NeuronUnityIntegration2.0. A Unity Based Application for Motion Capture and Gesture Recognition

Federico Becattini, Andrea Ferracani, Filippo Principi, Marioemanuele Ghianni, Alberto Del Bimbo

[name].[surname]@unifi.it

MICC Media Integration and Communication Center, University of Florence

Florence, Italy

ABSTRACT

NeuronUnityIntgration2.0 [1] is a plugin for Unity which provides gesture recognition functionalities through the Perception Neuron motion capture suit. The system offers a recording mode, which guides the user through the collection of a dataset of gestures, and a recognition mode, capable of detecting the recorded actions in real time. Gestures are recognized by training Support Vector Machines directly within our plugin. We demonstrate the effectiveness of our application through an experimental evaluation on a newly collected dataset. Furthermore, external applications can exploit NeuronUnityIntgration2.0's recognition capabilities thanks to a set of exposed API.

CCS CONCEPTS

• **Human-centered computing** → **Gestural input**; **Virtual reality**; *Interactive systems and tools.*

KEYWORDS

Motion Capture, Gesture Recognition, Support Vector Machines, Virtual Reality, Unity

1 INTRODUCTION

Motion capture has gained attention thanks to its vast use in developing products for two of the most profitable fields of entertaining: cinematography and gaming. Paired with motion recognition though, the focus has started to spread towards a broader spectrum of fields such as human-computer interaction [8], IoT [5] and medical rehabilitation [7]. In health-care, motion capture has also been used as a preemptive orthopedic technique [1] and for aiding deaf people by recognizing and translating sign language [6]. Motion capture is also useful for training athletes to improve the execution

---

[1]Demo video is avilable at http://tiny.cc/u1lz6y

of actions such as kicks or swings [2, 3]. The biggest obstacle for a pervasive diffusion of this technology though has been the cost of professional motion capture equipment which is usually around several tens of thousands of dollars, making the technology not accessible for casual users or even small-sized motion recognition companies.

Recently some more affordable solutions for motion capture have seen the light such as Rokoko[2] and Noitom's Perception Neuron[3]. These products offer low-cost affordable motion capture suits, yet still providing basic functionalities and integration with software development platforms such as Unity[4]. In this demo we are going to present *NeuronUnityIntegration2.0*, a library for motion recognition based on the Perception Neuron suit which can be both integrated in Unity as a custom build or used as an API for gesture recognition in external applications. Our system is fully customizable, allowing the users to record poses and gestures and train a discriminative model to recognize them.

2 PERCEPTION NEURON

Perception Neuron is a motion capture suit composed by 32 small wearable *Neuron* sensors which provide readings from 9-axis IMU (gyroscope, accelerometer and magnetometer) at 60 FPS. A micro-controller embedded in the suit acts as a power supplier for the sensors and aggregates the data before streaming it via USB or Wi-Fi to a workstation.

The suit is shipped with Axis Neuron, a tool for sensor calibration and real time visualization of the user's movements through a 3D avatar. Using Axis, motion can be recorded and exported to other frameworks such as Unity and Maya. An SDK for Unity is also provided which offers a series of scripts to communicate with Axis via socket for real time animation.

3 NEURONUNITYINTEGRATION2.0

We propose NeuronUnityIntegration2.0, a plugin for Unity which adds motion capture functionalities exploiting the Perception Neuron suit and custom gesture recognition. The plugin can run either in *record mode* or *recognition mode* and does not require any technical effort and knowledge in order to train the gesture classifier.

In record mode, the user can collect a dataset of gestures and poses (Fig. 1). Motion is shown in the UI in real time as the gestures are performed but pre-recorded poses can also be loaded to extract only relevant motion and precisely trim the gestures of interest. Through the interface the user can select if the recording is a pose (single frame) or a gesture (sequence of poses) and assign a label

---

[2]https://www.rokoko.com

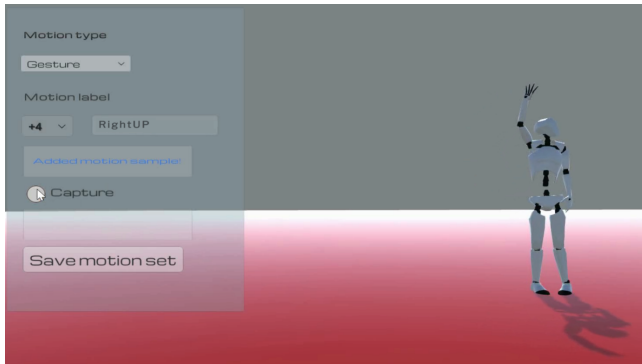[3]https://neuronmocap.com

[4]https://unity.com

Figure 1: In record mode gestures and poses can be recorded by the user. The dataset can then be used to train an SVM classifier. Optimal parameters for training are automatically chosen by the software, but can also be configured.

Figure 2: In recognition mode gestures and poses are automatically recognized in real time. Gesture recognition can be exploited in Unity or by other applications through the standalone tool.

to it. Once a set of samples has been collected, it can be saved as a dataset to train a pose/gesture classifier.

In recognition mode, the user is free to perform any activity while wearing the Perception Neuron suit and any known pose or gesture will be automatically recognized in real time (Fig. 2). When the action is recognized, a confidence score between 0 and 100 is also shown to the user. User defined callbacks can be triggered by recognition events.

NeuronUnityIntegration2.0 provides Unity scripts that can be attached to custom avatars in Unity. However, it can also communicate as a standalone tool with external applications providing gesture recognition API via web socket. Any application can connect to the socket and receive a stream of poses or simply be notified when an event occurs, namely a successfully recognized gesture or pose. A gesture notification can be used as a trigger event for instance in human computer interaction applications.
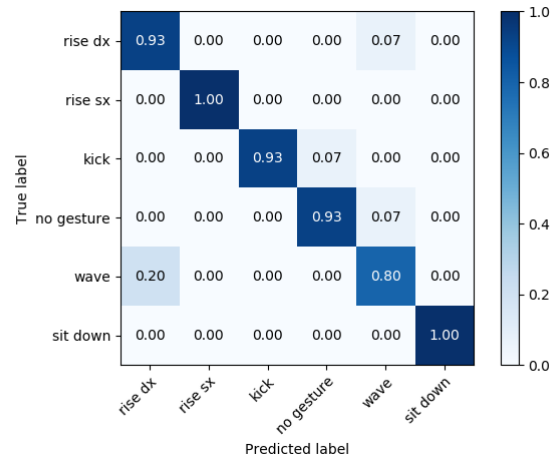

Figure 3: Confusion matrix of recognized gestures. The most difficult gesture recognition to be performed turned out to be *wave* which was sometimes confused with *rise_right_hand* (0.80 vs 0.20)

4 POSE AND GESTURE RECOGNITION

Once a dataset is acquired in record mode, a pose/gesture classifier can be trained and used in recognition mode. Through the interface of NeuronUnityIntegration2.0 the user can configure and train a multi-class Support Vector Machine (SVM) [4]. SVMs are learned with libSVM, which has been included in the application. Poses and gestures are treated differently, since for the former each frame has to be classified, while for the latter the classifier must look at sequences of frames.

Each pose acquired from the Perception Neuron suit is represented as a 96-dimensional feature vector encoding the position of each body joint since each of the 32 sensors are described by 3 spatial coordinates. The classifier therefore discriminates poses as points in a 96-dimensional space.

To classify gestures instead, we uniformly sample 10 poses from each training sample to represent different length sequences with a fixed-size descriptor. Our gesture classifier therefore operates in a $96 \times 10 = 960$-dimensional space. In recognition mode, gestures are recognized in real time by applying the classifier in a sliding window fashion, over the last N frames.

To experimentally evaluate the effectiveness of our proposed approach for gesture recognition, the system has been tested on a dataset with 2 poses and 5 gestures (plus a "no gesture" class), recorded by 13 users of different age, weight, height and gender. Poses are always perfectly recognized. As for gestures, the system obtained an accuracy of 93.3%. All gestures are recognized almost perfectly as shown in the confusion matrix in Fig. 3. The most difficult classes do distinguish have proven to be *wave* and *rise_right_hand* which may share common traits depending on how the action is executed.

ACKNOWLEDGMENTS

REFERENCES

[1] Myroslav Bachynskyi, Antti Oulasvirta, Gregorio Palmas, and Tino Weinkauf. 2014. Is motion capture-based biomechanical simulation valid for HCI studies?: study and implications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3215–3224.

[2] Joseph Baker and Damian Farrow. 2015. *Routledge handbook of sport expertise*. Routledge.

[3] Cristian García Bauza, Juan DâĂŹAmato, Andrés Gariglio, María José Abásolo, Marcelo Vénere, Cristina Manresa-Yee, and Ramon Mas-Sansó. 2012. A tennis training application using 3d gesture recognition. In *International Conference on Articulated Motion and Deformable Objects*. Springer, 239–249.

[4] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2, 3 (2011), 27.

[5] Kieran Gerard Holmes, Adam Coates, and Mohammad Hammoudeh. 2017. Motion capture using the internet of things technology: A tutorial. In *ACM International Conference Proceeding Series*. ACM New York.

[6] Rung-Huei Liang and Ming Ouhyoung. 1998. A real-time continuous gesture recognition system for sign language. In *Proceedings third IEEE international conference on automatic face and gesture recognition*. IEEE, 558–567.

[7] Hossein Mousavi Hondori and Maryam Khademi. 2014. A review on technical and clinical impact of microsoft kinect on physical therapy and rehabilitation. *Journal of medical engineering* 2014 (2014).

[8] Shuhei Sato, Hiroko Kamide, Yasushi Mae, Masaru Kojima, and Tatsuo Arai. 2018. Friendly Motion Learning towards Sustainable Human Robot Interaction. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 848–853.