Analysis of Robust Functions for Registration Algorithms

Philippe Babin, Philippe Giguère and François Pomerleau*

***Abstract*— Registration accuracy is influenced by the presence of outliers and numerous robust solutions have been developed over the years to mitigate their effect. However, without a large scale comparison of solutions to filter outliers, it is becoming tedious to select an appropriate algorithm for a given application. This paper presents a comprehensive analysis of the effects of outlier filters on the Iterative Closest Point (ICP) algorithm aimed at a mobile robotic application. Fourteen of the most common outlier filters (such as M-estimators) have been tested in different types of environments, for a total of more than two million registrations. Furthermore, the influence of tuning parameters has been thoroughly explored. The experimental results show that most outlier filters have a similar performance if they are correctly tuned. Nonetheless, filters such as *Var. Trim.*, *Cauchy*, and *Cauchy MAD* are more stable against different environment types. Interestingly, the simple norm L_1 produces comparable accuracy, while being parameterless.**

I. Introduction

A fundamental task in robotics is finding the rigid transformation between two overlapping point clouds. The most common solution to the point cloud registration problem is the ICP algorithm, which alternates between finding the best correspondence for the two point clouds and minimizing the distance between those correspondences [1], [2]. Based on the taxonomy of Rusinkiewicz *et al.* [3], Pomerleau *et al.* [4] proposed a protocol and a framework to test and compare the common configurations of ICP. They simplified the process to four stages: 1) data point filtering, 2) data association, 3) outlier filtering, and 4) error minimization. The stage on outlier filtering is necessary as the presence of a single outlier with a large enough error could have more influence on the minimization outcome than all the inliers combined. To solve this generic problem, Huber [5] extended classical statistics with robust cost functions. A robust cost function reduces the influence of outliers in the minimization process. The most common class of these robust functions is the maximum likelihood estimator, or M-estimator. Other solutions exist, which rely either on thresholds (i.e., hard rejection) or on continuous functions (i.e., soft rejection).

To the best of our knowledge, the current research literature is missing a comprehensive comparison of outlier filters for ICP (i.e., Stage 3 of the ICP pipeline). In the case of ICP used in mobile robotics, outliers are mainly caused by non-overlapping regions, sensor noises and shadow points produced by the sensor. Most papers about outlier filters compare their own algorithm with only two other algorithms [6]–[8] or only on a single dataset [7]–[10]. Few papers evaluate the influence of the overlap between the two point clouds and the initial perturbation, which are leading error causes for ICP [11]. Furthermore, the dataset selected for evaluation varies depending on research fields. For instance, papers targeting object reconstruction will use the Stanford dataset [8], [12], [13]. Results obtained with a dataset containing exclusively objects might be too specific and consequently not translate well to the field of mobile robotics, because of the difference in structure, density, and scale. Also, experiments on registration performances tend to modify multiple stages of ICP at once, making it difficult to estimate the impact of outlier rejection algorithms on the overall performance. Additionally, few papers on outlier filters evaluate the impact of the tuning parameters within those outlier rejection algorithms. The influence of an outlier for a given error can change drastically, as a function of the tuning parameter value. Finally, with the rise of the number of ICP variants [11], it is becoming tedious to select the appropriate robust solution for an application in mobile robotics.

To mitigate those problems, we propose a comprehensive analysis of outlier filter algorithms. To this effect, the main contributions of this paper are: 1) a large scale analysis investigating 14 outlier filters subject to more than two million registrations in different types of environment; 2) a consolidation of the notion of point cloud *crispness* [14] to a special case of a common M-estimator, leading to a better understanding of its use in the outlier filtering

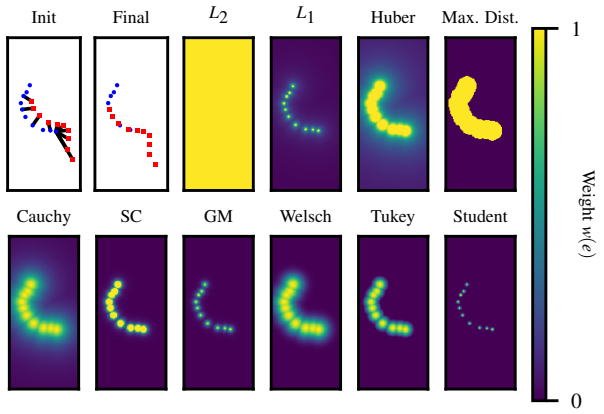

Fig. 1: The effect of different outlier filters on the registration of two overlapping curves. The blue circles and red squares are respectively the reference points and the reading points. The initial graph depicts the data association from the nearest neighbor at the first iteration. The final graph shows the registration at the last iteration. The other graphs depict scalar fields of the weight function $w(e)$ of a reading point. For L_2, all points have the same weight. L_1 gives an infinite weight to a point directly on top of a reference point.

*The authors are from the Northern Robotics Laboratory, Université Laval, Canada.
{philippe.babin.1@, philippe.giguere@ift., francois.pomerleau@ift.}ulaval.ca

stage; 3) a support to better replication of our results with the open-source implementations of tested outlier filters in `libpointmatcher`[1].

II. Related Works

Prior to the existence of ICP, a seminal work on M-estimators was conducted by Welsch [15], where he surveyed eight functions now known as the classic M-estimators. In the context of camera-based localization, MacTavish *et al.* [16] made a thorough effort by comparing seven robust cost functions, but their conclusions do not translate directly to point cloud registrations. For ICP, Pomerleau *et al.* [11] proposed an in-depth review of the literature explaining how outlier filters must be configured depending on the robotic application at hand. Unfortunately, their investigation is limited to listing current solutions, without comparison. These surveys guided us in the selection of our list of solutions analyzed in this paper.

When it comes to comparing outlier filters, the most common baseline is vanilla ICP (i.e., labeled L_2 hereafter), which does not have any outlier filter and directly minimizes a least-squared function.

In terms of hard rejections, Phillips *et al.* [12] compared a solution using an adaptive trimmed solution against a manually adjusted trimmed threshold [3] and L_2. However, they limited their analysis to a pair of point clouds with overlap larger than 75% and using simulated outliers. Another adaptive threshold solution, this time related to simulated annealing algorithms [9], was compared to five types of hard rejection algorithms. Unfortunately, they used a limited 2D dataset relying on ten pairs of scans with similar overlap. As for soft rejections, Bergström *et al.* [17] compared the effect of three M-estimators on ICP and proposed an algorithm to auto-tune them, but they only provided results based on simulated data of simple geometric shapes. Agamennoni *et al.* [18] proposed a soft rejection function based on the Student's T-distribution for registration between a sparse and a dense point cloud. But, they compare their algorithm to another complete ICP solution, where multiple stages changed. Bouaziz *et al.* [13] introduced a soft rejection function based on L_p norms, however, they did not provide qualitative evaluation and their solution does not use a standard least-squared minimizer. Closer to the topic of our analysis, Bosse *et al.* [10] compared five M-estimators and another robust function against L_2, for a variety of minimization problems, one of which was ICP. However, the ICP analysis was limited to examples relying on a single 2D pair of point clouds. In our analysis, a common 3D benchmark is used for all solutions allowing a fair comparison of robust cost functions.

When an outlier filter relies on fixed parameters, it is important to know their effects on registration performance. Segal *et al.* [19] removed matching points with residual errors higher than a given value (i.e., named *max. distance* hereafter) and evaluated the influence of this parameter on ICP. They concluded that the value used for this parameter is a trade-off between accuracy and robustness. Bosse *et al.* [20] compared the same outlier filter against one of the classic M-estimator *Cauchy* in an 2D outdoor environment. They concluded that *max. distance* is less robust to the parameter value than *Cauchy*, but it provided better accuracy when correctly tuned.

For the most part, a robust cost function relies on fix parameters, configured by trial and error. To sidestep this issue, some outlier filters are designed so as to be auto-tunable. For instance, Haralick *et al.* [21] and Bergström *et al.* [17] have both proposed an algorithm to auto-scale M-estimators. However, Bergström *et al.* [17] requires two additional hyper-parameters to tune the auto-scaler: one representing the sensor's standard deviation and the other specifying the decreasing rate to reach the standard deviation. In the case of outlier filters based on a threshold following a given quantile of the residual error, Chetverikov *et al.* [22] proposed an estimator to tune the overlap parameter, which uses an iterative minimizer. Unfortunately, having an iterative minimizer inside ICP's iterative loop becomes computationally intensive [23], thus most of the implementation resorts to manually fixing the quartile. Phillips *et al.* [12] improved on [22] by minimizing the Fractional Root Mean Squared Distance (FRMSD) to tune the parameter representing the overlap ratio. It removes the need for an inner loop to estimate the parameter, thus speeding up the execution by a factor of at least five. But, this algorithm performance also depends on a hyper-parameter that needs to be tuned [23]. In this paper, we also explore the influence of tuning parameters, testing even the stability of hyper-parameters, and provide a methodology to tune them.

III. Theory

The ICP algorithm aims at estimating a rigid transformation $\hat{\boldsymbol{T}}$ that best aligns a *reference* point cloud $\mathcal{Q}$ with a *reading* point cloud $\mathcal{P}$, given a prior transformation $\check{\boldsymbol{T}}$. The outlier filtering stage of ICP has strong ties to error minimization. The former reduces the influence of wrongful data association, while the latter finds a solution that respects the constraints of the previous stage. In the context of ICP, these two stages can be summarized as estimating the rigid transformation $\hat{\boldsymbol{T}}$ by minimizing

$$\hat{\boldsymbol{T}} = \arg\min_{\boldsymbol{T}} \sum_{i=1} \sum_{j=1} \rho\Big(e(\boldsymbol{T}, \boldsymbol{p_i}, \boldsymbol{q_j})\Big), \tag{1}$$

where $\rho(\cdot)$ is the cost function. The error function $e(\cdot)$ is the scaled distance between matched points, defined as

$$e(\boldsymbol{T}, \boldsymbol{p_i}, \boldsymbol{q_j}) = \frac{\|\boldsymbol{T}\boldsymbol{p_i} - \boldsymbol{q_j}\|}{s}, \tag{2}$$

where $\boldsymbol{p_i} \in \mathcal{P}$ and $\boldsymbol{q_j} \in \mathcal{Q}$. This is equivalent to a simplified version of the Mahalanobis distance, where the scale s is uniform on all dimensions. For the original version of ICP, s is set to one. The double summation of Equation 3 is expensive to compute and is typically approximated using a subset of pairs, using nearest neighbor points of each $\boldsymbol{p_i}$. To simplify the notation, we will use $e_m(\cdot)$ for each error to be minimized, with m being the index of this subset. Since $\rho(\cdot)$ is non-convex, the minimization must be solved iteratively, in a manner similar to the Iteratively Reweighted

[1] https://github.com/ethz-asl/libpointmatcher

Least-Squares (IRLS) [17] by decomposing $\rho(\cdot)$ as a weight w and a squared error term, such that

$$\hat{\boldsymbol{T}} \approx \arg\min_{\boldsymbol{T}} \sum_{m=1} w\big(e_m(\check{\boldsymbol{T}})\big)\ e_m(\boldsymbol{T})^2. \tag{3}$$

At each iteration, the prior transformation $\hat{\boldsymbol{T}}$ is assigned to the last estimated transformation until convergence. Figure 1 shows a toy example of registration. The subsequent scalar fields show the impact of different weight functions $w(\cdot)$ in the neighborhood of $\mathcal{Q}$. A notable example is L_2, where the lack of weights is equivalent to using a constant weight for all pairs. Beyond L_2, outlier filtering is all about the choice of this weight function $w(\cdot)$ and its configuration. Table I shows a list of outlier functions used in this paper, with a dedicated column highlighting their implementation of $w(\cdot)$.

TABLE I: Descriptive table of robust cost functions used in this analysis expressed with respect to their tuning parameter k and the scaled error e.

Functions	Conditions	Cost $\rho(e)$	Weight $w(e)$	M
L_2		$\frac{e^2}{2}$	1	✓
L_1		$\lvert e\rvert$	$\frac{1}{\lvert e\rvert}$	✗
Huber	$\begin{cases}\lvert e\rvert \le k\\ \text{otherwise}\end{cases}$	$\begin{cases}\frac{e^2}{2}\\ k(\lvert e\rvert - k/2)\end{cases}$	$\begin{cases}1\\ \frac{k}{\lvert e\rvert}\end{cases}$	✓
Cauchy		$\frac{k^2}{2}\log(1+(e/k)^2)$	$\frac{1}{1+(e/k)^2}$	✓
GM		$\frac{e^2/2}{k+e^2}$	$\frac{k^2}{(k+e^2)^2}$	✓
SC	$\begin{cases}e^2 \le k\\ \text{otherwise}\end{cases}$	$\begin{cases}\frac{e^2}{2}\\ \frac{2ke^2}{k+e^2} - k/2\end{cases}$	$\begin{cases}1\\ \frac{4k^2}{(k+e^2)^2}\end{cases}$	✓
Welsch		$\frac{k^2}{2}(1-\exp(-(\frac{e}{k})^2))$	$\exp(-(e/k)^2)$	✓
Tukey	$\begin{cases}\lvert e\rvert \le k\\ \text{otherwise}\end{cases}$	$\begin{cases}\frac{k^2(1-(1-(\frac{e}{k})^2)^3)}{2}\\ \frac{k^2}{2}\end{cases}$	$\begin{cases}(1-(e/k)^2)^2\\ 0\end{cases}$	✓
Student			$\frac{(k+3)(1+\frac{e^2}{k})^{-\frac{k+3}{2}}}{k+e^2}$	✗
Max. Dist.	$\begin{cases}\lvert e\rvert \le k\\ \text{otherwise}\end{cases}$	$\begin{cases}\frac{e^2}{2}\\ \frac{k^2}{2}\end{cases}$	$\begin{cases}1\\ 0\end{cases}$	✓
Trimmed	$\begin{cases}e \le P_f\\ \text{otherwise}\end{cases}$		$\begin{cases}1\\ 0\end{cases}$	✗

Legend: M = Is the function a M-Estimator?

A. Hard rejection

Outlier filters categorized as hard rejection define the result of $w(\cdot)$ to be binary (i.e., either zero or one). The two most commons solutions are *Max. distance* and *Trimmed*. The solution *Max. distance* rejects any match with a distance larger than a threshold. *Trimmed* only keeps the error below the f^{th} percentile P_f of the matches, where f is the *overlap ratio parameter*. This makes the registration accuracy directly related to how close this parameter f is to the actual overlap between reference and reading. In that sense, if f deviates from the true overlap, the accuracy will degrade. In applications where the overlap is unknown or changes often, selecting a fix overlap ratio f becomes challenging. A variant of *trimmed*, *Var. Trimmed* [12], calculates the FRMSD for all possible overlap ratios, and selects the ratio with the minimum FRMSD value as f. The *Median filter* is also used, but is a special case of *Trimmed*, where $f = 50\,\%$.

B. Soft Rejection and M-estimators

Outlier filters using soft rejection output a weight where $w(\cdot) \in \mathbb{R}^+$. The most common soft rejection algorithms are M-estimators. To be an M-estimator, a cost function $\rho(\cdot)$ must fulfill three conditions, which are to be *1)* symmetric, *2)* non-negative, and *3)* monotonically-increasing [21]. Those conditions do not limit M-estimators to robust cost functions as, for example, L_2 satisfies all of them. Moreover, cost functions associated with M-estimators are analyzed using their influence function $\psi(\cdot)$ and weight function $w(\cdot)$, such that

$$\psi(e) = \frac{\partial \rho(e)}{\partial e} \qquad \text{and} \qquad w(e) = \frac{\psi(e)}{e}.$$

The influence function $\psi(\cdot)$ is used to evaluate whether an M-estimator is robust or not. If $\psi(\cdot)$ is non-monotonic (i.e., redescending) and is null for an error that tends to infinity, the M-estimator is considered robust. As for the weight function $w(\cdot)$, it is given for convenience since it is the only part required to implement an M-estimator for an IRLS solution, as in Equation 3. In the soft rejection algorithms considered in this paper (shown in Table I), two are not M-estimators: L_1 has a singularity for an error that is equal to zero, and *Student* has an undefined $\rho(\cdot)$.

It is worth noting that some soft rejection functions are related. For instance, *Huber* uses a parameter k to combine L_1 and L_2, in order to avoid the singularity at $e = 0$ of L_1. Also, *Switchable-Constraint* (labeled *SC* hereafter), typically used in pose graph Simultaneous Localization and Mapping (SLAM), was expressed as a combination of L_2 and *Geman-McClure* (labeled *GM* hereafter) using a parameter k. It shares the same cost function as Dynamic Covariance Scaling [16]. *SC* is expected to have similar results to *GM* for extreme values of k.

C. Estimating the scale

As expressed in Equation 2, we used a scaled error in our ICP implementation. Contrary to the filter parameter k, which should be globally constant, the scale s is related to the point clouds and can be either fixed or estimated at every iteration. The scale s relates to the uncertainty for which paired points with a certain error should be considered as outliers. There are multiple estimators for the scale, two of the most interesting are: 1) Haralick *et al.* [21] used the Median of Absolute Deviation (MAD) as a scale estimator and calculated it at each iteration; 2) Bergström *et al.* [17] starts with $s = 1.9 \cdot \text{median}(e)$ and then gradually decrease s at each iteration, to asymptotically reach a standard deviation σ_*. The parameter ξ controls the convergence rate.

D. Relating crispness to the M-estimator Welsch

The notion of *crispness* as a measure of how well two point clouds are aligned was introduced by Sheehan *et al.* [24] in the context of sensor calibration. It originates from the use of a Gaussian kernel in a measurement of Rényi Quadratic Entropy (RQE) for a kernel correlation approach to registration [25]. RQE has the following cost function:

$$\rho_{\text{rqe}}(e_*) = \exp\left(-\frac{e_*^2}{4\sigma^2}\right), \tag{4}$$

where e_* is the unscaled error and σ is a tuning parameter. RQE has been described as an M-estimator by [25], however, it has not been related to an existing M-estimator. This cost

function is a special case of *Welsch* with $k = 2$ and $s = \sigma$. This means that minimizing for RQE is the same thing as using a *Welsch* M-estimator with this configuration. Thus, this configuration is expected to have good accuracy.

IV. Experiments

Our study focuses on registration-based localization in the context of mobile robotics. Our analysis follows a similar methodology as in Pomerleau *et al.* [4], with a strict focus on changing the outlier filtering stage of ICP. The other stages were kept the same. Table II describes the ICP pipeline used.

A number of key factors were selected to be tested jointly, in order to determine their influence. These factors were: **1)** the **outlier filter**, selected based on their popularity and their interesting properties (total of 12); **2)** the **configuration parameter** of the filters; **3)** the **environment** type (indoor, outdoor, etc.); and **4)** the **overlap** between the reference and reading scans. For each factor selection above, 128 registrations were computed with a **random perturbation** from the ground truth. How some of these factors were sampled during experiments is detailed below.

Configuration parameter sampling Filter parameter values were sampled in a way to ensure efficient exploration of their configuration space. For instance, the tuning parameters k of all M-estimators (i.e., *Huber*, *Cauchy*, *SC*, *GM*, *Welsch* and *Tukey*) and *Student* were sampled evenly on a log scale. Two sampling value regions were defined. The first region was $Z_1 \in \left[1 \times 10^{-6}, 0.1\right[$ and the second one was $Z_2 \in \left[0.1,\ 100\right]$. Z_1 explores the asymptotic behavior of the algorithm for near-zero k values, while Z_2 is where the parameter with the best accuracy is located for most estimators. They are sampled 20 times for Z_1 and 30 times for Z_2. For the error scale s for M-estimator (see Eq. 2), two auto-scalers (*Berg.* and MAD) and one fixed scale have been tested. All three permutations were tested on *Cauchy*, while all other M-estimators were tested only with MAD. In the case of the *Berg.* auto-scaler, we used a convergence rate of $\xi = 0.85$, as it was found to be the best one in our analysis. Its parameter σ_* was sampled in the same two zones as the M-estimator (Z_1 and Z_2). *Cauchy Berg.* used the tuning parameter $k_{cauchy} = 4.304$, and it was selected based on [17]. If the estimator used was MAD, then $s = MAD(\boldsymbol{e})$, otherwise $s = 1$. In the case of the *Trimmed* filter, its overlap parameter f has been sampled linearly 20 times in the range $\left[1 \times 10^{-4}, 100\right]\%$. For the *Var. Trim.* filter, the minimum and maximum overlap parameters have been set to 40% and 100% respectively. Its λ parameters have been sampled linearly 20 times between 0.8 and 5. The filter *Max. Dist.* has been sampled 20 times linearly in the range $\left[0.1, 2\right]$. Finally, the parameter-less L_1 and L_2 filters were used as-is.

Environments Experiments were performed on the *Challenging Datasets* [26]. These provide a ground truth with *mm*-level of precision. Our analysis used 3 sets of point clouds from this dataset, one per type of environment: structured (*Hauptgebäude*), semi-structured (*Gazebo Summer*) and unstructured (*Wood Summer*). As each of them contains around 35 point clouds, it allows a fine control of the ratio of overlap between point clouds. For each set, 12 pairs of point clouds where selected, to uniformly sample the overlap between 40% and 100%.

TABLE II: Listing of the configuration of `libpointmatcher` used.

Stage	Configuration	Description
Data association	`KDTree`	Three matches per point
Data filtering	`SurfaceNormal`	Density with 20 neighbors
	`MaxDensity`	Limit density to 10k pts/m^3
	`RandomSampling`	Keep 75% of points
Error Min.	`PointToPlane`	Point-to-plane error
Trans. Checking	`Differential`	Stop below 1 mm and 1 mrad
	`Counter`	Max. iteration count is 40

Initial perturbation on $\boldsymbol{T}_0$ For fine registration, ICP requires a prior on the transformation between the *reading* and *reference* point clouds. The performance accuracy of ICP is directly impacted by the distance between the initial transformation $\boldsymbol{T}_0$ and the ground truth. For each of the previous factors, 128 initial transformations $\boldsymbol{T}_0$ were generated by adding a random perturbation sampled from a uniform distribution, and centered at the ground truth. To stress-test ICP, we chose a perturbation as challenging as the hard perturbation of [4]. The perturbation in translation was generated by sampling a point in a sphere with a 1 m radius, while the one in rotation was generated by first sampling from an uniform angle distribution between 0 and 25 deg and then applying it around a random 3D vector.

Our evaluation of accuracy used the transformation error $\boldsymbol{\Delta}$ defined as $\boldsymbol{\Delta} = \boldsymbol{T}_{gt}^{-1}\boldsymbol{T}_{final}$ where $\boldsymbol{T}_{gt}$ is the ground truth and $\boldsymbol{T}_{final}$ is the transformation at the last iteration. $\boldsymbol{\Delta}$ is further separated into two components, for easier interpretation: $\boldsymbol{\Delta}_R$ for the 3x3 rotation matrix and $\boldsymbol{\Delta}_T$ for the 3x1 translation vector. Finally, the translation error was evaluated with the euclidean distance of $\boldsymbol{\Delta}_T$, and the rotation error metric is $\theta = \arccos(\frac{trace(\boldsymbol{\Delta}_R)-1}{2})$.

V. Results

In the first set of tests (Section V-A), we performed over 2.3 million pairwise registrations to exhaustively evaluate the outlier filter solutions. These registrations were computed offline using Compute Canada's Supercomputers. In the second set of experiments (Section V-B), we collected data with a mobile robot on an indoor-outdoor trajectory. We then tested on this trajectory the two best performing filters (Section V-A) for a real-time mapping task (Figure 4).

A. Pairwise Outlier Filter Tests

Pairwise registration results are broken down into three parts. We first discuss the performance (median error) of each filter as a function of its parameter, for all environments compounded (Figure 2). We then closely analyze the distribution of errors for the best parameter values found from this search, for the three environments (Figure 3). In the third part, we evaluate how robust the best parameters are to a change in the environment (Table III).

1) Parameter Search over all Environments Compounded: We computed the error metrics $\|\boldsymbol{\Delta}_T\|$ for all registrations, irrespective of the environment. Figure 2 shows the median translation error for M-estimators, as a function of the filter's tuning parameter value. We can see that all filters with parameters have a single global minimum, located in

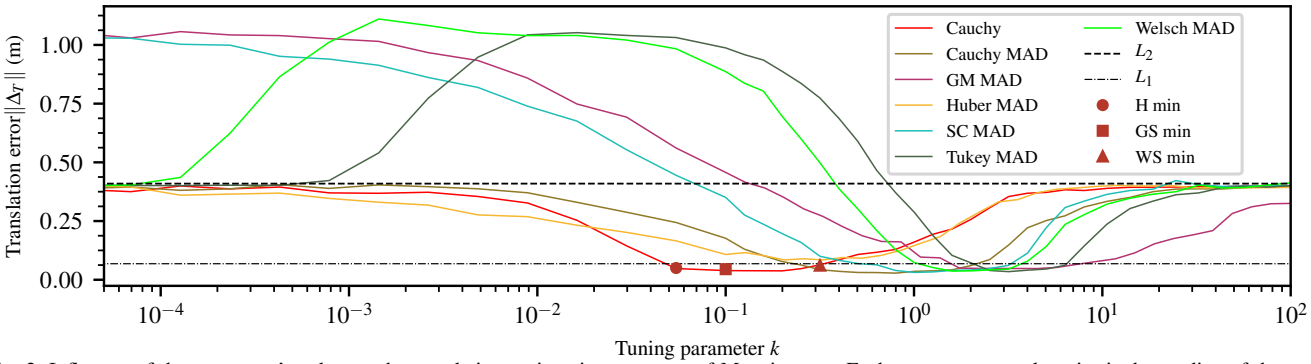

Fig. 2: Influence of the parameter's value on the translation registration accuracy of M-estimators. Each parameter sample point is the median of the error for all datasets for all overlap. The H min, GS min and WS min correspond to the parameter of *Cauchy* with the minimum median error for the environment *Hauptgebäude*, *Gazebo Summer*, and *Wood Summer*.

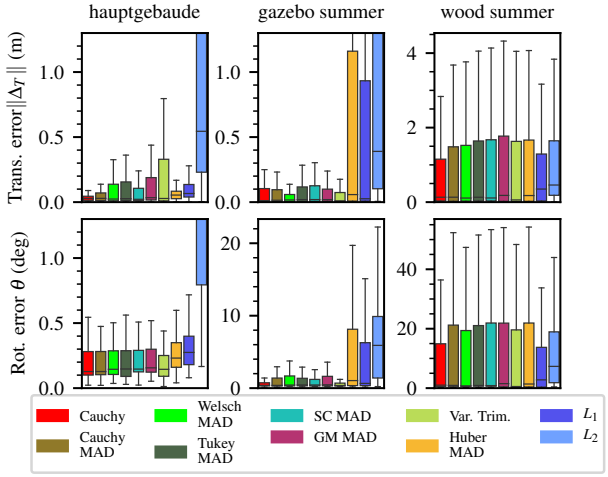

Fig. 3: Box plot of the performance of each outlier filter for the parameter value with lowest median error in that particular environment.

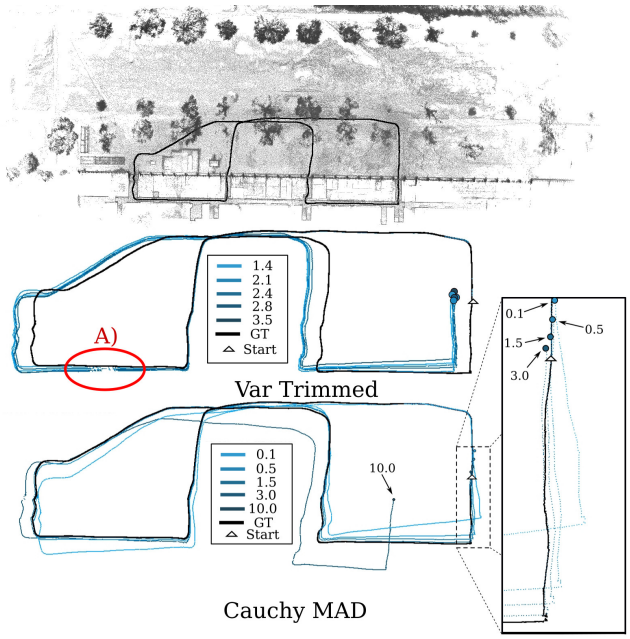

Fig. 4: Testing the ICP accuracy in a real time 3D SLAM application with a Husky A200 following a challenging indoor-outdoor route on Université Laval's campus. The start and end of the route are at the same location. The black trajectory (ground truth) was calculated offline. The end position of the trajectory for each parameter value is represented by a circle. In A), all configurations of *Var. Trimmed* could not converge correctly in the corridor.

a relatively flat valley. Most solutions have a similar best performance, while *Huber MAD* slightly under-performing.

Comparison with L_2 (Vanilla ICP): For large values of configuration parameter k, all filters (excepted *GM*) performed similarly to L_2 (top dashed line), i.e. outlier filtering is effectively disabled. This is not surprising, as these M-estimators have the property that $\lim_{k\to\infty} w(e,k) \approx 1$. For small values of k, the performance depends on the estimator used. It can nevertheless be categorized into two trends. For *Huber MAD*, *Cauchy* and *Cauchy MAD*, the translation error degrades smoothly towards L_2 as $k \to 0$, without surpassing it. For the four other filters, performance can become *much worse* than L_2 with $k \to 0$, although for *Welsch* and *Tukey* the performance eventually goes back down to L_2. From all this, we can conclude that: *1)* it is preferable to overestimate the parameter for all M-estimators than to underestimate it, to avoid too much rejection of inliers; *2)* peak performance varied little from one filter to another (less than 2 cm), except for *Huber MAD*; and *3) Huber MAD*, *Cauchy* and *Cauchy MAD* never performed worse than L_2.

Comparison with L_1: Despite being unsophisticated, L_1 almost always outperforms all outlier filters approaches, except for a narrow band of parameter values. *Huber* is the only M-estimator that does not outperform L_1. However, as we will see later in Figure 3, L_1 exhibits a much greater variance in certain environments, which could be problematic if one is interested in minimizing the risk of being lost.

Other observations: *Cauchy* and *Cauchy MAD* have the same curve, but with an offset, with *Cauchy MAD*'s minimum centered at $k = 1$ by the auto-scaler. Since *GM* and *SC* shares the same function, they have a similar curve, differing only by a slight offset. *Welsch* and *Tukey* also have similar curves with an offset, despite having very different weight functions $w(e)$.

2) Distribution of errors for the best fixed filter parameter, for three environments: In Figure 2 we showed only the median error, for all environments combined together. As such, this median does not tell the whole story. We thus computed in Figure 3 the distribution of errors of all filtering approaches, for three different environments. It is important to note that the filter parameters were fixed in these experiments, and were established from the best results found in Figure 2. From these experiments, the

environment influence on the registration is clear: the structured one (*Hauptgebäude*) is noticeably easier in rotation, while the unstructured one (*Wood Summer*) is by far the hardest in rotation and in translation. Furthermore, all error distributions are asymmetric (heavy-tailed). Apart from the unstructured environment of *Wood Summer*, all outlier filters have error spreads and median errors significantly smaller than L_2. This further confirms the lack of robustness of L_2. Although L_1 has a favorable median error in the semi-structured environment *Gazebo Summer*, its error spread is much greater that most other M-estimators. The performance of *Huber* is even worse than that of L_1 in that environment despite being a "robust" version of the former. Finally, we observed that *Cauchy* has an error spread noticeably smaller than its auto-scaled counterpart (*Cauchy MAD*).

3) Robustness of best fixed parameter across Environments: To determine if a filter is robust across environment changes, we used the following metric: if the best parameter of a filter for each three environments are all within the global flat valley, this filter is considered robust. We define this global flat valley as the range of parameters that perform better than L_1, when comparing the median translation error. For instance, Figure 2 shows the best parameters of *Cauchy* for three environments. Since all three parameter values have an error below L_1, *Cauchy* is robust as per our metric.

In Table III, the best performing parameters for all filters are presented, for 3 environments. The best overall filter for these experiments is *Var. Trim*. It has half of the error of the second best filter in *Wood Summer*. It is also the only hard rejection filter that met our robustness criteria describes above. Filters such as *Cauchy*, *Cauchy MAD*, *Welsch MAD*, and *Tukey MAD* are also robust across our environment changes, as they all have values below L_1 (indicated by bold notation). On the subject of auto-scaling, *Cauchy Berg* is out performed by *Cauchy* and *Cauchy MAD* for all environments except *Wood Summer*. The optimal parameter of *Welsch* for two environments is $k = 2$, the exact value for RQE kernel function (as discussed in Section III-D), which means that our experiments agree with the theory around kernel function. The parameters of *Max distance* and *Trim* depend on the environment, possibly indicating a lack of robustness. We can attribute this to the fact that they are hard rejection, and that they lack the adaptability of *Var. Trimmed*.

B. Test on a full Indoor-Outdoor Trajectory

In Figure 4, two filters (*Cauchy MAD* and *Var. Trim.*) were tested in a challenging route containing both indoor and outdoor portions. These filters were chosen for their good performance in our offline tests. The robot used a Velodyne HDL-32e for scanning and used a mix of wheel encoder and an Xsens MTi-30 IMU for odometry. To estimate the trajectory, an ICP-based SLAM was used with a moving window map. The moving window map was created by randomly decimating the map's point. For ground truth, the ICP-based SLAM was done offline without moving windows. The outlier filter performance was evaluated by comparing the cumulative registration error to the ground truth. Multiple parameter values were tested, all near the best value from Table III. For *Var. Trim.*, all parameter values had difficulties in the highlighted zone A in Figure 4, which occurs in a small corridor. This confirmed that *Var. Trim.* has problems in structured environments as in Figure 3-*Hauptgebäude*. For *Cauchy MAD*, all tested parameters finished close to the ground truth. The parameter with the best performance was $k = 3.0$, which demonstrates that the valley for this experiment is shifted to higher k values than our offline tests. We think that this change is caused by the different sensors used, and because this test is a registration between scan to (small) map, while our offline tests were between two scans.

TABLE III: All 14 outlier filters were tested on the 3 environments and for a variety of parameter values. The parameter corresponding to the smallest median error for each dataset is shown. In the median error columns, the outlier filter with the best performance is in bold. In the parameter columns, if the parameter value is located inside the global flat valley, then it is in bold. If all three environments are within the valley, then 'All' is in bold.

Outlier Filters	Median Error (mm)				Parameters			
	H	GS	WS	All	H	GS	WS	All
L_2	544	390	459	409	n/a	n/a	n/a	n/a
L_1	66	25	350	68	n/a	n/a	n/a	n/a
Huber MAD	54	58	176	84	0.05	0.33	0.67	0.33
Cauchy	28	11	131	37	**0.05**	**0.10**	**0.32**	**0.20**
Cauchy MAD	28	14	132	**28**	**0.40**	**1.00**	**1.59**	**0.80**
Cauchy Berg	39	18	99	39	**0.01**	**0.00**	0.05	0.01
SC MAD	**21**	19	111	31	0.50	**2.53**	**3.18**	1.00
GM MAD	34	19	180	47	1.08	**4.52**	11.72	4.52
Welsch MAD	23	17	109	36	**2.00**	**2.00**	**3.18**	**1.59**
Tukey MAD	26	19	130	34	**2.53**	**5.04**	**6.35**	**3.18**
Student	40	32	178	60	0.10	0.13	1.37	0.16
Max. Distance	45	38	112	58	0.30	**0.40**	0.60	0.40
Trim	39	24	284	63	0.63	**0.68**	0.89	0.68
Var. Trimmed	28	**9**	**58**	**27**	**1.91**	**2.35**	**2.35**	**1.91**

Legend: H = Hauptgebäude, GS = Gazebo Summer, WS = Wood Summer

VI. Conclusion

In this paper, we performed exhaustive robustness experiments on a wide range of outlier filters, in the context of ICP. After analysis, we concluded that all robust solutions have similar performances, with a number of particularities worth noting. For instance, L_1 exhibits good overall performance, despite having no parameter. Also, using MAD as auto-scale improves accuracy, but does not relieve from parameter tuning. When appropriately tuned, *Var. Trim.* has the best accuracy, with a translation error under $27\,\mathrm{mm}$, while the best M-estimator is *Cauchy MAD* with roughly the same error. Moreover, we demonstrated the necessity of a well-tuned outlier filter for robust registration. In particular, *Welsch*, *Tukey*, *GM*, and *SC* should be employed carefully, as they have the potential to produce worse estimate than L_2 when mis-tuned. This fact should be kept in mind when assessing the risks associated with parameter selection.

Encouraged by the result of L_1, further investigation will be made to adapt registration solutions related to L_p norms [13] into the standard ICP pipeline. Furthermore, the link between RQE, a Gaussian kernel and *Welsch* opens the door to a family of kernels to be studied for outlier filters along with an integration of the kernel correlation solution [25] and EM-ICP [7] into a generic version of ICP.

Acknowledgment

This work was partially financed by the Fonds de Recherche du Québec - Nature et technologies (FRQNT).

References

[1] P. J. Besl, "Geometric Modeling and Computer Vision," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 936–958, 1988.

[2] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, no. 3, pp. 2724–2729, 1992.

[3] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, pp. 145–152, 2001.

[4] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets: Open-source library and experimental protocol," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[5] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[6] S. Nobili, R. Scona, M. Caravagna, and M. Fallon, "Overlap-based ICP tuning for robust localization of a humanoid robot," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2017, pp. 4721–4728.

[7] S. Granger and P. Xavier, "Multi-scale EM-ICP : A Fast and Robust Approach for Surface Registration," *Proc. of ECCV*, pp. 418–432, 2002.

[8] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1145–1153, 2003.

[9] F. Pomerleau, F. Colas, F. Ferland, and F. Michaud, "Relative Motion Threshold for Rejection in ICP Registration," *Field and Service Robotics*, 2015.

[10] M. Bosse, G. Agamennoni, and I. Gilitschenski, "Robust Estimation and Applications in Robotics," *Foundations and Trends in Robotics*, vol. 4, no. 4, pp. 225–269, 2016.

[11] F. Pomerleau, 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.

[12] J. M. Phillips, R. Liu, and C. Tomasi, "Outlier robust ICP for minimizing fractional RMSD," in *3DIM 2007 - Proceedings 6th International Conference on 3-D Digital Imaging and Modeling*, 2007, pp. 427–434.

[13] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Eurographics Symposium on Geometry Processing*, vol. 32, no. 5, pp. 113–123, 2013.

[14] M. Sheehan, A. Harrison, and P. Newman, "Continuous vehicle localisation using sparse 3D sensing, kernelised rényi distance and fast Gauss transforms," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 398–405.

[15] R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics - Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[16] K. MacTavish and T. D. Barfoot, "At all Costs: A Comparison of Robust Cost Functions for Camera Correspondence Outliers," in *Proceedings -2015 12th Conference on Computer and Robot Vision, CRV 2015*, 2015, pp. 62–69.

[17] P. Bergström and O. Edlund, "Robust registration of point sets using iteratively reweighted least squares," *Computational Optimization and Applications*, vol. 58, no. 3, pp. 543–561, 2014.

[18] G. Agamennoni, S. Fontana, R. Y. Siegwart, and D. G. Sorrenti, "Point Clouds Registration with Probabilistic Data Association," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4092–4098, 2016.

[19] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Proc. of Robotics: Science and Systems*, vol. 2, p. 4, 2009.

[20] M. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based SLAM," *International Journal of Robotics Research*, vol. 27, no. 6, pp. 667–691, 2008.

[21] R. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim, "Pose Estimation from Corresponding Point Data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 19, no. 6, pp. 1426–1446, 1989.

[22] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, pp. 299–309, 2005.

[23] S. Du, J. Zhu, N. Zheng, Y. Liu, and C. Li, "Robust iterative closest point algorithm for registration of point sets with outliers," *Optical Engineering*, vol. 50, no. 8, p. 087 001, 2011.

[24] M. Sheehan, A. Harrison, and P. Newman, "Self-calibration for a 3D laser," *International Journal of Robotics Research*, vol. 31, no. 5, pp. 675–687, 2012.

[25] Y. Tsin and T. Kanade, "A Correlation-Based Approach to Robust Point Set Registration," in *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision*, vol. 0, 2004, pp. 558–569.

[26] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, "Challenging data sets for point cloud registration algorithms," *The International Journal of Robotics Research*, vol. 31, no. 14, pp. 1705–1711, 2012.