Recognizing Schaeffer's gestures for robot interaction

Francisco Gomez-Donoso and Miguel Cazorla

Computer Science Research Institute. University of Alicante
P.O. Box 99. E-03080. Alicante. Spain

Abstract. In this paper we present a new interaction system for Schaeffer's gesture language recognition. It uses the information provided by an RGBD camera to capture body motion and recognize gestures. Schaeffer's gestures are a reduced set of gestures designed for people with cognitive disabilities. The system is able to send alarms to an assistant or even a robot for human robot interaction.

Keywords: Shaeffer's gestures, 3d gesture recognition, human robot interaction

1 Introduction

This work focuses on part of the human-robot interaction, which is the branch of computer science and robotics that studies and develops new paths in the communication process between humans and robots. Human-robot interaction is a multidisciplinary field that includes topics such as artificial intelligence, robotics, design, social sciences and natural language understanding.

People with some kind of disability are a group that requires special attention from governments, and people with cognitive disabilities (learning difficulties, cerebral palsy, etc.) are a special group within that. Caregivers and educators need a way to communicate with these people. For that purpose, a special gesture set was developed by Schaeffer et al. [18]. To the best of our knowledge, there does not exist any system that is able to recognize these gestures. In this paper, we present a system that is specially designed for Schaeffer's gestures. The system, besides gesture recognition, is able to send messages to a person or robot to provide human robot interaction.

Gesture languages based on hand poses (i.e. static gestures) or movement patterns (i.e. dynamic gestures) have been used for implementing command and control interfaces [1–4]. Gestures, which involve spontaneous hand and arm movements that complement speech, have proven to be a very effective tool for multimodal user interfaces [5–9]. Objects manipulation interfaces [10–12] use the hand for navigation, selection and manipulation tasks in virtual environments.

Several applications, such as heavy machinery control or manipulators, handling computer-based avatars or musical interaction [13], use the hand as an efficient control device and with a high degree of freedom (DOF). And lastly,

some applications such as surgical immersive VR simulations [14] and training systems (VGX, nd), include the manipulation of complex objects in their own definition.

Almost all human computer interactions (HCI) systems based on gestures use hand movements as the main input. Currently, the most effective motion capture hand tools are electromechanical or magnetic detection devices (data gloves) [15,16]. These devices are placed on the hand to measure the location and angles of the finger joints. They offer the most comprehensive set of real-time measurements, are application-independent and allow full functionality of the hand in HCI systems. However, they have several disadvantages in terms of use, and are very expensive, hinder the movement of the hand, and require complex calibration and installation procedures to obtain accurate measurements.

Computer vision represents a promising alternative to data gloves because of its potential to provide a more natural interaction without intrusive devices. However, there are still several challenges to overcome for it to become more widely used, namely precision, speed processing, and generality, among others. Among the different parts of the body, the hand is the most effective tool for general purpose interaction, due to its communicative and manipulative functionality. Some trends in interaction tend to adopt the two modalities, thus allowing an intuitive and natural interaction.

The paper is structured as follows. In Section 2 we briefly introduce Schaeffer's gestures, what they are used for, and how they can help cognitively disabled people. Section 3 describes the general system architecture and explains the modules of the system: the input data, preprocessing and classification. Next, in Section 4 we present some experiments that run the gesture recognition system with different parameters over a set of recorded gestures in order to test its performance. Finally, we draw some conclusions and outline future work in Section 5.

2 Schaeffer's gestures

In 1980, Schaeffer, Musil and Kollinzas published a book entitled "Total Communication: A signed speech program for non-verbal children" [18] in which they lay the groundwork for interaction among people who are not able to speak, and they describe a complete sign language so that these people can relate to others more effectively.

The speech signed program is an example of a system of signs (as classified by Kiernan [19]) in which the therapist introduces the user to the speech signed language. It follows the structure of oral language, and some spoken words are accompanied with signs. The real strength of this system is that its use is based on the child's overall development framework. The study of common development enables us to understand the communicative disorders that certain diseases cause. We can use this speech signed program without special authorization or training and it can be modified to meet the personal needs of the people who might use it.

Fig. 1. Schaeffer gestures for "Sandwich" and "Water"

Its learning and use do not obstruct or hinder, or, therefore, slow the onset of language, quite the opposite, they promote and motivate language onset and / or development. Both this Alternative Communication System (ACS) and other alternative systems can be not only augmentative enhancers of speech but they "unlock" this way of communication as unique and allow others to be developed. The theoretical basis for ACS appeared in the USA in 1980, and a revised edition was published in 1994. Currently there is no Spanish translation of the original book or its revised edition, but there is an adaptation written by Antonia Rebollo Garcia [20].

This project can recognize a subset of Schaeffer gestures (Water, Help, Sandwich, Sleep, Shower, Sick, Clean, Mom, Dad, Want, Dirty), but we aim to recognize the complete Schaeffer language in the future. The system architecture, the modules that compose it and the information flow is detailed in the following sections.

3 Gestures recognition

3.1 System Overview

In Figure 2 we can see the overview diagram of the system. When a person makes a gesture in front of the camera, the motion is captured by a Kinect camera. This information, summarized and packaged, is what the system understands as a gesture. The "gesture" object is sent to the Gesture Class Pre Selection (GCPS) module, which quickly executes with a naive selection of a subset of possible classes for the gesture, discarding others to improve performance. Then, both the subset of possible-candidates classes and the gesture itself are sent to the classifier. The classifier compares the unknown gesture with every gesture present in the model using Dynamic Time Warping (DTW) [21], and it uses the Nearest Neighbor (NN) algorithm [22] to select the one with the shortest distance, and its class is returned as the tag for the unknown gesture. This result is then sent to a user or robot. The whole process is executed online.

There is also an offline stage that handles the training of the model, the editing process and the condensing process in order to obtain a fitted and error free model.

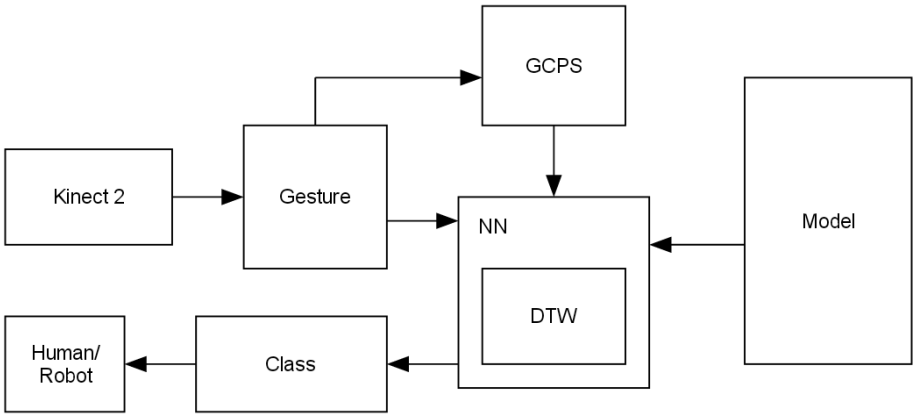

Fig. 2. System architecture

3.2 Kinect v2

Kinect is an RGB-D camera capable of capturing the color and depth of a scene separately. The color stream is obtained with a common high definition RGB digital camera and for the depth stream it sends light beams that reflect on the surfaces and return to the sensor. By measuring the time difference between the emission and reception it calculates the depth of the element reflecting the beam.

This device is capable of capturing color images at a resolution of 1920x1080, while the depth images are at a resolution of 512x424. But Kinect is not only capable of generating this information, it is also able to capture audio and its direction of origin, obtain point clouds with color, segment elements such as bodies and other objects, and most importantly for the task at hand, it is able to detect the joints of the skeleton of a person, which is ideal to detect gestures. Kinect v2 is able to capture up to 25 joints, although our recognition system only uses 11 joints for the upper part of the body of the person, the others are ignored.

3.3 Gesture

Once Kinect has captured these 11 joints of interest, two tasks are carried out before proceeding to the next module of the system: first, the points are grouped by joint type in order to facilitate the DTW comparison process in the classifier

module, and then it runs a downsampling process in order to speed up the system. Kinect captures information at 30 fps, which mean 330 points per second as long as a gesture lasts. Working with this amount of data means a high computational cost, so the system reduces the information. The downsampling method used is KMeans with 20 centroids.

In addition, it is necessary to obtain independence of the angle at which the subject is located when performing the gesture and its position in the scene. For this purpose, a change of the reference system to the points captured is performed. First, the system obtains two vectors, one from the neck joint to the right shoulder joint, and another from the neck joint to the head joint. These represent the X and Y axes of the new reference system. The Z axis is obtained by performing the cross product of these two vectors. Then, the transformation matrix is calculated from the rotation and translation between the new reference system and the camera reference system. Finally, the transformation matrix is multiplied by all the points that compose the gesture. In Figure 3 some gestures and their associated point cloud are shown.

Fig. 3. "Clean" and "Sleep" gestures with their associated joint point clouds

3.4 Gesture Class Pre Selection

The classification process compares the unknown gesture with all the gestures that make up the model, making it possible to accelerate the process by comparing only with a subset of gestures.

This Gesture Class Pre Selection (GCPS) module implements a series of naive but very fast to evaluate rules which examine the gesture features, such as hands position or point cloud centroids, and is able to discard some classes. For example, if the gesture for "Want" is performed with the hand below the shoulders at all time, all the gestures that are performed above them are automatically discarded. In this way some classes are taken as impossible and discarded, so when the comparison with the model occurs, it contains a reduced set of examples, which leads to an improved run time. The rules are evaluated in order and if the gesture does not meet any of the rules, the classifier runs with the full model.

3.5 Classifiers

In the classification process, distance of dynamic time warping between the unknown gesture and every gesture that composes the model is calculated. The distance between two gestures is calculated by adding the partial distances arising from comparing each point collection of each joint (that is why they are packed in such a way in the Gesture module). The system computes the distance from the unknown gesture and every gesture of the model. The nearest neighbor [22] algorithm is then executed and its class is returned as the label for the unknown gesture.

A philosophy of early abandon is also applied as follows: after each comparison between the unknown gesture and a gesture of the model, a check is run to see if the distance returned is the minimum distance so far, and if so it is stored. This distance is sent to the following comparison as a threshold value and if at some point of the comparison between two gestures the partial distance obtained is gretaer than that threshold, the algorithm finishes. In this way we improve the run time of this module [24].

3.6 Model

The model is composed of all the gestures that the system has learned. These gestures have been obtained from several people who were recorded with Kinect as they performed every gesture several times. Then the gestures were labeled and stored. Within the model there are gestures that were not properly performed, mislabeled or provide redundant or useless information. To eliminate all these problems two processes are carried out.

First, the editing algorithm [25] is applied so that mislabeled gestures are discarded, and then the CNN algorithm [26] is executed. The latter extracts only those examples that actually provide new information to form the final model. This whole training process is performed offline. The model used by the final recognition system is composed of 253 different gestures spread over 11 classes as shown in Fig 4.

4 Experiments

The experimentation consists in using the system to classify a collection of 264 gestures captured with Kinect. In this collection there exist 24 examples of each gesture type performed by five different persons. The classifier was set up with different parameters in order to discover the configuration that provides the best performance:

- Downsampling with 20Means, with GCPS activated and using kNN k=3 for the classifier
- Downsampling with 20Means, with GCPS deactivated and using NN for the classifier

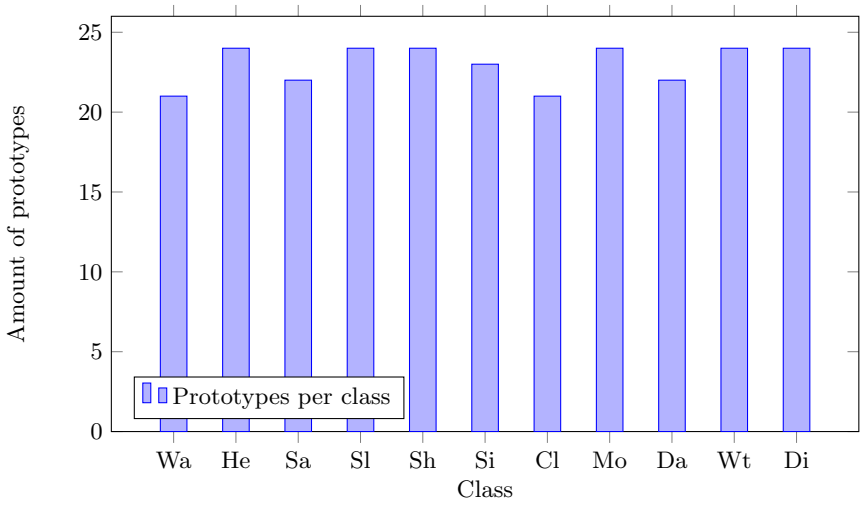

Fig. 4. Amount of prototypes.

– Downsampling with 20Means, with GCPS activated and using NN for the classifier
– Downsampling with 10Means, with GCPS activated and using NN for the classifier

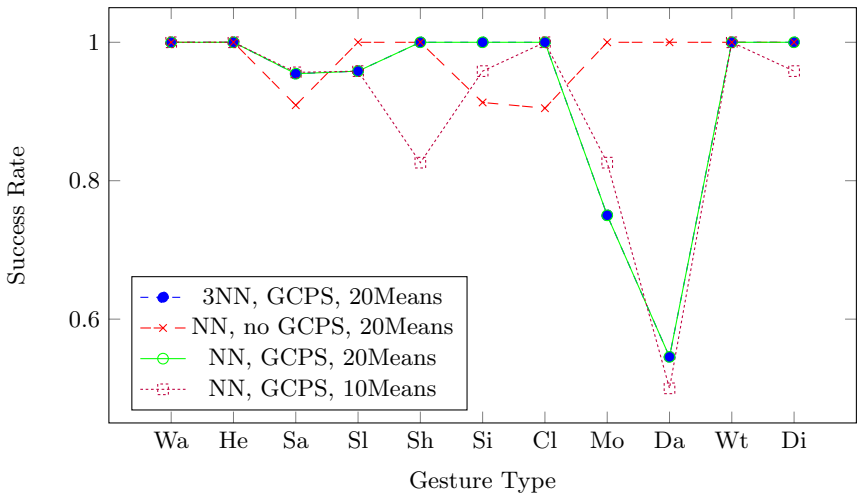

Fig. 5. Success rate with the different parameters.

Figure 5 shows the results for the different parameters. The system set-up that provides the best success rate is the one downsampled with 20Means, with the GCPS module deactivated and using NN for the classifier method. However this configuration requires too much run time, as the following plot shows, making it impractical for real time uses, which is what this project aim to address. Figure 6 shows that the fastest method for gesture classification tasks is the 10Means, with the GCPS module activated and using NN, but its success rate is below the threshold of acceptance. The second fastest system configuration, the one downsampled with 20Means with the GCPS module activated and using NN, provides a very high success rate, making it the best option with a reasonable ratio of success rate to elapsed time. In this figure the elapsed run time of a 5 cross validation lap for these four configurations is shown (a lap is composed of 55 unknown gestures classifications).

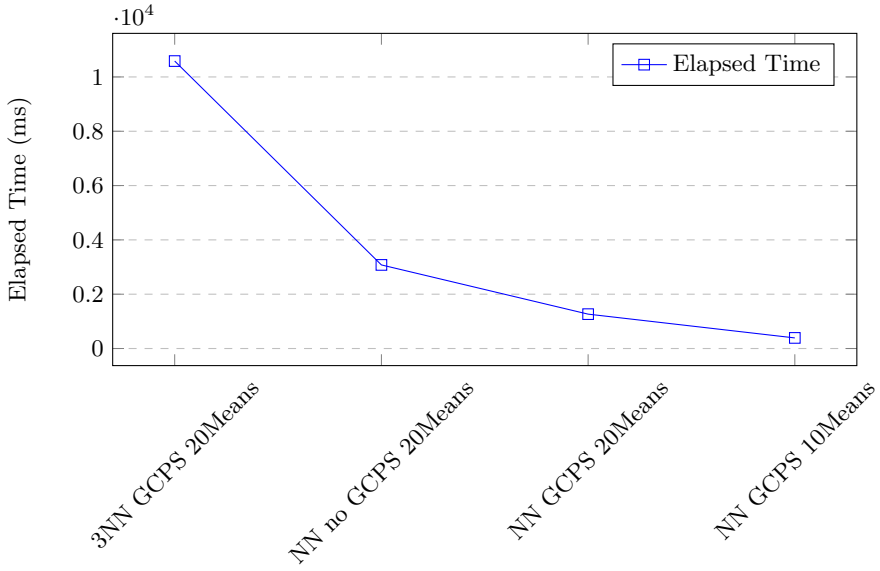

Fig. 6. Execution time for a five cross validation round.

Although the GCPS module introduces an error, it improves the execution time in every case while providing a high success rate, so its use is justified. We can see how the configurations with 20 means improve the execution time as well, while the 10 means provides the best run time but fails in terms of the success rate. Regarding the classification algorithm, the 3 nearest neighbours provides a high success rate but with a prohibitive processing time. Instead of that, the best option is to use the nearest neighbor algorithm, which not only provides a high success rate but is also faster.

So, in the light of the experiments, the best set-up for the gesture classification task is provided by the NN, GCPS activated and 20Means summarized system.

5 Conclusions

This approach provides an innovative, customizable and reliable system for Schaeffer's gesture language detection using Kinect, and oriented to human-robot interaction for everyone, including people with cognitive disabilities, who can use this system to communicate with a person or robot companion.

The system can only recognize a subset of 11 different gesture classes, but we aim, as future work, to recognize the whole of Schaeffer's sign language and implement a system to detect when a gesture starts and ends in order to create a continuous real-time classification system.

Acknowledgments.

This work has been supported by the Spanish Government DPI2013-40534-R Grant, supported with Feder funds.

References

1. F. Quek, "Unencumbered gestural interaction". IEEE Multi-Media 3, 1996.
2. M. Turk, "Handbook of Virtual Environments: Design, Implementation, and Applications". Hillsdale: K.M. Stanney, 2002.
3. S. Lenman, "Using marking menus to develop command sets for computer vision based hand gesture interfaces". NY: ACM Press, 2002.
4. M. Nielsen, "A procedurefor developing intuitive and ergonomic gesture interfaces for HCI". 5th International Gesture Workshop, 2003.
5. A. Wexelblat, "An approach to natural gesture in virtual environments". 1995.
6. F. Quek, "Multimodal human discourse: gesture and speech". 2002.
7. R. Bolt, "Put-that-there: voice and gesture at the graphics interface". NY: ACM Press, 1980.
8. D.B. Koons, "Iconic: speech and depictive gestures at the human-machine interface".NY: ACM Press, 1994.
9. M. Billinghurst," Put that where? Voice and gesture at the graphics interface". 1998.
10. D. Bowman, "Principles for the design of performance-oriented interaction techniques". Hillsdale: Lawrence Erlbaum Associates, 2002.
11. J. Gabbard, "A taxonomy of usability characteristics in virtual environments". Department of Computer Science, University of Western Australia, 1997.
12. V. Buchman, "ingARtips: gesture based direct manipulation in augmented reality". NY: ACM Press, 2004.
13. D. Sturman, "Whole hand input". MIT, 1992.
14. A. Liu, "A survey of surgical simulation: applications, technology, and education". 2003.

15. D. Sturman, "A survey of glove-based input". 1994.
16. E. Foxlin, "Motion tracking requirements and technologies". Hillsdale: Lawrence Erlbaum Associates, 2002.
17. I. Oikonomidis, N. Kyriazis and A.A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect", in Proceedings of the 22nd British Machine Vision Conference, BMVC 2011, University of Dundee, UK, Aug. 29-Sep. 1, 2011.
18. B. Schaeffer, A. Musil and G.Kollinzas, "Total Communication: A Signed Speech Program for Nonverbal Children". Research Press, 1980.
19. C. Kiernan, "Alternatives to speech: A review of research and manual and other forms of communication with the mentally handicapped and other noncommunication populations". British Journal of Mental Subnormality, 1977.
20. A. Rebollo et al., "Diccionario de signos para alumnos con necesidades educativas especiales en el área de comunicación/lenguaje : programa de comunicación total habla signada de B. Schaeffer". Conserjería de Educación y Universidades de la región de Murcia, 2011.
21. H. Sakoe and S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, Acoustics, Speech and Signal Processing, IEEE Transactions on, vol 26, 1978.
22. S. Arya, et al, "An optimal algorithm for approximate nearest neighbor searching in fixed dimensions". University of Maryland, 1994.
23. S. P. Lloyd, "Least square quantization un PCM". Bell Telephone Laboratories Paper, 1982.
24. J. Li and Y. Wang, "EA DTW: Early abandon to accelerate exactly warping matching of time series". College of Computer Science and Technology, Huazhong University of Science and Technology, 2007.
25. D. S. Wilson, "Asymptotic properties of nearest neighbor rules using edited data". IEEE Transactions on Systems, Man, and Cybernetics, vol. smc-2, no 3, 1972.
26. P.E. Hart, The condensed nearest neighbor rule. IEEE Transactions on Information Theory, IT-14(3):515516, 1968.