Article

Hand Posture Recognition Using Skeletal Data and Distance Descriptor

Tomasz Kapuściński * and Dawid Warchol

Faculty of Electrical and Computer Engineering, Department of Computer and Control Engineering, Rzeszów University of Technology, W. Pola 2, 35-959 Rzeszów, Poland; dawwar@kia.prz.edu.pl

* Correspondence: tomekkap@kia.prz.edu.pl; Tel.: +48-17-865-1614

Received: 27 February 2020; Accepted: 18 March 2020; Published: 20 March 2020

Abstract: In this paper, a method for the recognition of static hand postures based on skeletal data was presented. A novel descriptor was proposed. It encodes information about distances between particular hand points. Five different classifiers were tested, including four common methods and a proposed modification of nearest neighbor classifier, which can distinguish between posture classes differing mostly in hand orientation. The experiments were performed using three challenging datasets of gestures from Polish and American Sign Languages. The proposed method was compared with other approaches found in the literature. It outperforms every compared method, including our previous work, in terms of recognition rate.

Keywords: hand posture recognition; sign language; finger alphabet; skeletal data; leap motion

1. Introduction

Automatic hand posture recognition is an important research topic in computer science [1]. The overall goal is to understand the body language and then create more functional and efficient human–computer interfaces. Application areas are vast; from driver support via hand-controlled cockpit elements [2], through to home automation with consumer electronics driven by gestures [3], gaming industry applications [4], interaction with virtual objects [5], and finally, technological support for people with disabilities [6]. Solutions available on the market are either limited, very simple, or have background and illumination requirements that are difficult to meet in real-life scenarios. Therefore, it is desirable to continue research on the automatic interpretation of hand gestures.

Available approaches can be divided into two groups: (i) using special gloves with sensors [7] or cameras and (ii) computer vision methods [8]. Vision-based solutions seem to be more attractive because they are more comfortable, do not require any additional equipment limiting the user's freedom of movement, mimic natural interaction, and avoid stigmatization. However, building a reliable vision-based system is quite a challenge. Color-based methods, frequently used to segment hands, fail in the cases of complex backgrounds containing other skin-colored objects or users wearing short-sleeved clothing [9]. It is also very difficult to achieve the color constancy under varying scene illumination. Fortunately, new depth-sensing devices recently appeared in the market. They combine the visible and near-infrared part of the spectrum to obtain good quality depth maps. Some of them, based on time-of-flight principle, can even work in a completely dark room.

Therefore, in recent literature on hand gestures recognition, a shift to depth modality has been observed [10]. The new devices can acquire good quality 3D data, which can be then used to extract the hand skeleton containing information about the spatial configuration of bones corresponding to fingers. The main advantage of skeletal data—compared to images, point clouds, and depth maps—is its small size. The features calculated based on skeletons can also be a good addition to typical image-based or depth-based features. We can also expect that there will be more and more accurate devices providing

this type of data. Therefore, there is a need to develop new recognition algorithms based on hand skeletons.

In this paper, a problem of hand posture recognition based on skeletal data extracted by a depth sensor was tackled. The method is based on a novel hand descriptor combined with another one that was previously developed. The experimental tests were performed using four different classification methods and a proposed modification of the nearest neighbor classifier.

The main contributions of this paper are as follows:

1. The novel hand descriptor encoding information about distances between selected hand points;
2. The modified nearest neighbor classifier, suitable for posture recognition in the case where some of the classes differ mostly in hand orientation;
3. Experimental verification of the proposed methods using challenging datasets.

The remaining parts of this paper are organized as follows. The related works are characterized in Section 2. The proposed hand posture recognition method is presented in Section 3. Section 4 discusses the used datasets and performed experiments. Section 5 concludes the paper.

2. Related Work

One of the devices that can be used to obtain skeletal data for hands is the Leap Motion (LM) sensor [11]. In literature, there are several works devoted to the study of its usefulness for gesture recognition. In [12], the authors estimated the accuracy and repeatability of hand position measurement and found that this sensor outperforms competitive solutions with a similar price range. In [13], the sensor's usefulness for recognizing Australian Sign Language was assessed. Hand shapes for which the sensor does not work were identified. The authors concluded that the solution has great potential, but requires refining the API. In [14], the usefulness of the device for hand tracking was assessed. The authors noticed that further development of the sensor is needed to implement professional systems.

It is expected that sensors for the acquisition of hands skeletal data will be improved soon. Therefore, work is underway to use them to recognize sign languages: American [15–24], Arabic [25–28], Australian [13], Indian [29–31], Mexican [32], Pakistani [33], and Polish [34].

In [15], a subset of hand shapes from the American Finger Alphabet was recognized using features based on skeletal data: angle, distance, and elevation of the fingertips. The support vector machine (SVM) classifier was used. The recognition rate was 80.88%. After adding features obtained with the Kinect sensor—curvature and correlation—recognition efficiency increased to 91.28%.

The 26 letters of the American Finger Alphabet, shown by two people, were also recognized in [16]. The feature vector consisted of pinch strength, grab strength, average distance, spread and tri-spread between fingertips, determined from skeletal data. The recognition rate was 72.78% for the nearest neighbor (kNN) classifier and 79.83% for SVM.

In [25], 28 letters of the Arabic Finger Alphabet were recognized using 12 of 23 values measured by the LM sensor: finger length; finger width; average tip position with respect to x , y , and z -axis; hand sphere radius; palm position with respect to x , y , and z -axis; hand pitch, roll, and yaw. A 98% recognition rate was obtained for the Naive Bayes classifier, while it was 99% for Multilayer Perceptron.

Ten static hand shapes shown by 14 users were recognized in [18]. The following features were used: fingertip angles, distances, elevations, and positions. For the SVM classifier, the recognition rate was 81.5%. After adding features obtained from the Kinect's depth image, the recognition rate increased to 96.5%.

The Indian Finger Alphabet letters from A to Z and the numbers from 1 to 9, shown by ten users, were recognized in [29]. The feature vector consisted of the following distances: the fingertips—the middle of the palm, the index—the middle finger, the index—the ring finger, and the index—the little finger. For the kNN classifier, a recognition efficiency of 88.39% for the Euclidean metric and 90.32% for the cosine distance was obtained.

In [28], two LM controllers were used to prevent the individual fingers from being obstructed. The 28 letters of Arabic Finger Alphabet were recognized. The feature vector was a concatenation of the following features measured by two controllers: finger length; finger width; average tip position with respect to x , y , and z -axis; hand sphere radius; palm position with respect to x , y , and z -axis; hand pitch, roll, and yaw. The Linear Discriminant Analysis classifier was used. The recognition rate of 97.7% for a fusion of features and 97.1% for a fusion of classifiers was obtained.

Ten static hand shapes that can be used for rehabilitation after cerebral palsy were recognized in [35]. Features determined by the LM controller and three classification methods; decision tree, kNN, and SVM, were used. The obtained recognition rates of individual gestures ranged from 76.96% to 100%.

In [20], 26 letters of the American Finger Alphabet were recognized. The features measured by the LM controller and the Multilayer Perceptron (MLP) classifier were used. A recognition rate of 96.15% was achieved.

Forty-nine static gestures (twenty-six letters of the alphabet, ten numbers, and nine words) of Indian Sign Language were recognized in [31]. The feature vector was composed of the distances between the middle of the hand and the fingertips. The kNN classifier with four different measures of similarity: Euclidean distance measure, cosine similarity, Jaccard similarity, and Dice similarity was used. For gestures performed by ten people, recognition rates ranging from 83.11% to 90% were obtained.

In [27], forty-four static gestures (twenty-eight letters, ten numbers, and sixteen words) from the Arabic Sign Language were recognized. Two variants of the feature vector consisting of 85 and 70 scalar values measured by the LM controller were considered. The training set consisted of 200 performances of each gesture by two people. Tests were carried out on 200 executions of individual gestures by a third person. Three variants of the classifier were considered: SVM, kNN, and artificial neural network (ANN). The best recognition rate of 99% was obtained for the kNN classifier and the first considered feature vector variant.

Twenty-six letters and ten numbers of the American Finger Alphabet were recognized in [23]. Six different combinations of the following features were considered: standard deviation of palm position, palm curvature radius, the distance between the palm center and each fingertip, and the angle and distance between two adjacent fingertips. For gestures performed by twelve people and leave-one-subject-out protocol, the recognition rate was 72.79% for SVM and 88.79% for the deep network.

The 24 characters of the American Finger Alphabet shown by five people were recognized in [24]. Skeletal data in the form of angles between adjacent bones of the same finger and angles between adjacent fingers were used, as well as infrared images obtained with an LM sensor. For the classifier based on deep networks and leave-one-subject-out protocol, a recognition rate of 35.1% was obtained.

In [34], 48 static hand shapes from the Polish Finger Alphabet and Polish Sign Language were recognized. Gestures were shown 500 times by five users. Two different positions of the LM sensor and changes in the orientation of the hand were considered. Several classifiers, as well as their fusion, were tested. For the leave-one-subject-out protocol, the best recognition rate was 56.7%.

Twenty-four static gestures of the American Finger Alphabet, shown ten times by twelve people, were recognized in [22]. Features based on the skeletal data returned by the LM controller were used. The classification was carried out using hidden Markov models. For the leave-one-subject-out protocol, the recognition rate was 86.1%.

In [21], 12 dynamic and 18 static gestures of American Sign Language, shown by 20 people, were recognized. The feature vector was composed of the following: the internal angles of the joints between distal and intermediate phalanges, and intermediate and proximal phalanges; 3D displacements of the central point of the palm; 3D displacements of the fingertip positions; and the intrafinger angles. Recursive neural networks were used for the classification. The recognition rate was 96%.

The problem of fingers occlusion, occurring when performing gestures, was considered in [36]. For this purpose, three LM controllers were used. The feature vector consisted of a hand rotation matrix, y -component of the fingers directions, and distal phalanges rotation quaternion. Three classification methods available in the scikit-learn library were tested: logistic regression, SVC, and XGBClassifier. An 89.32% recognition rate was obtained for six selected hand gestures.

In [37], letters from the Israeli Sign Language (ISL) alphabet were recognized using the SVM classifier and feature vector consisting of the Euclidean distances between the fingertips and the center of the palm. The training dataset consisted of six letters performed 16 times by eight users. A system was able to translate fingerspelling into a written word with a recognition accuracy between 85%–92%.

The challenging problem of fistlike signs recognition, occurring in ASL, was described in [38]. In the proposed method, the area of several polygons defined by fingertip and palm positions, and estimated using the Shoelace formula, was used. The letters from the ASL alphabet were classified using the decision trees (DT). Seven letters performed 30 times by four persons were used as a training set. For 100 repetitions of each gesture by another user, the method achieved the recognition accuracy of 96.1%.

In [39], static hand postures recognition for a humanlike robot hand was described. Ten digits from ASL were recognized using the multiclass-SVM classifier. The feature vector consisted of the distances between the palm position and each fingertip and the distance between fingertips. The method was validated using a test dataset composed of 2000 static posture samples with an accuracy of 98.25%.

In [40], five one-handed and five two-handed static gestures of Turkish Sign Language, performed three times by two users, were recognized using artificial neural networks, deep learning, and decision trees. The 3D positions of all bones in the skeletal hand model, measured by the LM controller, were used. The recognition accuracies between 93% and 100% were achieved, depending on the classifier and the number of features used.

In [41], a deep-learning-based method for skeleton-based hand gesture recognition was described. The network architecture consists of a convolutional layer for extracting features and a long, short-term memory layer for modeling the temporal dimension. Ten static and ten dynamic hand gestures performed 30 times were recognized with an accuracy of 99%.

Based on the literature review, the following conclusions can be drawn:

- Despite that the currently available devices for obtaining skeletal data are imperfect, we have recently observed a significant increase in interest of using this modality for gesture recognition. In the last two years, several new publications were recorded.
- Most authors do not provide data used in experiments, which makes verification and comparative analysis difficult. Only three datasets available in works [15,18,24,34] are known to the authors.
- Many cited works omit tests using the leave-one-subject-out protocol. These tests are more reliable because they show the method's dependence on the person performing gestures.
- Some of the proposed feature vectors use features directly measured by the sensor. They are not independent of the size of the hand.
- Some of the works relate to the recognition of dynamic gestures, in which the hand movement trajectory is a great help.

The problem of recognizing hand postures based on skeletal data has not been fully solved, and further work in this area is advisable.

3. Proposed Method

Skeletal data obtained using the Leap Motion sensor was used [42]. The feature vector was the concatenation of the Point Pair Descriptor (PPD) introduced in [34] and the Distance Descriptor (DD) proposed in this paper. Five different classifiers were tested. Four of them proved to be the best among the 18 tested in [34]. The fifth classifier is a novel modification of the kNN method proposed in this paper. Details are given in Sections 3.1–3.3.

3.1. Point Pair Descriptor

Let P_c be the palm center, n_c normal to the palm at point P_c , P_i the tip of the i -th finger, and n_i the vector pointed by that finger (Figure 1).

Figure 1. Point Pair Descriptor construction.

The relative position of vectors n_c and n_i can be described by three values according to Formulas (1)–(3) [43]:

$$\alpha_i = \arccos(v_i \cdot n_i), \quad (1)$$

$$\phi_i = \arccos\left(u \cdot \frac{d_i}{|d_i|}\right), \quad (2)$$

$$\Theta_i = \arctan\left(\frac{w_i \cdot n_i}{u \cdot n_i}\right), \quad (3)$$

where the vectors u , v_i , and w_i define the so-called Darboux frame [44]:

$$u = n_c, \quad (4)$$

$$v_i = \frac{d_i}{|d_i|} \times u, \quad (5)$$

$$w_i = u \times v_i, \quad (6)$$

and $\cdot$ denotes the scalar and $\times$ —vector products. The Point Pair Descriptor consists of 15 features calculated for each finger using Formulas (1)–(3):

$$V = [\alpha_1, \phi_1, \Theta_1, \alpha_2, \phi_2, \Theta_2, \alpha_3, \phi_3, \Theta_3, \alpha_4, \phi_4, \Theta_4, \alpha_5, \phi_5, \Theta_5]. \quad (7)$$

The features were normalized to the interval [0–1].

Features α and Θ can be interpreted as pan and yaw angles between vectors pointed by fingers and palm normals. Feature ϕ is an angle between vectors pointed by fingers and the line connecting this vector with the initial point of palm normal. PPD is an alternative to other angular-based features describing hand skeleton. Such features are most often angles corresponding to the orientation of each fingertip projected onto the palm plane [15,18], between lines connecting the palm center with the fingers [45], between adjacent bones in the same finger, or between adjacent fingers [24]. Unlike PPD descriptor, these features do not use normals to the palm calculated at its center to determine angular relations.

3.2. Distance Descriptor

The Distance Descriptor is a novel method proposed in this paper. It encodes information about distances between hand points corresponding to fingertips and the palm center. It uses only information about the positions of these points. Normal vectors and vectors pointed by the fingers are not required. The descriptor can be computed as follows.

1. For each point P_i :

1.1. Compute distances (using Euclidean or city block metric) between the other points P_j , $j \neq i$.

- 1.2. Sort points P_j by the calculated distances in ascending order.
- 1.3. Assign consecutive integer values a_{ij} , starting from one, to the sorted points P_j .
2. Create a feature vector consisting of integer values, assigned to the points P_j in step 1.3, and ordered as follows $[a_{12}, a_{13}, a_{14}, a_{15}, a_{21}, \dots, a_{65}]$.
3. Create a reduced feature vector by adding together integer values corresponding to the same pair of indices i, j : $[a_{12} + a_{21}, a_{13} + a_{31}, \dots, a_{56} + a_{65}]$.

The purpose of step 3 is not only to reduce the number of features. After this step, for each point P_i , the descriptor determines not only which of the remaining points P_j are its nearest neighbors but also which of the P_j points consider P_i as their nearest neighbor. Features of the Distance Descriptor are normalized to the interval [0–1] by dividing them by $2(n - 1)$, where $n = 6$ is the number of points.

DD is an alternative to other positional-based and distance-based features describing hand skeleton. Such features are simple 3D or 2D coordinates or distances between fingertips and palm center [15,18,39,40,45] or between fingertips and palm plane [15,18]. Some of them are normalized with respect to the hand position and orientation, however, such normalization is not fully accurate. Moreover, even after the normalization, these features—unlike DD descriptor—are scale dependent, making methods difficult for recognizing gestures performed by people with different hand sizes (especially in the case of child hands). Some of these features are often not distinctive enough to differentiate between similar hand postures. It is because they do not include positional relations between fingertips (only between each fingertip and palm or plane center), whereas DD features include relations between each fingertip and between fingertips and palm center.

The Matlab codes for Point Pair Descriptor and Distance Descriptor can be downloaded from our website [46].

3.3. Classification

Four classifiers were tested: support vector machine with linear kernel function (SVM-Lin) [47], linear discriminant (LD) [48], ensemble of multiple decision trees—tree bagger (TreeBag) [49], and weighted k -nearest neighbors classifier with $k = 10$ (10NN-W) [50]. The parameters of the tested classifiers are listed in Table 1. We chose parameters starting from default values of our classification tool. We then tried to change each of them and observed if the changes caused improvement of results.

Table 1. The parameters of the tested classifiers.

Classifier	Parameter	Value
SVM-Lin	kernel function	linear
	box constraint level	1
	multiclass method	one-vs-one
LD	covariance structure	full
TreeBag	number of learners	30
10NN-W	number of neighbors	10
	distance metric	Euclidean
	distance weight	squared inverse

Some of the recognized shapes differ only in the spatial orientation of the hand. Therefore, the following modification of the nearest neighbor classifier was proposed. When searching for the nearest neighbor, only those samples from the training set whose orientation is similar to the tested one are taken into account. The training sample is then marked as a potential nearest neighbor only if

$$\text{acos} \left(\frac{n_{cx} \cdot n_{ct}}{|n_{cx}| |n_{ct}|} \right) \leq \gamma_t, \quad (8)$$

where n_{cx} , n_{ct} are normals to the palm of the classified and training sample respectively, $\cdot$ denotes scalar vector product, and γ_t is the threshold. We named the proposed method the nearest neighbor classifier with orientation restriction (NNOR). It should be considered as one of the novelties proposed in this paper.

4. Experiments

4.1. Datasets

The experiments were performed using two datasets recorded by the Leap Motion sensor. Dataset 1 was introduced in [34] and can be downloaded from our website [46] (see Figure 2).

Figure 2. Hand postures of Dataset 1.

It consists of 48 hand posture classes from the Polish Finger Alphabet (PFA) and Polish Sign Language (PSL). Each gesture was performed 500 times by 5 people, which is 120,000 executions in total. During the recordings, the sensor was lying horizontally on the table.

To perform an additional evaluation of our method and compare the results with other works, two additional datasets were used: Dataset 2 provided by Marin et al. [18] and Dataset 3 provided by Tao et al. [24]. Dataset 2 consists of 10 posture classes corresponding to letters from American

Sign Language. Each gesture was performed 10 times by 14 people, which is 1400 executions in total. Dataset 3 consists of 24 posture classes corresponding to letters from American Sign Language. Each gesture was performed 450 times by 5 people, which is 54,000 executions in total.

Gestures from each dataset are represented by the coordinates of fingertips, the palm center, the vector normal to the palm and the vectors coinciding with the fingers pointing direction.

4.2. Results

The results of 10-fold cross-validation obtained for Dataset 1 are shown in Table 2.

Table 2. Results of 10-fold cross-validation obtained for Dataset 1.

Features	SVM-Lin	LD	TreeBag	10NN-W
PPD + DD (Euclidean)	99.7	82.5	100.0	100.0
PPD + DD (city block)	99.9	86.0	100.0	100.0

The proposed method achieved 100% accuracy in the case of weighted k -nearest neighbors and tree bagger classifiers. However, the results of leave-one-subject-out, 5-fold cross-validation, presented in Table 3, are significantly worse.

Table 3. Results of leave-one-subject-out, 5-fold cross-validation tests obtained for the Dataset 1.

Features	SVM-Lin	LD	TreeBag	10NN-W
PPD + DD (Euclidean)	57.9	58.5	55.7	54.8
PPD + DD (city block)	55.7	57.0	55.8	49.6

Some posture classes from Dataset 1—e.g., 2, 100, TM, or H, U, or N, Nw—differ only in hand orientation. All of the used classifiers were unable to distinguish between them. Therefore, the NNOR classifier was proposed and tested in two variants: with Euclidean (NNOR-Euc) and city block (NNOR-CB) distance. γ_t was experimentally set to 35 degrees. The results of leave-one-subject-out validation, obtained for Dataset 1, are shown in Table 4. The best accuracy, 63.9%, is 5.4% higher than in the case of previously tested classifiers.

Table 4. Results of leave-one-subject-out, 5-fold cross-validation tests obtained for Dataset 1 using nearest neighbor classifier with orientation restriction (NNOR).

Features	NNOR-Euc	NNOR-CB
PPD + DD (Euclidean)	63.9	63.9
PPD + DD (city block)	56.7	58.1

4.3. Comparison with Other Works

Table 5 presents the comparison of the best results obtained by the method proposed in this paper and by the method from our previous work. The new method outperforms the previously proposed algorithm by 11.6%, which confirms the usefulness of the novel descriptor DD and the novel classifier NNOR.

Table 5. Comparison of leave-one-subject-out, 5-fold cross-validation test results obtained for Dataset 1.

Reference	Features	Classifier	Recognition Rate
previous work [34]	PPD	SVM-Lin	52.3
current work	PPD + DD (Euclidean)	NNOR-CB	63.9

The authors know of only two publicly available sets of static hand skeletal data for which comparative analysis using our method is possible. They are Dataset 2 and Dataset 3, described in Section 4.1.

Table 6 presents the comparison of the recognition rates obtained for Dataset 2 by our method (in two configurations) and by the methods proposed in two other works. The proposed algorithm outperforms other methods by more than 9%.

Table 6. Comparison of leave-one-subject-out, 14-fold cross-validation test results obtained for Dataset 2.

Reference	Features	Classifier	Recognition Rate
Marin et al. [15]	fingertips distances, angles, elevations	SVM-Gauss	80.9
Du et al. [45]	fingertips distances, angles, elevations + fingertips tip distance	SVM-Radial	81.1
Marin et al. [18]	fingertips positions	SVM-Gauss	81.5
our method	PPD + DD (Euclidean)	TreeBag	90.7
our method	PPD + DD (Euclidean)	10NN-W	90.7

Dataset 3 can be considered very challenging, since its authors [24] achieved a recognition rate of only 35.1% with skeletal data. Table 7 presents the comparison of the recognition rates obtained for this dataset by our method and the method proposed in other work.

Table 7. Comparison of leave-one-subject-out, 5-fold cross-validation test results obtained for the Dataset 3.

Reference	Features	Classifier	Recognition Rate
Tao et al. [24]	angles between bones and fingers	deep networks	35.1
our method	PPD + DD (city-block)	10NN-W	38.1
our method	PPD + DD (city-block)	NNOR-CB	40.9

The experiments were performed using Matlab R2018b software with Classification Learner toolbox on a PC with an Intel Core i5-8300H, 2.3 GHz CPU, and 16 GB RAM. The average time of feature extraction is about 5 ms. The total recognition time (feature extraction and classification) does not exceed 70 ms. Therefore, the time delay measured from gesture execution to the predicted response of the program is barely noticeable by the user.

5. Conclusions

Hand posture recognition is a classical task in computer vision [1,6,8]. Despite many methods, which perform robustly under some limitations, the problem is still exciting. Challenging barriers persist while creating a recognition system able to function in real-world conditions. The most important among them are occlusion of fingers related to the presence of affine transformation while projecting the 3D scene on the 2D image plane, scalability of considered gesture dictionaries, different background illumination, high computational cost, and repeatability of gesture execution by potential users. Currently, new devices working in the field of visible and near-infrared light are being developed, which will allow us to obtain accurate 3D information about the observed scene. There is a chance that the usage of these devices will eliminate some of the restrictions mentioned above. Therefore, in recent literature on hand gesture recognition, a shift to depth modality is observed. In this paper, a novel descriptor was proposed, which encodes information about distances between particular hand points. It is a scale-independent and distinctive alternative to other positional-based and distance-based features describing hand skeleton. Its features include relations between each fingertip and between

fingertips and palm center. Unlike most other works, the method has been tested on a data set containing 48 classes, among which many similar shapes can be identified. It has been observed that the independence of the proposed approach from hand orientation is not always desirable and can lead to difficulties in recognizing some hand configurations. Therefore, a modified version of the nearest neighbor classifier was proposed, which can distinguish between very similar or identical postures, differing only in hand orientation. The n -fold cross-validation tests were performed on three challenging datasets. For each dataset, the leave-one-subject-out protocol was used, which usually gives the worst results, but is the most trustworthy. It shows how the method deals with different gesture performances by individual users. The experimental results were compared with our previous work as well as with other methods found in the literature. A significant improvement of results over the compared methods was observed.

Summarizing, we did not find any methods for static hand posture recognition based only on hand skeletal data, with which we can compare our methods (mainly because of publicly available datasets) and which are better than our method in terms of recognition rate.

The proposed Distance Descriptor is invariant to position and scale. It is also invariant to rotation. However, this feature is desirable only for datasets not containing classes that differ only in orientation. Using the proposed nearest neighbor classifier with orientation restriction makes the recognition method partially dependent on orientation, enabling it to distinguish between such gestures. The threshold parameter γ_t has to be experimentally set based on the orientation of similar postures of the considered sign language; it should be less than the least angle between palm normals of any similar postures classes. However, it also cannot be too small, since that would make the recognition method too strongly dependent on hand orientation. It is worth noting that descriptors and features proposed in the literature are not always invariant to hand size and orientation.

The features of PPD contain the information about angular relations of skeletal joints. The information of DD features is positional and relies on distance between the joints. The results from Tables 3 and 5 show that the angular information of PPD and the positional information of DD complement each other, since the addition of DD to PPD features significantly improved the recognition rate.

The most commonly confused hand postures of the Dataset 1 are B-Bm, C-100, S-F, T-O, Z-Xm, Bz-Cm, and 4z-Cm. These postures are shown in pairs in Figure 3. Most of these hand-shapes are very similar, even for the human eye.

Figure 3. The most often confused postures of Dataset 1.

The proposed recognition method is fast and does not require a specific background, lighting conditions, or any special outfit—e.g., gloves. The main reason for the weaker results of leave-one-subject-out tests is the imperfection of the sensor, which has issues with proper detection of occluding fingers. Therefore, further work may include obtaining more accurate and reliable hand skeletal data using two calibrated depth sensors. Another future study topic may be recognition

of letter sequences (finger spelling), understood as quick, highly coarticulated motions. Finally, the developed PPD and DD descriptors can be adopted in a method for recognition of human actions based on whole-body skeletons (e.g., obtained from Kinect camera).

Author Contributions: Conceptualization, methodology, D.W. and T.K.; software, D.W. and T.K.; datasets, T.K.; experiments design, discussion of the results, D.W.; writing—review and editing, D.W. and T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This project is financed by the Minister of Science and Higher Education of the Republic of Poland within the “Regional Initiative of Excellence” program for years 2019–2022. Project number 027/RID/2018/19, amount granted 11 999 900 PLN.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PPD	Point Pair Descriptor
DD	Distance Descriptor
SVM-Lin	support vector machine with linear kernel function
LD	linear discriminant
TreeBag	ensemble of multiple decision trees—tree bagger
10NN-W	weighted k-nearest neighbors classifier with $k = 10$
NNOR-CB	nearest neighbor with orientation restriction and city block distance
NNOR-Euc	nearest neighbor with orientation restriction and Euclidean distance

References

1. Cheok, M.J.; Omar, Z.; Jaward, M.H. A review of hand gesture and sign language recognition techniques. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 131–153. [[CrossRef](#)]
2. Zengeler, N.; Kopinski, T.; Handmann, U. Hand Gesture Recognition in Automotive Human–Machine Interaction Using Depth Cameras. *Sensors* **2018**, *19*, 59. [[CrossRef](#)] [[PubMed](#)]
3. Purushothaman, A.; Palaniswamy, S. Development of Smart Home using Gesture Recognition for Elderly and Disabled. *J. Comput. Theor. Nanosci.* **2020**, *17*, 177–181. [[CrossRef](#)]
4. Khalaf, A.S.; Alharthi, S.A.; Dolgov, I.; Touns, Z.O. A Comparative Study of Hand Gesture Recognition Devices in the Context of Game Design. In Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces, Daejeon, Korea, 10–13 November 2019; Association for Computing Machinery: New York, NY, USA, 2019; ISS '19; pp. 397–402. [[CrossRef](#)]
5. Cardoso, J. A Review of Technologies for Gestural Interaction in Virtual Reality. In *Recent Perspectives on Gesture and Multimodality*; Cambridge Scholars Publishing: Newcastle-upon-Tyne, UK, 2019.
6. Bragg, D.; Koller, O.; Bellard, M.; Berke, L.; Boudreault, P.; Braffort, A.; Caselli, N.; Huenerfauth, M.; Kacorri, H.; Verhoef, T.; et al. Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. In Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility, Pittsburgh, PA, USA, 28–30 October 2019; ACM: New York, NY, USA, 2019; pp. 16–31.
7. Pezzuoli, F.; Corona, D.; Corradini, M.L.; Cristofaro, A. Development of a Wearable Device for Sign Language Translation. In *Human Friendly Robotics*; Ficuciello, F., Ruggiero, F., Finzi, A., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 115–126.
8. Rautaray, S.S.; Agrawal, A. Vision based hand gesture recognition for human computer interaction: A survey. *Artif. Intell. Rev.* **2015**, *43*, 1–54. [[CrossRef](#)]
9. Terrillon, J.; Shirazi, M.N.; Fukamachi, H.; Akamatsu, S. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), Grenoble, France, 28–30 March 2000; pp. 54–61. [[CrossRef](#)]
10. Smedt, Q.D.; Wannous, H.; Vandeborre, J.P. Heterogeneous hand gesture recognition using 3D dynamic skeletal data. *Comput. Vis. Image Underst.* **2019**, *181*, 60–72. [[CrossRef](#)]
11. Leap Motion. Available online: <https://www.leapmotion.com> (accessed on 20 March 2020).

12. Weichert, F.; Bachmann, D.; Rudak, B.; Fisseler, D. Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors* **2013**, *13*, 6380–6393. [[CrossRef](#)] [[PubMed](#)]
13. Potter, L.E.; Araullo, J.; Carter, L. The Leap Motion Controller: A View on Sign Language. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*; ACM: New York, NY, USA, 2013; OzCHI '13, pp. 175–178. [[CrossRef](#)]
14. Guna, J.; Jakus, G.; Pogačnik, M.; Tomažič, S.; Sodnik, J. An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *Sensors* **2014**, *14*, 3702–3720. [[CrossRef](#)] [[PubMed](#)]
15. Marin, G.; Dominio, F.; Zanuttigh, P. Hand gesture recognition with leap motion and kinect devices. In *Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP)*, Paris, France, 27–30 October 2014; pp. 1565–1569. [[CrossRef](#)]
16. Chuan, C.H.; Regina, E.; Guardino, C. American Sign Language Recognition Using Leap Motion Sensor. In *Proceedings of the 2014 13th International Conference on Machine Learning and Applications*, Detroit, MI, USA, 3–6 December 2014; pp. 541–544. [[CrossRef](#)]
17. Lu, W.; Tong, Z.; Chu, J. Dynamic Hand Gesture Recognition With Leap Motion Controller. *IEEE Signal Process. Lett.* **2016**, *23*, 1188–1192. [[CrossRef](#)]
18. Marin, G.; Dominio, F.; Zanuttigh, P. Hand gesture recognition with jointly calibrated Leap Motion and depth sensor. *Multimed. Tools Appl.* **2016**, *75*, 14991–15015. [[CrossRef](#)]
19. Fok, K.Y.; Ganganath, N.; Cheng, C.T.; Tse, C.K. A Real-Time ASL Recognition System Using Leap Motion Sensors. In *Proceedings of the 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Xi'an, China, 17–19 September 2015; pp. 411–414. [[CrossRef](#)]
20. Naglot, D.; Kulkarni, M. Real time sign language recognition using the leap motion controller. In *Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, 26–27 August 2016; Volume 3, pp. 1–5. [[CrossRef](#)]
21. Avola, D.; Bernardi, M.; Cinque, L.; Foresti, G.L.; Massaroni, C. Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric Hand Gestures. *IEEE Trans. Multimed.* **2019**, *21*, 234–245. [[CrossRef](#)]
22. Vaitkevičius, A.; Taroza, M.; Blažauskas, T.; Damaševičius, R.; Maskeliūnas, R.; Woźniak, M. Recognition of American Sign Language Gestures in a Virtual Reality Using Leap Motion. *Appl. Sci.* **2019**, *9*, 445. [[CrossRef](#)]
23. Chong, T.W.; Lee, B.G. American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach. *Sensors* **2018**, *18*, 3554. [[CrossRef](#)] [[PubMed](#)]
24. Tao, W.; Lai, Z.H.; Leu, M.C.; Yin, Z. American Sign Language Alphabet Recognition Using Leap Motion Controller. In *Proceedings of the 2018 IISE Annual Conference (IISE Annual Conference and Expo 2018)*, Orlando, FL, USA, 19–22 May 2018; pp. 599–604.
25. Mohandes, M.; Aliyu, S.; Deriche, M. Arabic sign language recognition using the leap motion controller. In *Proceedings of the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, Istanbul, Turkey, 1–4 June 2014; pp. 960–965. [[CrossRef](#)]
26. Elons, A.S.; Ahmed, M.; Shedid, H.; Tolba, M.F. Arabic sign language recognition using leap motion sensor. In *Proceedings of the 2014 9th International Conference on Computer Engineering Systems (ICCES)*, Cairo, Egypt, 22–23 December 2014; pp. 368–373. [[CrossRef](#)]
27. Hisham, B.; Hamouda, D. Arabic Static and Dynamic Gestures Recognition Using Leap Motion. *J. Comput. Sci.* **2017**, *13*. [[CrossRef](#)]
28. Mohandes, M.; Aliyu, S.; Deriche, M. Prototype Arabic Sign language recognition using multi-sensor data fusion of two leap motion controllers. In *Proceedings of the 2015 IEEE 12th International Multi-Conference on Systems, Signals Devices (SSD15)*, Mahdia, Tunisia, 16–19 March 2015; pp. 1–6. [[CrossRef](#)]
29. Mapari, R.B.; Kharat, G. Real time human pose recognition using leap motion sensor. In *Proceedings of the 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCIN)*, Kolkata, India, 20–22 November 2015; pp. 323–328. [[CrossRef](#)]
30. Kumar, P.; Saini, R.; Behera, S.K.; Dogra, D.P.; Roy, P.P. Real-time recognition of sign language gestures and air-writing using leap motion. In *Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, Nagoya, Japan, 8–12 May 2017; pp. 157–160. [[CrossRef](#)]
31. Naidu, C.; Ghotkar, A. Hand Gesture Recognition Using Leap Motion Controller. *Int. J. Sci. Res.* **2016**, *5*, 436–441.

32. Nájera, L.O.R.; Sánchez, M.L.; Serna, J.G.G.; Tapia, R.P.; Llanes, J.Y.A. Recognition of Mexican Sign Language through the Leap Motion Controller. In Proceedings of the International Conference on Scientific Computing, Las Vegas, NV, USA, 25–28 July 2016.
33. Raziq, N.; Latif, S. Pakistan Sign Language Recognition and Translation System using Leap Motion Device. In *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*; Xhafa, F., Barolli, L., Amato, F., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 895–902.
34. Kapuscinski, T.; Organisciak, P. Handshape Recognition Using Skeletal Data. *Sensors* **2018**, *18*, 2577. [[CrossRef](#)] [[PubMed](#)]
35. Gieser, S.N.; Boisselle, A.; Makedon, F. Real-Time Static Gesture Recognition for Upper Extremity Rehabilitation Using the Leap Motion. In *Digital Human Modeling. Applications in Health, Safety, Ergonomics and Risk Management: Ergonomics and Health*; Duffy, V.G., Ed.; Springer International Publishing: Cham, Switzerland, 2015; pp. 144–154.
36. Kiselev, V.; Khlamov, M.; Chuvilin, K. Hand Gesture Recognition with Multiple Leap Motion Devices. In Proceedings of the 2019 24th Conference of Open Innovations Association (FRUCT), Moscow, Russia, 9–10 April 2019; pp. 163–169. [[CrossRef](#)]
37. Cohen, M.W.; Zikri, N.B.; Velkovich, A. Recognition of Continuous Sign Language Alphabet Using Leap Motion Controller. In Proceedings of the 2018 11th International Conference on Human System Interaction (HSI), Gdansk, Poland, 4–6 July 2018; pp. 193–199. [[CrossRef](#)]
38. Chophuk, P.; Pattanaworapan, K.; Chamnongthai, K. Fist american sign language recognition using leap motion sensor. In Proceedings of the 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 7–10 January 2018; pp. 1–4. [[CrossRef](#)]
39. Zhi, D.; de Oliveira, T.E.A.; d. Fonseca, V.P.; Petriu, E.M. Teaching a Robot Sign Language using Vision-Based Hand Gesture Recognition. In Proceedings of the 2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Ottawa, ON, Canada, 12–13 June 2018; pp. 1–6. [[CrossRef](#)]
40. Karaci, A.; Akyol, K.; Gültepe, Y. Turkish Sign Language Alphabet Recognition with Leap Motion. In Proceedings of the International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES'18), Safranbolu, Turkey, 11–13 May 2018; pp. 189–192.
41. Alonso, D.G.; Teyseyre, A.; Berdun, L.; Schiaffino, S. A Deep Learning Approach for Hybrid Hand Gesture Recognition. In *Advances in Soft Computing*; Martínez-Villaseñor, L., Batyrshin, I., Marín-Hernández, A., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 87–99.
42. Leap Motion Data. Available online: <https://developer.leapmotion.com/documentation/v4/concepts.html> (accessed on 20 March 2020).
43. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Beetz, M. Learning informative point classes for the acquisition of object model maps. In Proceedings of the 2008 10th International Conference on Control, Automation, Robotics and Vision, Hanoi, Vietnam, 17–20 December 2008; pp. 643–650. [[CrossRef](#)]
44. Spivak, M. *A Comprehensive Introduction to Differential Geometry*, 3rd ed.; Publish or Perish: Houston, TX, USA, 1999; Volume 3.
45. Du, Y.; Liu, S.; Feng, L.; Chen, M.; Wu, J. Hand Gesture Recognition with Leap Motion. *arXiv* **2017**, arXiv:1711.04293.
46. Dataset and source codes of Distance Descriptor and Point Pair Descriptor. Available online: <http://vision.kia.prz.edu.pl> (accessed on 20 March 2020).
47. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
48. Rayens, W.S. Discriminant Analysis and Statistical Pattern Recognition. *Technometrics* **1993**, *35*, 324–326. [[CrossRef](#)]
49. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140.1018054314350. [[CrossRef](#)]
50. Dudani, S.A. The Distance-Weighted k-Nearest-Neighbor Rule. *IEEE Trans. Syst. Man Cybern.* **1976**, SMC-6, 325–327. [[CrossRef](#)]