

# Surface-based GICP

Michiel Vlamincx, Hiep Quang Luong and Wilfried Philips  
Department of Telecommunications and Information Processing  
Ghent University, imec, Belgium  
Email: michiel.vlaminck@ugent.be

**Abstract**—In this paper we present an extension of the Generalized ICP algorithm for the registration of point clouds for use in lidar-based SLAM applications. As opposed to the plane-to-plane cost function, which assumes that each point set is locally planar, we propose to incorporate additional information on the underlying surface into the GICP process. Doing so, we are able to deal better with the artefacts that are typically present in lidar point clouds, including an inhomogeneous and sparse point density, noise and missing data. Experiments on lidar sequences of the KITTI benchmark demonstrate that we are able to substantially reduce the positional error compared to the original GICP algorithm.

**Keywords**—3D registration; point cloud; GICP; surface reconstruction

## I. INTRODUCTION

Point cloud registration has been widely studied in the past two decades. A large share of this research was conducted in view of *scan matching* for use in *simultaneous localization and mapping* (SLAM) applications, especially with the advent of novel 3D sensing modalities. Many problems can be enumerated regarding the registration of point clouds originated from these depth sensors. Figure 1 depicts the most common issues. Point clouds acquired by laser scanners for example often suffer from an inhomogeneous and sparse point density, due to the angle of incidence of the laser beams in combination with the proximity of objects in the scene. Among the other artefacts are the presence of sensor noise or outliers as the result of respectively inaccurate or faulty measurements. Furthermore, it is possible that a large part of the scene is not sampled at all because of occlusion, which is on its turn leading to missing data. Finally, in case of incremental registration, previously acquired point clouds could be imperfectly aligned. Especially the latter problem is not yet well-addressed in state-of-the-art registration algorithms. Our goal is to perform excellent scan matching for use in SLAM applications. All of the aforementioned artefacts should thus be carefully taken into account as they are almost always present in scan data, especially in data originated from lidar scanners. To this end, we perform a preliminary surface reconstruction step followed by the re-projection of the points on the estimated surface. This will account for the majority of the artefacts, including reducing the noise, eliminating the outliers and filling small gaps. In addition we adapt the cost function in order to exploit the

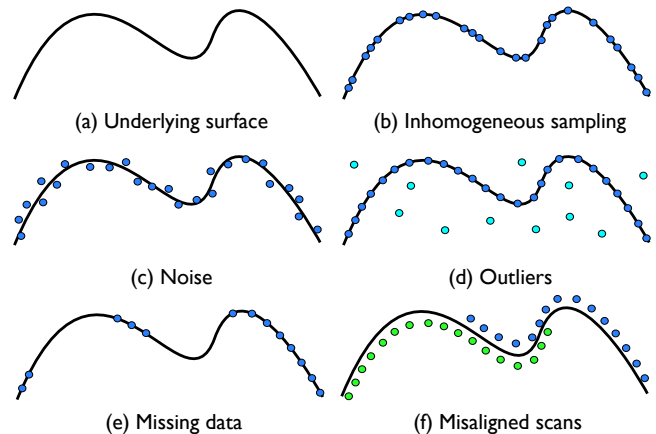


Figure 1. Several forms of artefacts in point cloud data, here represented in 2D.

knowledge about the underlying surface.

## II. RELATED WORK

Many laser-based SLAM systems use the *iterative closest point* (ICP) [1], [2] algorithm to estimate the transformation between consecutive point clouds produced by 3D sensors. This method has the main advantage that it is simple to use and easy to adapt. As a result, many researchers have proposed all kinds of improvements on this algorithm to cope with its initial weaknesses. One of the limitations is due to the ambiguous *data association* or correspondence estimation step. The original ICP assumes closest point correspondences but as point clouds are often the result of sampled surfaces, it is very unlikely that the exact same point is captured in the two different point clouds. He *et. al.* [3] therefore proposed to compute low-level geometrical features to guide the correspondence matching process in the ICP algorithm. In [4], the authors integrate an optimal neighbourhood selection step, examining the point cloud at several scales, to compute a similar set of low-level features. Marani *et. al.* [5] on their turn introduce the concept of a deletion mask, to remove the erroneous point correspondences which are extracted from ambiguous regions, i.e. regions where implicit differences can raise as a consequence of the change of the sensor view-point. Agamennoni *et. al.* [6] proposed another improvement, alle-

viating the correspondence association by adopting a probabilistic data association technique. Finally, as wrong point correspondences cause the ICP algorithm to suffer from local-minima, Yang *et al.* [7] proposed ‘Globally optimal ICP’, integrating a branch-and-bound (BnB) scheme which searches the 3D motion space. Besides the data association, one could also modify the actual cost function. One of the main developments in this respect was the generalization of the cost function, e.g. *point-to-point* and *point-to-plane*, by integrating it into a probabilistic framework, denoted as *Generalized ICP* (GICP), presented by Segal *et al.* [8]. As an alternative to point-to-point and point-to-plane, the authors propose the *plane-to-plane* cost function, based on the assumption that the environment is locally planar. Doing so they argue that the uncertainty of a point’s position along its normal is low - and depending on the sensor noise - while the uncertainty of its position in the plane is rather high. However, while having a closer look at artefact (f) in figure 1, we can see that (earlier) misaligned scans could violate this assumption and result in an offset exactly in the direction of the normal vector. This could be the case when the transformation was wrongly estimated, i.e. when the translation component of one of the directions was estimated ‘too small’ or ‘too large’. If we would use this concatenated point cloud, plane-to-plane GICP would perform poorly. To this end, we propose an extension on the GICP algorithm, which incorporates the information of the underlying surface in a better way. In [9], Holz *et al.* proposed to do something similar, but they adhere to the assumption that each point set is locally planar. As opposed to [8], they estimate the surface normals in a different manner. Instead of naively taking the closest neighbours to compute them, they determine them using approximate surface reconstruction by exploiting the topology of the points. Servos *et al.* further extended the GICP framework in [10] by integrating information from different sources, e.g. by incorporating color information in the registration process. Naturally, they denote their method by *Multi-Channel GICP*. The authors also adhere the local planarity assumption, but they propose to use additional *channels* of information to define the covariance of a point along the surface plane as well. Our proposed method follows the same strategy as this Multi-Channel GICP, in the sense that we augment the GICP framework by integrating additional information about the underlying surface. As a first contribution of this work, we propose to incorporate a surface reconstruction step into the registration pipeline based on fitting second degree polynomials. This step will on the hand one cope with the inaccuracies that are present in the point clouds due to outliers and noise. On the other hand, it will reduce faulty correspondences by filling small gaps in the point cloud as a result of missing data. As a second contribution, we propose to exploit the knowledge of the underlying surface by incorporating it as a prior in the actual cost function. This will also improve the data

association step by penalizing correspondences for which the underlying topology is too much different.

### III. APPROACH

#### A. GICP

In a nutshell, ICP iteratively determines *closest* point pairs between the two point clouds and subsequently use these to minimize a cost function. A widely-used cost function to estimate the transformation from one point cloud to another is the *point-to-plane* criterion:

$$E(\mathcal{P}_s, \mathcal{P}_t; \mathbf{T}) = \sum_{i=1}^N \mathbf{w}_i ((\mathbf{T}\mathbf{p}_i^s - \mathbf{p}_i^t) \cdot \mathbf{n}_i^t)^2, \quad (1)$$

for which a closed-form approximation was derived by Low *et al.* [11]. It is considered as a more robust alternative for the simpler *point-to-point* criterion given by:

$$E(\mathcal{P}_s, \mathcal{P}_t; \mathbf{T}) = \sum_{i=1}^N \mathbf{w}_i \|\mathbf{T}\mathbf{p}_i^s - \mathbf{p}_i^t\|^2. \quad (2)$$

In these equations,  $\mathbf{T}$  is the estimated transformation matrix,  $\mathcal{P}_s$  the *source* point cloud,  $\mathcal{P}_t$  the *target* point cloud,  $\mathbf{n}_i^t$  the surface normal according to target point  $\mathbf{p}_i^t$  and  $\mathbf{w}_i$  an optional weight vector. We consider  $i$  the index for the  $N$  corresponding points, making  $\mathbf{p}_i^s$  the corresponding point of  $\mathbf{p}_i^t$  for all  $i \in \{0 \dots N\}$ . Due to the sampling process, which leads to a discretization of the 3D space, perfect point-to-point matches are however nearly impossible to obtain. The point-to-plane cost criterion relaxes this constraint by allowing point offsets along the surface normal. Hence, it minimizes the distance from *all* points from the source point cloud  $\mathcal{P}_s$  to the tangent plane of its corresponding point in the target point cloud  $\mathcal{P}_t$ . However, this solution does still not take into account the fact that the source point cloud itself is also the result of a discretization of an (unknown) underlying geometric surface model. Segal *et al.* [8] therefore proposed *Generalized ICP* (GICP), which is a probabilistic generalization of the ICP algorithm that uses an objective function incorporating structural information from both the source and the target point cloud. Consider the source and target point clouds  $\mathcal{P}_s$  and  $\mathcal{P}_t$  to be sampled from underlying surface models, i.e. point clouds resulting from infinitely sampled surfaces -  $\hat{\mathcal{P}}_s$  and  $\hat{\mathcal{P}}_t$  - for which perfect point correspondences exist and let  $\hat{\mathbf{p}}_i^s$  and  $\hat{\mathbf{p}}_i^t$  be two points generated from respectively  $\hat{\mathcal{P}}_s$  and  $\hat{\mathcal{P}}_t$ . The points  $\mathbf{p}_i^s$  and  $\mathbf{p}_i^t$  are now assumed to be sampled from normal distributions  $\mathcal{N}(\hat{\mathbf{p}}_i^s, C_i^s)$  and  $\mathcal{N}(\hat{\mathbf{p}}_i^t, C_i^t)$ . Consider  $\mathbf{T}$  to be the transformation that aligns  $\hat{\mathcal{P}}_s$  and  $\hat{\mathcal{P}}_t$  such that  $\hat{\mathbf{p}}_i^t = \mathbf{T}\hat{\mathbf{p}}_i^s$  and  $\mathbf{d}_i = \mathbf{p}_i^t - \mathbf{T}\mathbf{p}_i^s$ . Because  $\mathbf{p}_i^t$  and  $\mathbf{p}_i^s$  are drawn from independent normal distributions,  $\mathbf{d}_i$  - which is a linear combination of  $\mathbf{p}_i^t$  and  $\mathbf{p}_i^s$  - is also drawn from a normal

distribution:

$$\mathbf{d}_i \sim \mathcal{N}(\hat{\mathbf{p}}_i^t - \mathbf{T}\hat{\mathbf{p}}_i^s, C_i^t + \mathbf{T}C_i^s\mathbf{T}^\top) \quad (3)$$

$$= \mathcal{N}(\mathbf{0}, C_i^t + \mathbf{T}C_i^s\mathbf{T}^\top). \quad (4)$$

Using MLE, the optimal transformation matrix  $\hat{\mathbf{T}}$  can be determined by

$$\hat{\mathbf{T}} = \underset{\hat{\mathbf{T}}}{\operatorname{argmax}} \prod_i p(\mathbf{d}_i) = \underset{\hat{\mathbf{T}}}{\operatorname{argmax}} \sum_i \log(p(\mathbf{d}_i)) \quad (5)$$

which can be simplified to

$$\hat{\mathbf{T}} = \underset{\hat{\mathbf{T}}}{\operatorname{argmin}} \sum_i \mathbf{d}_i^\top (C_i^t + \mathbf{T}C_i^s\mathbf{T}^\top)^{-1} \mathbf{d}_i. \quad (6)$$

The point-to-point ICP can be seen as a special case of this by setting  $C_i^t = I$  and  $C_i^s = 0$  leading to

$$\hat{\mathbf{T}} = \underset{\hat{\mathbf{T}}}{\operatorname{argmin}} \sum_i \mathbf{d}_i^\top \mathbf{d}_i \quad (7)$$

$$= \underset{\hat{\mathbf{T}}}{\operatorname{argmin}} \sum_i \|\mathbf{d}_i\|^2. \quad (8)$$

Similarly, the point-to-plane cost function can be seen as another special case where  $C_i^t = \mathbf{P}_i^{-1}$ ,  $\mathbf{P}_i$  representing the projection onto the span of the surface normal at  $\mathbf{p}_i^t$  (cfr. [8]). This *Generalized ICP* framework provides much more freedom in modelling the situation. As an alternative for the aforementioned cost functions, Segal *et. al.* propose a different choice for the covariances  $C_i^s$  and  $C_i^t$ . They assume that the point set is locally planar and consider each point to be distributed with high covariance along its local plane, and very low covariance in the surface normal direction. In line with this reasoning they propose to set the covariance matrices as follows:

$$C_i^t = \mathbf{R}_{\mathbf{n}_i^s} \cdot \begin{pmatrix} \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \mathbf{R}_{\mathbf{n}_i^s}^\top \quad (9)$$

$$C_i^s = \mathbf{R}_{\mathbf{n}_i^t} \cdot \begin{pmatrix} \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \mathbf{R}_{\mathbf{n}_i^t}^\top. \quad (10)$$

In these equations, the matrices  $\mathbf{R}_{\mathbf{n}_i^s}$  and  $\mathbf{R}_{\mathbf{n}_i^t}$  are respectively the rotation matrices that transform the basis vector  $\mathbf{e}_1$  onto the normal vectors  $\mathbf{n}_i^s$  and  $\mathbf{n}_i^t$  corresponding to respectively  $\mathbf{p}_i^s$  and  $\mathbf{p}_i^t$ . Doing so,  $\epsilon$  is reflecting the uncertainty along the approximated normals which should - under their assumption - be low. Its value accounts for the inaccurate measurements, or more specifically the noise of the sensor and should in general be small. In case of a Velodyne scanner, the value of  $\epsilon$  should be chosen somewhere between 0.01 and 0.03, approximating the accuracy of the scanner. The cost function of 6 with covariances defined as in eqn. 9 and 10 is often denoted as *plane-to-plane*.

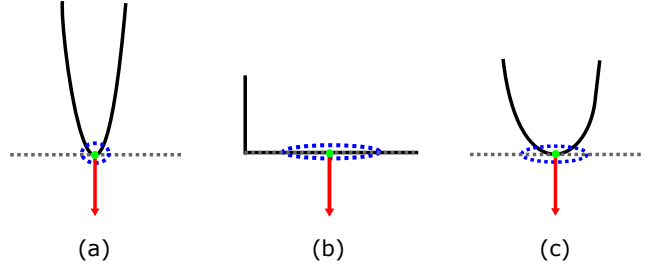


Figure 2. Three situations where different covariances should be set, here in 2D. Each of the green points has the same orientation for its normal vector. However, the uncertainty in the lateral direction differs for each of the points. In case the point is lying on a plane (b), the covariance in the direction of the plane is larger then when the point is lying on a curved surface, the higher the covariance in the direction perpendicular to the tangent plane, (a) vs. (c).

## B. Surface reconstruction

The method of Segal *et. al.* has the limitation that it does not take into account that there could be an uncertainty about the precise location of a point along its surface normal as well. This uncertainty could occur in the case of misaligned scans in successive registration as part of SLAM algorithms. Moreover, the uncertainty increases in sparser regions of the point cloud. These sparser areas are often present in data originated from 3D sensors, in particular in case of lidar scanners due to the angle of incidence of the laser beams. Also, the assumption that a point set is locally planar is rather simple. In that theory, every measured point provides only a single constraint along its surface normal. However, if we know more about the local surface, e.g. its curvature, we could incorporate additional constraints. For that reason, we propose to estimate the underlying surface of the point cloud in order to determine what the local variance is on the other dimensions. Figure 2 depicts three situations (in 2D) where different covariances should be picked. In case the point is lying on a plane (b), the covariance in the direction of the plane is higher than when the point is located on a curved surface. The higher the curvature the higher the covariance in the direction perpendicular to the tangent plane, (a) vs. (c). The main idea now is to estimate the surface depicted (in 2D) in Figure 2 by the black solid line. We propose to approximate this surface by fitting a second degree polynomial  $p_i$  through the distances of the neighbouring points to the tangent plane. The bivariate case of this quadratic function is given by  $p_i(x, y) = ax^2 + by^2 + cxy + dx + ey + f$  and has thus 6 parameters. To obtain an estimate on these parameters we minimize the following objective function:

$$\tilde{p}_i = \underset{p}{\operatorname{argmin}} \sum_{\mathbf{p}_j} (p(\mathbf{x}_j) - f_j)^2 \theta(\|\mathbf{p}_j - \mathbf{p}_i\|). \quad (11)$$

In this equation,  $\mathbf{p}_j$  are the neighbours of  $\mathbf{p}_i$ ,  $\mathbf{x}_j$  the orthogonal projections of  $\mathbf{p}_j$  onto the tangent plane  $H_i \triangleq (\mathbf{n}_i, d_i)$  and  $f_j \triangleq \langle \mathbf{p}_j, \mathbf{n}_i \rangle - d_i$  the distance of  $\mathbf{p}_j$  to  $H_i$ .

Finally,  $\theta(x) = e^{-\left(\frac{x}{\sigma_r}\right)^2}$  represents the weighting function that is based on the Euclidean distance between  $\mathbf{p}_j$  and  $\mathbf{p}_i$  and the average separation  $\sigma_r$  of the 3D points. In other words, neighbouring points located further away from  $\mathbf{p}_i$  get a lower weight than points located closer.

### C. Surface normals

The surface reconstruction detailed in the previous section still requires a preliminary estimate of surface normals (and hence tangent planes) for each point. In the original paper described in [8], the authors compute the normal vectors using PCA on the covariance matrix of the 20 closest neighbours. However, in case of an inhomogeneous point density, 20 neighbours are not appropriate in sparsely sampled areas. Some of the neighbours in that case could even lie on entirely different surfaces. Also, in case of discontinuities, this method can cause surface normals near the intersection of two planes to shape like an arc, as indicated by the blue line in drawing (a) of Figure 3. In some cases this would not be a problem for registration, for example when the incidence angle of the laser beams make the intersection visible in the two point clouds that need to be aligned. However, when the intersection is not visible in one point cloud, i.e. only one of the planes is visible, the closest neighbours will all be selected on the plane causing the normals to point in the ‘correct’ direction, cfr. image (b) in Figure 3. This means that for the same *physical* point, e.g. the blue point, the normal vectors in the two point clouds could be estimated differently. This would on its turn jeopardize the correct alignment as these correct correspondences would be penalized. Holz *et. al.* [9] therefore proposed *approximate surface reconstruction* to obtain a better estimate on the surface normals. Their method exploits the organized structure of the points as a result of the ordered acquisition by the different lasers. Doing so, they estimate the topology of the points. In our method we also exploit the knowledge about the ordering of the points, but we propose to perform a clustering of the points based on both their Euclidean distance as well as an initial guess of their normal vectors. Guided by this clustering we re-estimate the surface normals by selecting neighbouring points only within the same cluster. Doing so, we solve for the discontinuities in the point cloud, such as the one depicted in Figure 3. For the blue point, this means that only points of either one of the planes will be taken into account to estimate its true surface normal. Note that in reality it is very unlikely that exactly the intersection of the two planes is sampled. A point like the blue one can thus be assumed to always belong to either one of the planes. The actual derivation of the surface normal of a point  $\mathbf{p}_i$  is conducted using a PCA on the covariance matrix of the neighbours  $j \in 0, \dots, N$  of that point as given by equation

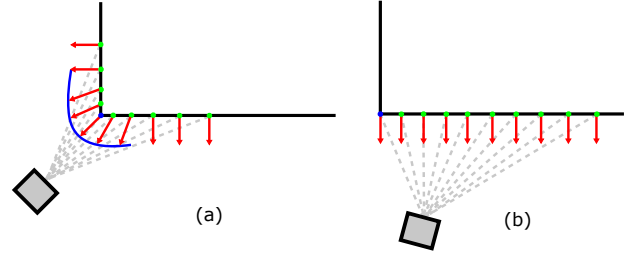


Figure 3. Image depicting the estimation of normal vectors using the k-nearest neighbour criterion. As can be seen in the left image, on the intersection of the two planes (here drawn in 2D), normal vectors are shaped like an arc (the blue line). The normal vectors depicted in the right image do not suffer from this artefact because points on the vertical plane are not sampled (due to occlusion) and are hence not used to compute the normal vector of the blue corner point. This results in a differently estimated surface normal for the blue point. In case of registration of the two point clouds, the (correct) matching of the corner point might be penalized by its discrepancy in normal vector, which should be avoided.

13:

$$C_i = \frac{1}{N} \sum_{j=1}^N (\mathbf{p}_j - \bar{\mathbf{p}}_i) \cdot (\mathbf{p}_j - \bar{\mathbf{p}}_i)^T \quad (12)$$

$$C_i \cdot \mathbf{v}_l = \lambda_l \cdot \mathbf{v}_l, \quad l \in 0, 1, 2 \quad (13)$$

In this equation,  $N$  is the number of neighbours of  $\mathbf{p}_i$  that we take into account,  $\lambda_l$  the  $l$ -th eigenvalue of the covariance matrix and  $\mathbf{v}_l$  the  $l$ -th eigenvector. The point  $\bar{\mathbf{p}}_i$  is the 3-dimensional centroid of the nearest neighbours. The surface normal can then be approximated by the eigenvector corresponding to the smallest eigenvalue, as this is exactly the direction in which the ‘least’ variation occurs.

### D. Surface-based GICP

Our actual surface-based GICP algorithm consists of a few improvements. First, we use a different neighbourhood function, as explained in the previous section. Instead of taking the 20 closest neighbours as in the original GICP algorithm, we select the closest neighbours that are within the same *cluster* as the point under consideration. Second, we project the point cloud onto the underlying surface, as explained in section III-B. This will reduce the noise and filter the outliers. Third, we conduct *voxel grid dilation* to deal with small gaps in the point cloud. More precisely, in sparser areas, we add points to the point cloud, initially at the centroid positions of a regularly spaced voxel grid. Subsequently, we project these centroid points on the surface estimated at their closest point in the point cloud. As a final improvement, we compute the covariance of the points in the directions perpendicular to the tangent plane, thereby using weights based on their distance to it. We thus perform a second projection of the data points, this time onto the tangent plane. Let  $\mathbf{x}_j \in \mathbb{R}^2$  be these orthogonal projections of the points  $\mathbf{p}_j$  onto the tangent plane. Note that after this transform, the new population covariance,  $\Sigma_i \in \mathbb{R}^{2 \times 2}$  is the

diagonal matrix of the largest eigenvalues of eq. 13. Now, we compute a weighted covariance using a Gaussian kernel that is defined as  $\mathcal{N}(\bar{f}_i, \sigma_i)$ , centered at the mean distance  $\bar{f}_i$  to the tangent plane and with the standard deviation  $\sigma$  for the distances. The kernel weights are then computed for each point as follows:

$$w_j = e^{-\frac{1}{2\sigma_i}(f_j - \bar{f}_i)^2} \quad (14)$$

Using these weights, the distance kernel weighted centroid and covariance,  $\bar{\mathbf{x}}_i$  and  $\Sigma_i^w$ , can be computed as:

$$\bar{\mathbf{x}}_i = \frac{1}{\sum_j w_j} \sum_j w_j \mathbf{x}_j \quad (15)$$

$$\Lambda_i = \frac{1}{\sum_j w_j} \sum_j w_j \cdot (\mathbf{x}_j - \bar{\mathbf{x}}_i) \cdot (\mathbf{x}_j - \bar{\mathbf{x}}_i)^\top \quad (16)$$

This distribution models the uncertainty of the distance of each point to the tangent plane locally. However, it can be biased if the original sample population was itself already biased. For that reason, we compensate the potential bias by normalizing the population covariance as follows:

$$\Omega_i = \Sigma_i^{-\frac{1}{2}} \Lambda_i \Sigma_i^{-\frac{1}{2}} \quad (17)$$

To use this information in the GICP framework,  $\Omega_i$  is used along the planar directions. Therefore, the resulting covariance used in S-GICP, cfr. eq. 6, is:

$$C_i^s = \mathbf{R}_{\mathbf{n}_i^s} \cdot \begin{pmatrix} \epsilon & \mathbf{0}_2 \\ \mathbf{0}_2^\top & \Omega_i^s \end{pmatrix} \cdot \mathbf{R}_{\mathbf{n}_i^s}^\top \quad (18)$$

$$C_i^t = \mathbf{R}_{\mathbf{n}_i^t} \cdot \begin{pmatrix} \epsilon & \mathbf{0}_2 \\ \mathbf{0}_2^\top & \Omega_i^t \end{pmatrix} \cdot \mathbf{R}_{\mathbf{n}_i^t}^\top. \quad (19)$$

In these equations,  $C_i^t$  and  $C_i^s$  represent the covariance matrices for point  $i$  in the target and source point cloud respectively. When the goal is to perform registration on consecutive point clouds as part of a SLAM algorithm, we suggest to project all points on the estimated surface to cope with the point cloud artefacts depicted in Figure 1. Ideally, the surface is re-estimated every time new points are added to the overall point cloud model.

#### E. Extreme cases

When we consider eqs. 15 and 16 for points lying on a perfect plane, cfr. the assumption made in [8], we see that the weights  $w_j$  are reduced to 1 as both the average distance to the tangent plane  $\bar{f}_i$  as well as the distance of a random neighbour point to the tangent plane  $f_j$  are 0. The covariance  $\Sigma_i^w$  will then approximate  $\Sigma_i$  and  $\Omega_i$  would degenerate to the identity matrix:  $\Omega_i = \Sigma_i^{-\frac{1}{2}} \Sigma_i \Sigma_i^{-\frac{1}{2}} = I_2 \in \mathbb{R}^{2 \times 2}$ . In summary, when the local neighborhood of a point is planar, our algorithms degrades to the original GICP algorithm.

## IV. EVALUATION

As the errors made by both our algorithm and the standard GICP are rather small, it is difficult to notice them visually when performing registration of only two point clouds. For that reason, we performed our experiments on a long lidar sequence that is part of the Kitti benchmark [12]. In that dataset, point cloud sequences were acquired by a Velodyne HDL-64e scanner. The dataset suffers from the main ‘artefacts’ that were enumerated in the introduction. For the experiment, we selected the 10th lidar sequence of the Kitti benchmark, as it contains roads predominantly surrounded by vegetation as well as roads in more urbanized areas. Figure 4 depicts two images taken from the concatenated point cloud after registration. The results of the experiment are depicted in Figure 5. The ‘standard’ GICP algorithm (blue line) was executed using the 20 nearest neighbours. Our own result (green line) was performed using a different neighbour function to generate surface normals as well as a different choice for the covariance matrices of 6 as explained in the previous section. The ground truth is depicted in red. In the experiment, there was no concatenation of point clouds, thus the registration performed for every new frame is the one that aligns the *currently* acquired point cloud with the *previous* one. As can be seen, the ground truth curve and the curve of our own result are lying almost entirely on each other, hence showing the high accuracy of our algorithm. In figure 6, the translation error in function of the path length is plotted for both standard GICP and our modified version of GICP. Both translational errors are increasing for increasing path lengths and have more or less the same slope. However, for small path lengths (100m), the translational error of our method approximates 1%, while the one of standard GICP is situated around 1.8%. For path lengths of 800m the translational errors are respectively 2.5% and 3.5%. Thus, we can conclude that our approach is leading to a gain of approximately 0.8 to 1%.

## V. CONCLUSIONS

In this paper we presented a registration framework that extends the GICP framework by incorporating information of the underlying surface. Our method is a natural extension that gracefully deals with all of the artefacts that come with point clouds acquired by lidar scanners including an inhomogeneous and sparse point density, noise and missing data. Experiments demonstrated that we can substantially reduce the positional error for lidar sequences of the KITTI benchmark without performing any kind of loop closing or Bayesian filtering.

## ACKNOWLEDGEMENT

This research is part of the ARIA project, an ICON project co-funded by imec, digital research institute founded by the Flemish Government.

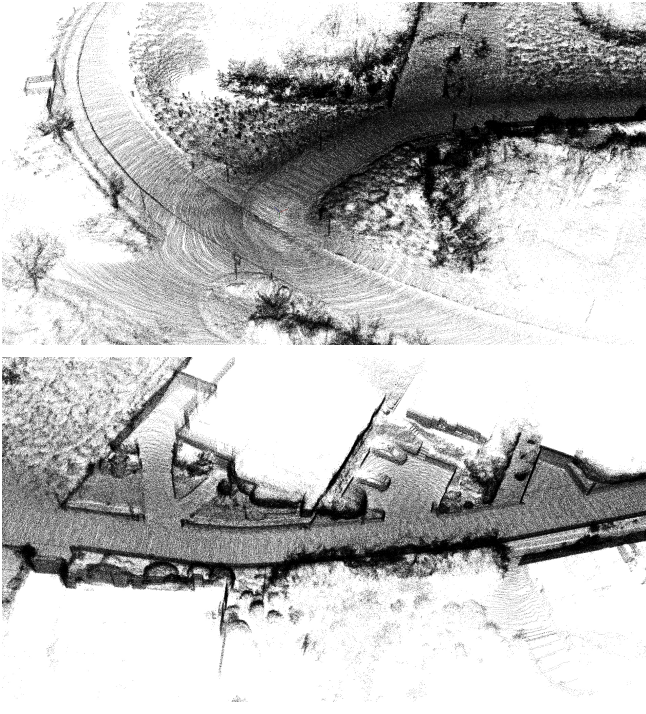


Figure 4. Figure depicting two images taken from the concatenated point cloud of lidar sequence 10 of the Kitti benchmark that was used for conducting experiments. The lidar sequence covers both, rural areas (with vegetation aside the road) as well as urban areas.

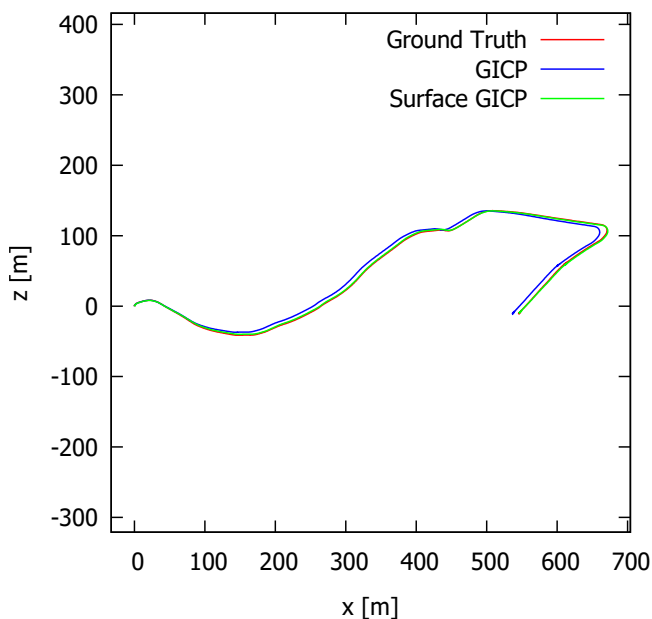


Figure 5. Plot of the estimated (blue) and ground truth (red) trajectory for the 10th sequence of the Kitti benchmark. The top trajectory was generated by running the ‘standard’ GICP algorithm taking into account the 20 nearest neighbours. There was no concatenation of point clouds, so the registration at every timestamp is only conducted from the *current* to the *previously* acquired point cloud. Obviously, there is no loop detection or closure.

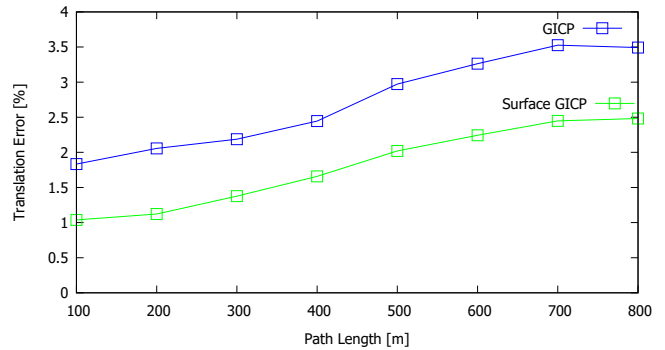


Figure 6. Translation error in function of the path length for our algorithm (green line) and the standard GICP (blue line) for the 10th sequence of the Kitti benchmark. Both curves are having more or less the same slope. However, the translational error of our method is approximately 1% below the one of standard GICP.

## REFERENCES

- [1] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992. [Online]. Available: <http://dx.doi.org/10.1109/34.121791> 1
- [2] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image Vision Comput.*, vol. 10, no. 3, pp. 145–155, Apr. 1992. [Online]. Available: [http://dx.doi.org/10.1016/0262-8856\(92\)90066-C](http://dx.doi.org/10.1016/0262-8856(92)90066-C) 1
- [3] Y. He, B. Liang, J. Yang, S. Li, and J. He, “An iterative closest points algorithm for registration of 3d laser scanner point clouds with geometric features,” *Sensors*, vol. 17, no. 8, 2017. 1
- [4] A. Gressin, C. Mallet, J. Demantké, and N. David, “Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 79, pp. 240–251, Apr. 2013. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00920922> 1
- [5] R. Marani, V. Renò, M. Nitti, T. D’Orazio, and E. Stella, “A modified iterative closest point algorithm for 3d point cloud registration,” *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 31, no. 7, pp. 515–534, Jul. 2016. [Online]. Available: <https://doi.org/10.1111/mice.12184> 1
- [6] G. Agamennoni, S. Fontana, R. Y. Siegwart, and D. G. Sorrenti, “Point clouds registration with probabilistic data association,” in *IROS*. IEEE, 2016, pp. 4092–4098. 1
- [7] J. Yang, H. Li, D. Campbell, and Y. Jia, “Go-icp: A globally optimal solution to 3d ICP point-set registration,” *CoRR*, vol. abs/1605.03344, 2016. 2
- [8] A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp,” in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009. 2, 3, 4, 5
- [9] D. Holz and S. Behnke, “Registration of non-uniform density 3d point clouds using approximate surface reconstruction,” in *ISR/Robotik 2014; 41st International Symposium on Robotics*, June 2014, pp. 1–7. 2, 4

- [10] J. Servos and S. L. Waslander, "Multi-channel generalized-icp: A robust framework for multi-channel scan registration," *Robotics and Autonomous Systems*, vol. 87, no. Complete, pp. 247–257, 2017. 2
- [11] K.-L. Low, "Linear least-squares optimization for point-to-plane icp surface registration," Tech. Rep., 2004. 2
- [12] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5