Approach for Improving User Interface Based on Gesture Recognition

Issam Elmagrouni^{1*}, Abdelaziz Ettaoufik², Siham Aouad³ and Abderrahim Maizate⁴

¹RITM ESTC/CED ENSEM, Hassan II University Casablanca Morocco

²LTIM, FS Ben M'SIK, Hassan II University Casablanca Morocco

³SSL ENSIAS, University Mohammed V Rabat Morocco

⁴RITM ESTC/CED ENSEM, University Hassan II Casablanca Morocco

Abstract. Gesture recognition technology based on visual detection to acquire gestures information is obtained in a non-contact manner. There are two types of gesture recognition: independent and continuous gesture recognition. The former aims to classify videos or other types of gesture sequences that only contain one isolated gesture instance in each sequence (e.g., RGB-D or skeleton data). In this study, we review existing research methods of visual gesture recognition and will be grouped according to the following family: static, dynamic, based on the supports (Kinect, Leap...etc), works that focus on the application of gesture recognition on robots and works on dealing with gesture recognition at the browser level. Following that, we take a look at the most common JavaScript-based deep learning frameworks. Then we present the idea of defining a process for improving user interface control based on gesture recognition to streamline the implementation of this mechanism.

1 Introduction

Understanding and classifying gestures from human gestural data is referred to as gesture recognition. Individual and distinct gestures are groups, and gestural instances containing gesture-data are instances, so gesture recognition is generally a pattern classification issue. There are both conventional and machine learning approaches to gesture recognition. State-machines [3], Secret Markov model (HMM) [4], and particle filter [5] are three conventional approaches worth noting.

On the other hand, a machine learning approach entails creating a mapping feature from gestural data to gesture-classes. To create the functional mapping from gestural instances to gesture-classes, many machine learning techniques have been used [6, 7]. Since the classes are identified ahead of time, gesture recognition is achieved by supervised learning, in which a supervisor modifies the training data collection, defining gestural instances and the corresponding class details for each gesture instance. We can use the already available mapping function to recognize an unknown gestural instance after the mapping function is constructed from the gestural instances and class details.

Human-computer interaction technology is gradually changing from computer-centric to human-centric. As an important human-computer interaction method, gesture control provides people with a natural and intuitive way of communication. It has been initially used in the entertainment industry and it has strong practicability in robot control [1]. Rich gestures are

difficult to recognize due to the small number and consistency of markers. Along with the development of visual technology, unmarked visual gesture recognition has become the current Mainstream.

The main process of visual gesture recognition includes: (1) Image acquisition: using a visual camera Collect gesture images; (2) Hand detection and segmentation: detect the position of the hand in the palm Gesture image and segment the hand area; (3) Gesture recognition: extract image features Hand area, and recognize the gesture type based on these characteristics.

The process of visual gestures recognition is shown in Figure 1.

Fig 1: The process of visual gestures recognition

Static gesture recognition and dynamic gesture recognition are two types of vision-based gesture recognition technology. The hand is in a static state for static gesture recognition, so the pose, shape, location, and other data details of the hand do not shift. It has the benefit of having a high recognition rate. Static gesture

* Corresponding author: magrouni@gmail.com

recognition, on the other hand, has its own drawbacks. Static movements, for example, can only convey a limited amount of knowledge and do not represent the characteristics of real human hand movement. Static gestures are a special condition of dynamic gestures, since they are composed of static gestures frame by frame.

Dynamic gesture recognition has the advantage of being able to convey more detail with continuously changing gestures, and it is commonly used in the field of human-computer interaction.

Gesture recognition can be classified into two types based on the theory and creation process: gesture recognition using conventional methods and gesture recognition using deep learning. This paper examines and analyzes recent research on visual gesture recognition, as well as the most popular gesture recognition methods. The key technologies used in each gesture recognition process are then compared and analyzed.

Gesture identification and segmentation, gesture monitoring, gesture feature extraction, and gesture classification are examples of these processes. Finally, this article addresses the problems and drawbacks of gesture recognition, as well as potential future research directions.

2 Related works

In recent years, the field of dynamic gesture recognition has gotten a lot of attention. A number of recent studies have attempted to create a more natural interface for interacting with computers and other devices. Recognizing a complex gesture is difficult, however, since a gesture can be interpreted in a variety of ways within the same context. Furthermore, the manner in which gestures are performed is influenced not only by the sequence of body movements, but also by the cultural background of those who perform the gestures. As a result, there is still a lot of work to be done in order to create an interface that allows humans and machines to communicate effectively using gestures.

However, it appears that gestures, their nature, applicability, fitness for function, and effectiveness were not always the primary focus in the articles reporting on the prototypes produced. Gestures were either the only mode of interaction or one of many, but neither mode seemed to integrate the functions that gestures usually serve in communication into the interfaces.

The review research methods of visual gesture recognition will be grouped according to the following family: static, dynamic, based on the supports (Kinect, Leap...etc), works that focus on the application of gesture recognition on robots and works on dealing with gesture recognition at the browser level

Ameur, Safa, et al [14], authors suggest a dynamic hand gesture recognition approach over a Leap Motion system in this paper, which uses touchless hand movements. To begin, they use recurrent neural networks with Long Short-Term Memory (LSTM) to

evaluate the sequential time series data collected from Leap Motion for recognition purposes. Basic unidirectional and bidirectional LSTM are used separately. The final prediction network, called Hybrid Bidirectional Unidirectional LSTM, is generated by combining the aforementioned models with additional components (HBU-LSTM)

Santos, Clebeson Canuto dos, et al [15] proposes the dynamic gesture recognition technique. The plan is made up of two key steps: • Pre-processing: using a modified version of the aforementioned star representation each input video is represented as an RGB image. • Classification: An ensemble of CNNs is used to train a dynamic gesture classifier. The image from the pre-processing phase is fed into two CNNs that have already been trained. After passing through a soft-attention mechanism, the effects of these two CNNs are weighted and sent to a completely connected sheet. Finally, a softmax classifier determines which class the gesture belongs.

Almasre et al[17] proposed a dynamic prototype model (DPM) that recognizes some ARSL gestured dynamic words using Kinect as a sensor. The DPM used eleven predictive models based on three algorithms' various parameter settings (SVM, RF, and KNN). The SVM models obtained the highest recognition accuracy rates for the complex terms gestured, according to the research findings.

Egemen et al[25] broadened the finite-state-machine (FSM) method in our hand gesture interface by allowing users to perform unique GUI actions by using gesture-specific attributes including distance between hands, distance from the camera, and time of occurrences. The RealSense SDK, which is used in our hand gesture interface, detects hand movements and extracts these attributes. These gesture-specific attributes allow users to trigger static gestures and execute them as dynamic gestures. We also added extra features to our hand gesture GUI to improve its performance, convenience, and user-friendliness.

In an Operating room environment without personal basis training and skeleton features from the Leap MotionTM sensor, A-reum Lee et al[27] used a gesture-based interface to identify five hand gestures and compared it to various deep learning algorithms such as DCNNs and CapsNet. CapsNet performed best in recognizing complex hand movements, which could be used as a non-contact interface in the OR to monitor clinical software.

Rihem Mahmoud et al [28] suggested a new recognition method to solve the issue of large-scale continuous gesture recognition using depth and gray-scale input images. The proposed recognition scheme is divided into three stages. To begin, continuous gesture sequences are segmented into isolated gestures using mean velocity information derived from deep optical flow estimation. Deep signature features are a collection of relevant descriptors extracted for each isolated segment in order to describe different intensities and spatial information describing the movement's location, velocity, and orientation.

M. Meghana et al [29] develops a robotic vehicle model that is powered by speech signals and hand movements.

The main component of this model will be an Android smartphone that will communicate with the robot through Bluetooth. This technique can be used to help people with disabilities or in industrial applications such as working robots powered by voice and hand movements.

The circuit is divided into two parts: hand motion recognition and voice recognition. Transmission and receiver modules make up the portion of the robot that deals with hand gestures. A software framework and a Bluetooth module are used for voice recognition. The voice and gesture recognition software is written using the Arduino IDE, which connects the microcontroller hardware to dump programs. This robot is around 5 different hand gesture inputs through a sensor MPU6050, namely the stop state, forward, backward, right, and left movements. When a user gives the voice commands "FRONT", "BACK", "LEFT", "RIGHT", and "STOP", it works similarly.

Yao Huang et al [30] propose a new real time hand gesture recognition method. Since fingers are the most important clue for hand gesture classification, a finger-emphasized multi-scale descriptor is proposed. To create a discriminative representation of the hand form, the proposed descriptor combines three types of parameters on multiple scales. For hand motion study, the characteristics of fingers are often emphasized. The DTW, SVM, and neural network are then used to investigate three hand gesture recognition solutions. Extensive tests have been carried out, with the findings demonstrating that the proposed approach is resistant to noise, articulations, and rigid transformations.

3 DEEP LEARNING FEATURES SUPPORTED IN BROWSERS

Even though body movements are the most natural means of human interaction, since the advent of the Web in the early 1990s, the computer mouse has become an indispensable tool for communicating with web pages. [26] With technical advances in computing power and Web browsers, it could be possible to substitute mouse commands with human gestures using a webcam.

3.1 Frameworks that have been chosen

Google's TensorFlow.js [18] is an in-browser machine learning library that allows users to define, train, and run models entirely in the browser using JavaScript. It replaces deeplearn.js, which is now known as TensorFlow.js Core. TensorFlow.js is a high-level API for defining models that is controlled by WebGL. Both Keras layers are supported by TensorFlow.js (including Dense, CNN, LSTM, and so on). Therefore, it is easy to import models pre-trained by the native TensorFlow and Keras into the browser and run with Tensorflow.js.

ConvNetJS [19] is a Javascript library created by Stanford's Andrej Karpathy. The entire library is focused on the transformation of three-dimensional numerical quantities. ConvNetJS currently supports common neural network models and cost functions for classification and regression. Furthermore, it supports convolutional networks, and an experimental

reinforcement learning. Unfortunately, although ConvNetJS might be the most famous framework before TensorFlow.js, it is no longer maintained after Nov. 2016.

Keras.js [20] abstracts away a variety of backend frameworks, including TensorFlow, CNTK, and others. It allows you to import Keras-trained models for inference. WebGL performs the computation in the GPU mode. This project, however, is no longer involved.

WebDNN [21], released by the University of Tokyo, claims to be the fastest DNN execution framework in browsers. It supports only the inference tasks. The framework supports 4 execution backends: WebGPU, WebGL, WebAssembly, and fallback pure JavaScript implementation.

Mind [22] is a neural network library with a lot of flexibility. The core system has only 247 lines of code and processes training data using a matrix implementation. It supports customization of the network topology and plugins to configure pre-trained models created by the mind community. However, this framework is no longer active.

Technological advancements [24] in computing power and Web browsers have potentially made it possible to replace mouse commands with human gestures through the usage of a webcam. To benefit from the potential, author created an open source library, NoTouch.js, for web developers which allows development of web pages that can be controlled with human gestures on existing Web browsers.

4 process for improving user interface control based on gesture recognition

It is necessary to define a process for improving user interface control based on gesture recognition to streamline the implementation of this mechanism.

This process should define the phases, activities and artifacts that facilitate the identification, specification and implementation of mechanisms.

This process which brings an answer to the constraints and precise criteria such as:

- The treatments must be real time.
- The methods must be robust to the acquisition conditions.
- The constraints imposed on users must be minimal.
- Make Web site Touchless without impacting the existing.

This process will consist of the following phases:

- Preparation

This phase captures requirements, in order to produce a model focused on the end-users' needs.

The results of the analysis are not dependent on any particular technology.

This branch is articulated on the two steps: Definition of requirements and identification of actions to be implemented.

- Adaptation

Following the appearance of new events: the user interfaces can at a time t_i can evolve to another user interface C_j at time t_j . Therefore this mechanism must change its behavior according to the context.

In order to solve the problem of adapting user interfaces based on gesture recognition to changes in context, namely information on the characteristics, preferences and knowledge of the user:

We propose some adaptation steps and among the steps: definition of context parameters

What will be the added value of a user interface improvement system based on gesture recognition if it does not take into account the possible contexts? With this in mind, we will identify all the variables that can modify the behavior of the system.

However, these variables must be classified in 3 categories:

Environment: the parameters of this category are determined mainly by the business analysts, since they are the most capable of defining information about the service infrastructure and the spatio-temporal environment that surrounds it.

User: this category of context specifies information about the example user (profile, language, preferences, location, etc.).

Device: contains the parameters that describe the device, it is composed of two categories software (e.g., operating system, navigator type, supported type of data, etc.) and hardware (e.g., Kinect, leap motion, camera).

- Construction:

This phase concerns the implementation of this mechanism, for this reason, we may propose in the future a framework based on TensorFlow.js

- Test:

Finally a phase of testing is fundamental to be able to avoid a risk of regression and to check the 2 constraints which concern the precision and real time.

Fig 2 : Process for improving user interface control based on gesture recognition

5 Discussion

5.1 Problems with gesture recognition on a technical level

In recent years, gesture recognition technology has advanced rapidly. However, due to the intervention of external environmental influences and the various shortcomings of the gesture itself, it is easy for the device to have a variety of consequences, making gesture recognition insurmountable. This paper summarizes the following technical difficulties, which are used as a guideline for relevant scientific research personnel, in order to improve the way of human-computer interaction using gesture recognition.

Segmentation of gestures: Most approaches have succeeded in recognizing isolated scenes at this time, but the dynamic context environment variables can be changed. When the illumination changes, the background have similar skin colours, etc., the detection, tracking and segmentation of gestures in the video will bring great difficulties.

Identification of gestures for occlusion or blind spots: When a human hand travels across space, there are often issues with objective occlusion or gesture recognition due to blind spots. Gesture monitoring and identification are greatly hampered as a result of this situation.

Gestural variety : The human hand has 27 degrees of freedom, which is the same as a highly deformable body. Changes in place and rotation are also part of the hand's movement. As a result, the human hand can perform a wide range of complex and varied movements, making hand feature analysis extremely difficult.

Recognition of real-time gestures : Since the input is a video image sequence with high image pixels and a large number of processed data, the device must be able to process a large amount of data rapidly in order to recognize the gesture model, which has high computer

hardware requirements. For real-time gesture recognition, this has also become a difficult issue.

5.2 The proposed process

The proposed approach provides a complete development cycle for improving user interface control based on gesture recognition to streamline the implementation of this mechanism.

However, the process is not detailed enough and is not illustrated through a case study which makes it difficult to analyse its real capabilities.

The process that we have proposed needs to be developed in several dimensions. Especially, to introduce other parameters related to the context and the user

6 Conclusion

This paper examines the most common vision-based gesture recognition methods in recent years, discusses several representative methods of gesture recognition based on conventional methods and gesture recognition based on deep learning, summarizes the benefits and drawbacks of different methods, and highlights current research technological challenges and development trends. With the advancement of human-computer interaction and the maturation of deep learning and other fields of study, efficient, accurate, and effective gesture recognition systems will undoubtedly be built to improve people's lives.

We're trying to figure out if touch-free inputs through human gestures will improve the usability of a web page and whether touch-free can improve the usability of a web page by making the interaction more natural by using the library

We have presented a process for improving user interface control based on gesture recognition. This process defines the phases, steps, activities and artifact.

The main phases of this process are: 1) Preparation; 2) Elaboration; 3) Construction; 4) Transition.

As part of our work on adaptable user interface control based on gesture recognition, we are currently finalizing this party.

We are also putting the last touches on another proposal taking into account several dimensions.

References

1. K Qian, Niu, H Yany. Developing a gesture based remote human-robot interaction system using kinect [J]. International Journal of Smart Home, 2013, 7 (4): 203-208.
2. S Bhowmick, A K Talukdar, K K Sarma. Continuous hand gesture recognition for English alphabets [C]//International Conference on Signal Processing and Integrated Networks, IEEE, 2015: 443-446.
3. D Tang, H J Chang, A Tejani, et al. Latent regression forest: structured estimation of 3d hand poses [J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2017, 39 (7): 1374-1234.
4. S Q Ren, K M He, R Girshick, et al. Faster R-CNN: Towards real-time object detection with region proposal networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39 (6): 1137-1149.
5. C. Li, Y. Hou, P. Wang, W. Li, Joint distance maps based action recognition with convolutional neural networks, IEEE Signal Process. Lett. 24 (5) (2017) 624–628, doi: 10.1109/LSP.2017.2678539 .
6. X. Liu, G. Zhao, 3d skeletal gesture recognition via sparse coding of time- warping invariant riemannian trajectories, in: Proceedings of the International Conference on Multimedia Modeling, Springer, 2019, pp. 678–690, doi: 10.29007/xhfp .
7. X. Liu, H. Shi, X. Hong, H. Chen, D. Tao, G. Zhao, Hidden states exploration for 3d skeleton-based gesture recognition, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019, pp. 1846–1855, doi: 10.1109/WACV.2019.00201 .
8. X. Chen, M. Koskela, Using appearance-based hand features for dynamic RGB-D gesture recognition, in: Proceedings of the International Conference on Pattern Recognition, 2014, pp. 411–416, doi: 10.1109/ICPR.2014.79 .
9. A. Yao, L.V. Gool, P. Kohli, Gesture recognition portfolios for personalization, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 1923–1930, doi: 10.1109/CVPR.2014.247 .
10. D. Wu, L. Shao, Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 724–731, doi: 10.1109/CVPR.2014.98 .
11. B. Fernando, E. Gavves, M. José Oramas, A. Ghodrati, T. Tuytelaars, Modeling video evolution for action recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June, 2015, pp. 5378–5387, doi: 10.1109/CVPR.2015.7299176 .
12. E. Escobedo-Cardenas, G. Camara-Chavez, A robust gesture recognition using hand local data and skeleton trajectory, in: Proceedings of the IEEE International Conference on Image Processing
13. A. Joshi, C. Monnier, M. Betke, S. Sclaroff, Comparing random forest approaches to segmenting and classifying gestures, Image Vis. Comput. 58 (2017) 86–95, doi: 10.1016/j.imavis.2016.06.001 .

14. Ameur, Safa, et al. « A Novel Hybrid Bidirectional Unidirectional LSTM Network for Dynamic Hand Gesture Recognition with Leap Motion ». *Entertainment Computing*, vol. 35, août 2020, p. 100373. DOI.org (Crossref), doi:10.1016/j.entcom.2020.100373.
15. Santos, Clebeson Canuto dos, et al. « Dynamic Gesture Recognition by Using CNNs and Star RGB: A Temporal Information Condensation ». *Neurocomputing*, vol. 400, août 2020, p. 238-54. DOI.org (Crossref), doi:10.1016/j.neucom.2020.03.038.
16. Mahmoud, Rihem, et al. « Deep Signature-Based Isolated and Large Scale Continuous Gesture Recognition Approach ». *Journal of King Saud University - Computer and Information Sciences*, septembre 2020, p. S1319157820304559. DOI.org (Crossref), doi:10.1016/j.jksuci.2020.08.017.
17. Almasre, Miada A., et Hana Al-Nuaim. « A Comparison of Arabic Sign Language Dynamic Gesture Recognition Models ». *Heliyon*, vol. 6, no 3, mars 2020, p. e03554. DOI.org (Crossref), doi:10.1016/j.heliyon.2020.e03554.
18. 2018. TensorFlow.js. <https://js.tensorflow.org/>.
19. 2018. ConvNetJS. <https://cs.stanford.edu/people/karpathy/convnetjs/>.
20. 2018. Keras.js. <https://github.com/transcranial/keras-js>.
21. 2018. WebDNN. <https://github.com/mil-tokyo/webdnn>.
22. 2018. Mind. <https://github.com/stevenmiller888/mind>.
23. JSHG."Javascript Hand Gesture Plugin", <https://nhudinhtuan.github.io/jshg/> (current March 5, 2018)
24. Korhan Akcura , NoTouch.js "A JavaScript Library for Touch-Free Web Browsing"
25. Egemen Ertugrul, Ping Li, Bin Sheng: On attaining user-friendly hand gesture interfaces to control existing GUIs. *Virtual Real. Intell. Hardw.* 2(2): 153-161 (2020)
26. Sharma, Ram Pratap, and Gyanendra K. Verma. "Human computer interaction using hand gesture." *Procedia Computer Science* 54 (2015): 721-727
27. Lee, A-reum, et al. « Enhancement of Surgical Hand Gesture Recognition Using a Capsule Network for a Contactless Interface in the Operating Room ». *Computer Methods and Programs in Biomedicine*, vol. 190, juillet 2020, p. 105385. DOI.org (Crossref), doi:10.1016/j.cmpb.2020.105385.
28. Mahmoud, Rihem, et al. « Deep Signature-Based Isolated and Large Scale Continuous Gesture Recognition Approach ». *Journal of King Saud University - Computer and Information Sciences*, septembre 2020, p. S1319157820304559. DOI.org (Crossref), doi:10.1016/j.jksuci.2020.08.017.
29. Meghana, M., et al. « Hand Gesture Recognition and Voice Controlled Robot ». *Materials Today: Proceedings*, vol. 33, 2020, p. 4121-23. DOI.org (Crossref), doi:10.1016/j.matpr.2020.06.553.
30. Huang, Yao, et Jianyu Yang. « A Multi-Scale Descriptor for Real Time RGB-D Hand Gesture Recognition ». *Pattern Recognition Letters*, vol. 144, avril 2021, p. 97-104. DOI.org (Crossref), doi:10.1016/j.patrec.2020.11.011.