A Gesture Recognition Framework for Exploring Museum Exhibitions

Vincenzo Agate, Salvatore Gaglio
{vincenzo.agate,salvatore.gaglio}@unipa.it
University of Palermo, Viale delle Scienze, ed. 6, 90128, Palermo, Italy

ABSTRACT

In this paper we present a gesture recognition framework for providing the visitors of a museum exhibition with a non intrusive interface for the multimedia enjoyment of digital contents. Early experiments were carried out at the Computer History Museum Exhibition of the University of Palermo.

CCS CONCEPTS

• **Human-centered computing → HCI theory, concepts and models**;

KEYWORDS

Gesture recognition, Human computer interaction, Ambient intelligence

ACM Reference Format:
Vincenzo Agate, Salvatore Gaglio. 2018. A Gesture Recognition Framework for Exploring Museum Exhibitions. In *AVI '18: 2018 International Conference on Advanced Visual Interfaces, AVI '18, May 29-June 1, 2018, Castiglione della Pescaia, Italy.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3206505.3206576

1 INTRODUCTION

This work presents a novel system enabling advanced and immersive fruition of digital cultural contents. The main idea is to provide the users, e.g., the visitors of a museum, with the opportunity to interact and immerse themselves in a virtual environment in which objects can be accessed and controlled [9].

We consider a scenario where a physical museum exhibition includes a collection of digital items representing artifacts of historical, artistic, or scientific interest, that can be enjoyed by interacting with a multimedia kiosk. To this aim, an unobtrusive RGB-D camera is used to capture color (RGB) and depth (D) images that allow to infer the user's posture in real-time. RGB-D information is analysed by means of three different machine learning techniques to recognize both the actions performed by the user, and the hand gestures used to interact with the virtual environment [12].

The next Section provides an in-depth description of the human gesture recognition algorithms underlying the system. A case study deployed at the University of Palermo is presented in Section 3. Conclusions and future work will follow in Section 4.

2 GESTURE RECOGNITION SYSTEM

Vision-based gesture recognition is a challenging task due to several constraints [1]. Some methods rely on wearable devices that can be used to explicitly track the hand gestures [10, 11]. In order to provide the users with an intuitive and easy to use interface, we focused on the adoption of a sensing device capable of capturing the users' movements by means of RGB-D motion sensors, namely the Microsoft Kinect [2].

The principle on which the whole recognition process is based is quite intuitive: assuming that a gesture can be described as a sequence of postures, we are interested in tracking some salient points of the human body to infer the corresponding postures. To this aim, we use a combination of three different machine learning techniques, similarly to what described in [8].

The tracking of a set of relevant points of the human body is performed by exploiting the skeleton detection algorithm provided by the Microsoft APIs. Such a tool allows to obtain the coordinates (x, y, d) of some body joints, where the couple (x, y) indicates the position of the joints with respect to the image plane, and d represents their distance from the acquisition device.

At each time step, a particular distribution of these joints defines a *posture*, whose sequence can be used to recognize a certain gesture. In order to reduce the number of all the possible postures, a clustering step based on K-means [13], is performed. The objective of this phase is to process the set of joints captured for a specific gesture so as to to identify the K most relevant postures. The K centroids computed by K-means are used to train a set of specialized multi-class Support Vector Machines (SVMs) [15] that will be able to classify future postures as one of the K already known. Combining K-means and SVMs [17] allows not only to obtain a more compact representation of the posture sequences, but also to merge all the consecutive instances of the same posture allowing to identify the same gesture performed at different speeds. This type of representation will therefore increase the capability the system of recognizing a gesture performed by different subjects.

The last step is to model the gestures as a sequence of postures. This modeling and recognition process is carried out by means of a classifier based on Hidden Markov Models (HMMs) [14]. If we want to recognize N gestures, N different HMMs are trained to capture a specific sequence of the transitions between one posture and the other. During the recognition phase, the movements of the users are expressed in terms of postures, each posture sequence is tested against the N available HMMs, and classified according to the highest posterior probability. If none of the HMMs returns a probability exceeding a certain threshold, the gesture is classified as unknown. The phases of the recognition process are summarized in Fig 1.

AVI '18, May 29-June 1, 2018, Castiglione della Pescaia, Italy

ACM ISBN 978-1-4503-5616-9/18/05.
https://doi.org/10.1145/3206505.3206576

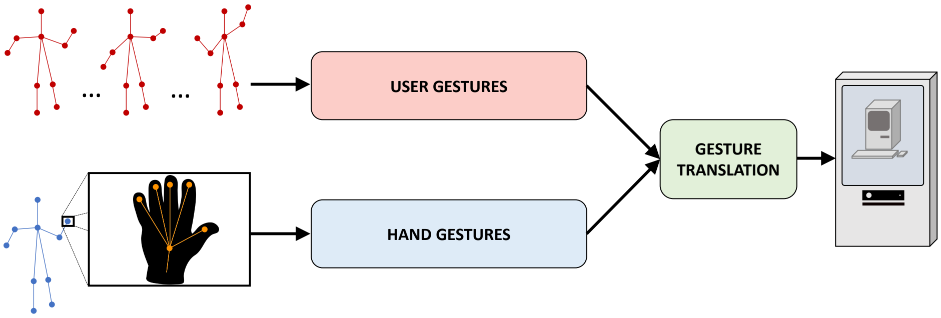

Figure 1: System architecture.

This processing chain allows to recognize the following user's gestures: *Horizontal arm wave*, *high arm wave*, *two hand wave*, *high throw*, *draw X*, and *draw tick*. These gestures enable the principal interactions between the user and the kiosk, e.g., allowing to activate the navigation menus, browse the catalogue of virtual objects, move from to one window to another, and so on.

A more specific set of gestures performed by the user's hand have also been considered to achieve advanced tasks, i.e., controlling the movements of a certain object in the virtual environment (i.e., *fingers open*, *fingers close*, *thumbs up*, *thumbs down*, *fist*, and *OK*).

In order to achieve this goal, the position of the user's wrist is detected by using the Kinect APIs; then, the region of pixels identifying the hand of the subject is isolated and some features of interest are extracted to describe the appearance of the hand [7, 16]. For example, to enlarge a photo, the user can place both hands open in front of the acquisition device, whilst an object can be grabbed by opening and closing one hand.

The gesture recognition process is summarized in Fig. 1. The gestures performed by the user are processed by two software modules responsible for recognizing the *user's gestures* and *the hand gestures* respectively. The output of both modules is interpreted by a translation module, which will transform high-level gestures into low-level navigation commands.

For instance, one of the gestures to be recognized could consists in the user drawing a tick, which could correspond to a low-level command that enables the selection of an item inside the virtual environment. The recognition of other, more complex, gestures allows to accomplish other tasks, such as detecting if and when a visitor wants to interact with a generic virtual object by rotating, moving, enlarging it, and so on. Moreover, the system can detect and manage the presence of more than one user at a time.

3 CASE STUDY

The system presented in this work will be used within the Computer History Museum Exhibition at the University of Palermo, where a collection of artifacts related to the history of computing are exposed. The purpose of this exhibition is to safeguard, preserve and show to the public the artifacts of the information age, while also investigating the history of modern information systems and their impact on the society. In this context, the proposed system will stimulate the visitors' curiosity by providing an easy to use interface to delve deeper into the secrets of the exposed devices. It is worth noticing that, thanks to its generality, our system may be successfully adopted in other smart environments, such as smart buildings [5, 6], in order to promote a non intrusive interaction with the users [3], and the design of context-aware systems [4].

A system prototype has been implemented by using the Microsoft Kinect v2 as acquisition device, whilst the gesture recognition algorithms were run on a single-board and low-cost computer, namely the Raspberry Pi 3 model B. The software modules for the acquisition, processing and recognition of gestures have been implemented on Raspbian OS by exploiting openFrameworks, an open source C++ development environment that allows to easily use some image processing libraries (e.g., OpenCV), and manage any video capture device. Nevertheless, this cross-platform framework would allow to quickly port our solution to different OSs, such as OS X, Windows, Linux, iOS, Android, embedded Linux ARM.

A prototype has been designed to recognize three classes of gestures: user gestures, hand gestures, and composite gestures. For example, the "horizontal arm wave" is used to activate or deactivate the kiosk, whilst the "fingers open" indicate that the user is ready to select an item. Composite gestures are obtained by combining the two sets. A practical example of this phenomenon occurs when the user tries to grab and move a virtual object. This action can be decomposed into a sequence of simpler gestures: i.e., *fingers open* (select), *fingers close* (grab), *horizontal arm wave* (move), and *fingers open* (release).

4 CONCLUSIONS AND FUTURE WORK

This paper presented a gesture recognition system to be used within a museum exhibition to provide the visitors with advanced, not intrusive, interfaces. The idea is to extend the physical collection by allowing the visitors to interact with virtual objects through a motion sensor device. Two different algorithms allow to detect and recognize both the users' movements and the hand gestures, so as to translate them into low-level commands enabling the navigation in a digital environment. A prototype of this system has been implemented at the University of Palermo and preliminary results are very promising. An in-depth experimental evaluation on a multi-user environment will be soon performed by exploiting real-data from the multimedia kiosks available at the Computer History Museum of the campus. This phase will allow to measure the performances of the systems according to the feedbacks provided by the users.

REFERENCES

[1] B. K. Chakraborty, D. Sarma, M. K. Bhuyan, and K. F. MacDorman. 2018. Review of constraints on vision-based gesture recognition for human-computer interaction. *IET Computer Vision* 12, 1 (2018), 3–15. https://doi.org/10.1049/iet-cvi.2017.0052

[2] P. Cottone, G. Lo Re, G. Maida, and M. Morana. 2013. Motion sensors for activity recognition in an ambient-intelligence scenario. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on.* IEEE, 646–651. https://doi.org/10.1109/PerComW.2013.6529573

[3] A. De Paola, A. Farruggia, S. Gaglio, G. Lo Re, and M. Ortolani. 2009. Exploiting the human factor in a WSN-based system for ambient intelligence. In *Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09. International Conference on.* IEEE, 748–753.

[4] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Lo Re, and S. K. Das. 2017. An adaptive bayesian system for context-aware data fusion in smart environments. *IEEE Transactions on Mobile Computing* 16, 6 (2017), 1502–1515.

[5] A. De Paola, S. Gaglio, G. Lo Re, and M. Ortolani. 2009. An ambient intelligence architecture for extracting knowledge from distributed sensors. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human.* ACM, 104–109.

[6] A. De Paola, G. Lo Re, M. Morana, and M. Ortolani. 2012. An intelligent system for energy efficiency in a complex of buildings. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2012.* IEEE, 1–5.

[7] S. Gaglio, G. Lo Re, M. Morana, and M. Ortolani. 2013. Gesture Recognition for Improved User Experience in a Smart Environment. In *AI*IA 2013: Advances in Artificial Intelligence.* Lecture Notes in Computer Science, Vol. 8249. Springer International Publishing, 493–504. https://doi.org/10.1007/978-3-319-03524-6_42

[8] S. Gaglio, G. Lo Re, and M. Morana. 2015. Human Activity Recognition Process Using 3-D Posture Data. *IEEE Transactions on Human-Machine Systems* 45, 5 (Oct 2015), 586–597. https://doi.org/10.1109/THMS.2014.2377111

[9] L. A. Hernández-Ibáñez, V. Barneche-Naya, and R. Mihura-López. 2016. A comparative study of walkthrough paradigms for virtual environments using kinect based natural interaction. In *2016 22nd International Conference on Virtual System Multimedia (VSMM).* 1–7. https://doi.org/10.1109/VSMM.2016.7863168

[10] B. G. Lee and S. M. Lee. 2018. Smart Wearable Hand Device for Sign Language Interpretation System With Sensors Fusion. *IEEE Sensors Journal* 18, 3 (Feb 2018), 1224–1232. https://doi.org/10.1109/JSEN.2017.2779466

[11] D. L. Lee and W. S. You. 2018. Recognition of complex static hand gestures by using the wristband-based contour features. *IET Image Processing* 12, 1 (2018), 80–87. https://doi.org/10.1049/iet-ipr.2016.1139

[12] G. Lo Re, M. Morana, and M. Ortolani. 2013. Improving User Experience via Motion Sensors in an Ambient Intelligence Scenario.. In *Pervasive and Embedded Computing and Communication Systems (PECCS), 2013.* 29–34.

[13] J. Macqueen. 1967. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability.* 281–297.

[14] L. R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (Feb 1989), 257–286. https://doi.org/10.1109/5.18626

[15] Bernhard Scholkopf and Alexander J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA.

[16] C. Wang, Z. Liu, and S. C. Chan. 2015. Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera. *IEEE Transactions on Multimedia* 17, 1 (Jan 2015), 29–39. https://doi.org/10.1109/TMM.2014.2374357

[17] Yukai Yao, Yang Liu, Yongqing Yu, Hong Xu, Weiming Lv, Zhao Li, and Xiaoyun Chen. 2013. K-SVM: An Effective SVM Algorithm Based on K-means Clustering. *JCP* 8 (2013), 2632–2639.