Hand gesture recognition with Leap Motion and Kinect devices

Giulio Marin, Fabio Dominio and Pietro Zanuttigh
Department of Information Engineering
University of Padova, Italy

Abstract—**The recent introduction of novel acquisition devices like the Leap Motion and the Kinect allows to obtain a very informative description of the hand pose that can be exploited for accurate gesture recognition. This paper proposes a novel hand gesture recognition scheme explicitly targeted to Leap Motion data. An ad-hoc feature set based on the positions and orientation of the fingertips is computed and fed into a multi-class SVM classifier in order to recognize the performed gestures. A set of features is also extracted from the depth computed from the Kinect and combined with the Leap Motion ones in order to improve the recognition performance. Experimental results present a comparison between the accuracy that can be obtained from the two devices on a subset of the American Manual Alphabet and show how, by combining the two features sets, it is possible to achieve a very high accuracy in real-time.**

I. INTRODUCTION

In recent years, hand gesture recognition [1] has attracted a growing interest due to its applications in many different fields, such as human-computer interaction, robotics, computer gaming, automatic sign-language interpretation and so on. The problem was originally tackled by the computer vision community by means of images and videos [1], [2]. More recently the introduction of low cost consumer depth cameras, like Time-Of-Flight cameras and Microsoft's Kinect$^{\text{TM}}$ [3], has opened the way to several different approaches that exploit the depth information acquired by these devices for improving gesture recognition performance. Most approaches recognize the gestures by applying machine-learning techniques to a set of relevant features extracted from the depth data. In [4] silhouette and cell occupancy features are used to build a shape descriptor that is then fed to a classifier based on action graphs. Volumetric shape descriptors and a classifier based on Support Vector Machines are used both by [5] and [6]. [7] and [8] compare, instead, the histograms of the distance of hand edge points from the hand center in order to recognize the gestures. Four different types of features are extracted and fed into a SVM classifier in the approach of [9] and [10].

The recent introduction of the Leap Motion device has opened new opportunities for gesture recognition. Differently from the Kinect, this device is explicitly targeted to hand gesture recognition and directly computes the position of the fingertips and the hand orientation. Compared to depth cameras like the Kinect and similar devices, it produces a more limited amount of information (only a few keypoints instead of the complete depth map) and its interaction zone is rather limited but the extracted data is more accurate (according to [11] the accuracy is of about $200\mu m$) and it is not necessary to perform image processing tasks to extract the relevant points. The Leap Motion software recognizes a few movement patterns only, like swipe and tap, but the exploitation of Leap Motion data for gesture recognition purposes is still an almost unexplored field. A preliminary study referring to sign language recognition has been presented in [12], while in [13] the authors use the device to control a robot arm.

This paper presents the first attempt to detect gestures from the data acquired by the Leap Motion. A set of relevant features is extracted from the data produced by the sensor and fed into a SVM classifier in order to recognize the performed gestures. The same gestures have also been acquired with a Kinect and this paper shows both the comparison of the performance that can be obtained from the data of the two devices, and how to combine them together to improve the recognition accuracy.

The paper is organized in the following way: Section II introduces the general architecture of the proposed system. Sections III and IV present the feature descriptors extracted from Leap Motion and Kinect data respectively. Then the classifying algorithm is described in Section V. Experimental results are in Section VI and finally Section VII draws the conclusions.

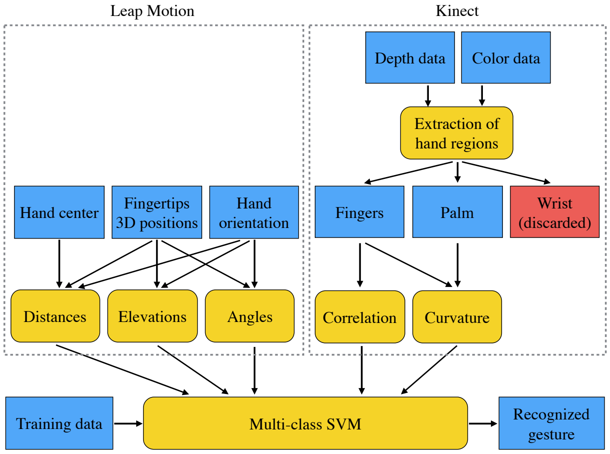

Fig. 1: Pipeline of the proposed approach.

II. GENERAL OVERVIEW

Fig. 1 shows a general overview of the two main pipelines in the proposed approach, the left side refers to the Leap Motion and the right one to the Kinect device. In the first step, the hand attributes are gathered from the Leap Motion and the depth acquired from the Kinect. Then a set of relevant features is extracted from the data acquired by the two devices. The two sensors provide different data, therefore different features set have been extracted from each of the two sensors. Finally a multi-class Support Vector Machine classifier is applied to the extracted features in order to recognize the performed gesture.

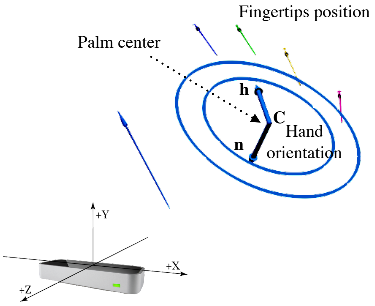

Fig. 2: Data acquired by the Leap Motion.

III. FEATURES EXTRACTION FROM LEAP MOTION DATA

Differently from the Kinect and other similar devices, the Leap Motion does not return a complete depth map but only a set of relevant hand points and some hand pose features. Fig. 2 highlights the data acquired by the Leap device that will be used in the proposed gesture recognition system, namely:

- **Position of the fingertips** $\mathbf{F}_i$, $i = 1, ..., N$ represent the 3D positions of the detected fingertips (N is the number of recognized fingers). Note that the device is not able to associate each 3D position to a particular finger.
- **Palm center** $\mathbf{C}$ roughly corresponds to the center of the palm region in the 3D space.
- **Hand orientation** based on two unit vectors, $\mathbf{h}$ is pointing from the palm center to the fingers, while $\mathbf{n}$ is perpendicular to the hand (palm) plane pointing downward from the palm center. However, their estimation is not very accurate and depends on the fingers arrangement.

An important observation is that, while the computed 3D positions are quite accurate (the error is about 200 μm according to the study in [11]), the sensor is not always able to recognize all the fingers. Not only fingers touching each other, folded over the hand or hidden from the camera viewpoint are not captured, but in many configurations some visible fingers could be lost, specially if the hand is not perpendicular to the camera. Furthermore, protruding objects near the hand, like bracelet or sleeves edges, are easily confused by fingers. This is quite critical since in different executions of the same gesture the number of captured fingers could vary. Approaches simply exploiting the number of captured fingers therefore do not work very well.

Note also that the current version of the Leap Motion software does not return any information about the matching between the acquired points and the corresponding fingers, therefore, values are randomly ordered. In the proposed approach, we deal with this issue by sorting the features on the bases of the fingertips angle with respect to the hand direction $\mathbf{h}$. This corresponds to sort them in the order from the thumb to the pinky. In order to account for fingers misalignment, as depicted in Fig. 3, we divide the plane described by $\mathbf{n}$ and passing through $\mathbf{C}$, into five angular regions S_i, $i = 1, ..., 5$, and assign each captured finger to a specific region according to the angle between the projection of the finger in the plane and the hand direction $\mathbf{h}$. Note that there is not a one-to-one matching between sectors and fingers, i.e., some of the sectors S_i could contain more than one finger and others could be empty. When two fingers lie in the same angular region, one of the two is assigned to the nearest adjacent sector if not already occupied, otherwise the maximum between the two feature values is selected.

All the features values (except for the angles) are normalized in the interval $[0, 1]$ by dividing the values for the distance between the hand center and the middle fingertip $S = ||\mathbf{F_{middle}} - \mathbf{C}||$ in order to make the approach robust to people with different hands of different sizes. The scale factor S can be computed when the user starts to use the system.

To this purpose we introduce the following features :

- **Fingertips angle** $A_i = \angle(\mathbf{F}_i^\pi - \mathbf{C}, \mathbf{h}), i = 1, ..., 5$, where $\mathbf{F_i^\pi}$ is the projection of $\mathbf{F_i}$ on the plane identified by $\mathbf{n}$, are the angles corresponding to the orientation of the projected fingertips with respect to the hand orientation. The actual number of captured fingers strongly affects the hand orientation $\mathbf{h}$ and so the fingertips angles. The obtained values A_i have been scaled to the interval $[0.5, 1]$ to better discriminate the valid interval from the missing values that have been set to 0. These values have also been used to assign each finger to the corresponding sector.
- **Fingertips distance** $D_i = ||\mathbf{F}_i - \mathbf{C}||/S, i = 1, ..., 5$, are the 3D distances of the fingertips from the hand center. Note that, as previously stated, there is at most one feature value for each sector and the missing values has been set to 0.
- **Fingertips elevation** $E_i = \text{sgn}((\mathbf{F}_i - \mathbf{F}_i^\pi) \cdot \mathbf{n})||\mathbf{F}_i - \mathbf{F}_i^\pi||/S, i = 1, ..., 5$, represent the distances of the fingertips from the plane corresponding to the palm region, accounting also for the fact that the fingertips can belong to any of the two semi-spaces defined by the palm plane. As for fingertips distances, there is at

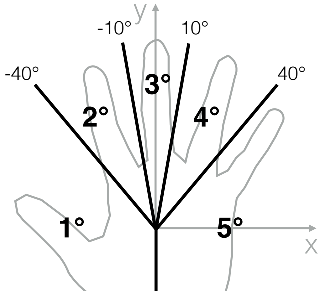

Fig. 3: Angular regions in the palm plane.

most one feature value for each sector and the missing values has been set to 0. Note that as for the fingertips angles, the values range has been scaled to the interval $[0.5, 1]$

IV. FEATURES EXTRACTION FROM KINECT DATA

Gestures have been acquired with both a Leap Motion and a Kinect device. For the features extraction from Kinect data, we employed a pipeline made of two main steps, i.e., the hand is firstly extracted from the acquired depth and color data and then two different types of features are computed from the 3D points corresponding to the hand. The extraction of the hand from color and depth data has been performed using the approach of [9]: the analysis starts from the closest point in the depth map and a thresholding on depth and 3D distances is used to extract the candidate hand region. A further check on hand color and size is performed to avoid to recognize closer objects as the hand.

Then, two different types of features are extracted. For the first set of features, an histogram of the distances of the hand points from the hand center is built as described in [9], i.e.:

$$L(\theta_q) = \max_{\mathbf{X}_i \in \mathcal{I}(\theta_q)} d_{\mathbf{X}_i} \tag{1}$$

where $\mathcal{I}(\theta_q)$ is the angular sector of the hand corresponding to the direction θ_q and $d_{\mathbf{X}_i}$ is the distance between point X_i and the hand center. For a detailed description of how the histogram are computed, see [9]. A set of reference histograms $L_g^r(\theta)$, one for each gesture g is also built. Differently from [9] where the maximum of the histograms were used, here the feature values are the maximum of the correlation between the current histogram $L(\theta_q)$ and a shifted version of the reference histogram $L_g^r(\theta)$

$$R_g = \max_{\Delta} \rho\left(L(\theta), L_g^r(\theta + \Delta)\right) \tag{2}$$

where $g = 1, ..., G$. Note that the computation is performed for each of the candidate gesture, thus obtaining a different feature value for each of the candidate gestures. Ideally the correlation with the correct gesture should have a larger value than the other features.

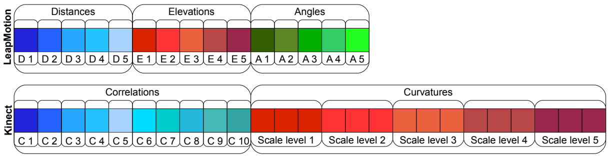

Fig. 4: Feature vectors extracted from the two devices.

The second feature set is based on the curvature of the hand contour. This descriptor is based on a multi-scale integral operator and is computed as described in [9] and [10]. The multi-scale descriptor is made of $B \times S$ entries $C_i, i = 1, ..., B \times S$, where B is the number of bins and S the number of employed scale levels.

V. GESTURE CLASSIFICATION

The feature extraction approach of Sections III and IV provides five feature vectors, each describing relevant properties of the hand samples extracted from the two sensors. In order to recognize the performed gestures, the extracted features are used into a multi-class Support Vector Machine classifier. Each acquired gesture is described by two feature sets. The set $\mathbf{V_{leap}} = [\mathbf{A}, \mathbf{D}, \mathbf{E}]$ contains all the features extracted from Leap Motion data while the set $\mathbf{V_{kin}} = [\mathbf{C}, \mathbf{R}]$ contains the features extracted from Kinect data. The complete feature set is obtained by concatenating the two sets $[\mathbf{V_{leap}}, \mathbf{V_{kin}}]$.

In order to recognize gestures, the five features vectors and their concatenation must be classified into G classes corresponding to the various gestures of the considered database. The employed classification algorithm exploits Support Vector Machines (SVM). A multi-class SVM classifier [14] based on the *one-against-one* approach has been used, i.e., a set of $G(G-1)/2$ binary SVM classifiers are used to test each class against each other and each output is chosen as a *vote* for a certain gesture. The gesture with the maximum number of votes is selected as the output of the classification. A non-linear Gaussian Radial Basis Function (RBF) kernel has been used while the classifier parameters have been selected by means of a grid search approach and cross-validation on the training set. Assume a training set containing data from M users, to perform the grid search we divide the space of parameters (C, γ) of the RBF kernel with a regular grid and for each couple of parameters the training set is divided into two parts, one containing $M - 1$ users for training and the other one the remaining user for validation and performance evaluation. We repeat the procedure changing each time the user used for the validation and we select the couple of parameters that give the best accuracy on average. Finally we train the SVM on all the M users of the training set with the optimal parameters. Alternative classification schemes also based on SVM for this task have been proposed in [15].

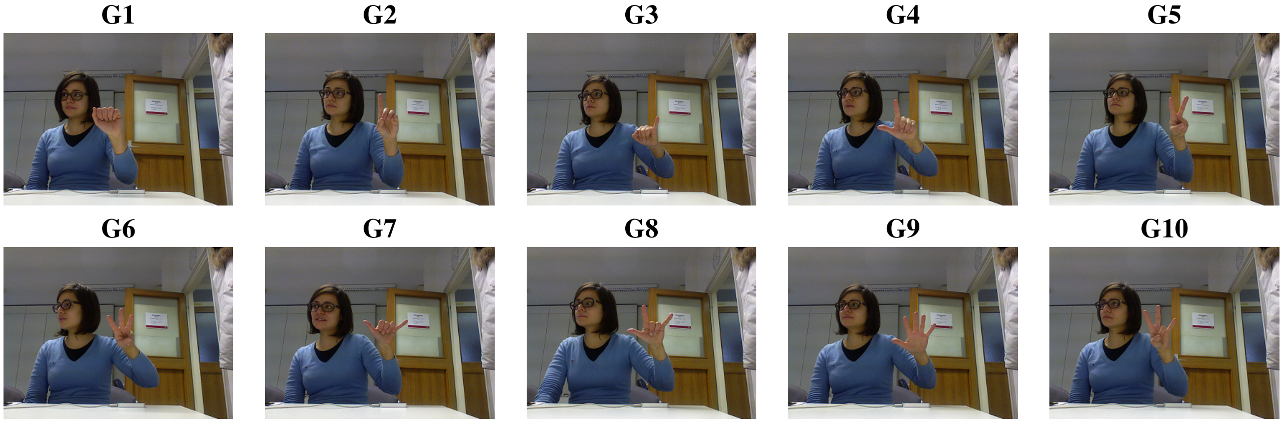

Fig. 5: Gestures from the American Sign Language (ASL) contained in the database that has been acquired for experimental results.

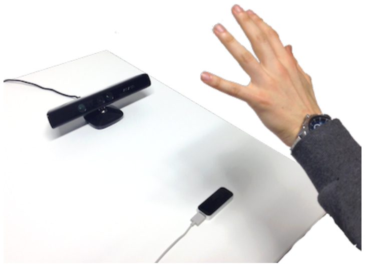

Fig. 6: Acquisition setup.

VI. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed approach, we acquired a dataset of gestures using the setup of Fig.6. It is available at http://lttm.dei.unipd.it/downloads/gesture. The performed gestures have been acquired at the same time with both a Leap Motion device and a Kinect sensor. The database contains 10 different gestures (see Fig.5) performed by 14 different people. Each gesture is repeated 10 times for a total of 1400 different data samples. For each sample, Leap Motion data reported in Section III have been acquired together with the depth maps and color images provided by the Kinect.

Table I shows the accuracy obtained from the Leap Motion and Kinect data using the classification algorithm of Section V. Fingertip distance features allow to obtain an accuracy of about 76%, they are able to recognize the majority of the gestures but G2 and G3 are easily confused. This is due to the limited accuracy of the hand direction estimation from the Leap Motion software that causes an unreliable matching between the fingertip of these gestures and the corresponding angular region. This inaccuracy is partially solved with the other two features but a slightly lower overall accuracy is obtained: 74.2% from the fingertip angles and 73% from the elevations. An interesting observation is that the 3 feature descriptors capture different properties of the performed gesture and by combining them together it is possible to improve the recognition accuracy, e.g., by combining distances and elevations, an accuracy of about 80% can be reached. By combining all the 3 features together, 81% of accuracy can be obtained, and this represents the best accuracy that can be extracted from Leap Motion data with the proposed approach.

Kinect data contain a more informative description (i.e., the complete 3D structure of the hand) but they are also less accurate and provide a lower-level scene description. Furthermore, the acquired data need to be preprocessed in order to segment the hand from the scene for features extraction. Correlation features allow to recognize a good number of gestures but the 65% accuracy is the lowest among the considered descriptors. As reported also in [9], curvature descriptor is instead a very accurate representation of the hand shape that allows to obtain a very high accuracy of 87.3%. It is the best single descriptor among the considered ones. Even if the performance of correlation is not too good, by combining this descriptor with the curvature it is possible to improve the results of the latter descriptor obtaining an accuracy of 89.7%. This is a proof of the good performance of the proposed machine learning strategy, that is able to obtain good results even when combining descriptors with very different performance, without being affected by the less performing features.

Table II reports the result obtained by combining all the various features from the two sensors. The complementarity of their descriptions is demonstrated by the optimal accuracy of 91.3% that is obtained. Even if the Kinect as a single sensor has better performance, the combined use of the two devices allows to obtain better performance than each of the two sensors alone.

Finally, Table III shows the confusion matrix when all the five features are combined together. The accuracy is over 90% for all the gestures but G6, G8 and G10, that are the gestures that more frequently fail the recognition. From this table it can also be noticed that gestures G2 and G3 that were critical for the Leap Motion, reveal a very high

Leap Motion		Kinect	
Feature set	Accuracy	Feature set	Accuracy
Fingertips distances (D)	**76.07%**	**Curvature (C)**	**87.28%**
Fingertips angles (A)	74.21%	Correlation (R)	65.00%
Fingertips elevations (E)	73.07%		
D + A	78.78%	**C + R**	**89.71%**
E + A	77.28%		
D + E	**80.20%**		
D + A + E	**80.86%**		

TABLE I: Performance of Leap Motion and Kinect features.

Feature set	Accuracy
D + A+ E + C + R	**91.28%**

TABLE II: Performance from the combined use of the two sensors.

accuracy when recognized from both the devices.

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
G1	**0.99**		0.01							
G2		**0.96**	0.03		0.01	0.01				
G3		0.02	**0.96**		0.01		0.01			
G4		0.01	0.01	**0.91**	0.01		0.01	0.03		0.01
G5		0.03		0.01	**0.94**	0.01		0.01		
G6		0.01	0.01		0.02	**0.86**			0.04	0.07
G7			0.01	0.02	0.01	0.01	**0.90**	0.05		
G8				0.03			0.07	**0.86**		0.04
G9						0.01		0.01	**0.97**	0.01
G10					0.01	0.19		0.03		**0.78**

TABLE III: Confusion matrix for the complete features set. *Yellow* cells represent true positive, while *gray* cells show false positive with failure rate greater than 5%.

VII. CONCLUSIONS

In this paper two different gesture recognition algorithms for the Leap Motion and Kinect devices have been proposed. Different feature sets have been used to deal with the different nature of data provided by the two devices, the Leap Motion provides a higher level but more limited data description while Kinect provides the full depth map. Even if the data provided by the Leap Motion is not completely reliable, since some fingers might not be detected, the proposed set of features and classification algorithm allows to obtain a good overall accuracy. The more complete description provided by the depth map of the Kinect allows to capture other properties missing in the Leap Motion output and by combining the two devices a very good accuracy can be obtained. Experimental results show also that the assignment of each finger to a specific angular region leads to a considerable increase of performance.

Future work will address the joint calibration of the two devices in order to compute new features based on the combination of the 3D positions computed by the two devices, and the recognition of dynamic gestures with the two sensors.

REFERENCES

[1] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, “Vision-based hand-gesture applications,” *Commun. ACM*, vol. 54, no. 2, pp. 60–71, Feb. 2011.

[2] D. Kosmopoulos, A. Doulamis, and N. Doulamis, “Gesture-based video summarization,” in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 3, 2005, pp. III–1220–3.

[3] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, *Time-of-Flight Cameras and Microsoft Kinect*, ser. SpringerBriefs in Electrical and Computer Engineering. Springer, 2012.

[4] A. Kurakin, Z. Zhang, and Z. Liu, “A real-time system for dynamic hand gesture recognition with a depth sensor,” in *Proc. of EUSIPCO*, 2012.

[5] P. Suryanarayan, A. Subramanian, and D. Mandalapu, “Dynamic hand pose recognition using depth data,” in *Proc. of ICPR*, aug. 2010, pp. 3105 –3108.

[6] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, “Robust 3d action recognition with random occupancy patterns,” in *Proc. of ECCV*, 2012.

[7] Z. Ren, J. Yuan, and Z. Zhang, “Robust hand gesture recognition based on finger-earth mover’s distance with a commodity depth camera,” in *Proc. of ACM Conference on Multimedia*. ACM, 2011, pp. 1093–1096.

[8] Z. Ren, J. Meng, and J. Yuan, “Depth camera based hand gesture recognition and its applications in human-computer-interaction,” in *Proc. of ICICS*, 2011, pp. 1 –5.

[9] F. Dominio, M. Donadeo, and P. Zanuttigh, “Combining multiple depth-based descriptors for hand gesture recognition,” *Pattern Recognition Letters*, 2013.

[10] F. Dominio, M. Donadeo, G. Marin, P. Zanuttigh, and G. M. Cortelazzo, “Hand gesture recognition with depth data,” in *Proceedings of the 4th ACM/IEEE international workshop on Analysis and retrieval of tracked events and motion in imagery stream*. ACM, 2013, pp. 9–16.

[11] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, “Analysis of the accuracy and robustness of the leap motion controller,” *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[12] L. E. Potter, J. Araullo, and L. Carter, “The leap motion controller: A view on sign language,” in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, ser. OzCHI ’13. New York, NY, USA: ACM, 2013, pp. 175–178.

[13] C. Guerrero-Rincon, A. Uribe-Quevedo, H. Leon-Rodriguez, and J.-O. Park, “Hand-based tracking animatronics interaction,” in *Robotics (ISR), 2013 44th International Symposium on*, 2013, pp. 1–3.

[14] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[15] L. Nanni, A. Lumini, F. Dominio, M. Donadeo, and P. Zanuttigh, “Ensemble to improve gesture recognition,” *International Journal of Automated Identification Technology*, 2014.