Assessing Losses for Point Set Registration

Anderson C. M. Tavares[1,2], Felix Järemo Lawin[1] and Per-Erik Forssén[1]

***Abstract*—This paper introduces a framework for evaluation of the losses used in point set registration. In order for a loss to be useful with a local optimizer, such as e.g. Levenberg-Marquardt, or expectation maximisation (EM), it must be monotonic with respect to the sought transformation. This motivates us to introduce *monotonicity violation probability* (MVP) curves, and use these to assess monotonicity empirically for many different local distances, such as point-to-point, point-to-plane, and plane-to-plane. We also introduce a local shape-to-shape distance, based on the Wasserstein distance of the local normal distributions. Evaluation is done on a comprehensive benchmark of terrestrial lidar scans from two publicly available datasets. It demonstrates that matching robustness can be improved significantly, by using kernel versions of local distances together with inverse density based sample weighting.**

***Index Terms*—Performance Evaluation and Benchmarking, Probability and Statistical Methods**

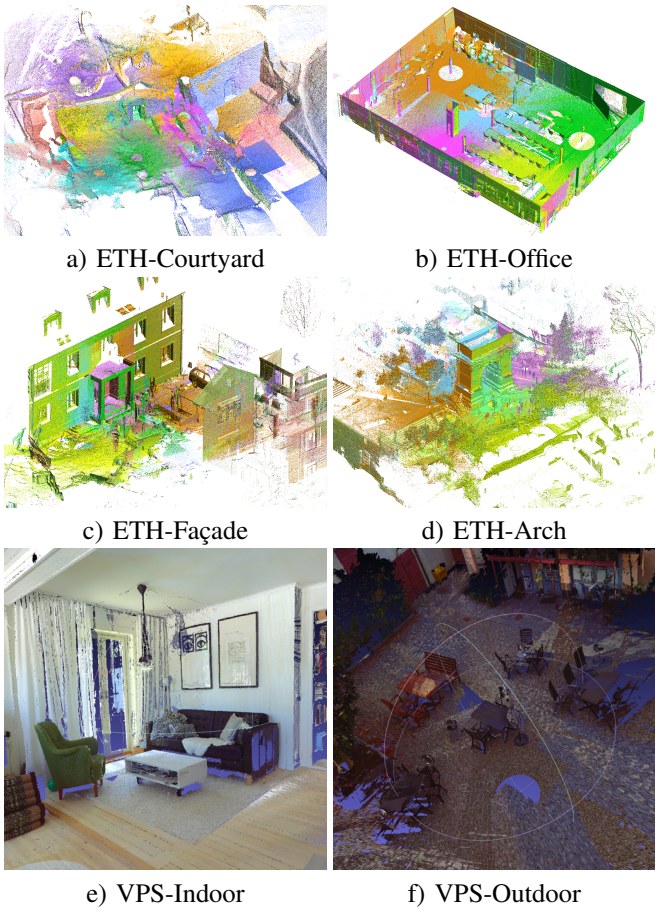
I. Introduction

OVER the years, a large number of methods for point set registration have been proposed. These have two main components: (1) A *registration loss* and (2) an *algorithm* that minimizes the registration loss (or, equivalently, maximizes a *registration score*). In order to obtain an efficient algorithm, the two components are typically deeply integrated, and for this reason the registration loss is rarely studied in isolation. Most of the practically useful registration losses are based on local distances, such as the *point-to-point* [1] and *point-to-plane* [2] distances commonly used in the iterative closest point algorithm (ICP).

In this paper we make an in-depth analysis of registration losses for point set alignment, and compare the different choices on challenging real data, see figure 1. We also experimentally demonstrate how performance differs between pairs with large and small point set overlap.

As many point clouds have dramatically higher density of samples near the sensor [5], another important aspect is how to reduce the bias caused by the varying density. To this end, we investigate the effect of density based sample weighting.

In order to be used in iterative minimization, a registration loss should be monotonically decreasing with better alignment of the point sets. This motivates us to measure monotonicity empirically, and for this purpose we introduce the concept of *monotonicity violation probability* (MVP) curves. We use such MVP curves as a criterion for assessing registration losses in our experiments.



a) ETH-Courtyard b) ETH-Office c) ETH-Façade d) ETH-Arch e) VPS-Indoor f) VPS-Outdoor

Fig. 1. Datasets [3], [4] used in the experiments. See table I for details.

II. Related Work

The objective functions used in point set registration can be divided into losses based on *maximum likelihood* (ML), and scores based on *kernel density estimation* (KDE).

The ML based losses descend from the *iterative closest point* (ICP) algorithm [2], [1]. Changes to the ICP algorithm have been proposed a countless number of times [6]. We mention only a select subset of notable ideas here, and refer to e.g. [7] for a more thorough treatment. The *point-to-plane* local metric was actually there from the start [2]. Robustification and Kd-tree speedup was introduced by Zhang [8]. LM-ICP [9] proposed the use of a distance field for

Manuscript received: September, 10, 2019; Revised December, 21, 2019; Accepted February, 3, 2020.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by ELLIIT, the Strategic Area for ICT research, funded by the Swedish Government, and by Vinnova through the Visual Sweden network.

[1]All authors are with Computer Vision Lab (CVL), department of Electrical Engineering (ISY), Linköping University (LiU), SE-581 83 Linköping, Sweden per-erik.forssen@liu.se

[2]Anderson C. M. Tavares is with RISE SICS East Linköping, SE-581 83 Linköping, Sweden anderson.tavares@ri.se

Digital Object Identifier (DOI): see top of this page.

speedup, an anticipation of cost volume based registration such as the TSDF [10] used in the Kinect Fusion [11] real-time registration method.

Robustified versions of ICP can also be defined using Gaussian mixture models [12]. Here a Gaussian mixture model and the tranformation parameters are inferred using *expectation maximization* (EM). This is further generalized in [13] to jointly handle multiple point sets. In order to better handle density variations in an EM framework, [5] further incorporates a density adaptive sample weighting into the probabilistic model.

Tsin and Kanade [14] introduced KDE based registration scores. They propose a score called *kernel correlation*. For efficiency, kernel correlation can use precomputed densities on a grid. This idea is generalized in the *normal distributions transform* (NDT) [15], where local *anisotropic* normal distributions are precomputed on the grid. A coarse-to-fine variation of grid cell sizes is also proposed in [15]. Another related development is to use a Gaussian Mixture (GM) model approximation of the probability density [16]. Here an L^2 distance between PDFs represented as GM is minimized.

One could also consider assessing the alignment quality *after* alignment [17]. This allows other aspects to be taken into account, such as visibility. While visibility is useful for assessing the final solution, it will cause the score to change rapidly with small shifts and rotations (whenever they occlude or disocclude points), and is thus detrimental when designing a locally convex score. Related ideas also appear in [18], where self-occlusisions are accounted for using rendering of candidate image views, and the use of image based distances on these.

III. Point set registration losses

We start by introducting the necessary terminology. Let V be a *normed vector space* (e.g., $V = \mathbb{R}^3$) with *norm* $\|\cdot\| : V \to \mathbb{R}_{\geq 0}$. Further, let $\mathcal{P}(V)$ be the set of all possible subsets of V. A *point set* $\mathcal{X} = \{\mathbf{x}_k\}_{k=1}^K \in \mathcal{P}(V)$ is a set of *points* $\mathbf{x}_k \in V$ with cardinality $|\mathcal{X}| = K$. A *transformation* $T : V \to V$, originally defined on points, acts on a point set as $T(\mathcal{X}) = \{T(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$.

For the rigid transformation group SE(3), a transformation T applied to a point $\mathbf{x}$ is expressed in matrix algebra using a rotation $\mathbf{R}$ and a translation $\mathbf{t}$ as:

$$T(\mathbf{x}) = \mathbf{R}\mathbf{x} + \mathbf{t}\,. \tag{1}$$

Thus, for convenience we use the notation $T = (\mathbf{R}, \mathbf{t})$.

Registration means aligning two point sets, which can be restated as finding the transformation, T^*, that minimizes a given *registration loss* $L : \mathcal{P}(V) \times \mathcal{P}(V) \to \mathbb{R}$:

$$T^* = \arg\min_T L(\mathcal{X}, T(\mathcal{Y})) \tag{2}$$

Such a loss can only be used to align point sets if it has a global minimum at the correct point set transformation T^*.

In order to use the loss in an iterative minimization, we further desire that the loss should be monotonically decreasing for small increments ΔT towards the correct transformation:

$$d(T^*, T \circ \Delta T) < d(T^*, T) \;\Rightarrow \tag{3}$$

$$L(\mathcal{X}, (T \circ \Delta T)(\mathcal{Y})) < L(\mathcal{X}, T(\mathcal{Y}))\,. \tag{4}$$

Here $d(T_1, T_2)$ is a *natural distance metric* on the transformation group and $\circ$ denotes the group operation on SE(3). E.g. for pure translations, the natural metric is the Euclidean distance $d(T_1, T_2) = \|\mathbf{t}_1 - \mathbf{t}_2\|$, and for pure rotations, it is the 3D rotation angle $d(T_1, T_2) = 2\sin^{-1}(\|\mathbf{R}_1 - \mathbf{R}_2\|/\sqrt{8})$, see [19]. For SE(3) there is unfortunately no unique natural metric [20], and thus it is common to present rotation and translation deviations separately.

IV. A General Loss Function

The point set registration losses considered here can all be expressed as a weighted sum of *local distances*. A local distance compares two points, or their neighbourhoods. Common examples are *point-to-point* [1], *point-to-plane* [2] and *plane-to-plane* [21] distances. In addition, we will also test the Wasserstein distance for normal distributions [22] as a local *shape-to-shape* distance.

For point sets $\mathcal{X} = \{\mathbf{x}_k\}_{k=1}^K$ and $\mathcal{Y} = \{\mathbf{y}_l\}_{l=1}^L$, we can collect all local distances in a $K \times L$ matrix $\mathbf{C}$, with elements:

$$c_{kl} = w_l d(\mathcal{N}_\mathcal{X}(\mathbf{x}_k), \mathcal{N}_\mathcal{Y}(\mathbf{y}_l))^2\,, \tag{5}$$

where $d(\cdot,\cdot)$ is a local distance, $\mathcal{N}_\mathcal{X}(\mathbf{x}_k)$ is the local neighbourhood in $\mathcal{X}$ around point $\mathbf{x}_k$ and w_l is an optional weight.

We can now express a general registration loss as:

$$L(\mathcal{X}, \mathcal{Y}) = \sum_{k,l} a_{kl} c_{kl} = \langle \mathbf{A} \mid \mathbf{C} \rangle_F\,, \tag{6}$$

where a_{kl} are *assignment weights*, and $\langle \cdot \mid \cdot \rangle_F$ is the Frobenius(matrix) scalar product. The weights a_{kl} encode the *assignment* (or more generally the *transport plan* [23], [24]) between points in $\mathcal{X}$ and $\mathcal{Y}$.

The assignments $\mathbf{A}$ are updated for each iteration, and thus a gradient of (6) is difficult or even impossible to define analytically. In e.g ICP [1], the assignment is binary and at most one-to-one, i.e. $a_{kl} \in \{0, 1\}$, and $\mathbf{A}$ is constrained to sum to either 1 or 0 along each row or column. This also implies that, in the ICP case, assignments could be stored as a set $\mathcal{A}$ of ordered index pairs:

$$\mathcal{A} \subset [1, K] \times [1, L]\,. \tag{7}$$

Such assignments are computed using nearest neighbours after applying the currently best transformation.

In order to increase robustness, all assignments for a point are set to zero if the nearest neighbour is beyond a maximum cutoff distance, $d_{\max}$, see e.g. [8], [21]. We can interpret this as a truncated quadratic loss, i.e. we keep the assignment, and instead replace (5) with:

$$c_{kl} = w_l \min(d_{\max}, d(\mathcal{N}_\mathcal{X}(\mathbf{x}_k), \mathcal{N}_\mathcal{Y}(\mathbf{y}_l)))^2\,. \tag{8}$$

For kernel based methods, more general, many-to-one assignments are used [14], and their contributions are weighted according to alignment distance.

A. Local distances

The ICP-based algorithms are all minimizing Mahalanobis distances between pairwise assignments:

$$d_{\Sigma_{k,l}}(\mathbf{x}_k, \mathbf{y}_l)^2 = \|\mathbf{x}_k - T(\mathbf{y}_l)\|^2_{\Sigma^{-1}_{k,l}} . \tag{9}$$

We may use different choices of $\Sigma^{-1}_{k,l}$ to recover the previously proposed local distances *point-to-point* [1], *point-to-plane* [2] and *plane-to-plane* [21]. For the point-to-point local distance [1], we have:

$$\Sigma^{-1}_{k,l} = \mathbf{I} , \tag{10}$$

and for the point-to-plane distance [2], we have:

$$\Sigma^{-1}_{k,l} = \mathbf{n}_k \mathbf{n}_k^T , \tag{11}$$

where $\mathbf{n}_k$ is the estimated surface normal at each point.

In contrast, the plane-to-plane local distance [21] depends on both point sets. It can be written as:

$$\Sigma^{-1}_{k,l} = (\hat{\mathbf{\Sigma}}_k + \mathbf{R}\hat{\mathbf{\Sigma}}_l\mathbf{R}^T)^{-1} , \tag{12}$$

where $\hat{\mathbf{\Sigma}}_k$ and $\hat{\mathbf{\Sigma}}_l$ are local covariance estimates at corresponding points, and $\mathbf{R}$ is the rotation part of the sought rigid transformation, see (1). To model the local 2D-structure of surfaces, [21] sets $\hat{\mathbf{\Sigma}} = \mathbf{U}\mathbf{D}\mathbf{U}^T$, where $\mathbf{U}$ contains the eigenvectors of the empirical covariance matrix and $\mathbf{D} = \text{diag}(\epsilon, 1, 1)$ with a small ϵ. This enforces a fixed density estimate along the plane estimated from the empirical covariance. We will test the effect of changing ϵ in the experiments.

Another way to utilize empirical covariances around points is to compare them and their locations directly, using the Wasserstein distance for Gaussian distributions [22]. This results in the expression:

$$d_{W_{k,l}}(\mathbf{x}_k, \mathbf{y}_l)^2 = \|\mathbf{x}_k - \mathbf{y}_l\|_2^2 + \text{tr}(\Sigma_k + \Sigma_l - 2(\Sigma_l^{\frac{1}{2}}\Sigma_k\Sigma_l^{\frac{1}{2}})^{\frac{1}{2}}) , \tag{13}$$

which we denote the local *shape-to-shape* distance. Figure 2 illustrates the components of each distance metric.

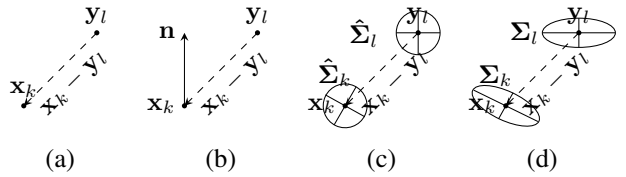


Fig. 2. Features used for distances between points $\mathbf{x}_k$ and $\mathbf{y}_l$ (a) point-to-point, (b) point-to-plane, (c) plane-to-plane, and (d) shape-to-shape.

B. Empirical covariances

We estimate empirical covariances and normals, for point-to-plane and plane-to-plane local distances, using the K-nearest neighbours (K-NN) of the respective point. The shape-to-shape distance, however, also depends on the scale difference between the two covariances, see (13) and thus a fixed radius computation of the covariance is needed. As a fixed radius neighbourhood may be empty of neighbours, we use the fallback formula:

$$\mathbf{\Sigma} = \mathbf{\Sigma}_{\text{prior}}\frac{1}{N} + \mathbf{\Sigma}_{\text{est}}\frac{N-1}{N} , \tag{14}$$

where N is the number of points in the neighbourhood (including the point itself). As prior, we use the expected covariance of an isotropic neighbourhood, $\mathbf{\Sigma}_{\text{prior}} = 0.2r^2\mathbf{I}$, where r is the neighbourhood radius. For non-empty neighbourhoods, we compute $\mathbf{\Sigma}_{\text{est}}$ using the regular unbiased estimation formula.

C. Sample density weighting

In order to reduce the impact of density variations in point sets (caused by the sampling pattern of 3D sensors, see [5]) we optionally use a sample weighting based on the inverse of a *kernel density estimate* (KDE) with an isotropic kernel:

$$w_k^{-1} = \sum_{i \in \mathcal{N}_\mathcal{X}(\mathbf{x}_k)} \exp\left(-\frac{1}{2h^2}\|\mathbf{x}_k - \mathbf{x}_i\|^2\right) . \tag{15}$$

Here, $\mathcal{N}_\mathcal{X}(\mathbf{x}_k)$ is the local neighborhood of $\mathbf{x}_k$ in $\mathcal{X}$, and h is the kernel bandwidth. The use of a neighborhood instead of all points allows rapid evaluation with a Kd-tree. For a variable number of points, the average over the points in the conventional KDE expression can be replaced by a sum here.

D. Registration Losses

In this work, we consider two types of losses; the maximum likelihood based loss used in the ICP-based algorithms and the density based loss used in [14]. We investigate the behaviour of both loss types for all the local distances introduced in section IV-A.

Maximum Likelihood (ML) loss: The registration loss functions used in the ICP based algorithms can be derived from maximum likelihood estimation [25]. First, we assume that $\mathcal{X}$ and $\mathcal{Y}$ are observations from random variables with associated probability density functions p_X and p_Y. We model p_X as a mixture of Gaussian distributions with means at the individual points $\mathbf{x}_k \in \mathcal{X}$ and covariances $\Sigma_{k,l}$:

$$p_X(\mathbf{y}_l|T) = \sum_k w_k \mathcal{N}(T(\mathbf{y}_l); \mathbf{x}_k, \Sigma_{k,l}) . \tag{16}$$

The registration problem can be formulated as maximizing the log-likelihood of the point set $\mathcal{X}$ given transformation T:

$$\mathcal{E} = \log p_Y(\mathcal{X}|T) . \tag{17}$$

This can be optimized over T using expectation maximization. We first introduce the latent variables Z and let $p_X(\mathbf{y}_l|T) = \sum_k p_{Z,X}(\mathbf{y}_l, k|T)$ be the marginal with $p_{Z,X}(\mathbf{y}_l, k|T) = \mathcal{N}(T(\mathbf{y}_l)); \mathbf{x}_k, \Sigma_{k,l})$. By setting the assignment weights to the latent posteriors [26]

$$a_{kl} = p_{Z|X}(k|\mathbf{y}_l, T) = \frac{p_{X,Z}(\mathbf{y}_l, k|T)}{\sum_{i=1}^{L} p_{X,Z}(\mathbf{y}_i, l|T)} \tag{18}$$

the negative log-likelihood loss becomes (cf. (6)):

$$L(\mathcal{X}, \mathcal{Y}) = -\sum_{kl} a_{kl}(\log p_{X,Z}(\mathbf{y}_l, k|T) - \log a_{kl}) = \tag{19}$$

$$\sum_{k,l} a_{kl}\left(\frac{1}{2}\|\mathbf{x}_k - T(\mathbf{y}_l)\|^2_{\Sigma^{-1}_{k,l}} + B_{k,l} + \log a_{kl}\right) , \tag{20}$$

where $B_{k,l} = 0.5\log((2\pi)^D|\Sigma_{k,l}|)$. In the ICP-case the term $\sum_i a_{kl}\log a_{kl}$ vanishes since the assignment is binary, see (7).

The negative log-likelihood is optimized iteratively by alternating between setting the assignments a_{kl} and minimizing L.

For both point-to-point and point-to-plane distances, the term $B_{k,l}$ is independent of the transformation T, thus it only contributes with a constant bias. In the plane-to-plane case, however, $\Sigma_{k,l}$ is updated for each iteration, thus $B_{k,l}$ will affect the likelihood estimate. However, this term is not useful in practise, as the scale of the covariance matrices is destroyed in the plane-to-plane calculation, see (12). Without correct scale, the magnitude of $B_{k,l}$ will be incorrect.

Kernel Density (KD) loss: The local distances in section IV-A are all based on the 2-norm, and are thus sensitive to outlier assignments. In the pursuit of increasing the robustness of the losses we also investigate kernel based versions. Following the work of [14] we compute the local Gaussian kernel density for each point. This means replacing the local distances in (5) with local similarities:

$$c_{k,l} = w_l \exp\left(-\frac{1}{2h^2}\|\mathbf{x}_k - T(\mathbf{y}_l)\|^2_{\Sigma^{-1}_{k,l}}\right). \tag{21}$$

Here $\Sigma_{k,l}$ is the Mahalanobis metric for the corresponding points x_k and y_l, h is the kernel bandwidth and w_l is the correspondence weight. We consider kernel versions for all local distances d described in section IV-A. The final registration score is then computed using (6). This score can be turned into a loss, by changing the sign of $c_{k,l}$ in (21).

V. Assessment of monotonicity

We assess monotonicity for a particular registration loss $L(\mathcal{X}_1, \mathcal{X}_2)$, by evaluating it for deviations from the correct alignment T^*. We do this for sequences of transformations $T_1, \ldots, T_N$, with $d(T^*, T_n) < d(T^*, T_{n+1})$, in the natural distance metric for the transformation group, see section III. The transformations are applied to one of the point sets before computing the loss. For each transformation sequence, this gives us a sequence of registration losses $L_0, L_1, \ldots, L_N$ where:

$$L_n = L(\mathcal{X}_1, T_n(\mathcal{X}_2)), \tag{22}$$

and L_0 is the registration loss at T^*.

A sufficient condition for monotonicity is that $I(L_n \leq L_{n-1}) = 0$, for all adjacent pairs of L_n. Here $I(x)$ is the indicator function:

$$I(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

We can test whether a violation has occurred in an interval of L_n, by using max(). This gives us a violation indicator function:

$$v_n = \max_{m \in [s,n]} I(L_m \leq L_{m-s}), \ n \geq s. \tag{24}$$

Here s is a step length parameter. If the step between subsequent transformations is too small, (24) will give spurious indications near the correct transformation. Causes of these include *assignment noise* (nearest neighbour assignment forces a neighbour to be selected, when the correct location may be between two points) and *inaccurate ground truth*. Spurious indications can be avoided by using a larger step, e.g. $s = 3$ instead of 1.

A. Monotonicity violation probability

For point sets depicting real scenes there is always a risk that a unique global minimum does not exist. This could e.g. be due to periodic structures in the scene, or too low an overlap between the point sets. This motivates an analysis of monotonicity violations in a statistical sense, i.e. how *likely* is it that a particular loss will have a violation compared to another one.

By averaging the violation indicator function (24) across a set of point set registrations and transformation sequences, we can obtain an empirical *monotonicity violation probability* (MVP) curve for each registration loss. We will use such MVP curves to compare the monotonicity of different registration losses in the experiment section.

A point $\hat{p}_n = E\{v_n\}$, on an MVP curve is an average of a set of violation indicators $\{v_n^k\}_1^K$, see (24). Thus $\hat{p}_n$ is binomially distributed

$$\hat{p}_n = \frac{1}{K}\sum_{k=1}^{K} v_n^k \sim \frac{1}{K} B(K, p_n). \tag{25}$$

In order to assess significance, we can add confidence bounds to the MVP curves. We use the *adjusted Wald test* proposed in [27]. This results intervals where p_n satisfies:

$$|p_n - \tilde{p}_n| < z_{\alpha/2}\sqrt{\tilde{p}_n(1-\tilde{p}_n)/(K+4)}. \tag{26}$$

Here $\tilde{p}_n = (K\hat{p}_n + 2)/(K+4)$, and $z_{\alpha/2} = 1.96$ for a 95% confidence interval. Note that for the MVP estimate, we still use $\hat{p}_n$ as defined in (25).

B. Sampling of transformation sequences

In the experiments we restrict the transformation sequences $T_1, \ldots, T_N$ to equidistant samplings away from the optimum, i.e. the special case where $T_n = (\Delta T)^n T^*$ for $n \in [1, N]$. As a loss could be violation free for a different trajectory from T_N to T^*, we may occasionally record a violation when there is none. Another option is to take steps in the gradient direction, but for losses of interest, see (6), the gradients are hard to obtain.

If sufficiently many transformation sequences are used, a simple way of choosing them is to do random sampling of directions in rotation and translation space. For lower numbers of samples, we recommend a symmetric arrangement of M directions. Symmetric directions are obtained from the vertices of the Platonic solids [28], when $M = 4, 6, 8, 12, 20$. For other values of M, a best fit can be found by minimizing the following cost, where $\hat{\mathbf{x}}_k$ are normalized columns of a $3 \times M$ parameter matrix $\mathbf{X}$:

$$\mathcal{J}(\mathbf{X}) = \sum_{k=1}^{M-1}\sum_{l=k+1}^{M} \pi - \cos^{-1}(\hat{\mathbf{x}}_k^T \hat{\mathbf{x}}_l). \tag{27}$$

Multiple equivalent minima of (27) exist, and they correspond to column permutations of $\mathbf{X}$. The solutions are vertices of the

Platonic solids for $M = 4, 6, 12$. For $M = 8, 20$ we get the Platonic solids as well as a twisted cube and dodecahedron. For other values of M we also get well separated vertices on the sphere.

C. Point set overlap

The difficulty of registration of two point sets depends on the amount of volume that they share, relative to their total volume. This common volume fraction is referred to as the point set *overlap*. Due to noise and sparsity of point sets, an overlap may be defined in a nearest neighbor sense.

We apply KDE using an isotropic normal kernel for one point set and query the density for points in the other set:

$$p_{\mathcal{Y}}(\mathbf{x}_k) = \frac{1}{L}\sum_{l=1}^{L}\mathcal{N}(\mathbf{x}_k - \mathbf{y}_l, \sigma) \tag{28}$$

We reverse the roles of $\mathcal{X}$ and $\mathcal{Y}$ and compute $p_{\mathcal{X}}(\mathbf{y}_l)$. Then we count the points whose densities are above a threshold t:

$$o_{\mathcal{XY}} = \frac{1}{K+L}\left(\sum_{k=1}^{K} I(p_{\mathcal{Y}}(\mathbf{x}_k) > t) + \sum_{l=1}^{L} I(p_{\mathcal{X}}(\mathbf{y}_l) > t)\right) \tag{29}$$

We calculate $o_{\mathcal{XY}} \in [0, 1]$ for all pairs of point sets $(\mathcal{X}, \mathcal{Y})$, and split them into two groups: easy pairs ($o_{\mathcal{XY}} \geq t_o$) and hard pairs ($o_{\mathcal{XY}} < t_o$) according to $t_o \in [0, 1]$.

VI. Experiments

In order to analyze the registration losses described in previous sections we perform extensive evaluation on a number of real-world lidar point sets. We present an evaluation of the monotonicity in accordance with section V.

A. Setup

We perform the evaluation by sampling the registration losses at different transformations T_n (see section V). For each pair of point sets we consider both rotation and translation errors.

For each point set pair $(\mathcal{X}, \mathcal{Y})$, we first select a target point set and apply the transformation $T_n(\mathcal{Y})$. Next, we compute assignment sets $\mathcal{A}_k$ for each point $x_k \in \mathcal{X}$ using a nearest neighbor (NN) search in point set $T_n(\mathcal{Y})$. For the NN search we use the Euclidean distance metric, as in [21]. Losses are then computed according to Section IV-D.

Rotation sensitivity is assessed by rotating one of the point sets in the range $[0°, 30°]$ in steps of $1°$. This is done along $M = 10$ different axes, one of which is the up-vector, see section V-B.

Translation sensitivity is assessed by shifting one of the point sets in the range $[0, 7.5\text{m}]$ in steps of 0.25 meter. This is done along the same $M = 10$ axes used for rotation.

In experiments with combined rotation and translation, we simply apply the above deviations in both rotation and translation, starting with rotation.

B. Datasets

We use pairs of point sets from two *terrestrial laser scanners* (TLS) [3], [4]. Each of the datasets comes with ground truth point set poses, which we use to generate controlled deviations from the optimum. The datasets used are described in table I.

TABLE I
DATASETS USED FOR EVALUATION, ALONG WITH NUMBER OF POINT SETS, THE NUMBER OF COMBINATIONS OF SET PAIRS, AVERAGE POINT SET RANGE (IN METERS) AND AVERAGE OVERLAP (SEE (29)).

Dataset name	#point sets	#point-set pairs	range	overlap
VPS Indoors [3]	4	6	1.6	99.8%
VPS Outdoors [3]	4	6	4.0	97.5%
ETH Façade [4]	7	21	3.0	98.9%
ETH Arch [4]	5	10	12.8	56.7%
ETH Courtyard [4]	8	28	10.6	91.8%
ETH Office [4]	5	10	4.6	100.0%
Total	33	81	6.1	91.3%

For efficiency, each set is downsampled to $S = 10\,000$ points. This is done by generating a random index permutation, and picking the S first points. This sampling preserves underlying structure and density variations in the data.

C. Parameters

Covariances are computed with K-NN with $K = 20$ as in [21] (point-to-point, point-to-plane and plane-to-plane), or fixed radius search (shape-to-shape) with $r = 3h$ where $h = \sigma = 0.3$. For ML methods we use a robust cutoff of $d_{\max} = 3h = 0.9$, which is close to the values found in by threshold evaluation in [21]. We split the set of pairs into two groups, **easy** and **hard** cases, by using the degree of overlap (29). We set $t = 0.1$ when computing the list $O_{\mathcal{XY}}$ of overlaps, and $t_o = \text{median}(O_{\mathcal{XY}})$, which gives groups of roughly equal sizes. We use the step $s = 3$ for differentiation in MVP. Therefore, the plots start at $3 \times 1° = 3°$ for rotation and $3 \times 0.25\text{m} = 0.75\text{m}$ for translation.

D. Results and discussion

We present MVP curves for loss functions based on the following local distances: point-to-point (**pt2pt**), point-to-plane (**pt2pl**), plane-to-plane (**pl2pl**) and shape-to-shape (**sh2sh**). For the plane-to-plane we use a value of $\epsilon = 0.01$. We also include a version of plane-to-plane (**pl2pl'**) with the PCL [29] default value of $\epsilon = 0.001$. Below, we present results with ML and kernelized versions separately.

Maximum Likelihood: Figures 3 and 4 show MVP curves for ML losses under rotation and translation respectively. The plots start at the differentiation step size (3° + 0.75m).

An interesting observation is that the **pl2pl** loss has significantly more violations than the others in almost all scenarios. This is particularly evident in the translation error case (see figure 4). The exception is when close to the solution for the easy pairs under only rotation deviations close to the solution (see figure 3, right column). This is in contrast to what has been found elsewhere [21]. A possible reason for this is that convergence under translation on real scenes is rarely considered. By adjusting the ϵ parameter of **pl2pl**, the

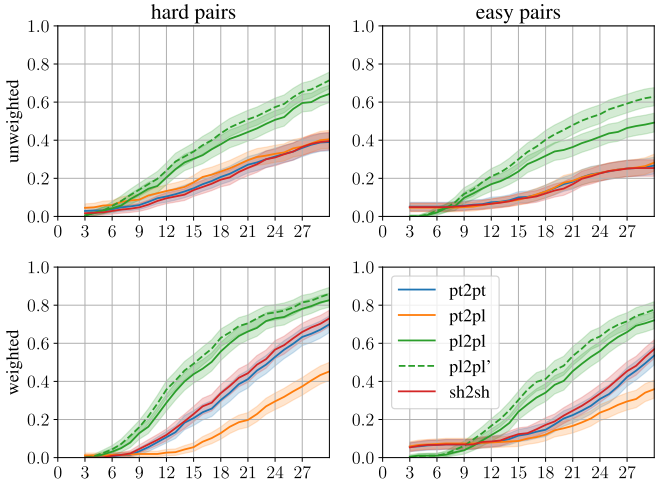


Fig. 3. MVP curves for ML losses under **rotation**. Unweighted (top) and weighted (bottom), for hard (left) and easy (right) cases, with point-to-point (pt2pt), point-to-plane (pt2pl), plane-to-plane (pl2pl has $\epsilon = 0.01$, pl2pl' has $\epsilon = 0.001$), and shape-to-shape (sh2sh). 95% confidence bands for each curve are shaded in the same colour. X-axis is degrees.

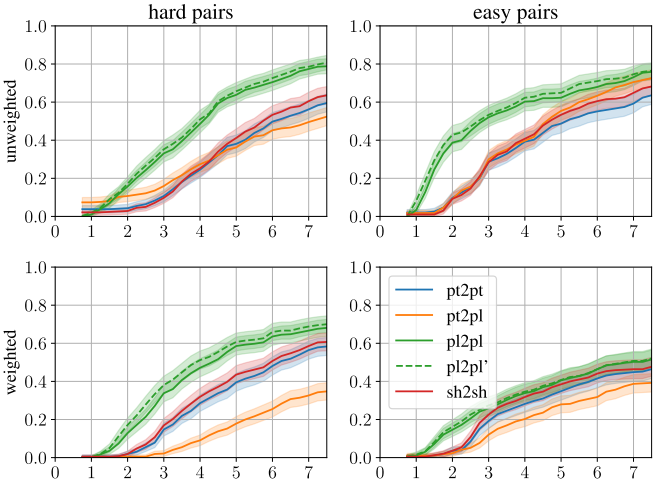


Fig. 4. MVP curves for ML losses for **translation** (see caption to Figure 3). X-axis is meters.

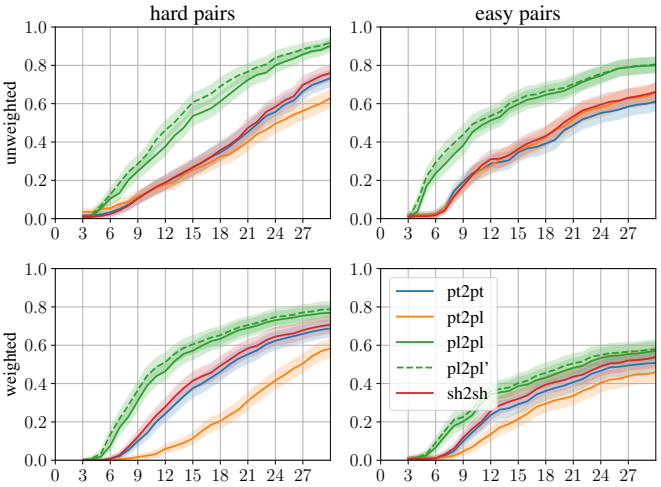


Fig. 5. MVP curves for ML losses under combined **rotation** and **translation** (see caption to Figure 3). X-axis is sample index.

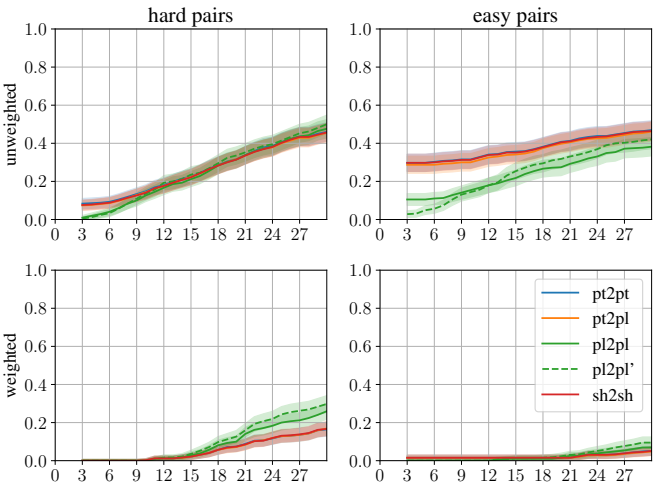


Fig. 6. MVP curves for KD losses under **rotation** (see caption to Figure 3).

performance can be improved somewhat over the PCL [29] default, but the general trend holds.

We can also see that weighting improves performance for all methods when close to the solution. In particular **pt2pl** benefits from weighting. When far from the solution, and under rotation, weighting results in more violations.

We also note that **sh2sh** behaves almost exactly like **pt2pt**, and is thus not worth the extra cost.

Finally, we present MVP curves under joint rotation and translation errors in Figure 5. Naturally, these curves show slightly more violations. Consistent with separating rotation and translation, these curves show that **pl2pl** has significantly more violations than the other methods, and that weighting improves performance.

Kernel Density: Figures 6 and 7 show MVP curves for KD losses under rotation and translation, respectively. Several conclusions from the ML-based methods also hold here: In the unweighted case, close to the solution, **pl2pl** has fewer violations than the other methods, while the behaviour for larger deviations is inconsistent. We also see again that **sh2sh** behaves almost exactly like **pt2pt**, and is thus not worth the extra cost. The performance change when changing the ϵ value in **pl2pl** is however insignificant in the KD case.

The most significant observation from 6 and 7, however is the large benefit from density based weighting in the KD case: *all* KD losses benefit from weighting, both under rotation and translation. Especially when close to the solution, we can see that the results are now nearly violation free. Although weighting is beneficial for **pl2pl**, it falls behind the others on the hard pairs. This demonstrates that weighting is beneficial in general, and should be used in all methods.

We also present MVP curves under joint rotation and translation errors in Figure 8. Similar to the MVP curves under rotation and translation separately, **pl2pl** has slightly more violations on the hard pairs. Moreover, the differences between methods on the easy pairs are insignificant.

VII. Conclusions

We have introduced monotonicity violation probability (MVP) curves as an analysis tool in point set registration.

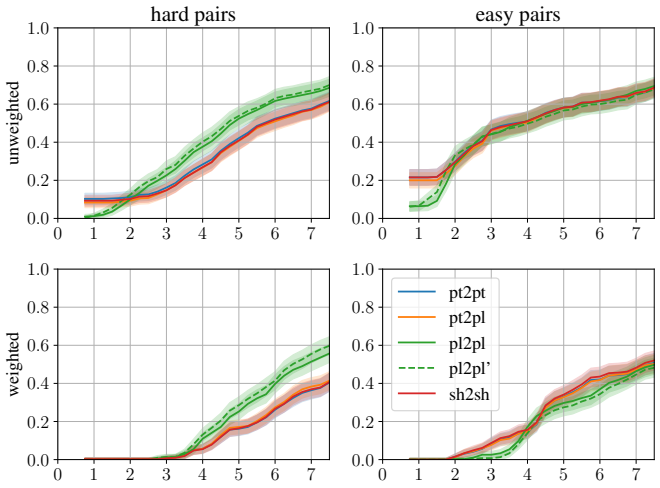


Fig. 7. MVP curves for KD losses for **translation** (see caption to Figure 3).

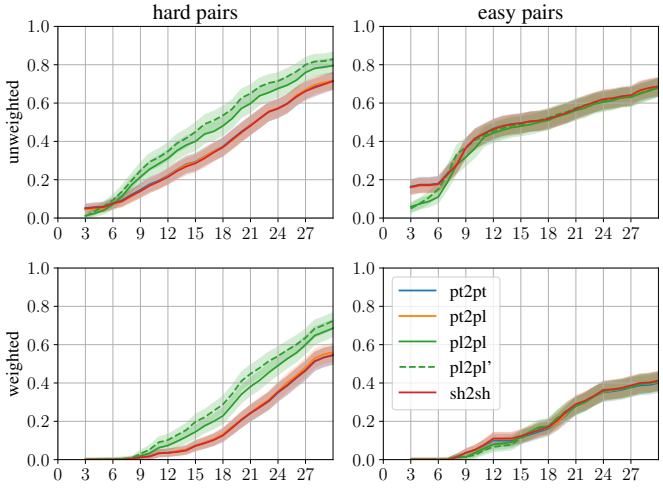


Fig. 8. MVP curves for KD losses under combined **rotation** and **translation** (see caption to Figure 3). X-axis is sample index.

They allow empirical analysis of only the cost instead of the entire algorithm. We have used the MVP curves to evaluate most common choices of local distances on a comprehensive dataset of 81 real terrestrial lidar scan pairs.

The experiments clearly demonstrate that density based weighting reduces the number of monotonicity violations in general. As density based weighting is not an expensive addition, it should thus be used in *all* registration methods.

We can further see that kernel based losses work very well together with density based weighting, resulting in very few violations in monotonicity. With a larger computational budget one could thus also consider running a weighted kernel method.

Further, our experiments indicate that plane-to-plane distances can increase robustness close to the solution, but in general introduce more violations than other local distances. This suggests a registration strategy, where initially a point-to-point based loss is used before switching to a plane-to-plane loss when the first optimizer converges.

We also introduced a local shape-to-shape distance, based on the Wasserstein distance for normal distributions. However, the evaluation shows it to perform near identical to point-to-point, and it is thus not worth the extra effort.

We hope that the introduction of MVP curves can help to improve the understanding of the losses used in point-set-registration. In the future they could also be a useful tool when designing novel loss functions.

REFERENCES

[1] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE TPAMI*, vol. 14, no. 2, p. 239256, 1992.
[2] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *IEEE ICRA'91*, April 1991.
[3] J. Unger, A. Gardner, P. Larsson, and F. Banterle, "Capturing reality for computer graphics applications," in *Siggraph Asia Course*, 2015.
[4] P. W. Theiler, J. D. Wegner, and K. Schindler, "Globally consistent registration of terrestrial laser scans via graph optimization," *ISPRS J. of Photog. and Remote Sens.*, vol. 109, pp. 126–138, 2015.
[5] F. J. Lawin, M. Danelljan, F. Khan, P.-E. Forssén, and M. Felsberg, "Density adaptive point set registration," in *IEEE CVPR'18*, June 2018.
[6] F. Pomerlau, F. Colas, R. Siegwart, and S. Magenant, "Comparing ICP variants on real-world datasets," *Aut. Rob.*, vol. 34, no. 3, April 2013.
[7] F. Pomerlau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Foundations and Trends in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.
[8] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *IJCV*, vol. 13, no. 2, pp. 119–152, 1994.
[9] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," in *British Machine Vision Conference*, 2001, pp. 662–670.
[10] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH'96*, 1996.
[11] R. A. Newcombe and et al., "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE ISMAR'11*, October 2011.
[12] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE TPAMI*, vol. 32, no. 12, pp. 2262–2275, 2010.
[13] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis, "A generative model for the joint registration of multiple point sets," in *ECCV'14*. Springer, 2014, pp. 109–122.
[14] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *ECCV'04*. Springer, 2004, pp. 558–569.
[15] M. Magnusson and et al., "Scan registration for autonomous mining vehicles using 3D-NDT," *JFR*, vol. 24, no. 10, pp. 803–827, 2007.
[16] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE TPAMI*, vol. 33, no. 8, pp. 1633–1645, 2011.
[17] I. Bogoslavskyi and C. Stachniss, "Analyzing the quality of matched 3d point clouds of objects," in *IROS'17*, 2017.
[18] C. Torre-Ferrero, J. Llata, S. Robla, and E. Sarabia, "A similarity measure for 3D rigid registration of point clouds using image-based descriptors with low overlap," in *IEEE ICCV Workshops*, 2009.
[19] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *IJCV*, vol. 103, no. 3, pp. 267–305, July 2013.
[20] M. Zefran, V. Kumar, and C. Croke, "Metrics and connections for rigid-body kinematics," *IJRR*, vol. 18, no. 2, pp. 242–1–242–16, 1999.
[21] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized ICP," in *Robotics: Science and Systems (RSS09)*, 2009.
[22] C. R. Givens and R. M. Shortt, "A class of Wasserstein metrics for probability distributions," *Michigan Math. Journal*, 1984.
[23] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *NIPS'13*, 2013.
[24] S. Koluri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE S. P. Magazine*, vol. 34, no. 4, July 2017.
[25] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *ECCV'02*. Springer, 2002.
[26] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
[27] A. Agresti and B. A. Coull, "Approximate is better than "exact" for interval estimation of binomial proportions," *The American Statistican*, vol. 52, no. 2, pp. 119–126, May 1998.
[28] E. W. Weisstein, "Platonic solid," From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/PlatonicSolid.html.
[29] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *ICRA'11*. IEEE, 2011, pp. 1–4.