Finger Spelling Recognition using Kernel Descriptors and Depth Images

K. Otiniano-Rodríguez, E. Cayllahua-Cahuina, A. Araújo de A.
Department of Computer Science
Federal University of Minas Gerais
Belo Horizonte, Brazil
Email: karlaotiniano@ufmg.br,ecayllahua@ufmg.br,arnaldo@dcc.ufmg.br

G. Cámara-Chávez
Department of Computer Science
Federal University of Ouro Preto
Ouro Preto, Brazil
Email: gcamarac@gmail.com

***Abstract*—Deaf people use systems of communication based on sign language and finger spelling. Finger spelling is a system where each letter of the alphabet is represented by a unique and discrete movement of the hand. RGB and depth images can be used to characterize hand shapes corresponding to letters of the alphabet. There exists an advantage of depth sensors, as Kinect, over color cameras for finger spelling recognition: depth images provide 3D information of the hand. In this paper, we propose a model for finger spelling recognition based on depth information using kernel descriptors, consisting of four stages. The performance of this approach is evaluated on a dataset of real images of the American Sign Language finger spelling. Different experiments were performed using a combination of both descriptors over depth information. Our approach obtains 92.92% of mean accuracy with 50% of samples for training; outperforming other state-of-the-art methods.**

***Keywords*-Finger spelling; Depth images; Kernel descriptors; Bag-of-Words.**

I. Introduction

Sign language and finger spelling systems are used by deaf people to communicate with each other. In sign language, the basic units are composed by a finite set of hand configurations, spatial locations and movements. Their complex spatial grammars are remarkably different from the grammars of spoken languages [1], [2].

Finger spelling is a system where each letter of the alphabet is represented by a unique and discrete movement of the hand. Finger spelling integrates a sign language due to many reasons: when a concept lacks a specific sign, for proper nouns, for loan signs (signs borrowed from other languages) or when a sign is ambiguous [3]. Each sign language has its own finger spelling, which may be similar to others of different languages.

At present, RGB and depth images can be used to characterize hand shapes corresponding to finger spelling system. However, there exists an advantage of depth sensors, over color cameras for finger spelling recognition: the depth maps provide 3D information of the hand. The best known sensor that captures this type of information is the Kinect™.

In this paper, we propose a model for finger spelling recognition using depth images. Motivated by the performance of kernel based features, due to its simplicity and the ability to turn any type of pixel attribute into patch-level features, we decided to use the gradient and Local Binary Pattern (LBP) kernel descriptors [4]. The experiments are performed using a public database composed of 60,000 depth images stating 24 symbols classes [5]. The obtained results show that the accuracy obtained by our method, using both descriptors, is greater than using each descriptor separately. Moreover, the accuracy obtained by the proposed method performs better than the methods proposed in [6], [7], [8], [9]. The results show that our method is promising.

Contributions: Unlike other methods that are based on depth information and RGB, this paper proposes an approach using only depth images. Thus, avoiding the use and storing of RGB images and the problems of aligning the kinect sensor. Also, we propose an algorithm to segment the hand based on depth information. In our experiments, we compare our proposed model to others models of state-of-the-art showing that our model has a higher performance relative to the other models.

A. Related work

Through the years, several techniques have been developed to achieve an adequate recognition rate of sign language. With the advance of technology, methods have been proposed in order to improve the data acquisition, processing or classification. In the case of image acquisition, there are three main approaches[7]: sensor-based, vision-based and hybrid systems using a combination of these systems. Sensor-based methods use sensory gloves and motion trackers to detect hand shapes and body movements. Vision-based methods, that use standard cameras, image processing, and feature extraction, are used for capturing and classifying hand shapes and body movements. Hybrid systems use information from vision-based camera and other type of sensors like infrared depth sensors.

These methods have some disadvantages, sensor-based methods, such as data gloves, restrict the natural movement of hands and are often very expensive. Video-based methods are less restrictive, but to locate the hands and segmenting them is a non-trivial task over RGB images. However, depth cameras have become popular at a commodity price. Segmenting the hand from the background using depth information makes the task much easier, as used in [10], [11], [12], [13].

Recently, the computer vision community has shown a great interest on depth cameras, due to their success in many

applications, such as pose estimation [14], [15], tracking [16], object recognition [16], etc. Depth cameras were also used for hand gesture recognition [6], [17], [18], [8], [7]. Uebersax et al. [17] present a system for recognizing letter and finger spelled words. Issacs & Foo [19] proposed an American Sign Language (ASL) finger spelling recognition system based on neural networks applied to wavelets features. Bergh & Van Gool [20] propose a method based on a concatenation of depth and color-segmented images, using a combination of Haar wavelets and neural networks for six hand poses recognition of a single user.

Pugeault & Bowden [6] use a Kinect™ device to collect RGB and depth images. This dataset is being used by several approaches for testing these models [6], [8], [7], [9]. Pugeault & Bowden [6] also proposed a model that consists in extracting features using Gabor filters and then a Random Forest predicts the letters from the ASL finger spelling alphabet. The model proposed by Estrela *et al.* [9] is based on the bag of features strategy combined with the Partial Least Squares (PLS) technique in order to create models of the letters in the manual alphabet. Otiniano & Cámara [7] proposed a method based on gradient kernel descriptor using intensity and depth information. Huynh [21] proposed a novel efficient LBP-based descriptor, specialized to encode the facial depth information.

B. Organization of the paper

The remainder of this paper is organized as follows. In Section II, our proposed model is introduced and detailed. The experiments are presented in Section III, where the results are discussed. Finally, conclusion and future work are presented in Section IV.

II. Proposed Model

This section describes the methodology developed to perform finger spelling recognition from depth information. The proposed model consists of four stages as shown in Figure 1. In the first stage, the hand area is segmented from background using the depth information from Kinect™ sensor and precise hand shape is extracted. The second stage consists in extracting the features from depth images. The gradient and the LBP kernel descriptors are used on the depth image in this step. The Gradient kernel descriptor captures image variations, it consists of three kernels: the normalized linear kernel weighs the contribution of each pixel using gradient magnitudes, an orientation kernel computes the similarity of gradient orientations and finally a position Gaussian kernel measures how close two pixels are spatially. The LBP descriptor is a local binary kernel descriptor that captures local shape information by using binary patterns. The third stage consists in capturing the semantic information. In order to do this, the Bag-of-Visual-Words (BoVW) model is applied. Finally, in the fourth stage, these features or histograms are used as input to our SVM classifier, obtaining the final results.

A. Segmentation

In order to segment the hand area from the background, the depth values corresponding to the hand need to be detected. Depth values corresponding to the hand are usually the smallest, this means that they are closer to the sensor (foreground). To detect these values and therefore obtain the hand, a clustering algorithm is used over the depth values. However, it is important to highlight that the detected five groups can not be too dispersed. Also, a small group will not always have all the hand values. To overcome these problems, we propose an algorithm based on [22]. We consider that groups that are too distant do not belong to the same hand. Thus, by using a threshold d_t, we make certain that only contiguous groups that possibly belong to the hand are labeled as foreground. The proposed algorithm can easily perform the hand segmentation using depth values:

1) Define a threshold d_t.
2) Cluster the depth values in *n* groups (clusters).
3) Sort the *n* clusters from closest to farthest. Let $d_{(1,2)}, d_{(2,3)}, ..., d_{(n-1,n)}$ be the distance between the center of consecutive clusters.
4) Label the closest cluster as FG (foreground).
5) For $i \in \{2, 3, 4...n\}$, if $d_{(i-1,i)} < d_t$, the *i*-th cluster is labeled as FG. Otherwise finish.

B. Feature Extraction

The feature extraction process is performed by applying gradient and LBP kernel descriptors.

1) Gradient kernel descriptor: In [23], a kernel descriptor that extracts the low-level image features is designed. It consists of three steps: a match kernel using some pixel attribute, learn compact basis vectors using Kernel Principal Component Analysis (KPCA) and construct kernel descriptor by projecting the infinite-dimensional feature vector to the learned basis vectors. The authors proposed three types of effective kernel descriptors using gradient, color and shape pixel attributes. Later, in [4], the gradient kernel descriptor is applied to depth images. Thereby, in order to capture edge cues in depth images, we used the gradient match kernel, K_{grad}:

$$K_{grad}(P,Q) = \sum_{p \in P} \sum_{q \in Q} \tilde{m}(p)\tilde{m}(q) k_o(\tilde{\theta}(p), \tilde{\theta}(q)) k_s(p,q) \tag{1}$$

The normalized linear kernel $\tilde{m}(p)\tilde{m}(q)$ weighs the contribution of each gradient where $\tilde{m}(p) = m(p)/\sqrt{\sum_{p \in P} m(p)^2 + \varepsilon_g}$ and ε_g is a small positive constant to ensure that the denominator is larger than 0 and $m(p)$ is the magnitude of the depth gradient at a pixel p. Then, $k_o(\tilde{\theta}(p), \tilde{\theta}(q)) = exp(-\gamma_o \|\tilde{\theta}(p) - \tilde{\theta}(q)\|^2)$ is a Gaussian kernel over orientations. The authors [23] suggest to set $\gamma_o = 5$. To estimate the difference between orientations at pixels p and q, we use the following normalized gradient vectors in the kernel function k_o:

$$\tilde{\theta}(p) = [sin(\theta(p))cos(\theta(p))]$$
$$\tilde{\theta}(q) = [sin(\theta(q))cos(\theta(q))]$$

where $\theta(p)$ is the orientation of the depth gradient at a pixel p. In Equation 2, a position Gaussian kernel, with p denoting

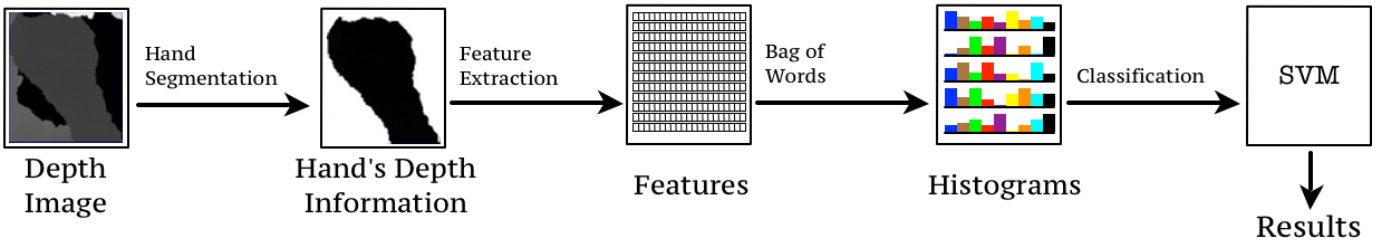

Fig. 1. Proposed model for finger spelling recognition.

the 2D position of a pixel in an image patch (normalized to [0,1]), measures how close two pixels are spatially. The value suggested for γ_s is 3.

$$k_s(p,q) = exp(-\gamma_s \|p-q\|^2) \tag{2}$$

To summarize, the gradient match kernel K_{grad} consists of three kernels: the normalized linear kernel weighs the contribution of each pixel using gradient magnitudes; the orientation kernel k_o computes the similarity of gradient orientations; and the position Gaussian kernel k_s measures how close two pixels are spatially.

Match kernels provide a principled way to measure the similarity of image patches, but evaluating kernels can be computationally expensive when image patch are large [23]. The corresponding kernel descriptor can be extracted from this match kernel by projecting the infinite-dimensional feature vector to a set of finite basis vectors, which are the edge features that we use in the next steps. For more details, the approach that extracts the compact low-dimensional features from match kernels is found in [23].

2) LBP: Local binary patterns: Let I be a depth image where P and Q are depth image patches from I, both $p \in P$ and $q \in Q$ are 2D position of a pixel in their corresponding depth image patch P and Q [23].

To extract the local binary patterns, a 3×3 patch around pixel p is used. Then, all values greater than p in the patch are marked with 1 and lower values with 0. The eight resulting values produce an 8-dimensional vector that represent the local binary pattern b_p.

The Gaussian Kernel defined in Equation 3 is used to measure the shape similarity between two local binary patterns b_p and b_q .

$$k_b(b_p,b_q) = exp(-\gamma_b \|b_p - b_q\|^2) \tag{3}$$

Using Equations 2 and 3, a local binary kernel descriptor k_{lbp}, is defined in Equation 4. This kernel has been proved to effectively capture local shape information [23].

$$K_{lbp}(P,Q) = \sum_{p\in P}\sum_{q\in Q} \widetilde{s}_p \widetilde{s}_q k_b(b_p,b_q) k_s(p,q) \tag{4}$$

where $\widetilde{s}_p = s_p / \sqrt{\sum_{p\in P} s_p^2 + \varepsilon_{lbp}}$.

s_p is computed by calculating the standard deviation of all the pixel values in a 3×3 neighborhood around pixel p. To make certain that the denominator will always be larger than zero, a small value is assigned to ε_{lbp}. Consequently, $\widetilde{s}_p \widetilde{s}_q$ is a normalized linear kernel that counterbalance the contribution of each local binary pattern.

C. Bag-of-Visual-Words

Bag-of-Visual-Words algorithm was first introduced by Sivic [24] for video retrieval. Due to its efficiency and effectiveness, it became very popular in the fields of image retrieval and categorization. Image categorization techniques rely either on unsupervised or supervised learning.

Our model uses the Bag-of-Visual-Words approach in order to search semantic information. The original method works with documents and words. Therefore, we consider an image as a document and the "words" will be the visual entities found in the image. The Bag-of-Visual-Words approach consists of three operations: feature description, visual word vocabulary generation and histogram generation. The feature description operation was explained in the previous subsection, now we will explain the remaining two steps, vocabulary and histogram generation.

1) Vocabulary Generation: A visual word vocabulary (codebook) is generated from feature vectors, where each visual word (codeword) represents a group of several similar features. The codebook defines a space of all entities occurring in the image.

2) Histogram Generation: A histogram of visual words is created by counting the occurrence of each codeword. These occurrences are counted and arranged in a vector. Each vector represents the features for an image.

D. Classification

Support vector machines (SVMs), introduced as a machine learning method by Cortes and Vapnik [25], 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. The SVMs have been developed as a robust tool for classification and regression in noisy and complex domains. SVMs can be used to extract valuable information from data sets and construct fast classification algorithms for massive data.

An important characteristic of the SVM classifier is to allow a non-linear classification without requiring explicitly a non-linear algorithm thanks to kernel theory.

Fig. 2. ASL Finger Spelling Dataset: 24 static signs by five users. It is an example of the variety of the dataset. This array shows one image from each user and from each letter.

In kernel framework, data points may be mapped into a higher dimensional feature space, where a separating hyperplane can be found. We can avoid to explicitly computing the mapping using the kernel trick which evaluates similarities between data $K(d_t, d_s)$ in the input space. Common kernel functions are: linear, polynomial, Radial Basis Function (RBF), χ^2 distance and triangular.

III. Experiments

In order to test our model, we use the ASL Finger Spelling Dataset [5], which contains 500 samples for each of 24 signs, recorded from 5 different persons (non-native to sign language), amounting to a total of 60,000 samples. Each sample has a RGB image and a depth image, but we only use a depth image, leaving aside the RGB image. Our proposed model works with static signs, therefore, the sign J and Z are not used (because these signs have motion). It is important to highlight that the dataset has a variety of background and viewing angles as we can see in Figure 2. It is possible to see the multiple variations in size, background and orientation.

Figure 3 shows the most similar signs *a*, *e*, *m*, *n*, *s* and *t*. The examples were taken from the same user, however, it is possible to see the variation in the background and the high similarity of the signs. All are represented by a closed fist, and differ only by the thumb position, leading to higher confusion levels. Consequently, these signs are the most difficult to differentiate in the classification task, obtaining the lowest accuracy values which will show later.

In order to test our proposed model, we conducted three experiments. In the first, a classification of the signs was performed using depth image features from the gradient descriptor. In the second, a classification was performed using the depth image features from the LBP descriptor. Finally, in the third experiment, the signs were classified using the features extracted from both: gradient and LBP descriptors.

For each of the three experiments, we have some specifications:

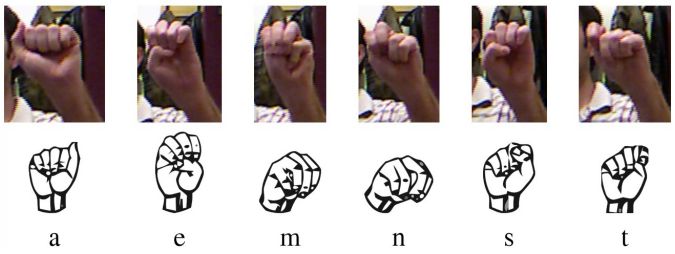

Fig. 3. Most conflictive similar signs in the dataset.

- In the segmentation stage, we use a threshold $d_t = 100$ and cluster the depth values in five groups, these values were obtained empirically.
- To extract all low level features using gradient and LBP kernel descriptors, approximately 12x13 patches are used over dense regular grid with spacing of eight pixels (images are not of uniform size), each patch has a size of 16x16.
- In order to produce the visual word vocabulary, the LBG (Linde-Buzo-Gray) [26] algorithm was used to detect one hundred clusters by taking a sample of 20% from the total features.
- Moreover, in the classification stage, we use a RBF kernel, whose values for *g* (gamma) and *c* (cost) are 0.25 and 5, respectively. We also use different percentages of samples for training and testing. For example, we use 10% of samples for training and the other 90% are used for testing, and this percentage varies up to 50% for training. In order to obtain more precise results, each experiment was performed 20 times and we show the mean accuracy for each one. The library LIBSVM (a library for Support Vector Machines)] [27] was used in our implementation.

First experiment: Gradient kernel descriptor: For this first experiment, an average accuracy of 85.18% was obtained

TABLE IV
ACCURACIES AND STANDARD DEVIATION OF THE CLASSIFICATION USING BOTH DESCRIPTORS.

% Training	% Testing	Accuracy	Standard Deviation
10	90	84.2%	0.34
20	80	88.81%	0.17
30	70	90.67%	0.14
40	60	92.11%	0.18
50	50	92.92%	0.16

when 50% of samples are used for training and 50% for testing. This accuracy is the mean of the values of the main diagonal of the confusion matrix and represents the signs correctly classified (true positives). This accuracy decreases when less samples are used for training. For example, when 10% of samples are used for training we obtain 75.13% of accuracy. All the results of this classification, with 50% of samples for training, are found in Table I. In this table, we can see that the signs *t* (68%), *n* (74%), *r* (76%), *s* (76%) and *m* (77%) are the most difficult to recognize. This happens because these signs are defined as the most conflictive similar signs, refer to Figure 3. Nonetheless, *a* and *e* signs, which are also conflictive similar signs, obtained a high accuracy, 89% and 82%, respectively, compared to the other conflictive signs. On the other hand, signs *b* and *l* obtained the highest accuracies, 95% for both.

Second experiment: LBP descriptor: The second experiment has an average accuracy of 86.53% and 76.68% when 50% and 10% of samples are used for training, respectively. Table II shows the results of this classification using 50% of samples for training. In this table, as in the previous experiment, we can see that the signs *t* (71%), *s* (74%), *n* (76%), *r* (78%) and *m* (78%) have the lowest values of recognition. Signs *a* and *e*, which are part of conflictive similar signs, obtained a high accuracy, 93% and 86%, respectively, compared to the other conflictive signs. Moreover, signs *b* and *l* obtained the highest accuracies, 96% and 95% respectively.

Third experiment: Both descriptors: The last experiment tests our method using both descriptors over depth information. The data was obtained by joining the features (histograms) from gradient and LBP descriptors, which were used independently in the previous experiments. An average accuracy of 92.92% and 84.2% is obtained when 50% and 10% of samples are used for training, respectively. Table III shows the results for each sign using 50% of samples for training. It is possible to see that we obtain signs with average accuracies of 98% (*b*) and 99% (*f*, *w* and *l*). On the contrary, signs *t* (82%), *n* (84%), *r* (86%) and *m* (88%) have the lowest values of recognition. Nevertheless, compared to previous experiments, the recognition rate of each sign has improved. These higher recognition rates corroborates the effectiveness of our proposed model.

In Table IV, we show the average accuracies for each experiment using different percentages of samples for training.

In order to compare the results, we summarize them in Tables V and VI. Both tables present the average accuracy and standard deviation for each experiment. We can see that using both descriptors we obtain the highest average accuracy, outperforming our isolate methods and also other methods proposed by Pugeault & Bowden [6], Otiniano & Cámara [7], Zhu & Wong [8] and Estrela *et al.* [9]. These last four methods are found in the state-of-the-art and use the same dataset, the principal difference between these methods is the number of samples used for training and the type of image (RGB, depth or both).

TABLE V
ACCURACIES AND STANDARD DEVIATION OF THE THREE EXPERIMENTS AND STATE-OF-THE-ART USING 10% OF SAMPLES FOR TRAINING.

Method	Accuracy	Standard Deviation
Gradient	75.13%	0.22
LBP	76.68%	0.17
G-LBP	84.2%	0.34
Zhu & Wong [8]	77.39%	0.13
Estrela *et al.* [9]	71.51%	-

TABLE VI
ACCURACIES AND STANDARD DEVIATION OF THE THREE EXPERIMENTS AND STATE-OF-THE-ART USING 50% OF SAMPLES FOR TRAINING.

Method	Accuracy	Standard Deviation
Gradient	85.18%	0.16
LBP	86.53%	0.17
G-LBP	92.92%	0.16
Pugeault & Bowden [6]	75.00%	-
Otiniano & Cámara [7]	91.26%	0.18

Zhu & Wong [8] and Estrela *et al.*[9] used approximately 10% of samples for training and RGB and depth images. In Table V, we present the results of these models and the results from our model also using 10% of samples for the training phase.

Pugeault & Bowden [6] and Otiniano & Cámara [7] used 50% of samples for training and they work with RGB and depth images. The results of these models are compared to our model in Table VI. We present the accuracies for each experiment of our model also using 50% of samples for training.

Finally, we highlight that our method outperforms the results obtained by the state-of-the-art. Furthermore, our method used only depth information while the other models presented above used RGB and depth information.

IV. CONCLUSION

In this paper, we proposed a method for Finger Spelling Recognition from depth information using kernel descriptors. The segmentation was performed based on depth maps. The Bag-of-Visual-Words model was applied in order to search for semantic information. Finally, the classification task is performed by a SVM classifier.

The combination of gradient kernel and LBP descriptors obtained the best results (92.92%) with a low variance.

TABLE I

CONFUSION MATRIX OF THE CLASSIFICATION OF 24 SIGNS USING GRADIENT KERNEL DESCRIPTOR.

	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	0.89	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.05	0.00	0.00	0.00	0.01	0.00
b	0.01	0.95	0.01	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	0.00	0.01	0.00	0.00	0.00	0.00
c	0.01	0.00	0.92	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
d	0.00	0.00	0.00	0.87	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.02	0.00	0.00	0.02	0.00	0.01	0.01	0.00	0.00	0.01	0.00
e	0.02	0.00	0.02	0.01	0.82	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.04	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.01	0.00
f	0.00	0.02	0.01	0.01	0.00	0.94	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
g	0.01	0.00	0.00	0.00	0.00	0.00	0.86	0.09	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	0.00	0.00	0.00
h	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.92	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	0.01	0.01	0.00	0.00	0.00	0.94	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	0.01
k	0.00	0.00	0.00	0.04	0.00	0.00	0.02	0.00	0.00	0.84	0.01	0.00	0.00	0.00	0.01	0.00	0.02	0.00	0.01	0.01	0.01	0.00	0.01	0.00
l	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.95	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
m	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.77	0.07	0.01	0.00	0.00	0.01	0.05	0.04	0.00	0.00	0.00	0.00	0.00
n	0.02	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.07	0.74	0.02	0.00	0.00	0.00	0.02	0.07	0.00	0.00	0.00	0.02	0.00
o	0.01	0.00	0.02	0.01	0.06	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.01	0.78	0.01	0.01	0.00	0.02	0.01	0.00	0.00	0.00	0.01	0.00
p	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	0.02	0.88	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
q	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.06	0.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
r	0.01	0.01	0.00	0.06	0.00	0.01	0.00	0.00	0.00	0.03	0.01	0.01	0.00	0.00	0.00	0.00	0.76	0.00	0.01	0.05	0.03	0.00	0.01	0.00
s	0.03	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.01	0.00	0.05	0.03	0.03	0.00	0.00	0.00	0.76	0.04	0.00	0.00	0.00	0.01	0.00
t	0.06	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.08	0.02	0.00	0.00	0.00	0.05	0.68	0.00	0.00	0.00	0.02	0.00
u	0.00	0.01	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.82	0.03	0.02	0.01	0.00
v	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.03	0.00	0.00	0.03	0.83	0.04	0.00	0.00
w	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.04	0.91	0.00	0.00
x	0.01	0.00	0.01	0.03	0.01	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.03	0.01	0.01	0.01	0.02	0.01	0.02	0.01	0.01	0.00	0.78	0.00
y	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	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	0.01	0.94

TABLE II

CONFUSION MATRIX OF THE CLASSIFICATION OF 24 SIGNS USING LBP DESCRIPTOR.

	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	0.93	0.01	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	0.01	0.03	0.00	0.00	0.00	0.00	0.00
b	0.01	0.96	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	0.00	0.00	0.00	0.00	0.00	0.00
c	0.01	0.00	0.93	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
d	0.00	0.00	0.01	0.89	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.01	0.00
e	0.02	0.00	0.02	0.01	0.86	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.02	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
f	0.00	0.02	0.01	0.01	0.01	0.94	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
g	0.01	0.00	0.00	0.00	0.00	0.00	0.91	0.05	0.00	0.01	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
h	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.92	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	0.00
i	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.94	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01
k	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.00	0.00	0.84	0.01	0.00	0.00	0.00	0.01	0.00	0.03	0.00	0.00	0.01	0.02	0.00	0.01	0.00
l	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.95	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
m	0.01	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.01	0.01	0.00	0.78	0.07	0.01	0.00	0.00	0.01	0.04	0.03	0.00	0.00	0.00	0.00	0.00
n	0.01	0.00	0.01	0.01	0.02	0.00	0.00	0.00	0.00	0.01	0.01	0.07	0.76	0.00	0.00	0.00	0.00	0.02	0.07	0.00	0.00	0.00	0.01	0.00
o	0.00	0.00	0.02	0.05	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.82	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
p	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.88	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01
q	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.09	0.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
r	0.01	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.78	0.01	0.01	0.05	0.01	0.00	0.01	0.01
s	0.03	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.04	0.03	0.02	0.00	0.00	0.00	0.74	0.07	0.00	0.00	0.00	0.00	0.01
t	0.06	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.03	0.07	0.02	0.00	0.00	0.00	0.02	0.71	0.00	0.00	0.00	0.02	0.00
u	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.84	0.02	0.00	0.01	0.00
v	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.02	0.85	0.04	0.00	0.00
w	0.00	0.01	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	0.01	0.00	0.00	0.01	0.05	0.91	0.00	0.00
x	0.01	0.00	0.01	0.04	0.00	0.01	0.00	0.00	0.00	0.02	0.01	0.00	0.02	0.00	0.03	0.01	0.01	0.00	0.02	0.01	0.00	0.00	0.78	0.01
y	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.92

TABLE III
CONFUSION MATRIX OF THE CLASSIFICATION OF 24 SIGN USING BOTH DESCRIPTORS WITH 50% FOR TRAINING.

	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	0.96	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	0.02	0.00	0.00	0.00	0.00	0.00
b	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	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
c	0.00	0.00	0.96	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	0.00	0.00	0.00
d	0.00	0.00	0.00	0.96	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	0.00	0.00	0.00
e	0.01	0.00	0.01	0.00	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
f	0.00	0.01	0.00	0.00	0.00	0.98	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	0.96	0.02	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	0.04	0.96	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	0.97	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
k	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	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	0.98	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	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.88	0.04	0.00	0.00	0.00	0.00	0.03	0.02	0.00	0.00	0.00	0.00	0.00
n	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.84	0.01	0.00	0.00	0.00	0.01	0.05	0.00	0.00	0.00	0.00	0.00
o	0.00	0.00	0.01	0.01	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.90	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.00
p	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	0.94	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
q	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	0.03	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
r	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.86	0.00	0.01	0.04	0.02	0.00	0.00	0.00
s	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.02	0.00	0.00	0.00	0.88	0.03	0.00	0.00	0.00	0.00	0.00
t	0.02	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.06	0.01	0.00	0.00	0.00	0.03	0.82	0.00	0.00	0.00	0.01	0.00
u	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.91	0.02	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	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.92	0.02	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	0.01	0.98	0.00	0.00
x	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.90	0.00
y	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	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	0.97

Although conflictive similar signs like, *m*, *n*, *r* and *t*, are the most difficult to recognize, our proposed model achieved over 82% accuracy, thus validating the robustness of the descriptors. Furthermore, for other conflictive similar signs, *a* and *e*, our model achieved accuracies of 96% and 94% respectively.

Unlike other descriptors, the kernel descriptors used in our model have the advantage that can be directly applied to depth images without having to compute the cloud of points, consequently, reducing the computation time. The extracted features effectively captured image variations and shape information.

As future work, we pretend to test other descriptors in depth and intensity images. We also intend to extend our method to recognize dynamic signs.

ACKNOWLEDGMENT

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

REFERENCES

[1] LIBRAS, "Brazilian sign language," http://www.libras.org.br/, last visit: March 10, 2012.

[2] P. W. Vamplew, "Recognition of sign language gestures using neural networks," *Australian Journal of Intelligent Information Processing Systems*, vol. 5, pp. 27–33, 1996.

[3] 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.

[4] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 821–826.

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

[6] 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.

[7] K. Otiniano-Rodríguez and G. Cámara-Chávez, "Finger spelling recognition from RGB-D information using kernel descriptor," in *Proceedings of the SIBGRAPI 2013 (XXVI Conference on Graphics, Patterns and Images)*, 2013.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] G. Fanelli, J. Gall, and L. V. Gool, "Real time head pose estimation with random regression forests," in *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR), June 2011, pp. 617–624.

[15] J. Shotton, T. Sharp, A. Kipman, A. W. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[16] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011, pp. 101.1–101.11.

[17] 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.

[18] 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.

[19] J. Isaacs and S. Foo, "Hand pose estimation for american sign language recognition," *36th Southeastern Symposium on System Theory*, pp. 132–136, 2004.

[20] 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.

[21] T. Huynh, R. Min, and J.-L. Dugelay, "An efficient lbp-based descriptor for facial depth images applied to gender recognition using rgb-d face data," in *Computer Vision-ACCV 2012 Workshops*. Springer, 2013, pp. 133–145.

[22] F. Prada, L. Cruz, and L. Velho, "Object extraction in rgbd images," in *Conference on Graphics, Patterns and Images. Ouro Preto, Brazil*, 2012.

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

[24] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 1470–1477.

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

[26] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.