10.5194/isprsannals-II-3-57-2014.

Huang, X., Fan, L., Wu, Q., Zhang, J., Yuan, C., 2019. Fast registration for cross-source point clouds by using weak regional affinity and pixel-wise refinement. In: IEEE Int. Conf. Multimed. and Expo. Los Alamitos, CA, USA: IEEE Computer Society, pp. 1552–1557. doi:10.1109/ICME.2019.00268.

Huang, X., Mei, G., Zhang, J., 2020. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In: IEEE Conf. Comput. Vis. Pattern Recognit.. doi:10.1109/CVPR42600.2020.01138.

Huang, X., Mei, G., Zhang, J., Abbas, R., 2021. A comprehensive survey on point cloud registration.

Huang, X., Zhang, J., Fan, L., Wu, Q., Yuan, C., 2017. A systematic approach for cross-source point cloud registration by preserving macro and micro structures. IEEE Trans. Image Process. 26, 3261–3276. https://doi.org/10.1109/TIP.2017.2695888.

Ku, T., Galanakis, S., Boom, B., Veltkamp, R.C., Bangera, D., Gangisetty, S., Stagakis, N., Arvanitis, G., Moustakas, K., 2021. SHREC 2021: 3D point cloud change detection for street scenes. Comput. Graph. 99, 192–200. https://doi.org/10.1016/j.cag.2021.07.004.

Lang, A., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2019. PointPillars: Fast Encoders for Object Detection from Point Clouds. In: IEEE Conf. Comput. Vis. Pattern Recognit.. Long Beach, CA, USA, pp. 12689–12697. doi:10.1109/CVPR.2019.01298.

Lehtola, V., Kaartinen, H., Nuchter, A., Kaijaluoto, R., Kukko, A., Litkey, P., Honkavaara, E., Rosnell, T., Vaaja, M., Virtanen, J.-P., Kurkela, M., Issaoui, A., Zhu, L., Jaakkola, A., Hyyppä, J., 2017. Comparison of the selected state-of-the-art 3D indoor scanning and point cloud generation methods. MDPI Remote Sens. 9, 796. https://doi.org/10.3390/rs9080796.

Li, J., Lee, G.H., 2019. USIP: Unsupervised stable interest point detection from 3D point clouds. In: IEEE/CVF Int. Conf. Comput. Vis. Seoul, Korea, pp. 361–370. doi:10.1109/ICCV.2019.00045.

Li, Y., Olson, E.B., 2010. Extracting general-purpose features from LIDAR data. In: IEEE Int. Conf. Robot. Autom. Anchorage, AK, USA, pp. 1388–1393. doi:10.1109/ROBOT.2010.5509690.

Liu, D., Li, D., Wang, M., Wang, Z., 2021. 3D change detection using adaptive thresholds based on local point cloud density. ISPRS Int. J. Geo-Inf. 10 https://doi.org/10.3390/ijgi10030127.

Liu, K., Boehm, J., Alis, C., 2016. Change detection of mobile lidar data using cloud computing. ISPRS Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLI-B3, 309–313.

Lu, F., Chen, G., Liu, Y., Qu, Z., Knoll, A., 2020. RSKDD-Net: Random sample-based keypoint detector and descriptor. In: Adv. Neural Inf. Process. Syst. virtual conference volume 33, pp. 21297–21308.

Ma, W., Tartavull, I., Barsan, I.A., Wang, S., Bai, M., Mattyus, G., Homayounfar, N., Lakshmikanth, S.K., Pokrovsky, A., Urtasun, R., 2019. Exploiting sparse semantic HD maps for self-driving vehicle localization. In: IEEE/RSJ Int. Conf. Intell. Robot. Syst. Macau, China, pp. 5304–5311. doi:10.1109/IROS40897.2019.8968122.

Mi, X., Yang, B., Dong, Z., Chen, C., Gu, J., 2021. Automated 3D road boundary extraction and vectorization using MLS point clouds. IEEE Trans. Intell. Transp. Syst. PP, 1–11. https://doi.org/10.1109/TITS.2021.3052882.

Nagy, B., Benedek, C., 2019. 3D CNN-based semantic labeling approach for mobile laser scanning data. IEEE Sens. J. 19, 10034–10045. https://doi.org/10.1109/JSEN.2019.2927269.

Nagy, B., Benedek, C., 2019. Real-time point cloud alignment for vehicle localization in a high resolution 3D map. In: Eur. Conf. Comput. Vis. Workshop. Munich, Germany volume 11129 of LNCS, pp. 226–239. doi:10.1007/978-3-030-11009-3_13.

Park, J., Kim, P., Cho, Y., Kang, J., 2019. Framework for automated registration of UAV and UGV point clouds using local features in images. Autom. Constr. 98, 175–182. https://doi.org/10.1016/j.autcon.2018.11.024.

Qiu, D., May, S., Nüchter, A., 2009. GPU-accelerated nearest neighbor search for 3D registration. In: Int. Conf. Comput. Vis. Syst. Liege, Belgium, pp. 194–203. doi:10.1007/978-3-642-04667-4_20.

Ratha, N.K., Karu, K., Chen, S., Jain, A.K., 1996. A real-time matching system for large fingerprint databases. IEEE Trans. Pattern Anal. Mach. Intell. 18, 799–813. https://doi.org/10.1109/34.531800.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: IEEE Int. Conf. Robot. Autom. Kobe, Japan, pp. 3212–3217. doi:10.1109/ROBOT.2009.5152473.

Rusu, R.B., Cousins, S., 2011. 3D is here: Point cloud library (PCL). In: IEEE Int. Conf. Robot. Autom. Shanghai, China, pp. 1–4. doi:10.1109/ICRA.2011.5980567.

Schlichting, A., Brenner, C., 2016. Vehicle localization by lidar point correlation improved by change detection. ISPRS Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLI-B1, 703–710. https://doi.org/10.5194/isprsarchives-XLI-B1-703-2016.

Sipiran, I., Bustos, B., 2011. Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes. The Vis. Comput. 27, 963–976. https://doi.org/10.1007/s00371-011-0610-y.

Theiler, P.W., Wegner, J.D., Schindler, K., 2014. Keypoint-based 4-points congruent sets – automated marker-less registration of laser scans. ISPRS J. Photogramm. Remote Sens. 96, 149–163. https://doi.org/10.1016/j.isprsjprs.2014.06.015.

Voelsen, M., Schachtschneider, J., Brenner, C., 2021. Classification and change detection in mobile mapping lidar point clouds. J. Photogramm. Remote Sens. Geoinf. Sci. 89, 195–207. https://doi.org/10.1007/s41064-021-00148-x.

Xiao, W., Vallet, B., Brédif, M., Paparoditis, N., 2015. Street environment change detection from mobile laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 107, 38–49. https://doi.org/10.1016/j.isprsjprs.2015.04.011.

Yang, H., Shi, J., Carlone, L., 2021. Teaser: Fast and certifiable point cloud registration. IEEE Trans. Robot. 37, 314–333. https://doi.org/10.1109/TRO.2020.3033695.

Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. Int. J. Comput. Vis. 13, 119–152. https://doi.org/10.1007/BF01427149.

Zhao, J., He, X., Li, J., Feng, T., Ye, C., Xiong, L., 2019. Automatic Vector-Based Road Structure Mapping Using Multibeam LiDAR. MDPI Remote Sens. 11, 1726. https://doi.org/10.3390/rs11141726.

Zheng, H., Wang, R., Xu, S., 2017. Recognizing street lighting poles from mobile lidar data. IEEE Trans. Geosci. Remote Sens. 55, 407–420. https://doi.org/10.1109/TGRS.2016.2607521.

Zhong, Y. (2009). Intrinsic shape signatures: A shape descriptor for 3D object recognition. In Int. Conf. Comput. Vis. Workshops (pp. 689–696). Kyoto, Japan. doi:10.1109/ICCVW.2009.5457637.