6DOF Point Cloud Alignment using Geometric Algebra-based Adaptive Filtering

Anas Al-Nuaimi
Technische Universität München
anas.alnuaimi@tum.de

Wilder B. Lopes*
Universidade de São Paulo
wilder@usp.br

Eckehard Steinbach
Technische Universität München
eckehard.steinbach@tum.de

Cassio G. Lopes
Universidade de São Paulo
cassio@lps.usp.br

Abstract

In this paper we show that a Geometric Algebra-based least-mean-squares adaptive filter (GA-LMS) can be used to recover the 6-degree-of-freedom alignment of two point clouds related by a set of point correspondences. We present a series of techniques that endow the GA-LMS with outlier (false correspondence) resilience to outperform standard least squares (LS) methods that are based on Singular Value Decomposition (SVD). We furthermore show how to derive and compute the step size of the GA-LMS.

1. Introduction

3D-Point cloud alignment is the problem of recovering the 6-degree-of-freedom (6DOF) transformation that aligns two overlapping point clouds (PCDs) in a common frame.

One principal approach to point cloud alignment requires determining point correspondences between the points of the PCDs to be aligned. Once correspondences have been computed, a least-squares (LS) estimator can be used to estimate the transformation that minimizes the Euclidean distance of the correspondence pairs after alignment [8].

In practical systems, the correspondences are usually automatically determined using for instance local-shape descriptor matching [28, 1]. Many correspondences can be false in this case and a standard LS estimator may not lead to satisfactory results as it treats all correspondences equally.

In our previous work [18] we devised a completely new approach for estimating the relative rotation between PCDs related by point correspondences. It is based on a least-mean-squares (LMS) adaptive filter (AF), derived in light of Geometric Algebra (GA), a mathematical tool that allows for representing geometric transformations in a very general fashion [12, 11, 14]. Supported by results of Geometric Calculus [12], the alignment problem is posed as a *rotor* estimation problem [7, 31, 27] which is addressed from an AF-theory perspective [25, 6]. This novel formulation allows for a tractable derivation while emphasizing the geometrical intuition behind rotors, a notion that is less intuitive when using matrix algebra to represent geometric transformations. Experiments have determined the GA-LMS alignment accuracy to be comparable to that of standard SVD-based estimators [18].

Motivated by the results in [18], our goal in this paper is to present an alternative based on adaptive learning for orientation estimation. The contributions of this paper are:

- The use of an adaptive filter (based on GA) as the error minimizer of the 6DOF alignment algorithm;
- A rule to set the GA-LMS step-size value as a function of the PCDs dimensions and the point correspondences in Section 3;
- The adaptive nature of the GA-LMS is exploited to make it more outlier-resilient than standard SVD-based LS estimator (Section 4).

The notation throughout the manuscript is as follows. Matrices are represented as boldface capital letters. Scalars and vectors have both lower-case normal-type font. Multivectors (the general elements of a GA) are represented by normal-type capital letters, unless otherwise noted.

2. Point cloud alignment problem

Take two PCDs in the $\mathbb{R}^3$, Y (Target) and X (Source), related via a 1-to-1 correspondence, in which at least a part of X is a translated and rotated version of Y. Each PCD has K correspondence points, $\{y_n\} \in Y$ and $\{x_n\} \in X$, $n = 1...K$, obtained via a local-shape feature-based matching system. In the registration process, the goal is to find a rigid transformation to align Y and X. In other words, what is the rotation and the translation that should be applied to each point in X to map it onto Y?

*W. B. Lopes was supported by CAPES Foundation, Ministry of Education of Brazil, under Grant BEX 14601/13-3. This work was conducted during his research stay at the Technische Universität München.

2.1. Standard approach

This question can be posed as a least-squares problem [3, 33], in which one should minimize the following cost function

$$\mathcal{F}(\mathbf{R},t)=\frac{1}{K}\sum_{n=1}^{K}\left\|y_n^{'}-\mathbf{R}x_n^{'}-t\right\|_2^2, \tag{1}$$

where $y_n^{'}$ and $x_n^{'}$ are 3×1 vectors representing (presumably) *corresponding* points in the target and source PCDs, respectively; $\mathbf{R}$ is a 3×3 rotation matrix; and t is a 3×1 translation vector.

Defining the centroids of the target and the source PCDs as $\bar{y}$ and $\bar{x}$ respectively, the coordinate of each point in a PCD with respect to its centroid is given by

$$\begin{aligned} y_n &= y_n^{'}-\bar{y} \\ x_n &= x_n^{'}-\bar{x}. \end{aligned} \tag{2}$$

Substituting (2) into (1) and setting $t=\bar{y}-\mathbf{R}\bar{x}$ (translating the source PCD's centroid to match that of the target PCD), the cost function (1) can be rewritten as

$$\mathcal{F}(\mathbf{R})=\frac{1}{K}\sum_{n=1}^{K}\|y_n-\mathbf{R}x_n\|_2^2. \tag{3}$$

Finding the matrix $\mathbf{R}$ that minimizes (3) is known as the *orthogonal procrustes problem*. Methods available in the literature rely on a standard least-squares estimator which is typically based on the SVD algorithm [3, 33, 9]. After estimating $\mathbf{R}$, the translation is recovered by $t=\bar{y}-\mathbf{R}\bar{x}$.

2.2. Geometric Algebra approach

Geometric (Clifford) Algebra was developed as a mathematical language to unify all the different algebraic systems trying to express geometric relations. Some GA concepts are introduced throughout the paper, when necessary. For an in-depth discussion, see [12, 11, 14].

Given a vector space $\mathcal{A}$, the Geometric Algebra $\mathcal{G}(\mathcal{A})$ is comprised by the so-called *multivectors*, generated by multiplying the vectors in $\mathcal{A}$ via the *geometric product* [14], together with the multiplicative identity (scalar) 1. For example, the geometric multiplication of two vectors $a,b\in\mathcal{A}$ is defined as $ab=a\cdot b+a\wedge b$, in terms of the *inner* $(a\cdot b)$ and *outer* $(a\wedge b)$ products [12]. Thus, whenever dealing with multivectors, the use of the geometric product is implicit.

Our previous work [18] restates (3) from the perspective of Geometric Algebra. Particularly, the matrix $\mathbf{R}$ is substituted by the rotor r and its *reversed* version $\widetilde{r}$ (the analogous to conjugation in GA) in order to represent rotations. Thus, the GA least-squares cost function is [18]

$$J(r)=\frac{1}{K}\sum_{n=1}^{K}|y_n-rx_n\widetilde{r}|^2, \tag{4}$$

in which the magnitude $|\cdot|$ in GA generalizes the norm operator $\|.\|$ of linear algebra [14].

In [18] we applied Geometric Calculus [12, 13] to devise an LMS AF to estimate the best rotor that recursively minimizes (4). The simulations in [18] show that the GA-LMS is able to perform the task, providing similar results to a standard SVD-based method [9].

In the sequel, motivated by the good results in [18], the estimation capabilities of the GA-LMS are improved by implementing a number of techniques which explicitly exploit the adaptive nature of GA-LMS wherein each correspondence is individually considered while searching for the optimum rotor (Section 4). Additionally, we derive a rule to select the GA-LMS step size (Section 3.1). The experiments in Sections 4 and 5 show that the proposed GA-LMS based registration provides lower errors than the SVD-based method used for comparison.

3. GA-LMS adaptive filter revisited

In [18] the GA-LMS is derived by calculating the gradient $\nabla J(r)$ (via Geometric Calculus [12]) and substituting it into the *steepest-descent rule* [25], which will recursively provide a new estimate for the rotor r (the only lower-case multivector in this paper) at each iteration,

$$r_i=r_{i-1}-\mu\widetilde{\nabla}J(r_{i-1}),\text{ where} \tag{5}$$

$$\nabla J(r_{i-1})=4\widetilde{r}_{i-1}\sum_{n=1}^{K}y_n\wedge(r_{i-1}x_n\widetilde{r}_{i-1}), \tag{6}$$

and $\widetilde{\nabla}J(r_{i-1})$ is the reversed gradient. Note that $\nabla J(r_{i-1})$ is a function of all the correspondence pairs $\{y_n,x_n\}$, $n=1,\cdots,K$, i.e., it needs all the available correspondences, at each i, to update the estimate for r.

One of the contributions of [18] is to approximate the gradient by its *current value* $\nabla J(r_{i-1})\approx 4\widetilde{r}_{i-1}\left[y_i\wedge(r_{i-1}x_i\widetilde{r}_{i-1})\right]$, which reduces the computational complexity of the algorithm since now it needs only the correspondence pair at the current iteration $\{y_i,x_i\}$. Adopting that in (5) results in the GA-LMS update rule,

$$\boxed{r_i=r_{i-1}+\mu\left[y_i\wedge(r_{i-1}x_i\widetilde{r}_{i-1})\right]r_{i-1}}. \tag{7}$$

As shown in [18], Equation (7) is able to converge using only part of the available correspondences, delivering an estimate as good as the SVD method used for comparison.

3.1. Selecting the step-size value

Selecting the step size within an appropriate range of values prevents the AF from diverging and enables us to take full advantage of the GA-LMS capabilities. This way, we devise a simple formula, refined by empirical data, for the step-size bounds as a function of the PCDs dimensions and the point correspondences.

Embedding $1/K$ into $J(r)$ in (4), we plug (5) into (4),

$$\begin{aligned} J(r_i) &= \sum_{n=1}^{K}|y_n|^2+|x_n|^2-2\langle y_n r_i x_n\widetilde{r_i}\rangle \\ &= c-2\sum_{n=1}^{K}\left\langle y_n\Big(r_{i-1}-\mu\widetilde{\nabla}J\Big)x_n\Big(r_{i-1}-\mu\widetilde{\nabla}J\Big)\widetilde{\ }\right\rangle, \end{aligned} \tag{8}$$

where $c = \sum_{n=1}^{K} |y_n|^2 + |x_n|^2$. Notice, we omit the index $i-1$ of $\widetilde{\nabla} J$ for the sake of readability and we use $\langle ab \rangle \triangleq a \cdot b$, $a, b \in \mathbb{R}^3$ (*0-grade operator*) [18].

The second term of the right-hand side of (8) can be expanded as

$$2 \sum_{n=1}^{K} \begin{array}{l} \langle y_n r_{i-1} x_n \widetilde{r}_{i-1} - \mu y_n r_{i-1} x_n \nabla J \\ -\mu y_n \widetilde{\nabla} J x_n \widetilde{r}_{i-1} + \mu^2 y_n \widetilde{\nabla} J x_n \nabla J \rangle. \end{array} \tag{9}$$

Thus we have a recursive equation for the cost function value

$$\begin{array}{rl} J(r_i) = J(r_{i-1}) & +2\mu \sum_{n=1}^{K} \langle y_n (r_{i-1} x_n \nabla J + \widetilde{\nabla} J x_n \widetilde{r}_{i-1}) \rangle \\ & -\mu^2 \sum_{n=1}^{K} \langle y_n \widetilde{\nabla} J x_n \nabla J \rangle, \end{array} \tag{10}$$

in which $J(r_{i-1}) = c - \sum_{n=1}^{K} 2 \langle y_n r_{i-1} x_n \widetilde{r}_{i-1} \rangle$.

In order to make $J(r_i) < J(r_{i-1})$ at each iteration [25], one should select the step size within the interval bounded by the roots of the following second-order equation in μ,

$$2\mu \sum_{n=1}^{K} \langle y_n (r_{i-1} x_n \nabla J + \widetilde{\nabla} J x_n \widetilde{r}_{i-1}) \rangle - \mu^2 \sum_{n=1}^{K} \langle y_n \widetilde{\nabla} J x_n \nabla J \rangle = 0, \tag{11}$$

which is comprised by the second and third terms on the right-hand side of (10). Note that one of the roots of (11) is $\mu = 0$. The second root is calculated via

$$\mu(i) = 4 \sum_{n=1}^{K} \left\langle y_n r_{i-1} x_n \nabla J \right\rangle \Big/ \sum_{n=1}^{K} \left\langle y_n \widetilde{\nabla} J x_n \nabla J \right\rangle, \tag{12}$$

where the use of the *cyclic reordering property* [18] and the relation $\langle A \rangle = \langle \widetilde{A} \rangle$ (which holds for any multivector A [18]) allows for rewriting the first term of (11) as $2 \sum_{n=1}^{K} \left\langle y_n \left[r_{i-1} x_n \nabla J + \widetilde{\nabla} J x_n \widetilde{r}_{i-1} \right] \right\rangle = 4 \sum_{n=1}^{K} \langle y_n r_{i-1} x_n \nabla J \rangle$. Note that μ is iteration dependent (it is a function of the rotor r_{i-1}).

To represent μ solely as a function of the Target (Y) and Source (X) PCDs dimensions and their point correspondences, one can simplify (12) by removing the dependence on r_{i-1}. The underlying idea is to consider r_{i-1} as a "dummy" rotation, i.e., make $r_{i-1} = 1$ in (12). As a consequence, (6) becomes $\nabla J = 4 \sum_{n=1}^{K} y_n \wedge x_n$. The intuition is that it will provide the necessary μ value for the next iteration given that the PCDs were not rotated in the previous iteration. Applying this to (12) allows us to express μ as

$$\mu(Y, X) = \sum_{n=1}^{K} \langle y_n x_n Q \rangle \Big/ \sum_{n=1}^{K} \langle y_n \widetilde{Q} x_n Q \rangle, \tag{13}$$

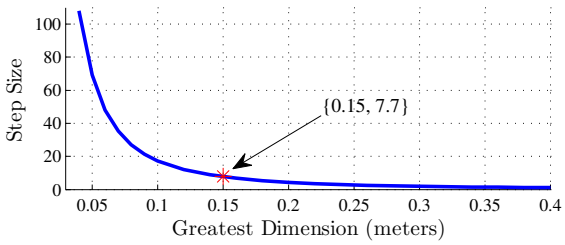


Figure 1: Simple rule for selecting μ.

in which $Q = \sum_{n=1}^{K} y_n \wedge x_n$. As an example, using (13) to compute the step-size value for the Stanford Bunny dataset (discussed in details in Section 5) results in $\mu = 0.51$.

As aforementioned, (13) is obtained to make $J(r_i) < J(r_{i-1})$ at each iteration. However, this approach is very conservative, prioritizing monotonic convergence at the expense of speed (the step-size upper limit depicted by (13) is very small, making the filter adaptation very slow). Indeed, the simulations in [18] show that the GA-LMS is able to align the Stanford Bunny PCDs [30], without diverging, using $\mu = 8$, approximately fifteen times the conservative value $\mu = 0.51$. Using this fact, we relax the constraint $J(r_i) < J(r_{i-1}) \ \ \forall i$ and multiply (13) by $\rho \geq 1$ to obtain a simple rule for selecting μ,

$$\boxed{\mu(Y, X) = \rho \sum_{n=1}^{K} \langle y_n x_n Q \rangle \Big/ \sum_{n=1}^{K} \langle y_n \widetilde{Q} x_n Q \rangle}. \tag{14}$$

In this paper we use $\rho = 15$. Figure 1 shows the step size values calculated via (14) as a function of the PCDs greatest dimension. Note the inverse proportion between PCD size and μ, i.e., for big PCDs one should use small step sizes, and vice versa. This is the case because the input data $\{y, x\}$ provided by big PCDs has higher power (greater magnitudes $|y|, |x|$) than the one from small PCDs, resulting in higher values for the gradient term (6). Thus, to avoid divergence, we must compensate for it by making μ smaller.

Since the selection of μ is not required to be exact, one should think of Figure 1 as a chart of possible values for μ. For example, the greatest dimension in the PCDs of the Stanford Bunny dataset is $15cm$, which in Figure 1 corresponds to $\mu = 7.7$. One may select a slightly higher value to increase speed, however, this also increases the chances of divergence. Choosing a lower value is also possible, with convergence speed reduction as a side effect. Therefore, Figure 1 depicts the recommended *superior limit* for μ.

4. Implementation and robustification

4.1. Filter implementation

The filter was implemented in C++ as part of the retrieval system explained in [1] that computes point cor-

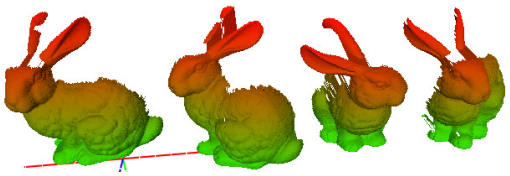


Figure 2: Bunny dataset point clouds. From left to right: Bunny000, Bunny045, Bunny270 and Bunny315.

respondences using local-shape features of 3D keypoints. The GA-LMS update rule (7) was implemented using the *Geometric Algebra ALgorithms Expression Templates* (Gaalet) [26]. The filter update code is as simple as shown in Listing 1. Notice the rotor normalization.

Listing 1: Implementation of (7) using Gaalet. (∧) performs outer product and (∼) computes the reverse of a multivector.

```
//Update rotor
rotAfter = eval(rotBefore+mu*(y^(rotBefore*
    x*(~rotBefore)))*rotBefore );
//Normalizing rotor (important!)
rotAfter = normalize_mv(rotAfter);
```

4.2. Filter robustification

Registration systems that establish correspondence pairs are not perfect. Many correspondences can be invalid (*outliers*). Our experiments have shown that the GA-LMS in its basic form is not outlier-resilient. Its accuracy matches that of standard SVD-based methods and degrades proportionally to an increasing outlier rate. To robustify the filter and achieve outlier resilience we show here a series of improvements that increase the accuracy significantly.

Adopting established techniques of the adaptive filtering community, as well as simple geometric heuristics to improve the GA-LMS performance, we achieve the desired goal of outperforming standard SVD-based registration strategies, while not significantly compromising speed. We exploit the fact that the GA-LMS is an adaptive filter that processes individual sample points (correspondences in our case) individually and iteratively. Hence, we can choose *how much* (Section 4.2.3) and *how often* (Section 4.2.2) data samples are allowed to contribute or *whether* to consider them at all in the first place (Section 4.2.1).

To explain the proposed robustification techniques we use a source-target pair of the Stanford Bunny dataset [30] which are commonly used in alignment experiments [22, 17, 10]. Specifically, we use Bunny315 and Bunny000, as source and target PCDs, respectively. These two PCDs are shown among others in Figure 2. The Bunny dataset is introduced in greater detail in Section 5. The figure of merit used to evaluate the AF performance is the Mean-Square Error (MSE). The *true MSE* uses *only inliers* to calculate the estimation error. In practice, the AF can not compute

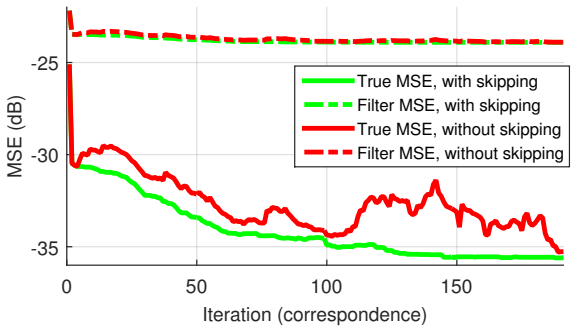


Figure 3: Skipping iterations with increasing filter MSE (MSE computed using all correspondences including inliers and outliers) vs. no skipping. When skipping, the true MSE (solid lines) is clearly lower than when not skipping, proving that skipping reduces the contributions from outliers leading to better pose estimation. The strong reduction in MSE after the first iteration is due to the subtraction of the centroids (2).

the *true MSE* since it does not know what correspondence subset constitutes only inliers. What the AF evaluates is what we call the *filter MSE*, which is calculated using all correspondences (inliers and outliers).

4.2.1 Iteration skipping

Having observed that outliers typically lead to an instantaneous increase in the MSE, we propose skipping the filter update that would lead to such an increase, thus discarding the otherwise harmful contribution of the correspondence. The skipping is done based on the filter MSE value. Notice, this is something we are strictly capable of doing due to the sequential nature of the filter.

Our experiments have shown that skipping iterations in which the filter MSE increases improves the true MSE. As depicted in Figure 3, the true MSE is always lower when skipping and mostly monotonically decreasing. Also, the frequent degradation in the true MSE when not skipping (solid red line in Figure 3) no longer appears when skipping (solid green line in Figure 3), suggesting that the contribution of the outliers is indeed being discarded.

4.2.2 Sample refeeding

From information theory, we know that data samples (correspondences in our case) that have already been processed can provide new information if processed again at a later stage [25]. The intuitive explanation is that, especially when the step size is relatively small, a sample's residual error, given the current filter state, can lead to another filter update in the opposite direction of the reversed gradient, according to (7). Hence, it is common in adaptive filtering literature to refeed (reuse) the same set of data samples to obtain a better final outcome [19, 4, 20, 5].

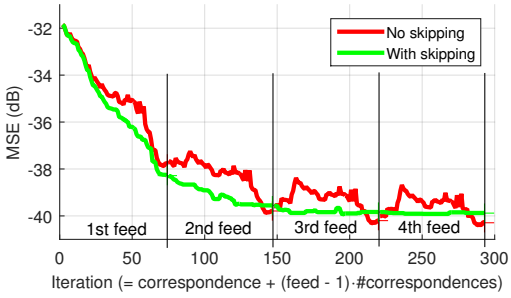


Figure 4: Refeeding the same set of correspondences multiple times can extract further information from already processed samples. In this case, the correspondence pairs are fed 4 times into the GA-LMS. The (true) MSE at the end of each cycle is better than that of the previous cycle. The gain in the 3rd & 4th cycles is marginal since the orientation has been already recovered accurately in the 2nd stage. Here we limited the number of correspondences to 33% of those in Figure 3 as this technique is particularly useful when the number of correspondences is limited.

This technique can lead to a noticeable improvement in the MSE in the case where the number of correspondences is not enough for the filter to converge, or the step size is not large enough to achieve convergence – μ must be designed according to Section 3.1 to avoid filter divergence.

Figure 4 shows the MSE progression for the Bunny pair. However, in this experiment only 33% of the correspondences used in Section 4.2.1 are used. Clearly, the multiple feeding strategy benefits the alignment. Notice, additional iterations from the same data set do not statistically degrade the solution and hence it is always safe to apply this technique. Given that the filter is light-weight (as seen in the results in Section 5.2) the extra computational cost can be accommodated. Also notice that more exquisite refeeding techniques can be implemented that include randomization and prioritazation of correspondences.

4.2.3 Geometric correspondence weighting

The idea behind this technique is to compute an individual *weight* for each correspondence, which is used to weight the AF step size exploiting the capability to treat data points selectively. This weight reflects the likelihood of the correspondence being an inlier. It is explained in Figure 5. Once the weight of each correspondence $\alpha_i (0 < \alpha_i \leq 1)$ is computed, (7) is slightly changed to

$$r_i = r_{i-1} + \alpha_i \mu \left[y_i \wedge (r_{i-1} x_i \widetilde{r}_{i-1}) \right] r_{i-1}. \qquad (15)$$

The results in Table 1 suggest that geometric correspondence weighting should not be enabled as the only robustification mechanism. The fact that each correspondence is weighted by a factor smaller than one means that the filter step sizes are generally decreased. Hence, its alignment improvement effect is only observed when combined with refeeding (see Section 4.2.2).

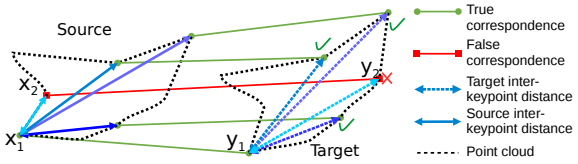


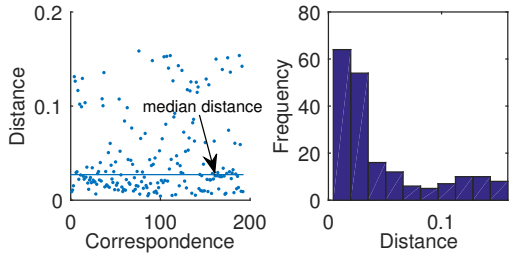
Figure 5: The geometric weight of correspondence $\{y_i, x_i\}$ is computed by first computing the Euclidean distance of x_i to all other keypoints in the source cloud (solid blue lines), $d^s_{i,j} \, \forall j \in \{1, .., K\} \backslash i$, and computing the corresponding distances $d^t_{i,j} \, \forall j \in \{1, .., K\} \setminus i$ on the target side (dashed blue lines). The number of keypoints which satisfy $|d^s_{i,j} - d^t_{i,j}| < \epsilon$ (green ticks) is a vote for correspondence i being an inlier. In this example, $\{y_1, x_1\}$ gets $v_i = 3$ votes. The weight is a function of the votes: $\alpha_i = v_i / \max_j v_j$. Hence, $0 < \alpha_i \leq 1$.

4.2.4 Statistical correspondence filtering

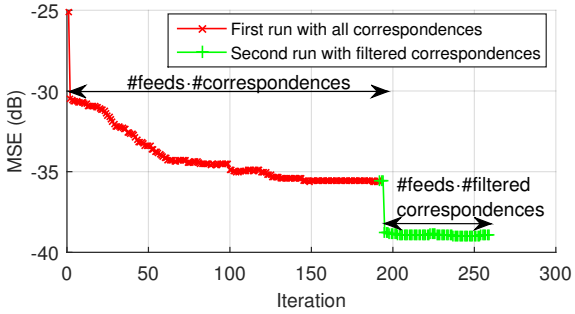
As explained in Section 4.2.2, after a sufficient number of correspondence refeeds, no further improvement to the MSE can be obtained. Given the used correspondence set, a lower limit on the MSE will be reached and further improvements might be achievable if a new less outlier-corrupted correspondence set is available. Assuming that once the filter has converged the recovered orientation is accurate up to a few degrees the remaining error in the pose is composed mainly of a 3D translation error. Hence, when transforming the source keypoints using the obtained transformation, and subsequently computing the Euclidean distance to their respective target keypoints, the set of inlier correspondences should be identifiable. It is specifically the set of correspondence pairs whose Euclidean distances are roughly similar. The other subset should be composed of outliers.

Indeed, Figure 6a shows a concentration of correspondences whose spatial distances, after applying the obtained GA-LMS transformation, are similar. We compute the median distance d and retain only those correspondences whose spatial distances are within $d \pm \lambda \sigma_d$, where σ_d is the standard deviation in the spatial distances and we set $\lambda = 0.25$ in this paper. The GA-LMS is then fed with those filtered correspondences, and is initialized with the most recently obtained rotor. Figure 6b shows that a significant MSE improvement happens as soon as the second GA-LMS run commences, as the computed centroids coincide more accurately.

The results of the collective contributions of the robustification techniques on the bunny set (introduced at the beginning of Section 4) are shown in Table 1. It can be seen that while using some of them individually might lead to a deterioration of the alignment, collectively they reinforce one another as each one of them possesses different properties leading to substantially improved alignment.



(a) Distances between target and rotated source keypoints after the first GA-LMS run and before correspondence filtering.



(b) The true MSE curve.

Figure 6: The GA-LMS is first run using all correspondences (red). For the obtained rotor, the residual error is assumed to be mainly a translatory component causing a constant shift in inlier source-target keypoint pairs. The inliers are determined statistically (correspondences with a distance close to the median distance). The filtered correspondences comprising mainly inliers are fed again into the GA-LMS for a 2nd run. The second run converges to a better MSE (green).

Table 1: The contributions of the filter robustification techniques explained in Section 4.2. Comparing B and C to A it can be seen that skipping and statistical correspondence filtering lead to improvements individually. When combined, they lead to even better alignment (D). Comparing E to A shows that geometric correspondence weighting alone deteriorates performance, suggesting it should not be used. However, this is expected since weighting each step by $\alpha_i < 1$ means the step size is reduced and the filter might not converge. Indeed, comparing E to F shows that geometric correspondence weighting is only meaningful once combined with refeeding as it compensates for the reduced step size while benefiting from the selective weighting. Columns G, H and J show that the combination of these mechanisms with increasing refeeding values reduces the angular alignment error. Column I, again shows that geometric coresondence weighting improves results when combined with the other mechanisms. The error values are averages of 10 runs, each with a different set of correspondences.

Test	A	B	C	D	E	F	G	H	I	J
Nr. Feeds	1	1	1	1	1	2	1	2	2	4
Skipping	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓
Stat. filt.	✗	✗	✓	✓	✗	✗	✓	✓	✓	✓
Geom. w.	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓
Angle error	16.7	14.0	10.8	8.4	24.0	14.9	15.1	6.2	7.0	5.4

5. Evaluation

For our experiments we distinguish between two variants of GA-LMS: GA-LMS+ is the standard GA-LMS with 4x refeed, statistical correspondence filtering and skipping; GA-LMS++ is GA-LMS+ with geometric correspondence weighting. To compute correspondences we use our local shape feature matching system, explained in [1], with Harris3D keypoints and SHOT descriptors [29].

5.1. Computing alignment error

The computed alignment is checked for correctness. To that end we first compute the error transformation

$$\mathbf{T}_e = \begin{pmatrix} \mathbf{R}_e & t_e \\ 0\,0\,0 & 1 \end{pmatrix} = \mathbf{T}^{-1}\hat{\mathbf{T}}. \tag{16}$$

where $\mathbf{T}$ and $\hat{\mathbf{T}}$ are the 3D homogenous transforms of the ground truth and the estimate, respectively. Hence, ideally $\mathbf{T}_e$ should be identity. Any departure from that indicates a residual alignment error. Using the *log map*, the error rotation matrix $\mathbf{R}_e$ is broken down into a rotation axis and an error rotation angle ϕ_e.

The translation error d_e is computed by determining the query's centroid, $\mathbf{c}$, translation when applying the ground truth transform compared to when applying the estimated transform. So, it is computed as follows:

$$d_e = ||(\mathbf{R} - \hat{\mathbf{R}})\mathbf{c} + \mathbf{t} - \hat{\mathbf{t}}||_2 \tag{17}$$

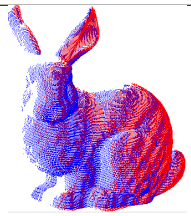
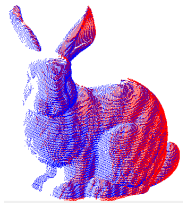
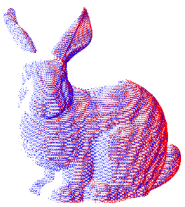
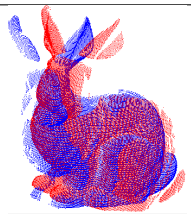
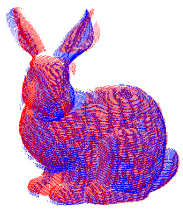
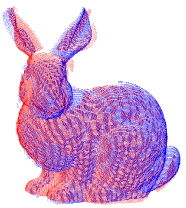
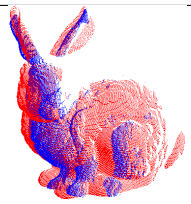
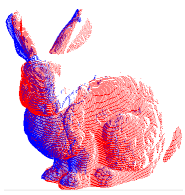
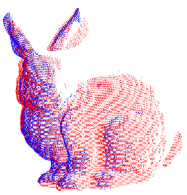
5.2. Bunny dataset

For our experiments we use the Stanford Bunny dataset [30]. It is composed of 10 scans of a rabbit model obtained from different view points. Each point cloud has an average nearest (NN) distance of around 0.5 mm. We pick four specific scans: Bunny000, Bunny045, Bunny270 and Bunny315, where the numeric postfix identifies the angle in the ground plane from which the bunny was scanned. The four scans are shown in Figure 2.

After establishing correspondences using feature matching each of the two previously introduced alignment estimators (SVD and GA-LMS) are run to retrieve the alignment.

The results in Table 2 clearly show that the GA-LMS+ and GA-LMS++ are more accurate on average, in terms of both orientation estimation and translation estimation, than the SVD estimator. GA-LMS++, which employs the geometric correspondence weighting, is better than GA-LMS+, as expected. However, the gain is small. This is mainly due to the dispersion of the outliers throughout the bunny shape (Section 5.3 shows a case with substantial gain). The GA-LMS+ is slower than the SVD-based method provided by the Point Cloud Library (PCL) [24]. Nonetheless it is typically done in a few milliseconds making it suitable for real-time applications. Moreover, we

Table 2: Experiments using the Stanford Bunny dataset. Three pairs used from the dataset introduced in Section 5 (1: bunny045 vs bunny000; 2: bunny315 vs bunny000; 3: bunny315 vs bunny270). The alignment angle and translation errors as well as the required time are shown for each case. These values represent the average values over 10 different alignment attempts with different correspondence sets. The true correspondence rate (TCR) varies from 20% to 80%. One representative visualization is shown for each case. Clearly, both GA-LMS variants are better than the SVD-based approach, in terms of angle and translation accuracy while remaining suitable for real-time applications.

Set	SVD (LS)	GA-LMS+	GA-LMS++
1	 2.4° 1.8 mm 0.04 ms	 0.8° 0.3 mm 23ms	 0.8° 0.3 mm 23ms
2	 9.9° 10.7 mm 0.025 ms	 6.2° 4.0 mm 8.0ms	 5.5° 3.3 mm 8.6ms
3	 12.9° 12.0 mm 0.02 ms	 9.5° 2.6 mm 6.4ms	 6.4° 1.7 mm 7.0ms

have determined that $> 95\%$ of the time is spent on the MSE computation which can be spared if no skipping is performed.

Using the first bunny pair shown in Table 2, the resilience against outliers is depicted in Figure 7. The outlier correspondences are randomly added to true correspondences to generate tougher alignment scenarios. The results clearly show the GA-LMS superiority as compared to SVD. The same holds true even if half as many correspondences is used. Similar experiments on the two other bunny pairs (not shown) confirmed these conclusions.

5.3. Alignment with few correspondences

The experiment shown in Figure 8 demonstrates the GA-LMS+ capability to deal with cases that have a small number of correspondences, out of which many are false and distributed in an unfavorable geometric fashion. The GA-

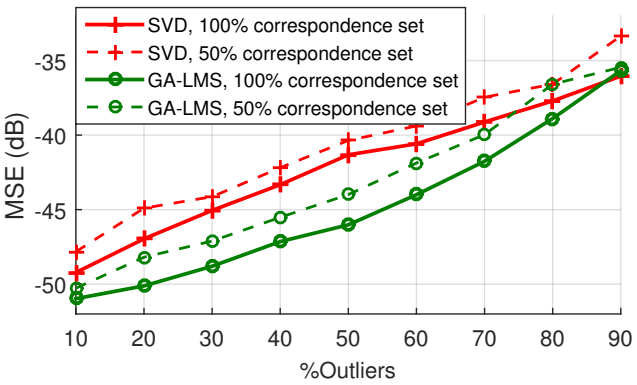
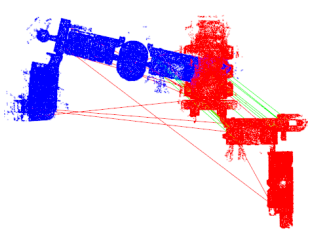
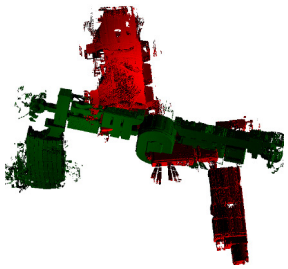


Figure 7: Estimation resilience against outliers. A set of randomly chosen outliers is added to confuse the estimators. The curves are averaged over 50 realizations to account for different constellations of outliers. Conducting the experiment using a subset involving only half of the correspondences shows a similar superiority of the GA-LMS w.r.t. SVD. The GA-LMS uses a 4x refeed (Section 4.2.2) with geometric correspondence weighting (Section 4.2.3).



(a) Correspondences between the indoor scans. Red lines show outliers (false correspondences) as opposed to inliers shown in green.



(b) SVD-based alignment fails due to the high false-correspondence rate and their geometric distribution.

Figure 8: Aligning two large-scale indoor point clouds obtained inside a museum obtained using the scanning trolley of [16]. The number of correspondences is reduced to 25, out of which 11 are wrong (outliers).

LMS+ does not benefit from the multiple feeds before the statistical correspondence filtering is performed. Once this is done, the multiple feeds are critical to overcome the problem with the limited number of correspondences (see MSE progression after the statistical correspondence filtering in Figure 9). The statistical correspondence filtering is seen to significantly reduce the number and ratio of outliers. In effect, the GA-LMS+ retrieves the alignment. The residual orientation error is around 7°. The GA-LMS++ shows even better performance. Due to the geometric constellation of the outliers, the geometric correspondence weighting leads to effective downweighting of the outliers contribution. The final outcome has an angle error of 0.79°and nearly optimal translatory alignment. The SVD-based approach, on the other hand, fails completely as seen in Figure 8b.

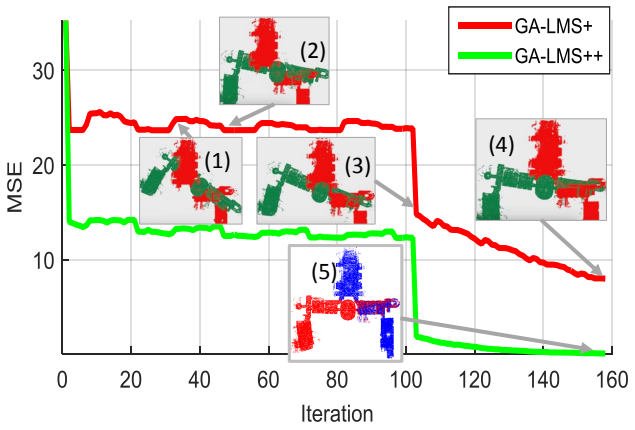


Figure 9: The true MSE of the GA-LMS+ initially rises (1). The MSE then reduces as more inliers are encountered (2). Nevertheless the orientation error remains high until the statistical correspondence filtering (see Section 4.2.4) is performed (3). This filtering increases the inlier rate and the AF keeps minimizing the MSE. The final alignment (4) is correct in terms of translation and has a relatively small angle error as opposed to the SVD-based approach, which fails completely (Figure 8b). Due to the geometric correspondence weighting, GA-LMS++ performs even better, providing nearly optimal translatory and rotational alignment (5).

6. Literature review

Point cloud alignment deals with the problem of finding the transformation between two overlapping point clouds which is important for many applications including combining individual scans into a more comprehensive 3D model (scan matching). In general, there are two different alignment problems: *fine alignment*, which requires a good pre-alignment, for which the Iterated Closest Point (ICP) is considered the standard solution [23, 33, 3]; *general alignment*, which does not make any requirements on the initial alignment. This paper is concerned with the latter problem.

According to Gelfand et al. [10] point cloud alignment methods can be mainly grouped into two categories. The first involves voting-based algorithms in which the 6DOF parameter space is quantized and exhaustive point-triplet alignment is used to vote in the parameter space. These methods are guaranteed to find the optimum but suffer from inefficiency due to their brute force nature. The second category involves computing point correspondences and using them to estimate the transformation that best maps the source points to their respective target points. This paper is concerned with the latter category. A novel approach that uses 4-point sets as correspondences [21], as opposed to 1-point correspondences, exists and is beyond the scope of this paper as are statistical approaches that don't establish hard correspondences such as [17, 22].

This paper does not address correspondence estimation but rather the problem of computing the alignment once such correspondences have been established which is occasionally called as the *absolute orientation problem*. The most straight-forward solution is the least squares (LS) minimization. Different formulations of the LS minimization exist. The main formulations use either 3×3 rotation matrices [2], quaternions [15] or dual quaternion [32] to represent the rotation. All three result in closed-form solutions to the minimization problem and have been found to perform similarly accurate in non-degenerate cases [8].

The solution to the formulation using rotation matrices involves the singular value decomposition (SVD) [2] which is fast to compute. The landmark researches of Zhang [33] and Besl and McKay [3] use this alignment approach.

The accuracy of the SVD estimator is degraded by outliers as its treats all points equally. At best individual correspondences can be weighted [8], however, all points are considered jointly and once.

In our previous work [18], we presented a wholly new solution to the alignment error minimization problem using an adaptive filter devised on top of a Geometric Algebraic formulation employing rotors (which are isomorphic to quaternions) to represent rotation. The formulation allows us to perform Geometric Calculus on the cost function to derive a Least-Mean-Square (LMS) adaptive filter, which we termed the Geometric Algebra LMS (GA-LMS). The GA-LMS distinguishes itself by acting on each correspondence individually in an iterative procedure. This allows to treat correspondences selectively, a property we exploit in this paper. It has the fundamental advantage of being deterministic and its complexity being linear with the number of correspondences. In terms of accuracy, the GA-LMS is similar to SVD (or generally LS) estimators.

In this paper we build upon the theory presented in our previous work [18] and show that the filter can handle full 6DOF estimation. We also introduce a series of improvements that make the GA-LMS clearly outperform standard LS estimators (Section 4) while remaining suitable for real-time applications. Moreover, we show how to properly set the step size (Section 3.1).

7. Conclusion

In this paper we extended our prior work on Geometric-Algebra-based LMS adaptive filter (GA-LMS) to estimate the 6DOF alignment between two point clouds, related by 1-to-1 correspondences, without prior alignment. We first show how to design the step size of the GA-LMS adaptive filter to enforce convergence. Subsequently, we exploit adaptive-filtering properties to evolve our previous approach by applying a series of techniques that equip the GA-LMS with outlier resilience. Ultimately, the improved GA-LMS outperforms the standard LS 6DOF estimator based on the SVD while maintaining a low computational footprint.

References

[1] A. Al-Nuaimi, M. Piccolrovazzi, S. Gedikli, E. Steinbach, and G. Schroth. Indoor location retrieval using shape matching of kinectfusion scans to large-scale indoor point clouds. In *2015 Eurographics Workshop on 3D Object Retrieval*, 3DOR, pages 31–38, Aire-la-Ville, Switzerland, Switzerland, 2015. Eurographics Association.

[2] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 9(5):698–700, Sep 1987.

[3] P. Besl and N. D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992.

[4] L. Chamon, W. B. Lopes, and C. G. Lopes. Combination of adaptive filters with coefficients feedback. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3785 –3788, march 2012.

[5] L. F. O. Chamon and C. G. Lopes. There's plenty of room at the bottom: Incremental combinations of sign-error lms filters. In *ICASSP*, pages 7248–7252, 2014.

[6] P. S. Diniz. *Adaptive Filtering: Algorithms and Practical Implementation*. Kluwer Academic Publishers, Norwell, MA, USA, 2 edition, 2002.

[7] L. Dorst, D. Fontijne, and S. Mann. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.

[8] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5-6):272–290, 1997.

[9] G. Forsythe and P. Henrici. The cyclic jacobi method for computing the principal values of a complex matrix. In *Trans. Amer. Math. Soc., Volume 94, Issue 1, Pages 1-23*, 1960.

[10] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *3rd Eurographics Symposium on Geometry Processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[11] D. Hestenes. *New Foundations for Classical Mechanics*. Fundamental Theories of Physics. Springer, 1999.

[12] D. Hestenes and G. Sobczyk. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Fundamental Theories of Physics. Springer, 1987.

[13] E. Hitzer. Multivector differential calculus. *Advances in Applied Clifford Algebras*, 12(2):135–182, 2002.

[14] E. Hitzer. Introduction to Clifford's Geometric Algebra. *Journal of the Society of Instrument and Control Engineers*, 51(4):338–350, 2012.

[15] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[16] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach. Tumindoor: An extensive image and point cloud dataset for visual indoor localization and mapping. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1773–1776, Sept 2012.

[17] B. Jian and B. C. Vemuri. Robust point set registration using gaussian mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1633–1645, 2011.

[18] W. B. Lopes, A. Al-Nuaimi, and C. G. Lopes. Geometric-algebra lms adaptive filter and its application to rotation estimation. In *IEEE Signal Processing Letters (Accepted for publication). Preprint on arxiv.org, arXiv:1601.06044*.

[19] W. B. Lopes and C. G. Lopes. Incremental-cooperative strategies in combination of adaptive filters. In *Int. Conf. in Acoust. Speech and Signal Process.*, pages 4132 –4135. IEEE, 2011.

[20] W. B. Lopes and C. G. Lopes. Incremental combination of rls and lms adaptive filters in nonstationary scenarios. In *Int. Conf. in Acoust. Speech and Signal Process.* IEEE, 2013.

[21] N. Mellado, D. Aiger, and N. J. Mitra. Super 4pcs fast global pointcloud registration via smart indexing. *Computer Graphics Forum*, 33(5):205–215, 2014.

[22] A. Myronenko and X. Song. Point set registration: Coherent point drift. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2262–2275, 2010.

[23] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. 3rd Intern. Conf.*, pages 145–152. IEEE, 2001.

[24] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[25] A. Sayed. *Adaptive filters*. Wiley-IEEE Press, 2008.

[26] F. Seybold and U. Wössner. Gaalet-a c++ expression template library for implementing geometric algebra. In *6th High-End Visualization Workshop*, 2010.

[27] M. J. Stanway and J. C. Kinsey. Rotation identification in geometric algebra: Theory and application to the navigation of underwater robots in the field. *Journal of Field Robotics*, 2015.

[28] J. Tangelder and R. Veltkamp. A survey of content based 3d shape retrieval methods. In *Proc. of 2004 Internat. Conf. on Shape Modeling Applications*, pages 145–156, June 2004.

[29] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *Proc. of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, pages 356–369, Berlin, Heidelberg, 2010. Springer-Verlag.

[30] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 311–318, New York, NY, USA, 1994. ACM.

[31] R. Valkenburg and L. Dorst. Estimating motors from a variety of geometric data in 3d conformal geometric algebra. In L. Dorst and J. Lasenby, editors, *Guide to Geometric Algebra in Practice*, pages 25–45. Springer London, 2011.

[32] M. W. Walker, L. Shao, and R. A. Volz. Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Underst.*, 54(3):358–367, Oct. 1991.

[33] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, 13(2):119–152, Oct. 1994.