# Finger Spelling Recognition from Depth data using Direction Cosines and Histogram of Cumulative Magnitudes

Edwin Escobedo; Guillermo Camara;
Department of Computer Science (DECOM)
Federal University of Ouro Preto
Ouro Preto,MG, Brazil
Email: edu.escobedo88,gcamarac@gmail.com

***Abstract*—In this paper, we propose a new approach for finger spelling recognition using depth information captured by Kinect™ sensor. We only use depth information to characterize hand configurations corresponding to alphabet letters. First, we use depth data to generate a binary hand mask which is used to segment the hand area from background. Then, the major hand axis is determined and aligned with $Y$ axis in order to achieve rotation invariance. Later, we convert the depth data in a *3D* point cloud. The point cloud is divided into subregions and in each one, using direction cosines, we calculated three histograms of cumulative magnitudes $H_x$, $H_y$ and $H_z$ corresponding to each axis. Finally, these histograms were concatenated and used as input to our Support Vector Machine (SVM) classifier. The performance of this approach is quantitatively and qualitatively evaluated on a dataset of real images of American Sign Language (ASL) hand shapes. The dataset used is composed of 60000 depth images. According to our experiments, our approach has an accuracy rate of 99.37%, outperforming other state-of-the-art methods.**

***Keywords*-Finger spelling recognition; depth information; points cloud; directional cosines; support vector machine (SVM).**

## I. Introduction

For hearing impaired, sign language is the only way of communication, where signs are performed by moving the hands in combination with facial expressions and posture of the body. Sign language, contrary to common gestures, has complex spatial grammars that are highly structured. Deaf people use systems of communication based on sign language and finger spelling which is a system where each letter of the alphabet is represented by an unique and discrete hand movement.

In order to improve the quality of deaf community life, research in sign language recognition has played a significant role. As spoken languages, sign languages have its own variations, even in countries where people spoke the same language. For example, English-Spoken countries such as the United States of America, United Kingdom, and Australia, each of them has created their own sign language: ASL, BSL and Australia Sign language, respectively. This means that a same word may be represented by different signs. Unfortunately, these languages are barely known outside of the deaf community, meaning a communication barrier.

Manual spelling, or finger spelling, is a system where each letter of the alphabet is represented by an unique and discrete movement of the hand. The finger spelling integrates a sign language due to many reasons: when a concept lacks of a specific sign, for proper nouns, for loan signs (signs borrowed from other languages), for finger spelled compounds or when a sign is ambiguous [1]. Each sign language has its own finger spelling similar to different characters in different languages.

Nowadays, several techniques have been developed to achieve an adequate recognition rate of sign language. Two approaches are commonly used to interpret them: sensor-based and vision-based [2]. Sensor-based methods use sensory gloves and motion trackers to detect hand shapes and body movement. Vision-based methods use standard cameras to capture and classify hand shapes and body movements. Unfortunately, the Sensor-based methods require extensive calibration, also they restrict the natural movement of the hands and they are often very expensive. Therefore, video-based methods are more used because they are less intrusive, but new problems are arose: locating the hands and segmenting them is a non-trivial task also intensity images are vulnerable to illumination variations and to cluttered backgrounds, hindering hand detection and tracking.

However, with the recent appearance of low price depth sensors, such as Microsoft Kinect™ [3] which provides intensity and depth data, and skeleton joint positions, these problems are overcome because depth information can be used to improve the segmentation process [4], [5], [6], [7] and it is also invariant to illumination changes. In addition, depth cameras were also used for hand gesture recognition [8], [9], [10]. Pugeault & Bowden [8] use a Microsoft Kinect™ device to collect RGB and depth images. They extracted features using Gabor filters and then a Random Forest predicted the letters from the American Sign Language (ASL) finger spelling alphabet. Bergh & Van Gool [11] propose a method based on a concatenation of depth and color-segmented images, using a combination of Haar wavelets and neural networks for 6 hand

poses recognition of a single user. Otiniano & Camara [12] proposed a method for ASL recognition using RGB-D information, which combines SIFT feature from intensity images and gradient kernel descriptor [13] from depth images. Then, the bag-of-visual-words (BOW) model [14] is employed, and finally, the histogram of BOW features are fed into SVM classifier. Zhu et al. [15] propose a method that combines color and depth kernel descriptors. Estrela *et al.* [16] use a light-feature called Binary Appearance and Shape Elements (BASE) to fuse intensity information with shape information. The proposed framework is based on BOVW, and the partial least squares technique is used to train models for each individual letter. A novel method of pattern recognition is presented in [17] for recognizing 36 different gestures using SIFT features with PCA and template matching methods. Silva *et al.* [18] explores the spatial pyramid matching descriptor for finger spelling recognition, achieving a high accuracy rate. Li *et al.* [19] propose to combine a feature learning approach based on sparse auto-encoder (SAE) with a convolutional neural network (CNN). Then, the learned features from both channels are concatenated and fed into a multiple layer PCA to get the final feature. This approach was tested in a ASL finger-spelling data set.

*Contributions:* In this paper, we propose a novel method for finger spelling recognition, exploiting the depth information advantages such as illumination invariance and facility to perform hand segmentation. First, hand depth information is segmented from the background and converted to a Point Cloud ( $PC_{depth}$). Then, the $PC_{depth}$ is divided into subregions and using the direction cosine concept, for each point in a subregion, we estimate its magnitude and their three directional angles formed with axis $X$, $Y$ and $Z$. Then, three oriented magnitude histograms which represent the local features of a sign, are computed.

The experiments are performed using a public database composed of 60,000 depth images stating 24 symbols classes [20]. The obtained results show that the accuracy obtained by our method, using depth data only, outperforms other methods that use multimodal data [8], [16], [15], [12], [19] and [18]. The results show that our method is promising.

The remainder of this paper is organized as follows. In Section II, we describe our proposed finger spelling recognition system. Experiments and results are presented in Section III. Conclusions and future works are presented in section IV.

## II. Proposed Model

This section describes our proposed method in order to perform finger spelling recognition. The proposed method is divided in four main stages, as shown in Fig. 1. In the first stage, hand segmentation is performed on depth data. In the second stage, the segmented hand is aligned with $Y$ axis. Then, depth data is converted into a point cloud. In the third stage, we compute the local features on the aligned hand. Finally, these features are used as input to our SVM classifier.

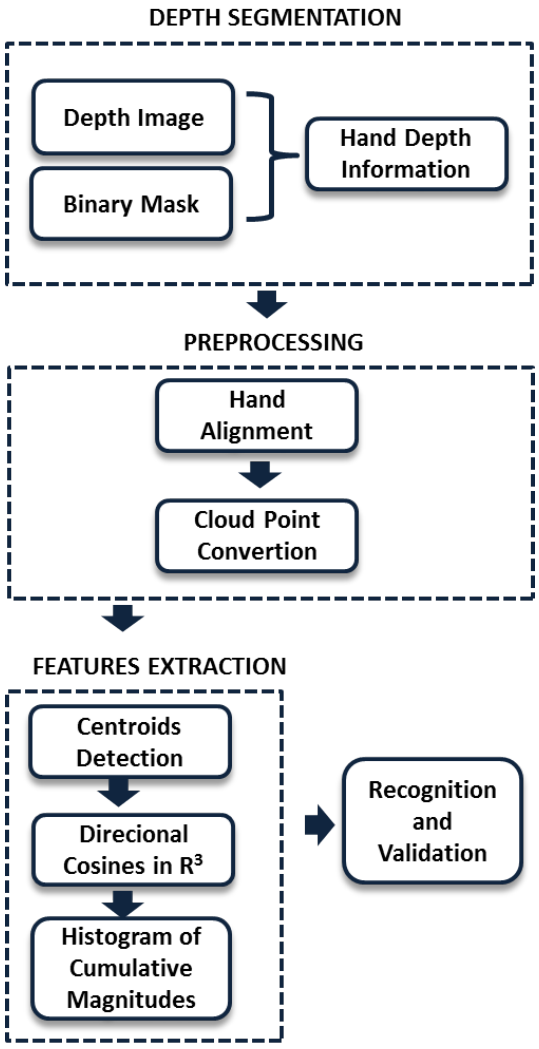


Fig. 1. Proposed model for our finger spelling recognition system.

### A. Depth Segmentation

The hand segmentation was performed using depth data, as hands appear closer to the sensor, a threshold was used to segment the hand. Then, the segmented hand is used as a binary mask over the depth image to only get the hand depth values. Fig. 2 shows the result of the hand depth segmentation.

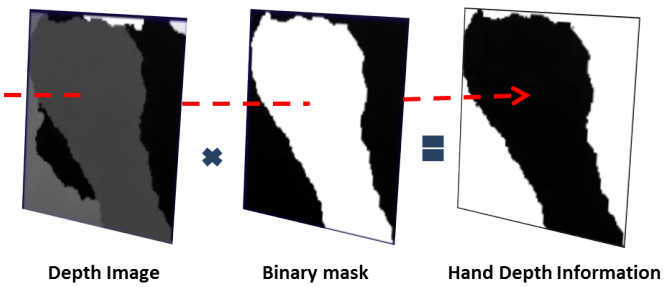


Fig. 2. Example of hand depth segmentation process.

### B. Preprocessing

After hand segmentation, to avoid rotation problems, the hand is aligned in relation to $Y$ axis. For this, we calculate the major diagonal and its orientation $\theta$. If the hand is inclined to the left side, its angle is negative and if it is inclined to the right its angle is positive. Then, we calculated the difference $\Delta\theta$ as follows:

$$\Delta\theta = \begin{cases} 90 - \theta & \text{if} \quad \theta > 0 \\ 270 - \theta & \text{if} \quad \theta < 0 \end{cases} \tag{1}$$

Then, after the hand alignment, depth data that belongs to the hand is converted into a point cloud $PC_{depth}$. Fig. 3 shows the preprocessing step of hand depth data.

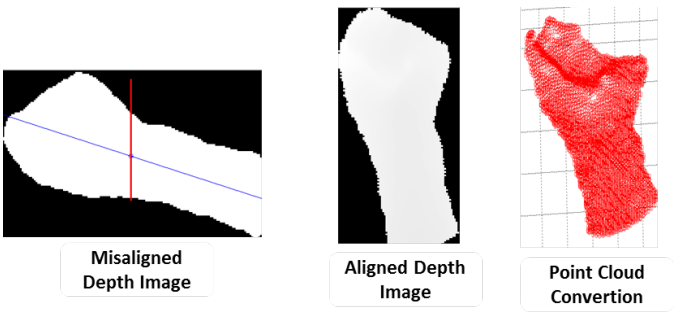


Fig. 3. Preprocessing. First, the hand is aligned with $Y$ axis and then converted into a point cloud.

### C. Features Extraction

For the feature extraction, we use the *direction cosine concept*:

**Definition 1** (Direction Cosines)**.** *The direction cosines of a vector* $\mathbf{V}$ *are the cosines of the angles between the vector and the three coordinate axis (see Fig. 4). In three-dimensional Cartesian coordinates, if* $\mathbf{V}$ *is a vector in the Euclidean space,* $\mathbb{R}^3$*, then:*

$$V = v_x e_x + v_y e_y + v_z e_z \tag{2}$$

*where* $e_x$*,* $e_y$ *and* $e_z$ *are the standard basis in Cartesian notation and the scalars* $v_x$*,* $v_y$*,* $v_z$ *being the scalar components of the vector* $\mathbf{V}$*. Then, the direction cosines are:*

$$|V| = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{3}$$

$$\alpha = \cos a = \frac{V \cdot e_x}{|V|} = \frac{v_x}{|v|} \tag{4}$$

$$\beta = \cos b = \frac{V \cdot e_y}{|V|} = \frac{v_y}{|v|} \tag{5}$$

$$\gamma = \cos c = \frac{V \cdot e_z}{|V|} = \frac{v_z}{|v|} \tag{6}$$

*Furthermore,* $\cos a$*,* $\cos b$ *and* $\cos c$ *must meet the follow equality:*

$$\cos^2 a + \cos^2 b + \cos^2 c = 1 \tag{7}$$

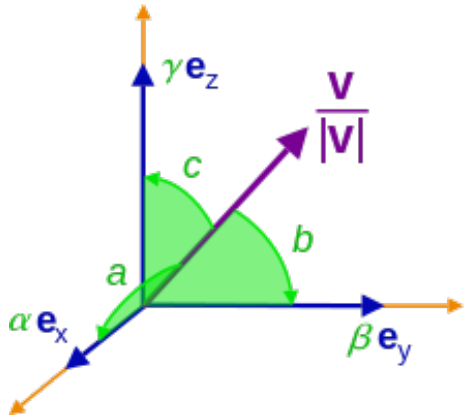


Fig. 4. Spatial representation of the direction cosines.

Based on this concept and taking advantage of the point cloud data, we propose a new method for local feature extraction, using spacial information to generate a vector $V_{depth}$ which is the concatenation of several histograms of cumulative magnitudes. The steps to generate $V_{depth}$ are as follows.

1) The point cloud $PC_{depth}$ is divided in $N \times N$ spatial subregions $S_i$.
2) For each subregion $S_i$ calculate the central point $CP_{S_i}$. Then, the directional vectors $VS_{P_d}$ are generated between $CP_{S_i}$ and the points $P_d \in S_i$.

$$VS_{P_d} = \{CP_S - P_d | \forall P_d \in S_i\} \tag{8}$$

3) For each directional vector $VS_{P_d}$ in $S_i$, decompose it into its directional cosines for each Cartesian axis ($\alpha$, $\beta$ and $\gamma$) and calculate its magnitude $|VS_{P_d}|$. For this, Equations 4, 5, 6 and 3 are used, respectively. Then, by an inverse function, we obtain the angles $a, b$ and $c$.
4) For each subregion $S_i$ calculate the cumulative magnitude orientation histograms, one for each coordinate axis $(X, Y, Z)$. Each vector $VS_{P_d}$ casts a weighted vote for an orientation-based histogram, calculated in the previous step. The histogram bins are evenly spread over 0 to 180 degrees. Thus, three cumulative magnitude orientation histograms $H_x$, $H_y$ and $H_z$ are generated for each subregion $S_i$.
5) Finally, the local feature vector $V_{depth}$ is created concatenating the cumulative histograms from each subregions $S_i$.

$$V_{depth} = \bigcup_{i=1}^{i=NxN} \{H_x^i, H_y^i, H_z^i\} \tag{9}$$

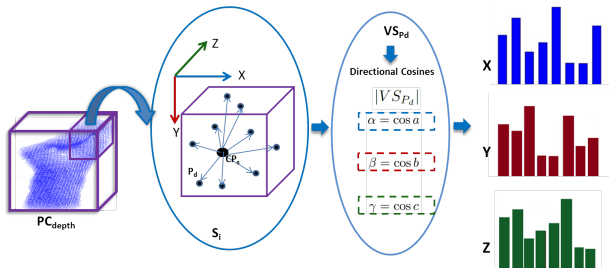


Fig. 5. Cumulative histogram generation process for $H_x$, $H_y$ and $H_z$.

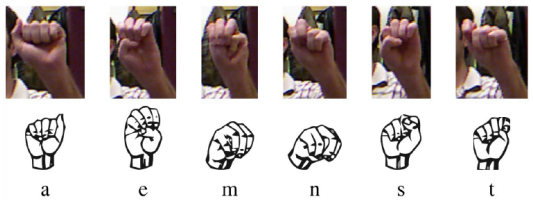


Fig. 7. Illustration of the similarity in the dataset between different classes. All are represented by a closed fist, and differ only by the thumb position.

### D. Recognition and validation

We use Support Vector Machines (SVM) to classify the features. The SVMs [21] are a useful classification method. Furthermore, SVMs have been successfully applied in many real world problems and in several areas: text categorization, handwritten digit recognition, object recognition, etc. An important characteristic of the SVM classifier is to allow a non-linear classification without requiring explicitly a nonlinear algorithm. In kernel framework data points may be mapped into a higher dimensional feature space, where a separating hyperplane can be found. Common kernel functions are: linear, polynomial, Radial Basis Function (RBF), *etc*.

## III. Experiments

In this section, we describe the database used as well as the experiments performed. We also make an analysis of obtained results.

### A. ASL Finger Spelling Dataset

The ASL Finger Spelling Data set [20] contains 500 samples for 24 signs, recorded from 5 different persons, amounting to a total of 60,000 samples. Each sample has a RGB image and a depth image, making a total of 120,000 images. The sign J and Z are not used, because these signs have motion and the proposed model only works with static signs. The data set has variety of background and viewing angles. The Fig. 6 shows some examples where is possible to see the variety in size, background and orientation.

Due to the variety in the orientation when the signal is performed, signs became strongly similar. Fig. 7 shows the most similar signs *a*, *e*, *m*, *n*, *s* and *t*. The examples are taken from the same user. It is easy to identify the similarity between these signs, all are represented by a closed fist, and differ only by the thumb position, leading to higher confusion levels. Therefore, these signs are the most difficult to differentiate in the classification task. Being necessary a descriptor able to detect the lows differences between them.

### B. Experiments

To validate our method, we work with the following experimental protocol configuration.

- We used $N = 5$ to divide the points cloud $PC_{depth}$ in 25 subregions.
- The number of bins for each histogram was 8. We work with orientations between $0°$ and $180°$.
- The total size of the vector $V_{depth}$ was: $5 \times 5 \times (8 \times 3) = 600$.
- The classification was performed using the LIBSVM (A library for Support Vector Machines) library [22] using a RBF kernel, whose values for $g$ (gamma) and $c$ (cost) are $0.5$ and $0.6$, respectively.

The classification was performed varying the amounts of training samples. The objective is to evaluate the robustness of the features extracted by our approach. The training set size used here are of: 10%, 40%, 55%, 60%, 65%, 70% and 75%. The smallest training size is defined based on the experimental protocol used by Zhu *et al.* [15]. The authors used 40 samples for each sign, this represents more and less 10% of the dataset. The obtained results are shown in the Table I.

TABLE I
Accuracy and standard deviation ($SD$) for different percentages of training and test.

| % Train | % Test | Accuracy | DS |
|---|---|---|---|
| 75 | 25 | **99.37** | 0.0608 |
| 70 | 30 | 99.34 | 0.0596 |
| 65 | 35 | 99.30 | 0.0526 |
| 60 | 40 | 99.26 | 0.0570 |
| 50 | 50 | 99.21 | 0.0518 |
| 40 | 60 | 99.03 | 0.0548 |
| 10 | 90 | 97.29 | 0.1183 |

### C. Results

We can see in Table II the comparison of our approach in two different scenarios. In the first, a 10% dataset is used for training our SVM classifier. Our approach obtain the highest accuracy, outperforming the methods based on intensity and depth information proposed in [16], [15], [18]. While Estrela *et al.* [16] use a BOW based approach on intensity and depth data, Zhu et al. [15] propose a method that combines color and depth kernel descriptors and Silva *et al.* [18] explores the spatial pyramid matching descriptor on intensity images, we use local point cloud features only. Our recognition rate is of 97.29%, overcoming other approaches. The recognition rate of our approach differs from the second best result in almost 10%.

In the second experiment, a 50% of the data is used as training set, again our approach outperforms other methods [8], [12], [18], [19]. Pugeault & Bowden [8] use Gabor filters


Fig. 6. Illustration of the variety in the ASL dataset. This array shows one image from each user and from each letter. The size, orientation and background can change to a large extent.

on intensity and depth images and a Random Forest to predict the letters. Otiniano & Camara [12] combine SIFT feature from intensity image and gradient kernel descriptor from depth image, then, a BOW approach is used to generate histograms of BOW features and fed into a SVM classifier. Li *et al.* [19] combine a feature learning approach based on sparse auto-encoder (SAE) with a convolutional neural network (CNN). This approach achieves the second best results. Again, our approach, with simple local point cloud features achieves the best results. While the accuracy rate difference between the two highest results (our approach and Li *et al.*) is small, the difference of the approach complexity is big. Our method uses simple operations, but performs slightly better than the renowned CNN deep learning algorithm.

Table III presents the confusion matrix of our proposed method. The sign with the lowest recognition rate is about 97%, a high accuracy rate. This demonstrates the efficiency of our method when working with the depth information. The letters $M$ and $N$, a small number, are still being wrongly classified.

Finally, to identify the high accuracy of our method, we compare the signals presented in the Fig. 7, where we can see that the hand configurations are similar. However, when drawing the grid graph of these signals, we can easily detect their differences. These differences are modeled by our local hand descriptor. Fig. 8 shows a grid graph of the signs showed in Fig. 7. The difference are now more evident for these signs.

## IV. Conclusion

In this paper, we propose a new method for finger spelling recognition exploiting the depth information. The difference between our method and others is the conversion of depth data into a point cloud $PC_{depth}$ which is used to extract histograms of cumulative magnitudes for each axis $(X, Y, Z)$ based in the direction cosine concept. All these operations are simple and can be computed rapidly.

TABLE II
Accuracies of the classification.

| Protocol | Method | Accuracy (%) |
|---|---|---|
| A | Silva *et al.* [18] | 92.50 |
| | Zhu *et al.* [15] | 88.94 |
| | Estrela *et al.* [16] | 71.51 |
| | **proposed approach** | **97.29** |
| B | Pugeault & Bowden[8] | 75.00 |
| | Otiniano & Camara [12] | 91.26 |
| | Silva *et al.* [18] | 97.90 |
| | Li *et al.* [19] | 99.10 |
| | **proposed approach** | **99.21** |

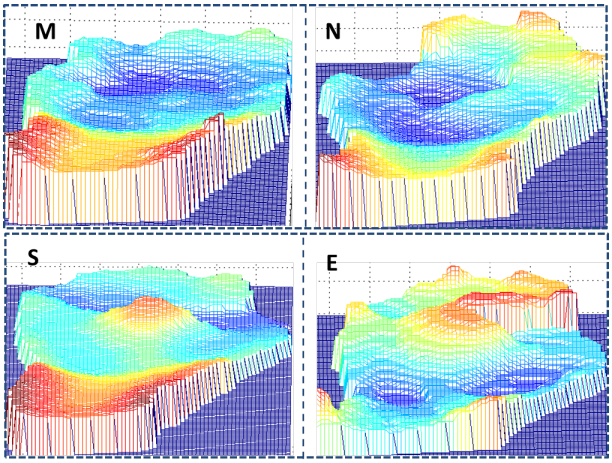


Fig. 8. Samples of the signals $M, N, S$ and $E$ graphed in the space.

To evaluate our method, we tested it in two different scenarios. In the first, we used 10% of the dataset for training, obtaining a rate recognition of 97.29%, outperforming other proposed methods. In the second, we used 50% of the dataset for training, obtaining a rate recognition of 97.29%,

TABLE III
CONFUSION MATRIX OF THE CLASSIFICATION OF 24 SIGNALS USING OUR METHOD BASED ON LOCAL FEATURES OF DEPTH IMAGES.

| | a | b | c | d | e | f | g | h | i | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| b | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| c | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| d | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| e | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| f | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| g | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| h | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| i | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| k | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| l | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| m | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| n | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.98 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| o | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| p | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.98 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| q | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.98 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| r | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| s | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.99 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| t | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.97 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| u | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.99 | 0.00 | 0.00 | 0.00 | 0.00 |
| v | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | 0.00 | 0.00 | 0.00 |
| w | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 |
| x | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | 0.00 |
| y | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |

outperforming other state of the art methods. Also, our method achieves a better differentiation of similar signs like *N, R, A, T, S* and *E*, increasing the recognition rate.

Our method is able to detect the lows variations between similar signs, achieving a high recognition rate.

As future work, we expect to use, evaluate and validate our local descriptor with dynamic hand gestures. We will combine these local features with global features that describe the trajectory of a dynamic sign.

## ACKNOWLEDGMENT

The authors are thankful to the Brazilian funding agencies CNPq, CAPES and FAPEMIG (Grant APQ-02292-12) and to the Federal University of Ouro Preto (UFOP) for supporting this work

## REFERENCES

[1] A. Puente, J. M. Alvarado, and V. Herrera, "Fingerspelling and sign language as alternative codes for reading and writing words for Chilean deaf signers," *American Annals of the Deaf*, vol. 151, no. 3, pp. 299–310, 2006.

[2] G. Murthy and R. Jadon, "A review of vision based hand gestures recognition," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 405–410, 2009.

[3] Z. Zhang, "Microsoft kinect sensor and its effect," *MultiMedia, IEEE*, vol. 19, no. 2, pp. 4–10, 2012.

[4] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of the 19th ACM International Conference on Multimedia*. ACM, 2011, pp. 1093–1096.

[5] V. Frati and D. Prattichizzo, "Using Kinect for hand tracking and rendering in wearable haptics," in *Proceedings of the IEEE World Haptics Conference (WHC)*. IEEE, 2011, pp. 317–321.

[6] Y. Li, "Hand gesture recognition using Kinect," in *Proceedings of the 3rd IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2012, pp. 196–199.

[7] Z. Mo and U. Neumann, "Real-time hand pose recognition using low-resolution depth images," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2006, pp. 1499–1505.

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

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

[10] M. d. S. Anjo, E. B. Pizzolato, and S. Feuerstack, "A real-time system to recognize static gestures of brazilian sign language (libras) alphabet using kinect," in *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems*. Brazilian Computer Society, 2012, pp. 259–268.

[11] M. Van den Bergh and L. Van Gool, "Combining RGB and ToF cameras for real-time 3D hand gesture interaction," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, ser. WACV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 66–72.

[12] K. Otiniano Rodriguez and G. Camara Chavez, "Finger spelling recognition from rgb-d information using kernel descriptor," in *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*. IEEE, 2013, pp. 1–7.

[13] L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," *Advances in Neural Information Processing Systems*, vol. 7, 2010.

[14] J. Mukherjee, J. Mukhopadhyay, and P. Mitra, "A survey on image retrieval performance of different bag of visual words indexing techniques," in *Students' Technology Symposium (TechSym), 2014 IEEE*. IEEE, 2014, pp. 99–104.

[15] X. Zhu and K. Wong, "Single-frame hand gesture recognition using color and depth kernel descriptors," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2989–2992.

[16] B. Estrela, G. Cámara-Chávez, M. F. Campos, W. R. Schwartz, and E. R. Nascimento, "Sign language recognition using partial least squares and rgb-d information," in *Proceedings of the IX Workshop de Visão Computacional, WVC*, 2013.

[17] D. G. Patel, "Point pattern matching algorithm for recognition of 36 asl gestures," *International Journal of Science and Modern Engineering*, vol. 1, no. 7, June 2013.

[18] S. Silva, W. R. Schwartz, and G. Camara-Chavez, "Spatial pyramid matching for finger spelling recognition in intensity images," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2014, pp. 629–636.

[19] S.-Z. Li, B. Yu, W. Wu, S.-Z. Su, and R.-R. Ji, "Feature learning based on sae–pca network for human gesture recognition in rgbd images," *Neurocomputing*, vol. 151, pp. 565–573, 2015.

[20] R. B. Nicolas Pugeault, "ASL finger spelling dataset," http://personal.ee.surrey.ac.uk/Personal/N.Pugeault/index.php?section=FingerSpellingDataset, last visit: April 29, 2013.

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[22] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.