Accuracy and Efficiency Performance of the ICP Procedure Applied to Sign Language Recognition

Juarez Paulino da Silva Júnior
University of Brasília, Computer Science Department,
Brasília, DF, Brazil, 70910-900
juarez.paulino@gmail.com

and

Marcus Vinicius Lamar
University of Brasília, Computer Science Department,
Brasília, DF, Brazil, 70910-900
lamar@unb.br

and

Jacir Luiz Bordim
University of Brasília, Computer Science Department,
Brasília, DF, Brazil, 70910-900
bordim@unb.br

Abstract

This work addresses the problem of recognizing the American Sign Language (ASL) hand alphabet relying only on depth information acquired from an RGB-D sensor. To accomplish this goal, a novel Iterative Closest Point (ICP) based recognition methodology is proposed where it comprehensively analyzes the inputs and outputs of the alignment as efficiency and accuracy determinants. Next, a novel classification technique, denoted *Approximated KB-fit*, is proposed to efficiently handle the space complexity of the database template matching. The overall accuracy of the recognition reached a performance of 99.04% in a cross-validation workbench with 520 distinct input depth images. The achieved frame rate was 7.41 FPS performed on a 2.4 GHz single processor based machine.

Keywords: 3D Shape Congruence, ASL Hand Alphabet Recognition, ICP Alignment, Pattern Recognition, Template Matching Architecture.

1 Introduction

The recent introduction of low-cost sensor devices, empowered with real-time RGB-D image acquisition mechanisms, favored the appearance of many innovative computer vision works. In particular, the use of depth data can robustly simplify typical and difficult tasks such as image segmentation, occlusion handling or data interpretation in environments with poor illumination properties [1, 2].

Furthermore, the spatial information of the viewed scenes substantially increases the application possibilities of expressive vision-based algorithms in research areas like 3D reconstruction, augmented reality and 3D body and object pose tracking [1, 3, 4].

Depth data is also of particular interest to hand gesture recognition. Although one can find accurate and efficient solutions in the literature, they often require additional hardware or accessories which may be expensive or demand a complex setup. Moreover, these solutions present constraints which reduce the natural user interface with the system [5]. Alternatively, solutions based only on intensity images usually lack robustness when applied to different light conditions or have low accuracy to distinguish gestures images with similar colors and shapes even when they present distinct tridimensional compositions [2]. Thus, it should be expected that spatial data acquisition through a low cost RGB-D

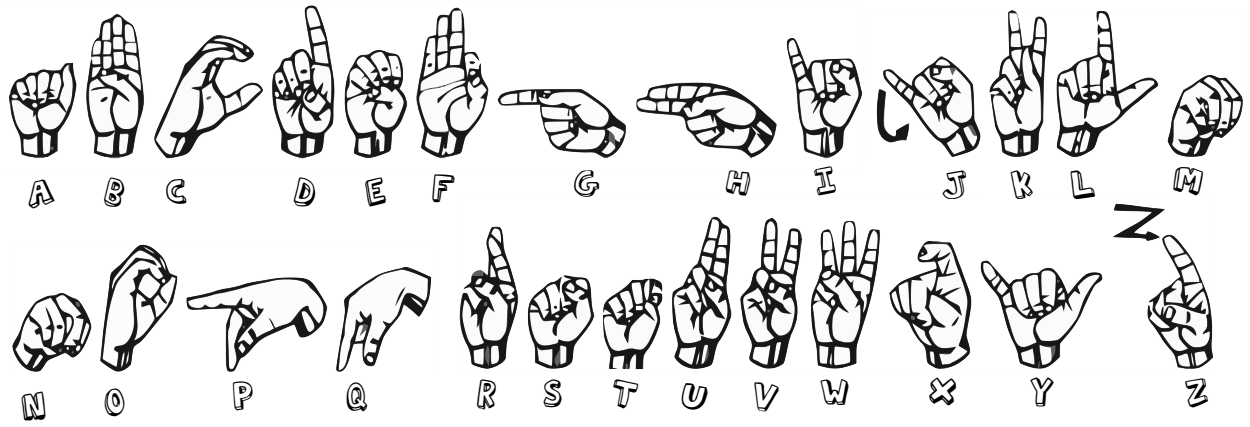


Figure 1: ASL hand alphabet. Reproduced from [7].

sensor may be a more natural and robust way to handle the aforementioned problems.

Sign language recognition is considered the most complex category in the gesture recognition domain [6], since it deals with a large amount of static and dynamic postures which can be very similar in shape and involve not only hands, but also face, torso and arms. This work is restricted to the recognition of the American Sign Language (ASL) alphabet letters, a set consisting of 26 static hand posture gestures [1], as shown in Figure 1.

In fact, to solve this less-general problem, by using spatial information, a system is required to process an input depth image from the sensor and assign it to one of the 26 possible representative classes of the ASL letter alphabet. One simple, but effective, approach is to apply the template matching architecture [8, 9] as a classification tool which consists in: (i) *matching* the test data to a selected representative set of the database models; and (ii) *comparing* these images in a pairwise fashion to deduce the level of similarity between both images. The reference models that provide the best values for a predefined *metric of correspondence* are used to identify the class to which the test data belongs. Though it is not the best process for fast image recognition, this architecture allows a full and detailed analysis of a given matching procedure [2]. From this scope, a natural matching procedure is to directly compare the aligned test and model images. Iterative Closest Point [10] (ICP) is an algorithm that performs such alignment and allows the quantification of the correspondence between the test and model pair. Other works have already tried the ICP as a matching procedure [11] in the same context, but have also discarded it, claiming that the ICP was not suitable to retrieve good comparison metrics while performing the classification.

This work proposes an investigation of the ICP as a 3D shape matching algorithm applied to the recognition of the ASL alphabet letters. The presented results and methodology aim to explore the inputs and outputs of the ICP procedure and to identify a set of parameters to improve *accuracy* (how many correct matches it can identify) and *efficiency* (average recognition speed).

Once the ICP has properly performed the data alignment, the next step is to establish a comparison mechanism to infer similarities. As it will be shown, the correspondence metrics choice directly affects the recognition accuracy. Experimental results show that the proposed methodology attains an accuracy level over 99%. According to Table 1, this accuracy surpasses other similar works in ASL alphabet recognition in at least 10 percentage points. Besides that, most real applications of the hand alphabet recognition should be consistent with the notion of an almost real-time processing (≈ 15 FPS). In fact, the ICP is a slow iterative technique and, unless it is somewhat improved, it is not recommended to real-time applications [12]. By imposing performance constraints as input parameters to the ICP, the proposed implementation obtained an improvement of 10 fold while maintaining a reasonable accuracy. Likewise, based on the described approach, the template matching architecture should be efficiently modified since its naive brute force (denoted here as *best-fit*) runs only in 0.20 FPS (Table 5). In this aspect, the proposed Approximated KBucket-*fit* classification limits the space complexity of the applied database and contributes with a 40 time improvement in the recognition speed.

Regarding the considerations above, the main contributions in this paper are:

1. The proposal and analysis of 5 output metrics (*Alignment Matrix Norm*, *Maximum Distance Threshold*, *Minimum Inliers*, *Maximum Inliers* and *Mean Inliers*),

2. Evaluation of the ICP input parameters and their effects on accuracy and recognition speed;

3. The proposal of an efficient algorithm for the template matching (*Approximated KBucket*-fit) as a classification tool.

[1] In order to adapt the temporal sequence gestures from letters 'J' and 'Z' to a static form, it is presumed that they are captured on their final posture positions.

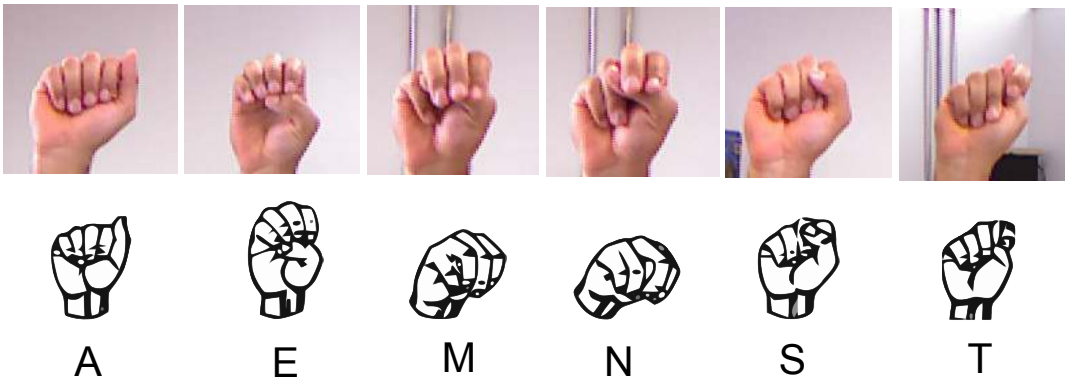


Figure 2: Illustration of ambiguous classes in the ASL alphabet. Letters are represented by a closed fist, and differ only by the thumb position, leading to higher confusion levels. Adapted from [19].

The rest of the paper is organized as follows. Section 2 presents the related works while Section 3 gives an overview of the ICP. Section 5 details the ICP enhancements for ASL recognition. Section 6 presents the experimental scenarios along with their respective simulations and detailed results. Finally, Section 7 concludes the paper, including future directions.

2 Related Work

Many related works share common problems and objectives with the present paper. Although sign language recognition is a well studied topic in computer vision research [2, 6], most of the early solid works did not establish complete solutions to the problem when it requires accuracy, efficiency and natural user interaction altogether. In this context, the use of depth data with real-time acquisition allows not only the fast object segmentation, as it might be naively used, but also a significant improvement on the accuracy of the recognition. This section presents a brief compilation of related works which successfully apply depth data on the recognition of hand and sign language gestures.

2.1 General Techniques

Some techniques explore the properties of 3D data acquisition, applying them on the gesture recognition. In [13], the authors acquire the depth stream data, analyze the morphology, position and orientation of hand and fingers, and apply a constraint-matching to recognize a few common hand shapes in Japanese Sign Language. Although it is possible to find optimal constraints that improve system accuracy on small sets, it is still a hard task to scale this type of classification in larger scenarios.

Another set of approaches exists to deal with sign recognition through depth images, which includes Support Vector Machines (SVM) [14], K-Nearest Neighbors (KNN) [15], Average Neighborhood Margin Maximization (ANMM) [16], Hidden Markov Models (HMM) [17], and Artificial Neural Networks (ANN) [14, 18]. These approaches rely on specific feature extraction, extensive offline training and produce fast estimation for real-time applications. However, augmenting or diversifying the number of gestures often requires a new training process, and a poor feature specification tends to be a main drawback in obtaining high accuracy.

A recent employed technique is due to the use of *Random Decision Forests* (RDF) [14, 19]. It performs classification in real-time frame rates but retrieves only a confidence level for each trained class, which, most of the times, is not accurate enough given a defined set of features. Pugeault and Bowden [19] built an interactive system that uses Gabor Filter to extract features prior to RDF classification. Their results presented interesting ambiguity classes for the ASL alphabet letters (Figure 2), highlighted by the sets $\{A, E, M, N, S, T\}$ and $\{U, R\}$, from which they got low confidence levels and, even with the use of depth data, could not be addressed.

2.2 ICP Specific Related Methodologies

Besl and McKay [10] figured out that ICP could be useful in the congruence of two 3D distinct geometric forms and not only as an alignment procedure. However, due mainly to the slow iterative process, few researches have investigated the indirect outputs of the ICP procedure and applied them on hand shape matching.

A more remarkable set of contributions from the ICP procedure may be borrowed from researches exploring biometrics. Amor *et al.* [20] built a classical probe-and-gallery model and successfully recognized face depth images from arbitrary points of view using the ICP shape matching. In [21], concerned with efficiency, the authors proposed a spatial voxel indexation model associated with a database repository. This association was then used to perform

Table 1: Comparative results between related works and techniques.

Related Work	Recognition Application	Applied Technique	Best Reported Accuracy
Trindade *et al.* [11]	ASL alphabet	Template Matching + ICP	N/A
Keskin *et al.* [14]	ASL digits	ANN + Joint Position	98.81%
		SVM + Joint Position	99.90%
Uebersax *et al.* [16]	ASL alphabet	ANMM + Hand Orientation	89.60%
Liwicki and Everingham [17]	BSL alphabet	HMM + Orientation Descriptors	84.10%
Pugeault and Bowden [19]	ASL alphabet	RDF + Gabor Filter	75.00%
Amor *et al.* [20]	Face Biometrics	Template Matching + ICP	97.25%
Ping and Bowyer [21]	Face Biometrics	Template Matching + ICP	94.10%
	Ear Biometrics		97.30%

constant time computation of the closest points in an ICP iteration. Both works use only the common mean square distance error to identify shape congruences. Moreover, ICP studies on 3D shape biometrics often rely on body parts which are more statically fixed when compared to the high degree of freedom (DOF) context of hand gestures, so it is simpler to achieve higher accuracy (Table 1).

On a more recent work, Trindade *et al.* [11] developed a system to recognize manual alphabet letters in the Portuguese Sign Language. The authors conducted experiments with acquired depth data and used them on shape matching applying a naive ICP approach. They stated that, because of the limited information from the acquired point cloud, their ICP implementation was not well suited to perform same class sign matching. However, they did not present concrete results of this analysis and neither described their experiments in depth.

Table 1 summarizes the main works discussed in this section. The underlying approaches of these works are presented along with the recognition scenario where they were applied and the accuracy reported by their experiments.

From the discussion above, none of the related works have systematically analyzed the ICP as a possible matching procedure for 3D hand shape recognition. Also, only a few works explore the ICP efficiency on near real-time contexts, as it requires optimization improvements on the naive processing. None of them, however, are directly applied to the sign language recognition.

3 Background Overview

Iterative Closest Point (ICP) [10] is the dominant fine registration algorithm in the literature and it aims at the retrieval of an accurate solution to the Euclidean rigid motion between two 3D point surfaces. The rigid motion (transformation) is usually recovered by means of the scale, rotation and translation components that bring the two point sets to a same spatial orientation. This way, the ICP algorithm works by iteratively minimizing the cost function of the distances computed between selected corresponding points in the two surfaces (Figure 3). In turn, the cost function is usually associated with a mean square distance metric which it is mathematically stated to always converge to the nearest local minimum [10].

In its most simple approach, and assuming a cost function based on point-to-point distance between correspondences, an iterative step of the alignment includes the following:

- Starting with an estimate T of the rigid transformation of a previous iteration, each point of the first point set, $p_i \in P$, is brought to an induced point $T(p_i)$ in the same oriented system of the second point set. Therefore, along one iterative step, the method needs to search for another set of points on the second model, $m_i \in M$, which minimizes the distance cost function between $T(p_i)$ and m_i (Equation 1).

$$\text{point-to-point}_{\text{ rms-error}} = \sqrt{\frac{1}{L}\sum_{i=1}^{L}\left\|\overrightarrow{m_i} - (R\overrightarrow{p_i} + \overrightarrow{t})\right\|^2}, \tag{1}$$

  where L is the number of assigned correspondences, p_i and m_i are the corresponding pairs selected from both test and model images, R is the rotation matrix, and $\overrightarrow{t}$ is the translation vector, both obtained by the transformation T acquired in the last iteration.

- One of the possible solutions to minimize the previous equation, by means of unit quaternions [22], is to build a symmetric matrix $Q(\Sigma_{pm})$, of size 4×4:

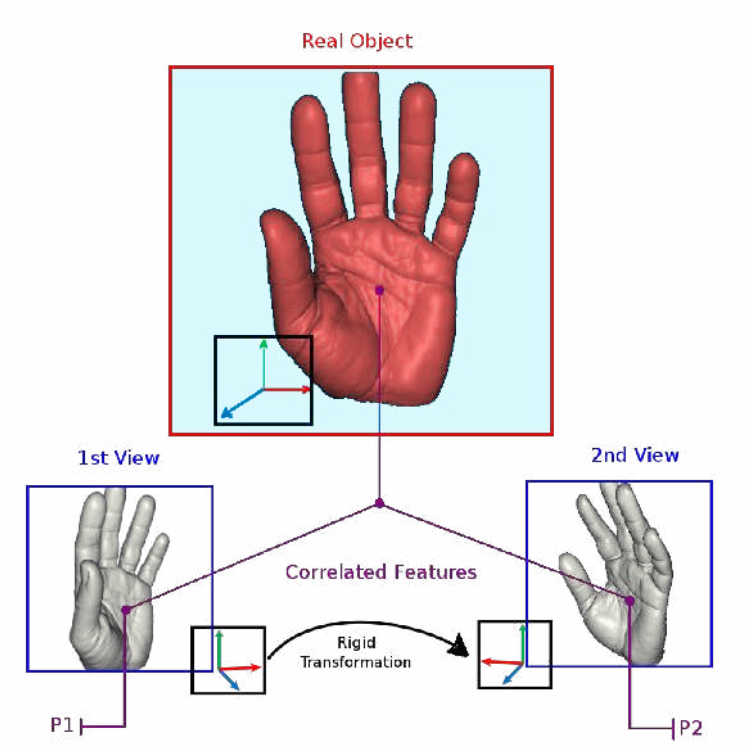


Figure 3: Illustration of the rigid motion acquisition from corresponding points.

$$Q(\Sigma_{pm}) = \begin{bmatrix} tr(\Sigma_{pm}) & \Delta^T \\ \Delta & \Sigma_{pm} + \Sigma_{pm}^T - tr(\Sigma_{pm}) I_3 \end{bmatrix}, \tag{2}$$

where tr is the *trace* function, $\Delta = \begin{bmatrix} A_{23} A_{31} A_{12} \end{bmatrix}^T$ is computed from the skew-symmetric matrix given by $A_{ij} = \left(\Sigma_{pm} - \Sigma_{pm}^T\right)_{ij}$; Δ^T is the transpose of Δ; I_3 is the identity matrix; and Σ_{pm} is the cross-variance matrix of the point sets p_i and m_i given by:

$$\Sigma_{pm} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left[\overrightarrow{p}_i \overrightarrow{m}_i \right] - \overrightarrow{\mu}_p \overrightarrow{\mu}_m, \tag{3}$$

with:

$$\overrightarrow{\mu}_p = \frac{1}{N_p} \sum_{i=1}^{N_p} \overrightarrow{p}_i, \tag{4}$$

$$\overrightarrow{\mu}_m = \frac{1}{N_m} \sum_{i=1}^{N_m} \overrightarrow{m}_i, \tag{5}$$

the centroids of the point sets p_i and m_i, respectively.

- The unit eigenvector $\overrightarrow{q}_R = \begin{bmatrix} q_0 \; q_1 \; q_2 \; q_3 \end{bmatrix}^T$ correlated to the greatest eigenvalue of the matrix Q is chosen as the new rotation expressed in terms of a quaternion. The rotation matrix R can, then, be retrieved and the new translation vector t is easily computed by the difference vector between the centroids as shown below:

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}, \tag{6}$$

$$t = \overrightarrow{\mu}_m - R \overrightarrow{\mu}_p. \tag{7}$$

- The method should iterate until it converges to an optimal solution, where the cost function is minimized, in other words, is below a cutoff value (*threshold*).

Once the iterative procedure has been completed, the last acquired alignment transformation (T) should give a correspondence map between the two different views (coordinate systems) of the aligned point surfaces. It means that, given a point $p_i \in P$, it is corresponded to the best point p' given by the alignment T on the surface of M, as shown by Equation 8:

$$p' = T(\overrightarrow{p_i}) = R(\overrightarrow{p_i}) + \overrightarrow{t}. \tag{8}$$

As a fine alignment mechanism, the ICP supposes that a good initial rigid motion between the shapes has been provided. This way, it is possible to either increase the convergence speed as to achieve a global minimum for the cost function. It is necessary to highlight that when a reasonable estimate is not provided, the processing may incorrectly converge to a local minimum and fail to retrieve a satisfactory solution. Coarse registration techniques [23, 24] are frequently used to find rough estimates. In the proposed methodology, a rough estimate is simply taken from the translation vector between the two shape centroids.

An important work on the ICP procedure is due to Rusinkiewicz and Levoy [12] who optimized the algorithm for efficiency. They divided the procedure in stages and analyzed many of the proposed variants to achieve the best convergence speed. From their set of variants definitions, this work applies:

i) a classical select-match-minimize type of iteration;

ii) a kd-tree data structure, with normal point compatibility, is used to find corresponding points in logarithmic time;

iii) threshold rejection for the distant corresponded pairs of points;

iv) point-to-point minimization in the first steps to assure stability;

v) point-to-plane iterations in the main loop body;

vi) uniform random subsampling from both the point clouds.

The overall complexity of this proposed ICP variant is in the order of:

$$O(KL(\log N_p + \log N_m)), \tag{9}$$

where K is the performed number of iterations, L is the required number of corresponding pairs selected in each iteration, N_p and N_m are the number of vertices from test and model data, respectively.

4 ICP Enhancement for Sign Language Recognition

One simple observation when performing a template matching recognition is that there must be some mechanism to correspond the test and model data [8]. In this sense, it is natural to consider that if both data are aligned, i.e. they are in a same global orientation, it should be easier to compare them. This is a clear motivation to apply the ICP registration, but there still must be a way to relate both the aligned test and model information. In fact, this type of correspondence may be deduced from the ICP processing itself.

Another important concern to the design of the recognition system is how long the ICP procedure will take to correspond the test data with the entire model database. That is, how much it is possible to improve the ICP *efficiency* without compromising its *accuracy* performance and how to efficiently handle the number of comparisons under the database size. In this case, the overall procedure to perform a simple recognition is the application of M instances (related with database size) of the ICP alignment, as in Equation 9, resulting in a total time complexity of:

$$O(MKL(\log N_p + \log N_m)). \tag{10}$$

This section introduces the main contributions of this work, starting with the ICP enhancements to quickly produce reliable metrics for the sign language recognition (Section 4.1), and presenting, later, an efficient design for classification techniques based on template matching (Section 4.2).

4.1 Alignment Analysis

The first contribution is the improvement of the ICP alignment processing. Such improvement can be achieved by manipulating the variable inputs (the iterative elements and procedure modifiers) and inspecting what may be measured (correspondence metrics) in a single instance of the registration. The hypothesis is that each of these selected properties may directly contribute to the accuracy or efficiency performance of the ICP shape matching.

4.1.1 Input Parameters

Iterative Elements: As already mentioned, the cost of the ICP algorithm is a main problem when applying it to real-time systems. This fact can be derived from the ICP algorithm complexity, as shown in Equation 9. For instance, with large K or L values, the ICP running time is not expected to complete its task in a reasonable time to support real-time applications. So there are basically two identified types of iterative elements in the proposed implementation which have clear influence in the ICP efficiency:

1. **The maximum allowed number of main loop iterations**:

   As previously presented, the main loop body of the implemented ICP procedure is composed of point-to-plane steps of the select-match-minimize iterations. If the two point surfaces are already closely related, the ICP processing can find the best alignment in fewer iterations;

2. **The maximum allowed number of correspondence points selected**:

   The restriction of the points selection in each iteration is straight correlated to a subsampling of the original ICP problem. As already stated, an uniform random subsampling is implemented. The proposed implementation includes a parameter to set the chosen number of selected points to correspond in each iteration, so only a reduced size of the original alignment problem needs to be solved.

From both the input parameters, it is expected that a restriction on their values contributes to speed the recognition process without affecting the shape matching accuracy.

Modifiers: The basic ICP implementation retrieves only the rigid transformation (rotation and translation) from two shape geometries. In this sense, the procedure modifiers are implementation flags which decide whether or not to compute an approximate scale in the minimization stage [22]. Three types of scale transformation are investigated:

- rigid motion with no modifier;
- motion with a computed uniform scaling;
- motion with a computed non-uniform scaling.

It is expected that the applied scale modifiers can achieve a slightly better accuracy since it can handle different hand shapes and positions from the acquired images.

4.1.2 Correspondence Metrics

The definition of correspondence metrics emerges from the need to compare and match different alignments of the test data with the template models. In ICP-based sign language recognition, the choice of a reliable metric may contribute to identify the right matching.

The first chosen metric (point-to-point RMS Error) is usually known and, in most of the works, is the only metric used to implement *correspondence evaluations*. The second metric (point-to-plane RMS Error) is common in ICP works but is mainly related to alignment convergence and is not usually applied in recognition.

1. **Root Mean Square *Point-to-point* Error (Point-to-point RMS Error)**

   This metric has been employed in a number of related works [11, 12, 20, 21]. The Point-to-point RMS error computation is performed after the final iteration with the same function required for the minimization step (Equation 1);

2. **Root Mean Square *Point-to-plane* Error (Point-to-plane RMS Error)**

   The *point-to-plane error* is usually indicated to align flat surfaces and frequently has better convergence than the point-to-point minimization [12]. Besides its importance to convergence and efficiency criteria, it is not usually found in works that evaluates correspondence between two shapes.

These commonly applied *root mean square* metrics are derived directly from the error value of the iterative steps in ICP processing, and may not be the most appropriate parameters to achieve high accuracy in the ASL recognition. In this context, this work proposes another set of metrics for evaluating correspondences, as follows:

3. **Alignment Matrix Norm**

   The *alignment matrix norm* used in this text is relative to the Fobrenius norm, which is a natural extension of the vector norms applied to matrices. This metric quantifies how much the last minimization step proceeded in the direction of a local minimum.

Given the 4 x 4 transformation matrix T_{aligned} of the last step and assuming I_4 as the identity matrix, the required computation is done by:

$$\|T_{\text{residual}}\| = \sqrt{\sum_{i=1}^{4}\sum_{j=1}^{4} |(T_{\text{aligned}} - I_4)_{ij}|^2}; \tag{11}$$

4. **Maximum Distance Threshold (Max-dist Pairs)**

   In the proposed ICP implementation, a *maximum distance threshold* is iteratively computed while performing the correspondence rejection. This metric starts with a rough estimation (the minimum of the diagonal bounding box distance from both the point sets) and is updated by a factor of the median distance of the corresponding points in each iteration.

   In practical terms, as the data images are acquired from a fixed system, it is expected that the smaller the maximum distance estimate is, the better the match between the given shapes will be;

5. **Minimum Inliers (Min inliers)**

   A *minimum inliers* metric is proposed as a similarity metric which estimates the spatial overlapping parts in the final matched surfaces. It may be computed efficiently through a proper voxel indexation algorithm or, with less performance, by using kd-trees. The search procedure identifies a given overlapping point if its nearest neighbor on the other image is under the max distance threshold.

   The semantic of overlapping metrics is trivial since the more the two shapes overlap, the better the match is expected to be. The word "minimum" in this metric indicates that the overlap computation is conducted from the test data or from the model data, applying the acquired rigid transformation T (for test data) or its inverse T_{inv} (for model data), choosing the type of data which gives the minimum fraction of correlated features (pessimistic view of the correspondence overlap);

6. **Maximum Inliers (Max Inliers)**

   The *maximum inliers* is similar to the minimum inliers value but it acquires the greatest fraction of correlated features from both directions of overlapping computation (optimistic view of the correspondence overlap);

7. **Mean Inliers**

   The *mean inliers* metric is just the acquisition of the mean value of the overlapping represented by the minimum and maximum inliers. In this way it is expected from statistics that a more representative value is obtained.

The correspondence metrics are a good mechanism to estimate the 3D shape congruence between two depth images. By extracting such values, it is stated that a sign language system based on ICP processing can accurately identify the 26 ASL alphabet letters given an enrolled set of template models.

4.2 Classifier

In a basic verification scenario, a test data P is matched against a set of template models $\{M_i\}$ uniformly distributed in the evaluated classes so that a classifier can infer which of the ASL letters the test belongs to. To perform such deduction, a classification technique should give a consistent interpretation of the extracted correspondence metrics.

This work proposes, as contributions, the *best-fit* (Section 4.2.1) and the *Approximate KBucket-fit* (Section 4.2.2) techniques to the classification stage. The *best-fit* was primarily stated to verify the accuracy of the 3D shape matching. The *Approximate KBucket-fit*, alternatively, has a strong bias to improve the methodology efficiency, reducing the space complexity for database comparisons.

4.2.1 Best-fit Classification

A first simple approach is to consider, in each instance, the complete set of database models and to run the ICP procedure to each possible enrolled image. The template model which provides the best fit value given a selected correspondence metric is chosen as the one that labels the test data in its respective class.

This strategy could be related to a form of 1-NN classification [25, 9], where a test sample is recognized with its nearest neighbor data given a set of features. However, in the context of the ICP, there is no proper feature vector from a single image, since the evaluation metrics are acquired in a pairwise manner. So, there is no possible nearest-neighbor context and the complete search of the best similarity must be employed. From now on, this first procedure is referred as the *best-fit* technique.

Algorithm 1 Approximate KBucket-fit.

Require: $\mathbb{M}$: Full set of models;
P: Test data instance;
K: Class bucket size.
Ensure: $\mathbb{C}$: Recognized class.

1. $U \leftarrow \emptyset$ {Stores random samples from the 26 classes}
2. $B[26] \leftarrow \{\emptyset\}$ {Bucket lists with K samples of metric values per class}
3. $M[26] \leftarrow \{0\}$ {Mean metric value per class}
4. $U \leftarrow$ GENERATESUBSET($\mathbb{M}, \mathbb{K}$)
5. **for all** model $u \in U$ **do**
6. $B[u.\text{class}] \leftarrow B[u.\text{class}] \cup$ EVALUATEICPMETRIC(P, u)
7. **end for**
8. $\mathbb{C} \leftarrow \emptyset$, BestMean $\leftarrow 0$
9. **for all** class $c \in$ {A-Z} **do**
10. $M[c] \leftarrow$ COMPUTEMEANVALUE($B[c]$);
11. **if** $\mathbb{C} = \emptyset$ **or** BestMean $< M[c]$ **then**
12. $\mathbb{C} \leftarrow c$, BestMean $\leftarrow M[c]$
13. **end if**
14. **end for**
15. **return** $\mathbb{C}$;

4.2.2 Approximate KBucket-fit Classification

One of the obvious drawbacks of the *best-fit* technique is its slow comparison process of one test data against all the template models. That is, before the ICP registration, there is no *a priori* knowledge of the correspondence metrics between the input instance and the enrolled database. Concerned with the efficiency requirements, this work proposes the *approximate KBucket-fit* (KB-fit) algorithm.

A sketch of the procedure is listed in Algorithm 1. In this technique, a subset of the database models are selected in a non-deterministic fashion so that K samples from each class are chosen (line 4). Next, processing involves $\|U\|$ instances of the ICP alignment (lines 5-7), which are responsible for retrieving similarity values regarding a predetermined correspondence metric. The final computation consists in obtaining, for each class, the mean similarity value of each class (lines 8-14). The recognition process terminates, in line 15, when the class with the best mean value is found.

As the KB-fit technique relies in a randomized and statistical procedure, an empirical set of experiments are conducted in Section 6 to analyze its accuracy.

5 Proposed Methodology

The specific paradigm, applied throughout the work, consists of a template matching architecture where classification is performed by comparing a given "testcase" against an enrolled set of depth images models. The complete diagram of the proposed methodology is shown in Figure 4. Recognition is performed by a series of three stages, each of them with its own requirements and providing the respective output elements to be used in the subsequent stage.

An initial sequence of pre-processing steps are executed to prepare the raw depth images acquired from the sensor device (1st Stage). Next, the ICP procedure aligns a given test image with a database models subset which is evenly distributed from the ASL alphabet classes (2nd Stage). Then, the sub-products of the ICP alignment are used as evaluation metrics in the proposed classification scheme (3rd Stage). The following sections describes each of these stages in detail.

5.1 Pre-processing Steps

The image acquisition is always performed with an off-the-shelf Kinect sensor [26], OpenNI SDK [27] and NITE middleware [28]. The sensor is fixed at a half body height position and only the acquired depth channel is used to apply the proposed techniques.

Hand image segmentation has always been a crucial step on the sign language recognition [2]. By using the spatial localization, depth data based algorithms simplify this hard task providing robust and accurate results. Since segmentation is not the main focus, this paper applies a common depth thresholding approach, which defines a fixed boxed frame from where all its contained points are determined as hands. The segmented images are acquired as a

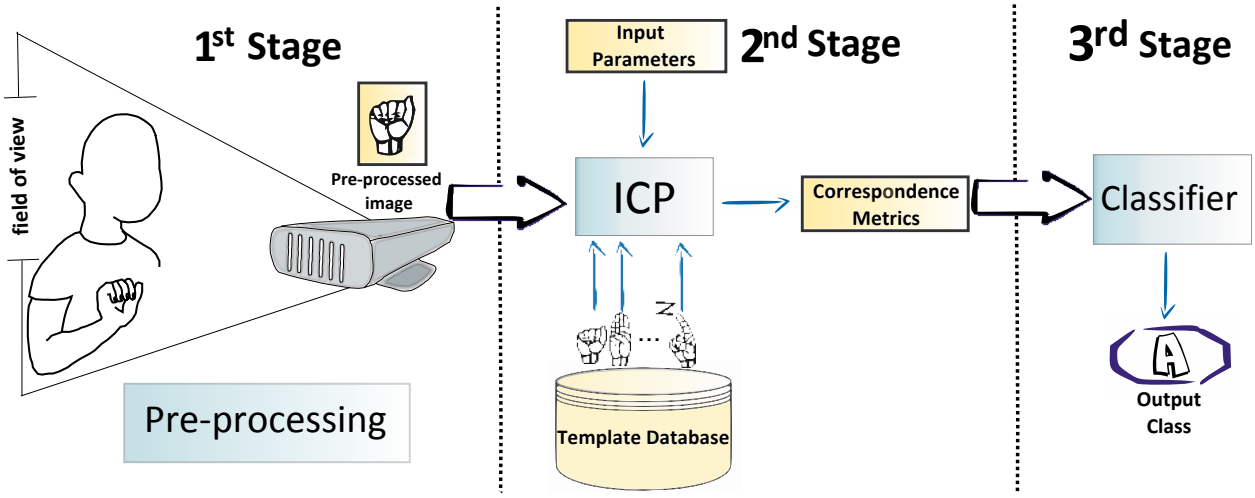


Figure 4: Diagram of the applied recognition methodology.

128×128 pixel window where each pixel represents the hand depth distance ranging from 70cm to 110cm of the sensor device. To normalize the acquisition and to obtain a more uniform image representation, the user of the system is required to perform the hand gesture trying to fill the entire box frame during segmentation.

The coordinates in the acquired 128×128 window are given in pixels for the x,y-axis and in millimeters for the depth information (z-axis). The *depth coordinate system* is the Kinect's native data representation. Such representation, however, is not appropriate for recognition purposes. Hence, a *world coordinate system* must be defined to match with a true 3D Cartesian coordinate system where at least the distance metrics are compatible with the three spatial axis. In this world system, every point is specified by 3 axis values x, y and z. The x and y axis run along a line in the same direction and with the same origin as the x,y-axis of the projected pixel image but with a proper metric space. The z axis runs into the scene, perpendicular to both the x and y axis, with the same semantics as the old depth coordinate system. Although the OpenNI SDK [27] provides conversion procedures to precisely estimate the coordinate metrics from the camera point of view, the use of its functions require expensive computation which should not be practical for real-time applications. Due to this condition, it is proposed a simpler conversion function from the native *depth coordinate system* to an approximate *world coordinate system*. The conversion retrieves the horizontal and vertical maximum field of view extension (Figure 4) from the camera and performs an image scaling, as defined in Equation 12. From this conversion, it is expected that the metric space becomes more consistent and the Euclidean properties are preserved without compromising efficiency.

$$\begin{pmatrix} x_{\text{world}} = x_{\text{depth}} \times \frac{\text{HorizontalFieldOfView}}{\text{HorizontalPixelResolution}} \\ y_{\text{world}} = y_{\text{depth}} \times \frac{\text{VerticalFieldOfView}}{\text{VerticalPixelResolution}} \\ z_{\text{world}} = z_{\text{depth}} \end{pmatrix} \tag{12}$$

The output of the pre-processing stage is a 128×128 (16K) 3D point image consisting of the essential hand information from where all the other recognition steps are applied. No additional pre-processing is performed.

5.2 ICP Processing

This stage is responsible to apply the ICP alignment between the pre-processed test image and the template models in the database, retrieving the correspondence metrics required to perform the final classification. The most significant part of the processing time in the methodology is spent in this stage, therefore the majority of the efficiency issues are also dealt here.

To instantiate the ICP processing, it is required to choose: (i) a set of the input parameters (Section 4.1.1); and (ii) an appropriate correspondence metric (Section 4.1.2) to keep track of the performed alignments. The main purpose of selecting the input parameters is to enhance the overall efficiency by limiting the complexity of the procedure. In contrast, the choice of a correspondence metric (analysis in Section 6.1) will lead to better accuracy results.

The applied template database was built from 20 samples of each of the 26 ASL alphabet letter (a total of 520 model samples) acquired under different ambient light conditions, with some of the possible variants of the same hand gesture and from a single user. Even though the data was intentionally acquired under different light conditions, it did not influence the accuracy results since all the processing is produced by the depth channel (light has almost no influence in

Table 2: Baseline values for measuring the ICP performance.

Maximum number of ICP iterations:	10
Maximum correspondence pairs by iteration:	50
Procedure modifier:	Rigid motion with no modifiers
Evaluation metric:	Mean Inliers metric
Classification technique:	best-fit

the depth data). On the other hand, the diversification of the database, by means of the accepted postures variants, was selected from minor degree rotations of each of the standard postures, as described in Figure 1.

Finally, the use of the proposed template database should consider the specific classification technique that will be applied in the next stage. The *best-fit* classification (Section 4.2.1) tries to perform the alignment of the test image against all the samples in the database, whereas the *KB-fit* technique (Section 4.2.2) smartly selects a database subsampling, thus improving efficiency.

5.3 Classification

Classification is the last stage in the recognition methodology. It basically gathers all the correspondence metrics from each test-model pair and indicates the best class of representation for the test image. This stage is efficiently computed in the template matching architecture, and it is even faster than image acquisition from the pre-processing stage. It runs in $O(M)$, where M is the number of pairwise alignments performed in the previous ICP processing stage.

In terms of accuracy, the *best-fit* technique (Section 4.2.1) presents the best results since it searches the entire database samples. Its use is justified by the hypothesis that the closest template model should have a more valuable correspondence metric with the test data. This way, if the database has a larger number of samples, there will also be a better chance for the *best-fit* classification finding the right output class.

In cases where efficiency matters, like in real-time applications, the *KB-fit* technique (Section 4.2.2) can be employed. It downsizes the database samples required for alignment in the previous stage and attempts to maintain the accuracy level. In the context of this technique, a specific analysis is performed, indicating the best tradeoff between accuracy and efficiency while varying the bucket size values (Section 6).

The next section encompasses a full set of experimental scenarios and their results are discussed so the expected hypothesis are verified.

6 Experiments and Discussion

The proposed methodology is evaluated based on (i) *accuracy*: the percentage of correct matches for a given test scenario; and (ii) *efficiency*: average ICP runtime and average recognition frame rate. For this purpose, both *offline* simulations and *online* experiments have been conduced. While offline simulations were used to compute the average ICP runtime and accuracy, online experiments were used to compute the average frame processing rate.

In order to restrict the scope of the analysis, and avoid a combinatorial explosion in the number of possibilities, a baseline combination for the studied parameters has been defined. These values are shown in Table 2 and the reasoning behind their choice is discussed in the following sections.

Figure 5: Successful ICP matching of the acquired test images (aligned on gray-scale) with the 26 ASL template models (referenced in the RGB channel).

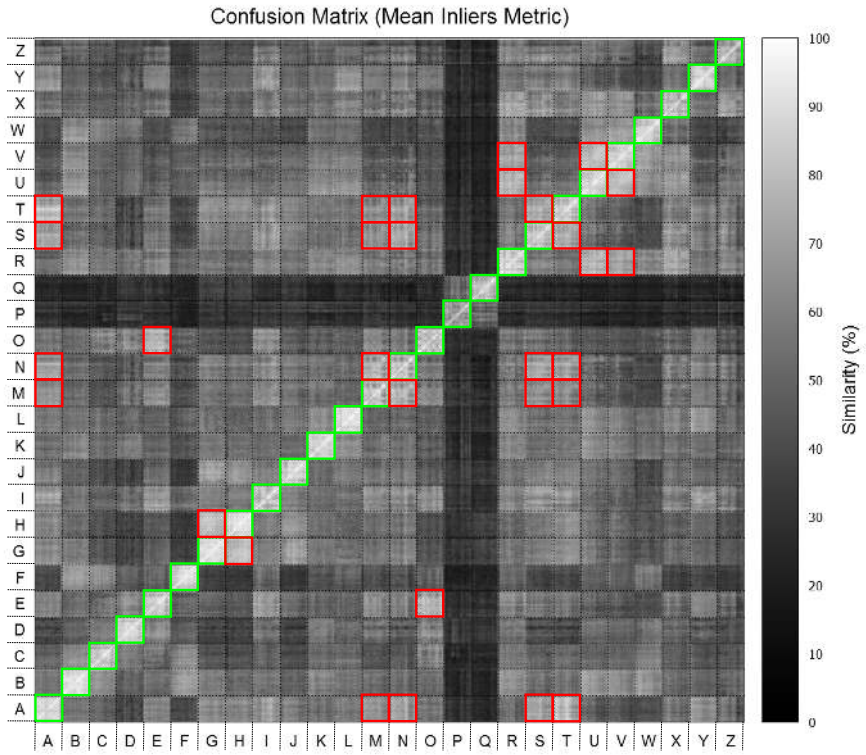


Figure 6: Confusion matrix for a simulated scenario with the entire database.

6.1 Accuracy Verification

As can be observed in Figure 5, the proposed ICP procedure can effectively match any of the 26 shapes of the ASL hand alphabet. This result provides evidence that alignment can be properly established under the given capture and pre-processing circumstances. As discussed in Section 3, computation of coarse registration was not necessary prior to ICP application. The results in Figure 5 also contradict one of the statements in the work of Trindade *et al.* [11] which says that ICP fails to match some of the ASL letters given the incomplete point surfaces acquired from the Kinect sensor. In fact, it may be empirically verified that the depth map usually lacks information where the observed extension of an object is parallel to the z-axis of the sensor coordinate system (e.g., a forefinger pointing to the camera). However, this lack of information does not degenerate the retrieval of the ICP rigid alignment for the purposes of ASL alphabet recognition. This behavior can be seen from the presented alignments where incomplete depth maps, shown in grayscale, did not cover the entire model, shown in the RGB channel, but were correctly aligned in the proposed environment.

Although promising, these results do not show the actual rate of correct matches (accuracy). Thus, a more detailed verification is needed before taking further conclusions.

6.1.1 Handling Ambiguities

In order to verify the input parameters and the correspondence metrics, a template matching scenario is established where each model data is aligned with all the others in a quadratic complexity fashion. Figure 6 presents the confusion matrix of the raw *mean-inliers* metric values given the baseline configuration in Table 2. In this confusion matrix, the classes are divided in 26×26 clusters and each cluster is equivalent to the confusion values between two different letters. The analysis of each cluster is given from the decomposition of 20×20 elements, where each element means the comparison of a specific letter model (L_1) with another letter model (L_2). The expressive dark cross marks denote the poses of the letters 'P' and 'Q' which were distinctly acquired with significant portion of the forearms. Acquisition of the forearms in the input data makes the models of these letters particularly different from any other in the database.

In the figure, The squared-marked clusters in the anti-diagonal represents the *Mean Inliers* metric values of the elements within the same class. Thus, these clusters have the brightest colors compared to others on their respective rows. On the other hand, the remaining sparse squared-marked clusters in the figure illustrate the most ambiguous correspondences values for incorrect matches. These clusters still have a bright color but but they have less intensity than those on the anti-diagonal. Table 3 shows the average similarity for these elements. With some small differences from the ambiguities suggested in [19], the most ambiguous sets in terms of *Mean Inliers* similarities are $\{A, M, N, S, T\}$, $\{E, O\}$, $\{G, H\}$ and $\{R, U, V\}$ (shown in Table 3). As can be verified, even in the worst case, the computed values for

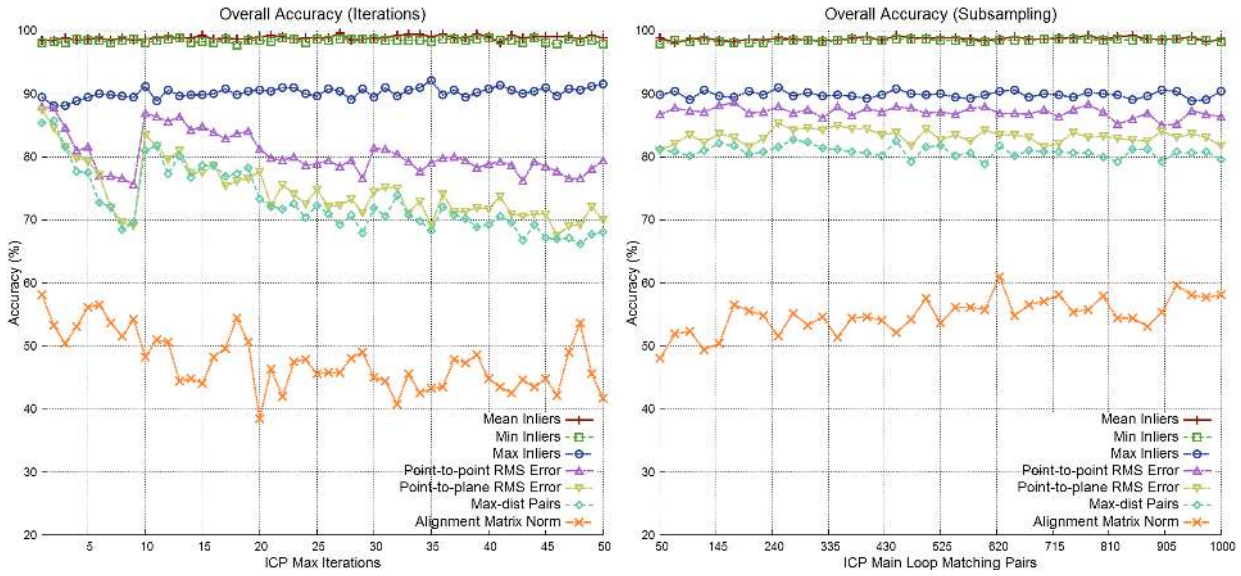
Table 3: Averaged similarity values (in %) for the sets of the most ambiguous clusters in Figure 6.

%	A	M	N	S	T
A	87.72	64.77	68.50	71.32	77.13
M	64.77	78.26	73.06	64.63	63.13
N	68.50	73.06	78.08	71.05	68.97
S	71.32	64.63	71.05	82.73	74.10
T	77.13	63.13	68.97	74.10	83.86

%	E	O
E	83.60	75.14
O	75.14	80.74

%	G	H
G	90.06	79.62
H	79.62	89.38

%	R	U	V
R	85.74	76.61	70.90
U	76.61	86.63	79.35
V	70.90	79.35	86.02



(a) Maximum iterations results

(b) Subsampling results

Figure 7: Accuracy performance through iterative elements.

a correct matching differ in at least 5 percentage points. Such difference allows for correct recognition of the output classes while handling ambiguities.

6.1.2 Methodology Evaluation

The results presented so far have shown that ICP can produce correspondence metrics and represent them as similarities among the possible classes. These results are achieved during the 1st and 2nd stages of the proposed methodology (Figure 4). The 3rd stage's role is to take advantage of the positive difference between correct and incorrect class similarities. This difference is what the classifier implicit uses to correctly assign a given data to its representative class.

The *best-fit* technique is applied to count the number of correct classifications by varying the ICP input parameters and evaluation metrics. In Figure 7, the iterative elements are considered with respect to their achieved accuracies under different evaluation metrics. A first observation is that, for the majority of the proposed metrics, the reported accuracy is weakly correlated to either the number of maximum iterations and the iterative subsampling. These results show that the *Alignment Matrix Norm* gives the worst and most unstable results under the studied elements. On the other hand, the *Min Inliers* and *Mean Inliers* attain an accuracy rate of $\approx 100\%$, with a slight advantage in favor of the *Mean Inliers*.

Such observations regarding the evaluation metrics are also confirmed when investigating the overall accuracy of the proposed ICP modifiers (Table 4). With 99.04% of correct matches, the *Mean Inliers* metric successfully recognized 515 cross-validation inquiries, mismatching only 5 instances, all of them related to the alignment from letter models 'A' and 'T'. Furthermore, it is verified that, at least for the given sample space, the proposed modifiers have no such interesting contribution to accuracy improvements.

To correctly evaluate accuracy when applying the KB*-fit* technique, a different set of simulations was performed. To deal with the randomization factor of the non-deterministically selections, 100 distinct experiments were conducted with $1..19$ possible K values for the bucket size. On each experiment, for a fixed instance of bucket samples (training data), the accuracy of the KB*-fit* algorithm was verified by classifying all the out of the bucket database images (test

Table 4: Overall accuracy relating metrics and modifiers.

Metrics \ Modifiers	no-modifier	uniform-scale	non-uniform scale
Mean Inliers	98.85%	98.85%	99.04%
Min Inliers	97.88%	98.46%	98.46%
Max Inliers	89.81%	89.23%	89.81%
Point-to-point RMS Error	86.73%	87.12%	86.35%
Point-to-plane RMS Error	80.96%	80.96%	80.96%
Max-dist Pairs	81.15%	82.50%	81.15%
Alignment Matrix Norm	28.08%	25.58%	24.81%

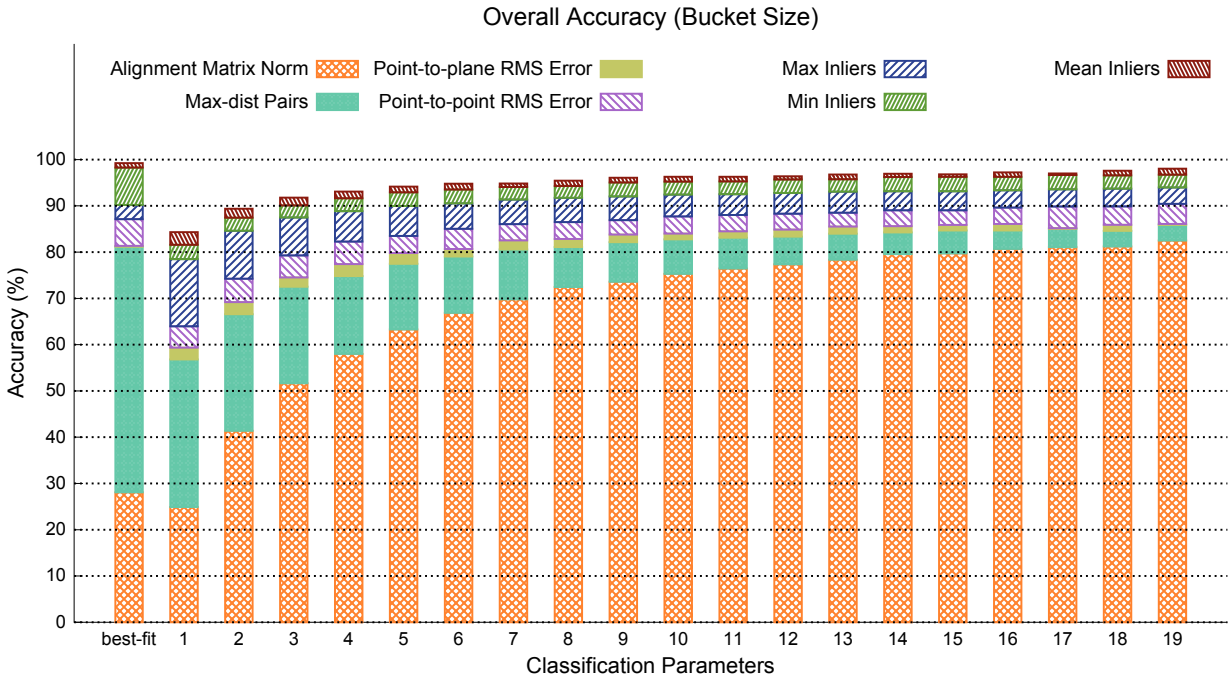


Figure 8: Average accuracy with varying KB-*fit* bucket sizes.

data). Figure 8 presents the average accuracy for all the performed experiments. The first bar on the x axis, labeled as *best-fit*, represents the comparable value of accuracy using the baseline parameters (Table 2). The results show that, even for $K \approx 1$, the proposed KB-*fit* technique attains a remarkable performance for all of the proposed metrics. In this sense, the statistical analysis applied in KB-*fit*, through the mean value computation, allows it to build a representative metric value for each class prior to the recognition. This presents a significant improvement even for unstable metrics, such as *Alignment Matrix Norm*.

The accuracy results allow to state an order of the studied correspondence metrics, determined as follows: (1^{st}) *Mean Inliers*, (2^{nd}) *Minimum Inliers*, (3^{rd}) *Maximum Inliers*, (4^{th}) *Root Mean Square Point-to-point Error*, (5^{th}) *Root Mean Square Point-to-plane Error*, (6^{th}) *Maximum Distance Threshold*, and (7^{th}) *Alignment Matrix Norm*. The obtained order indicates that the proposed inliers based metrics outperform in accuracy all other RMS error based metrics, commonly applied when using ICP [11, 20, 21].

6.2 Efficiency Performance

Through a dimensional point of view, the efficiency of the ICP procedure is dominated by the change of the iterative elements of its input parameters. This is justifiable since other ICP elements, like the proposed application of modifiers, are considered a natural extension of the processing, with their time complexities dominated by the overall procedure. Figure 9 presents the correlation between the iterative elements and the average processing time of an instance of the ICP alignment in a 2.4 GHz single processor based machine. It can be verified that the maximum allowed number of iterations has a stronger driven force when compared to the maximum allowed number of iteration pairs. Thus, increasing the number of iterations has a worse impact on the processing time. When both variable parameters are selected with a large value, the ICP running time can be up to 10 times slower ($\approx$ 320ms). By checking the template

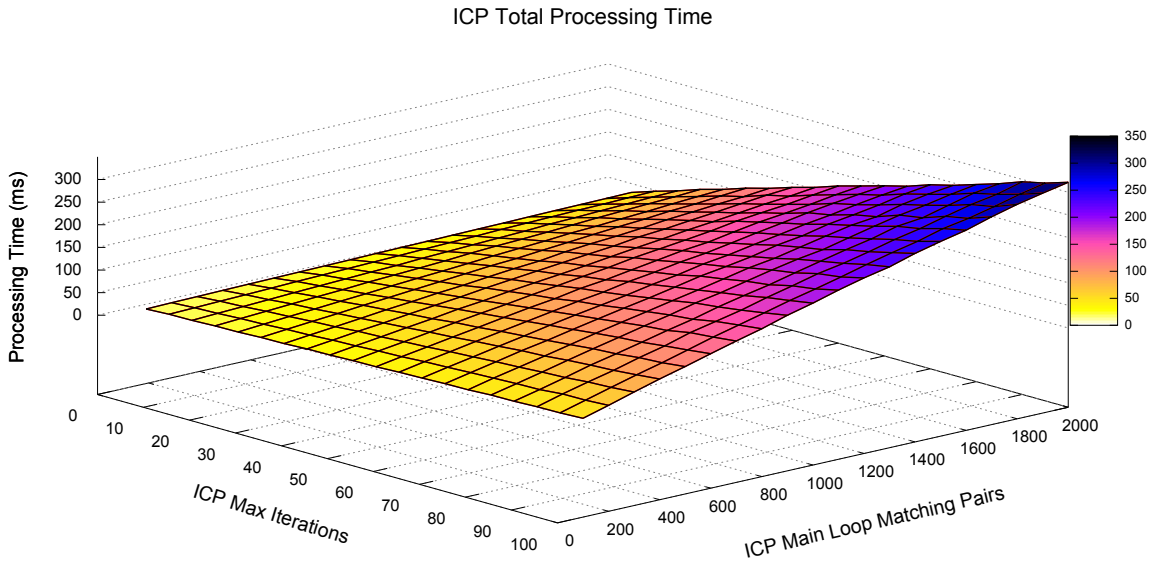


Figure 9: Average ICP processing time by varying the iterative elements. Performed in a 2.4 GHz single processor machine.

Table 5: Average annotated frame per second (FPS) rates for different classification parameters. Performed in a 2.4 GHz single processor machine.

Parameter	Rate	Parameter	Rate	Parameter	Rate	Parameter	Rate
best-fit	0.20	***15B-fit***	0.41	***10B-fit***	0.97	***5B-fit***	3.70
19B-fit	0.27	***14B-fit***	0.48	***9B-fit***	1.33	***4B-fit***	4.53
18B-fit	0.29	***13B-fit***	0.55	***8B-fit***	1.68	***3B-fit***	5.33
17B-fit	0.33	***12B-fit***	0.67	***7B-fit***	2.20	***2B-fit***	6.29
16B-fit	0.36	***11B-fit***	0.81	***6B-fit***	2.91	***1B-fit***	7.41

matching time complexity in Equation 10, when M instances of the ICP alignment are required, this scenario would make the application of the proposed methodology in real-time contexts impractical. In contrast, as the accuracy results (Figure 7) have shown that no significant improvement was achieved by increasing the number of iterative elements, a minimal configuration value, as in Table 2, positively decreases the processing time of the ICP alignment (15ms) and efficiently handles the ASL recognition needs.

Another important analysis of the speed of the ASL recognition can be done by examining the frame rate in a practical online implementation of the proposed methodology. The results presented in Table 5 show that the KB-*fit* technique may achieve almost two orders of magnitude in the recognition speed when compared to the *best-fit* approach. From the accuracy results, it can be verified that the KB-*fit* does not substantially degrades the performance, even for small bucket sizes.

From the results in Figure 8 and in Table 5, the main correlation between accuracy and efficiency can be summarized as follows:

- Accuracy always achieves its best performance with the *Mean Inliers* metric, so it is the best choice for the ICP correspondence metric on the proposed methodology;
- If accuracy is the main focus, the *best-fit* classification will achieve the highest performance in exchange for a slow recognition process (0.20FPS): 99.04% of correct matches in the evaluated scenarios;
- If efficiency is the goal, the *1B-fit* classifier will outperform any of the proposed techniques for the classification stage. As shown in the results, it will still maintain a reasonable recognition accuracy (84.31%) while reaching an average reported rate of 7.41 FPS;
- A good balance of accuracy and efficiency is to apply the *5B-fit*: it will allow a high accuracy (94.16%) with an efficiency rate of 3.70 FPS.

In summary, the use of the proposed methodology where the ICP procedure is applied has proven to be a reliable system for recognition, achieving high accuracy performance when compared to other state of the art solutions (Table 1). At the same time, it has a drawback, result of to the application of the template matching architecture. The proposed KBucket-*fit* classification limits this problem by reducing the space complexity of the database samples, making it possible to provide good accuracy results and support online applications.

7 Conclusion and Future Works

The results presented show that the ICP algorithm can be used to produce accurate matches even with a very similar set of gestures poses. With a best achieved accuracy of 99.04%, the methodology has shown to be accurate enough to the sign language recognition. However, as ICP processing is always conditioned to the pairwise data alignments, the general template matching paradigm is still a bottleneck to its application in real-time contexts (≈ 15 FPS).

As a future work, coding the ICP procedure to work in accelerated hardware, such as GPU's, is a plausible alternative to apply this technique in real-time. Given the high reported accuracy, another possibility is to combine existing classification tools, like random decision forests, to coarsely reduce the space of possible matches and let the proposed methodology resolve only the tricky cases.

By proposing a diversified set of correspondence metrics, it was possible to build a comprehensive analysis of the applicability of the Kinect's depth data in the 3D shape recognition. Furthermore, the proposed KBucket-*fit* algorithm has contributed to significantly increase the achieved performance when compared with the time consuption of the brute-force *best-fit* algorithm, without compromising accuracy. From this point of view, the KBucket-*fit* also contributes to many other interesting fields, such as biometrics. Besides that, the described methodology and techniques can be promptly adapted to any other sign language hand alphabet with minor modifications, requiring only the replacement of the template models in the dataset.

Finally, the use of the ICP procedure has a great potential to deal with dynamic gesture recognition, since it can robustly track the alignments of two spatial and temporal proximal geometry shapes. This may complete the requirements for the full sign language machine recognition and easily approach this work to applications in human computer interaction and robotics.

References

[1] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 559–568.

[2] J. Suarez and R. Murphy, "Hand gesture recognition with depth images: A review," in *RO-MAN, 2012 IEEE*, 2012, pp. 411–417.

[3] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ser. ISMAR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 127–136.

[4] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *Computer Vision and Pattern Recognition*, Providence, Rhode Island, USA, 2012.

[5] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 3, pp. 311–324, 2007.

[6] M. Al-Ahdal and N. Tahir, "Review in sign language recognition systems," in *Computers Informatics (ISCI), 2012 IEEE Symposium on*, 2012, pp. 52–57.

[7] Wikipedia. American manual alphabet. [Online]. Available: http://en.wikipedia.org/wiki/American_manual_alphabet

[8] X. Zabulis, H. Baltzakis, and A. A. Argyros, *Vision-based hand gesture recognition for human computer interaction*, ser. on Human Factors and Ergonomics. Lawrence Erlbaum Associates, Inc. (LEA), 2009, ch. 34, pp. 34.1–34.30.

[9] S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artificial Intelligence Review*, pp. 1–54, 2012.

[10] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, 1992.

[11] P. Trindade, J. Lobo, and J. Barreto, "Hand gesture recognition using color and depth images enhanced with hand angular pose data," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, 2012, pp. 71–76.

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

[13] K. Fujimura and X. Liu, "Sign recognition using depth image streams," in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, 2006, pp. 381–386.

[14] C. Keskin, F. Kirac, Y. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 1228–1234.

[15] M. Van den Bergh, D. Carton, R. de Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss, "Real-time 3D hand gesture interaction with a robot for understanding directions from humans," in *RO-MAN, 2011 IEEE*, 2011, pp. 357–362.

[16] D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool, "Real-time sign language letter and word recognition from depth data," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 383–390.

[17] S. Liwicki and M. Everingham, "Automatic recognition of fingerspelled words in british sign language," *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 0, pp. 50–57, 2009.

[18] K. R. Konda, A. Königs, H. Schulz, and D. Schulz, "Real time interaction with mobile robots using hand gestures," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, ser. HRI '12. New York, NY, USA: ACM, 2012, pp. 177–178.

[19] N. Pugeault and R. Bowden, "Spelling it out: Real-time ASL fingerspelling recognition," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 1114–1119.

[20] B. Amor, M. Ardabilian, and L. Chen, "New experiments on ICP-based 3D face recognition and authentication," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, 2006, pp. 1195–1199.

[21] P. Yan and K. Bowyer, "A fast algorithm for ICP-based 3D shape biometrics," in *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*, 2005, pp. 213–218.

[22] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America. A*, vol. 4, no. 4, pp. 629–642, Apr. 1987.

[23] J. P. da Silva Júnior, D. L. Borges, and F. de Barros Vidal, "A dynamic approach for approximate pairwise alignment based on 4-points congruence sets of 3D points," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. 889–892.

[24] Z. Zhang, S. H. Ong, and K. Foong, "Improved spin images for 3D surface matching using signed angles," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 537–540.

[25] S. Malassiotis, N. Aifanti, and M. Strintzis, "A gesture recognition system using 3D data," in *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, 2002, pp. 190–193.

[26] Microsoft Corp. Redmond WA. Kinect for Xbox 360.

[27] OpenNI SDK. [Online]. Available: http://www.openni.org

[28] PrimeSense. NITE Middleware. [Online]. Available: http://www.primesense.com/solutions/nite-middleware