




Gesture Recognition by Leap Motion Controller and LSTM Networks for CAD-oriented Interfaces

Lisa Mazzini^(✉), Annalisa Franco, and Davide Maltoni

Department of Computer Science and Engineering, University of Bologna,
Via dell'Università 50, 47521 Cesena, Italy
{lisa.mazzini,annalisa.franco,davide.maltoni}@unibo.it

Abstract. This paper presents a gesture recognition approach for CAD interfaces where the Leap Motion Controller is used for its high precision in modelling user hands. A simple, compact and effective hand representation is proposed to encode trajectory and pose across time. Recognition is based on Recurrent Neural Networks, particularly suited for processing data sequences. An effective data augmentation technique is also described to increase the size of the training set. Experiments conducted on a novel dataset of gesture performed by 30 volunteers show the effectiveness of the proposed technique; the dataset will be made available to the community for future studies.

Keywords: Gesture recognition · Leap Motion Controller · LSTM networks · Computer-Aided Design

1 Introduction

Gestures are one of the most common and natural ways people use to communicate; humans move arms, hands, fingers or even the whole body to transmit information or interact with the environment. In recent years, the development of Human-Computer Interaction systems received great attention from the research community with the aim of developing natural and unobtrusive interfaces, and making users able to interact with the system without any hand-held device. Gesture recognition systems can be profitably used in a variety of applications [4]; among others, sign language translation, daily assistance to elders or disabled people, security application and gaming are probably the most relevant.

This work focuses on the development of a gesture recognition system for CAD interfaces. Although the realisation of a complete 3D model requires fine user movements quite difficult to realise outside the sophisticated traditional CAD interfaces, more intuitive and natural interactions can be useful for initial prototyping or successive interaction with existing models. The widespread diffusion of low-cost RGB-D sensors (e.g. Kinect) and their ability to track users' movements greatly fostered research in this field. The approach proposed in this

paper is based on the use of the Leap Motion Controller (LMC) [13,17], which provides interesting functionalities for detecting and tracking user's hands; being it designed to work at short distance, hands information is provided with a noticeably higher level of precision with respect to previous devices operating at larger distances and tracking the whole human body.

The proposed gesture recognition approach is based on a novel, compact but effective hand representation coupled with Long-Short Term Memory networks (LSTM), which represent a natural choice for their ability of managing sequences of inputs over time. When dealing with networks, the level of accuracy reachable is often influenced by the availability of training data; while, for its nature, gesture recognition is in general considered a small-scale problem, the set of data for network training can be incremented by artificially generated data. One further contribution of this paper is the definition of data augmentation techniques able to produce additional data for training while keeping unaltered the semantic of gestures. Finally, a new dataset of gestures will be made available to the research community to allow for future comparisons.

The paper is organised as follow: Sect. 2 presents the state of the art, with particular reference to gesture recognition for CAD applications, Sect. 3 describes the proposed approach, the experiments are described in Sect. 4 and Sect. 5 draws some conclusions.

2 Related Works

The recent literature on human gesture recognition is huge and a comprehensive review goes beyond the scope of this work; interested readers can refer to [3,4,15] for recent surveys on 3D hand gesture recognition. Several solutions for natural CAD interfaces have been proposed in the literature. Many works propose contact-based solutions where the user interacts with the system by means of ad-hoc input devices. In [10] different techniques for sketch-based modeling are described, where the users interact with CAD applications by means of sketches; in [19] a Virtual Reality based system is described, where an electronic data glove is suggested as input device. Several vision-based techniques have also been proposed as an alternative to contact-based solutions, with the aim of providing to the user a more natural interface. No direct interactions with input devices are requested in this case; gesture interpretation is based on data streams acquired by cameras of different nature (e.g. RGB or depth). One of the most interesting sensors in this context is Microsoft Kinect [12,18], a low-cost device able to capture in parallel RGB and Depth data streams; its success is largely related to the skeleton representation provided by the SDK which allows to easily track subjects and analyze their behaviour. The use of Kinect for gesture recognition in CAD applications is proposed in some works [7,8,16]; however it is worth noting that the fine hand gestures needed to precisely interact with the system are difficult to capture with Kinect due to its simplified skeleton model where hands are simply identified by a single joint (in the palm) and no information about fingers is provided. Leap Motion Controller works at smaller distances

with respect to Kinect and offers a much more detailed hand representation, where each finger is represented by several joints. In [2] LMC multiple applications in Human-Computer Interaction are described, ranging from the medical field to human-robot interaction, from games and gamification to sign language recognition. A CAD interface based on LMC is described in [14] where a proof of concept system able to recognize a set of gestures is described; the details of the recognition approach are not given and the dataset used for testing is not available, thus making impossible a comparison with our proposal. A relevant work for our study is [1] where the use of LMC coupled with recurrent neural networks is discussed for sign language and semaphoric gesture recognition. The authors adopt a complex hand model and a deep network to deal with gestures of different nature with interesting results; we will show in our experiments that, for the specific CAD context, also a simplified representation and a relatively small network allow to reach fully satisfactory results.

3 Proposed Approach

This paper proposes a novel approach for gesture recognition based on the use of Leap Motion Controller. The Leap Motion Controller is a device designed to detect and track user hands, usually placed on the user physical desktop in front of the computer, or mounted on a headset for virtual reality. The device has two monochromatic IR cameras and three infrared LEDs. The IR light emitted from the LEDs is reflected by the user hands and then read by the cameras. Thanks to these tools, the device is able to perceive user hands inside a hemispherical area until a distance of 1 m, with a precision of 0,7 mm and a frame rate up to 200 fps. The information acquired by the sensor is then used to create an internal representation of the two hands, easily accessible thanks to the provided SDK.

3.1 Hand Representation

The hand skeleton information extracted by the LMC consists of a set of attributes, providing geometric data about the user palm, fingers and arm, but also high-level information like acquisition confidence and grabbing or pinching strength. Among the different data provided, the geometric ones are more relevant to our model. Our objective is to define a representation capturing the gesture evolution represented by hand pose, without including any information related to hand shape which is user-specific and not meaningful for gesture recognition. For this reason we neglect most of the data related to the hand position in space (except palm position, used as a reference to evaluate hand translation in time), and we mainly rely on the directions characterizing hand and fingerprints. In particular we exploited for our representation (see Fig. 1a):

- *arm*: described by its direction \mathbf{d}_a ;
- *palm*: described by its 3D position \mathbf{p} and its direction \mathbf{d}_p ;

- *fingers*: each finger is a complex object, consisting of a list of bones representing the single phalanges. We consider the direction of each bone $\mathbf{d}_{\mathbf{b}_{f,p}}$, with f being the finger index ($f = 1, \dots, 5$) and p the phalanx index ($p = 1, \dots, 3$).

Starting from the hand information provided by LMC, we defined a set of numerical features able to encode the hand pose as well as its movement in space across time. The use of angle values, instead of joint positions, allows to achieve a good level of invariance with respect to users' specific hand characteristics. In this work only gestures involving a single hand are considered, but the proposed model can be easily extended to a more general case where the user exploits both hands.

Using the above described raw data, different types of features are extracted for each frame i :

- the translation $\Delta \mathbf{p}(i)$ of the palm position with respect to frame $i - 1$:

$$\Delta \mathbf{p}(i) = \mathbf{p}(i) - \mathbf{p}(i - 1)$$

- the angle $\omega(i)$ between the palm direction and the arm direction, computed as:

$$\omega(i) = \arccos\left(\frac{\mathbf{d}_{\mathbf{p}}(i) \cdot \mathbf{d}_{\mathbf{a}}(i)}{|\mathbf{d}_{\mathbf{p}}(i)| \cdot |\mathbf{d}_{\mathbf{a}}(i)|}\right)$$

- a set of angles $\alpha_{f,p}(i)$, with $f = 1, \dots, 5$ and $p = 1, \dots, 3$, representing for each finger the angle between the palm direction and each finger phalanx:

$$\alpha_{f,p}(i) = \arccos\left(\frac{\mathbf{d}_{\mathbf{p}}(i) \cdot \mathbf{d}_{\mathbf{b}_{f,p}}(i)}{|\mathbf{d}_{\mathbf{p}}(i)| \cdot |\mathbf{d}_{\mathbf{b}_{f,p}}(i)|}\right)$$

Please note that for the thumb finger, only the $\alpha_{f,p}$ angles referred to two phalanges can be computed (i.e. for $f = 1, p = 1, 2$).

The angles $\alpha_{f,p}$ are computed to capture the finger extension or closure; ω angle can detect the wrist movement during the gesture. Each angle is measured in the plane formed by the two directions involved. In order to keep track of the hand spatial movement, we decided to consider only the variation of the palm center coordinates; by considering only point variation and not its absolute coordinates, the resulting features are invariant from the initial hand position. Each frame of the video sequence is therefore represented by a 18-dimensional vector obtained by the ordered concatenation of the above described values (3 values for translation on the three axis, 1 ω angle, and 14 $\alpha_{f,p}$ values). The sequence length is fixed to 60 frames per gesture.

3.2 Network Structure

Our approach exploits Recurrent Neural Network to recognize gestures; in particular we evaluated two variants: Long Short-Term Memory [9] and Gated Recurrent Unit [5]. All RNNs have internal state vectors than can store past events

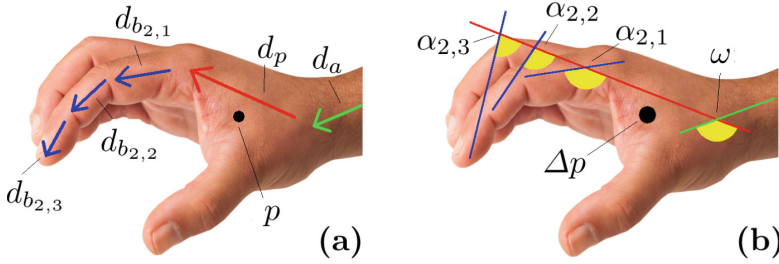


Fig. 1. Hand model: the palm direction (red) intersects the different phalanges directions (blue) and the arm direction (green), forming the angles used to model the hand pose (for instance, the features are represented for each phalanx $i = 1, \dots, 3$ for $f = 2$). The palm position (black dot) is used to keep track of the hand movement. (Color figure online)

and process current data based on the past, but in particular LSTM and GRU are able to handle longer-term dependencies characterising longer sequences of data. The results obtained using LSTM or GRU are often comparable in terms of accuracy [6]. We chose a *many-to-one* network model; in fact the network processes all the sequence elements before returning the predicted class. We chose a fixed length of 60 frames the sake of simplicity, because it has proved to be a sufficient time span for every gesture (about 2–3s per gesture). The model can be easily adapted to different frame lengths or even variable lengths among samples. For our problem, we sized the network as shown in Fig. 2: the input layer has 18 neurons, corresponding to the size of feature vectors; it is then connected to two hidden layers, each one composed by 200 LSTM neurons. The final layer

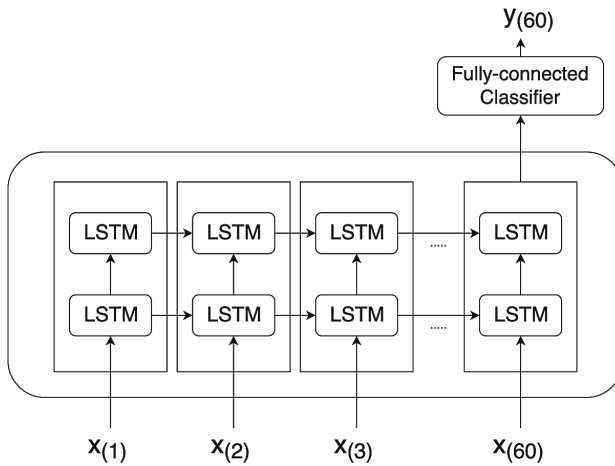


Fig. 2. Network structure unrolled through time.

is a fully-connected layer, which takes as input the last output of the second hidden layer; this layer works as a classifier and it will return the probability of each class for the current gesture. As optimizing algorithm to minimize the loss function during the training phase, we chose Adam Optimizer because it provides in several contexts better performance than other optimizers [11]. The learning rate is fixed to 0.0005.

3.3 Data Augmentation

In order to increase the data available for network training, a data augmentation technique is proposed; in particular, some transformations to the original data are applied to produce new gestures which reproduce the main gesture characteristics without introducing “unnatural” movements or hand poses.

Please note that the same random transformations are applied to the whole gesture since applying independent variations to the single frames would produce a noisy, non-smooth pattern.

Trajectory Rotation and Scaling. The first transformation applies to hand trajectory, described by the palm position \mathbf{p}_i across time. An affine transform is applied to produce trajectory rotation and scaling; trajectory translation would be totally ineffective, since the trajectory is finally encoded in terms of pose variations ($\Delta\mathbf{p}_i$ features) to achieve independence from the absolute coordinates. The affine transform given in Eq. (1) produces:

- a trajectory rotation of θ_x , θ_y and θ_z degrees on the X, Y and Z axis, respectively;
- a trajectory scaling of s_x , s_y , s_z on the three axis.

The transformation parameters are randomly generated within the ranges given in Table 1. The rotation on the X axis is quite small, because higher values would affect excessively the gesture nature; larger variations on the Y and Z axes can be applied. Moreover a uniform scaling is applied.

$$\begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (1)$$

Hand Pose Variation. The second transformation applies to hand pose, represented by the $\alpha_{f,p}$ angles. Each angle is slightly modified to generate a new pose that is still natural and realistic. In particular, to emulate effectively the natural movement of fingers, the amplitude of the variation applied is directly proportional to the phalanx distance from the palm (see v_1 , v_2 and v_3 in Table 1). In fact, the farther the phalanx is from the palm, the wider is the angle resultant from the variation applied.

For this reason, the transformation factor chosen for the fingers is slightly modified from phalanx to phalanx. Let $\alpha'_{f,p}$ be the generated angle from the original $\alpha_{f,p}$, it can be computed as:

$$\alpha'_{f,p} = v_p \cdot \alpha_{f,p}$$

Variations can be applied in both directions evenly (extending all the fingers or making them more closed).

Table 1. Transformations applied in data augmentation for trajectory rotation and scaling and for hand pose modification.

Variation	Range	Variation	Range
θ_x	$\pm(5^\circ - 10^\circ)$	v_1	$[0.95, 0.99] \cup [1.01, 1.05]$
θ_y, θ_z	$\pm(10^\circ - 15^\circ)$	v_2	$[0.945, 0.989] \cup [1.011, 1.055]$
s_x, s_y, s_z	$[0.85, 0.9] \cup [1.1, 1.15]$	v_3	$[0.94, 0.988] \cup [1.012, 1.06]$

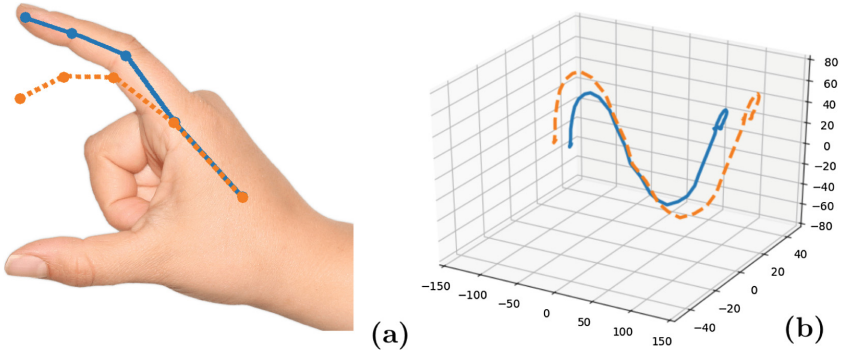


Fig. 3. Examples of data augmentation: (a) hand pose variation (example on a single finger) and (b) gesture trajectory scaling. Solid blue lines represent the original data, orange dotted lines the derived one. (Color figure online)

4 Experiments

4.1 Dataset

To the best of our knowledge no public benchmarks including raw data acquired by LMC are available. The authors of [1] share their dataset but only in terms of extracted features; the raw data needed to derive our representation are not available thus making impossible the evaluation. We decided therefore to collect a new dataset including gestures that can be applied in a hypothetical CAD software. In particular, starting from the interface described in [14], we defined 8 gesture:

- *Translation*: using only index finger, the user draws a straight trajectory.
- *Rotation*: extending both index and middle finger, the user rotates the hand of 180° , facing the palm upwards.
- *Extrusion*: extending thumb, index and middle finger, the user draws a straight or undulating trajectory.
- *Left swipe*: using only index finger, the user moves the hand quickly from right to left.
- *Right swipe*: using only index finger, the user moves the hand quickly from left to right.
- *Close*: using only index finger, the user moves down the hand quickly.
- *Scale enlargement*: starting with thumb, index and middle finger tips close together, the user moves them apart.
- *Scale reduction*: starting with thumb, index and middle finger far apart, the user moves them close together.

Each of the 30 volunteers used his/her dominant hand to perform the gesture, so there are also left-handed samples; a small training about the gestures is provided by letting them to watch a short video (available at <https://youtu.be/ZWPTjusyao0>). Then, they proceeded to perform the gestures, at their chosen speed (keeping the difference between standard speed gestures and quick gestures). Each person performed each gesture twice, so 16 gestures were obtained per person, overall 480 gesture samples. The dataset is available at <http://biolab.csr.unibo.it/CADGestures.html>.

4.2 Result and Discussion

The main indicator used for performance evaluation is accuracy, which is simply computed as the number of correct predictions C made by the network over the total number of examined instances N : $accuracy = \frac{C}{N}$. Furthermore, to extract more precise and class-specific information about the recognition accuracy, we also analyzed the confusion matrix where the rows refer to the real gesture class and the columns to the predicted one. All tests have been performed on a PC with Linux OS, on a GeForce GTX1070 GPU with 8 GB of dedicated memory and 16 GB RAM. We implemented the LSTM and GRU networks using Tensorflow, while Scikit-learn was used test SVM.

The dataset is partitioned in training set and test set in proportion 80–20, so we have 384 gestures for network training, and 96 for testing purpose. This basic training set is referred to as TS_{Base} . Moreover, to evaluate the effectiveness of data augmentation, we derived two additional training set, TS_{A1} and TS_{A2} , obtained generating respectively 1 or 2 gestures for each original gesture in TS_{Base} ; the resulting cardinality is then $|TS_{A1}| = 768$ and $|TS_{A2}| = 1152$.

We tested two versions of the proposed network, i.e. built with LSTM and GRU cells; moreover, as a term of comparison, we also evaluated the proposed hand model coupled with a SVM classifier. Since SVMs are not able to process data sequences, we concatenated in a single vector all the sequence feature vectors (overall 1080 features).

The results obtained with the base and augmented training sets are summarized in Table 2 and Fig. 3. Both LSTM and GRU reach 100% accuracy on the training set, but the first one better generalizes its knowledge on the test set, thus producing overall better results. SVMs are not designed to evaluate the sequential nature of the input, which is significant in this particular problem and this may be the reason of their lower accuracy.

In general, even if a good testing accuracy is reached with TS_{Base} , the results clearly show that data augmentation is important and significantly impacts performance for all the tested classifiers (+6% accuracy for LSTM). We can then deduce that the proposed data augmentation allows to produce new instances maintaining the nature and the spontaneity of the gesture performed.

Table 2. Results obtained using different algorithms and training sets.

Algorithm	Training set	Acc. on test set
LSTM network	TS_{Base}	87,3%
	TS_{A1}	91,6%
	TS_{A2}	93,7%
GRU network	TS_{Base}	84,3%
	TS_{A1}	87,5%
	TS_{A2}	88,5%
SVM	TS_{Base}	70,8%
	TS_{A1}	75,0%
	TS_{A2}	71,8%

An analysis of the confusion matrices in Fig. 3 shows that the most difficult gesture to recognize is Extrusion, probably due to its similarity with the Rotation gesture pose (the only difference is the extension of the thumb), even if Extrusion requires a well defined trajectory in the space, whilst Rotation is almost static. This is comprehensible if we consider that in the proposed model, only one feature value is related to trajectory and the pose information has a much higher influence on the final decision.

Even though a direct comparison with [1] is not possible since different gesture datasets are used, we can observe that our compact representation, coupled with proper data augmentation techniques, allows to reach an overall accuracy of 93,7%, comparable to that of more complex systems, like the one proposed in [1] where the reached accuracy is 96,4% (Fig. 4).

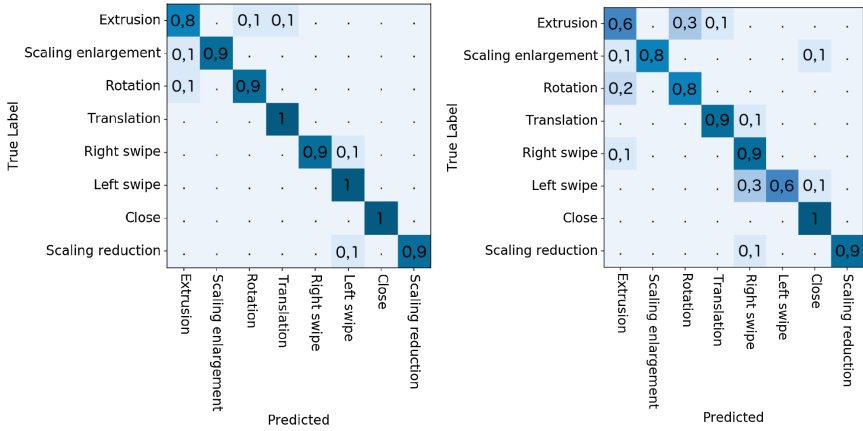


Fig. 4. Confusion matrix of the LSTM network (on the left) and the GRU network (on the right).

5 Conclusions

In this paper a new approach to gesture recognition has been proposed, based on LSTM recurrent networks and Leap Motion Controller. The results obtained are overall quite satisfactory; the fine representation of user hands allows to discriminate precise gestures with a good accuracy. Moreover, the data augmentation technique proposed to increase the set of data for network training allowed to achieve a further performance improvement. An analysis of the main causes of errors suggests some possible future works; in particular, the extracted features are mainly related to hand pose, while hand trajectory contributes to a little extent to the whole representation. Improving this aspect would allow to better discriminate gestures characterized by a similar hand posture by different trajectories across space.

References

1. 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. Multimedia* **21**(1), 234–245 (2019). <https://doi.org/10.1109/TMM.2018.2856094>
2. Bachmann, D., Weichert, F., Rinkenauer, G.: Review of three-dimensional human-computer interaction with focus on the leap motion controller. **18**(7), 2194 (2018). <https://doi.org/10.3390/s18072194>
3. Chaudhary, A., Raheja, J.L., Das, K., Raheja, S.: A survey on hand gesture recognition in context of soft computing. In: Meghanathan, N., Kaushik, B.K., Nagamalai, D. (eds.) *CCSIT 2011. CCIS*, vol. 133, pp. 46–55. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17881-8_5

4. Cheng, H., Yang, L., Liu, Z.: Survey on 3D hand gesture recognition. *IEEE Trans. Circuits Syst. Video Technol.* **26**(9), 1659–1673 (2016). <https://doi.org/10.1109/TCSVT.2015.2469551>
5. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078) (2014)
6. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint [arXiv:1412.3555](https://arxiv.org/abs/1412.3555) (2014)
7. Cohen, M.W., Frid, A., Malin, M., Ladijenski, V.: Generating 3D cad art from human gestures using kinect depth sensor. *Comput.-Aided Des. Appl.* **12**(5), 608–616 (2015). <https://doi.org/10.1080/16864360.2015.1014740>
8. Dave, D., Chowriappa, A., Kesavadas, T.: Gesture interface for 3D cad modeling using kinect. *Comput.-Aided Des. Appl.* **10**(4), 663–669 (2013). <https://doi.org/10.3722/cadaps.2013.663-669>. <https://www.tandfonline.com/doi/abs/10.3722/cadaps.2013.663-669>
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
10. Kazmi, I.K., You, L., Zhang, J.J.: A survey of sketch based modeling systems. In: 2014 11th International Conference on Computer Graphics, Imaging and Visualization, pp. 27–36, August 2014. <https://doi.org/10.1109/CGIV.2014.27>
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
12. Microsoft: Kinect for windows. <https://developer.microsoft.com/en-us/windows/kinect>
13. Leap Motion: Leap motion controller. <https://developer.leapmotion.com/get-started/>
14. Pareek, S., Sharma, V.: Development of cad interface using leap motion (2014)
15. Shabnam, S.: Real time hand gesture recognition system: a review (2015)
16. Shiratuddin, M.F., Wong, K.W.: Non-contact multi-hand gestures interaction techniques for architectural design in a virtual environment. In: ICIMU 2011: Proceedings of the 5th International Conference on Information Technology Multimedia, pp. 1–6, November 2011. <https://doi.org/10.1109/ICIMU.2011.6122761>
17. Weichert, F., Bachmann, D., Rudak, B., Fisseler, D.: Analysis of the accuracy and robustness of the leap motion controller. *Sensors* **13**(5), 6380–6393 (2013)
18. Zhang, Z.: Microsoft kinect sensor and its effect. *IEEE Multimedia* **19**(2), 4–10 (2012)
19. Zheng, J.M., Chan, K.W., Gibson, I.: A VR-based CAD system. In: Mo, J.P.T., Nemes, L. (eds.) *Global Engineering, Manufacturing and Enterprise Networks*. ITIFIP, vol. 63, pp. 264–274. Springer, Boston, MA (2001). https://doi.org/10.1007/978-0-387-35412-5_31