



**HAL**  
open science

# Distributed Accelerometers for Gesture Recognition and Visualization

Pedro Trindade, Jorge Lobo

► **To cite this version:**

Pedro Trindade, Jorge Lobo. Distributed Accelerometers for Gesture Recognition and Visualization. 2nd Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS), Feb 2011, Costa de Caparica, Portugal. pp.215-223, 10.1007/978-3-642-19170-1\_24 . hal-01566546

**HAL Id: hal-01566546**

**<https://inria.hal.science/hal-01566546>**

Submitted on 21 Jul 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Distributed Accelerometers for Gesture Recognition and Visualization

Pedro Trindade\*, Jorge Lobo

Institute of Systems and Robotics  
Department of Electrical and Computer Engineering, Faculty of Science and Technology  
University of Coimbra – Polo II, 3030-290 Coimbra, Portugal  
{pedrotrindade,jlobo}@isr.uc.pt

**Abstract.** Acceleration information captured from inertial sensors on the hand can provide valuable information of its 3D angular pose. From this we can recognize hand gestures and visualize them. The applications for this technology range from touchless human-machine interfaces to aiding gesture communication for humans not familiar with sign language. The development of silicon chip manufacture allowed these sensors to fit in the top of a nail or implanted in the skin and still wirelessly communicate to a processing unit. Our work demonstrates that it is possible to have gesture recognition from a clutter-free system by wearing very small devices and connect them to a nearby processing unit. This work will focus on the processing of the acceleration information. Methods are shown to estimate hand pose, finger joints' position, and from that recognize gestures. A visualization of the angular pose of the hand is also presented. This visualization can show a single render of the pose of a recognized gesture or it can provide a simple real-time (low-latency) rendering of the hand pose. The processing of the acceleration information uses the gravity acceleration as a vertical reference.

**Keywords:** Accelerometers; Hand; Gesture Recognition; Gesture Visualization.

## 1 Introduction

Nowadays the skills of communication are vital to the society. Whether in industrial environments or plain social human activities there is a constant need of good communication skills. In a work environment there may be a need to communicate to a machine in a remote, secure, practical or non-intrusive manner. Such requirements are achieved when it is possible to represent the normal human activity by a virtual representation. In the particular case of human gestures, it is possible to have a virtual representation of the human gesture and be able to communicate in a noisy, possibly at a long distance or with low visibility environment.

When related to human-to-human activities, communication can become complicated when one of the interlocutors does not know the other's language. This is the case when two persons try to communicate and one is hearing impaired and knows sign language, while the other is not and does not know such language. Being possible to intermediate the communication between this two persons allows a great

---

\* This work is partially supported by the collaborative project HANDLE (grant agreement no 231640) within the Seventh Framework Programme FP7 - Cognitive Systems, Interaction and Robotics.

social achievement with an immeasurable value to those who carry that limitation. Science and Industry are constantly evolving and today it is possible to create a system capable of facilitating communications, as mentioned above, and yet be portable, reliable, eventually self-powered and very simple to use. The development of silicon chip manufacture enabled the development of low-cost single chip inertial sensors. By using gravity as a vertical reference these inertial sensors can provide acceleration information from gravity to be used for gestures' angular pose estimation. These sensors can fit in a person's thumbnail or even implanted on the skin. Having them connected to some terminal available to the user, it is possible to have a system available to the user's mobile devices like a smartphone or an iPad.

## 2 Contribution to Technological Innovation

This work mainly addresses the problem of recognizing simple static sign gestures from the information provided by inertial sensors. By using a triaxial accelerometer in each finger and one on the palm, we can measure the acceleration of gravity in relation to each axis of the sensor. Based on this we are able to estimate the pose of all the hand joints. This enables us to have a clear representation of the hand.

From the representation we developed an algorithm to recognize the gesture against a pre-defined library of gestures. From this algorithm it will be shown that it is possible to have only one reference gesture in the library and still achieve useful results. This avoids a cluster-based approach for recognition and simplifies the process without compromising the results.

Key contributions:

- Study of the information extracted from hand distributed accelerometers.
- Algorithm to estimate the 3D angular pose of each finger, despite of the inobservance of rotation in the gravities axis.
- Visual representation in real-time (low latency) and offline of the hand using Python and Blender 3D software package.

## 3 Related Work

Several types of gesture capture systems are possible. It can be an optical capture system using vision for recognizing the configuration of a hand and it can be a hand sensor- based recognition. Based on the extended analysis of [1], an overview of these systems is given next.

*Color markers:* Color markers systems for gesture recognition use color patterns to estimate the pose of the hand. This estimation is obtained with inverse kinematics. It has been demonstrated that it is possible to have a low-cost and effective recognition with this approach yet it requires an camera suitably positioned and proper lightning conditions. Our approach only requires minute sensors distributed on the hand.

*Bare-hand tracking:* These systems typically rely on edge detection and silhouettes and are generally robust to lightning conditions. Reasoning from them involves inference algorithms to search the high-dimensional pose space of the hand. That is computationally expensive and goes far from real-time and becomes unsuitable for Human-Machine interfaces.

*Marker-based motion-capture:* Marker-based systems involve the use of retro-reflective markers or LED and expensive many-camera setups. Such systems are

highly accurate but are expensive. Our proposed system only requires low-cost sensors on the hand.

*Data-driven pose estimation:* The Data-driven pose estimation makes use of the values from sensors attached to the hand to define the pose of the hand. Such approach allows simple computation to estimate the hand pose. This type of system can easily become very intrusive since it needs the user to somehow wear the sensors. In this work we show that only a few minute non intrusive sensors are enough.

## 4 Implementation

### 4.2 Gestures and Portuguese Sign Language

Gestures allow the representation of simple sign language expressions of actions. They can be understood as the static positioning of the fingers in relation to the hand’s wrist. It may also include some rotation of the wrist. In Portuguese Sign Language the most elementary words such as the alphabet and the numbers are represented by static gestures. This work is focused in these static gestures. Fig. 1 exemplifies one of these gestures.



**Fig. 1** Example of a static gesture from the Portuguese Sign Language.

### 4.3 Acceleration Sensors and Motion Classifier

Accelerometers measure linear acceleration, and current sensors integrate three orthogonal sensing axis on a single chip. These can be distributed onto each finger and the palm as shown in Fig. 2.

The sensed acceleration results from gravity and hand motion. In order to use gravity as a vertical reference to infer hand and finger pose, the hand has to be static. The motion classifier looks in the neighborhood of each sample to detect this, assuming sustained steady accelerations of the hand and fingers do not occur.

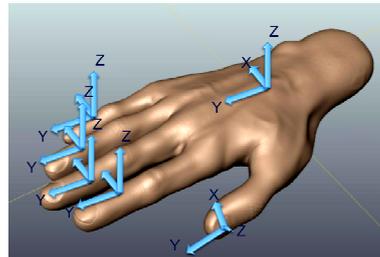
The modulus of the acceleration vector is used and three thresholds are defined  $\{M_1, M_2, M_3\}$ . Let  $W$  be the depth of neighboring search, for a given sample  $s_i$ , its neighbors are defined between  $[s_{(i-W)}, \dots, s_{(i+W)}]$  and let  $L_i$  be the level of motion of sample  $s_i$ , then for each sample  $s_i$

$$L_i = l, M_{l-1} \leq (\max\{s_{(i-W)}, \dots, s_{(i+W)}\} - \min\{s_{(i-W)}, \dots, s_{(i+W)}\}) < M_l, \quad (1)$$

where  $l = \{1, 2, 3\}$  and  $M_0 = 0$ .

### 4.4 Gesture recognition

When there is no acceleration, the gravity vector provides a vertical reference so that the measured acceleration can provide information about the pose of the triaxial accelerometer. However this provides only two inclination angles, but no azimuth



**Fig. 2** The different frames of reference for a distributed accelerometer system with a sensor in each finger and the palm.

since rotations about the vertical are not sensed. To overcome this limitation a method, based on [6] is used, where a method to determine the rotation between two frames of references is proposed by using Horn’s [14] closed-form solution for absolute orientation using unit quaternions. Each triad of accelerometers can be seen as an observer of the gravity vertical reference. When the sensor is static the measurements provide a vertical reference vector in the sensor’s local frame of reference.

So, by using two set of vectors, one given by the accelerometer on a finger and the other set of vectors given by the accelerometer in the palm it is possible to obtain the full 3D pose of the finger’s accelerometer relative to the palm accelerometer.

This results in a feature space of 20-dimensional real vector space (*roll*, *pitch* variable for the thumb and *roll*, *pitch*, *yaw* for the other fingers). A library of gestures is created, so that a nearest neighbor approach can be used to identify an observed gesture. Manhattan distance is used: let  $p$  be the observed gesture and  $q_i$  the set of gestures in the library, we get a set of distances  $d_i$  given by:

$$d_i = \sum_{k=1}^{17} |p_k - q_{i,k}|, \tag{2}$$

and the shortest one indicates the matched gesture.

Fig. 3 shows a general overview for the process of recognizing gestures. The raw sensor output needs some filtering, bias and scale factor compensation. The next step is the motion classifier as explained above. When there is no motion, the process enters the final step of the classifier, where the static gesture is recognized. The detection of sudden motion can also be used to signal the start and end of a gesture for the interface..

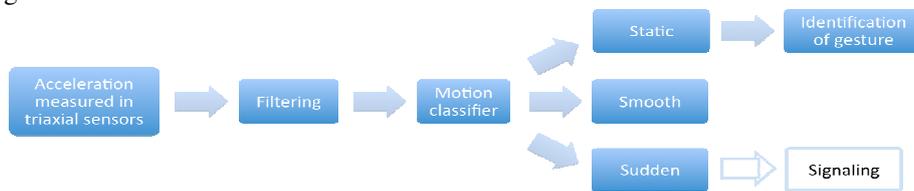


Fig. 3 Overview of the phases for gesture recognition.

#### 4.5 Visualization

##### Relationship between finger joints

For the gesture recognition presented above only the fingertip orientation relative to the palm was considered. For rendering a human hand model some assumptions are required. According to [2] the total active range of motion of a typical finger is 260°, in which is the sum of active flexion at the MP (metacarpophalangeal) (85°), PIP (proximal interphalangeal) (110°), and DIP (distal interphalangeal) (65°) joints. The range of active extension at the MP joint varies between people but can reach 30- 40° [3]. Although there is a considerable axial rotation in addition to flexion/extension and abduction/adduction movements, this is constrained and so it is not considered a true third degree of freedom. In total, the human hand, including the wrist has 21 degrees of freedom of movement [2][4]. A common reduction of the number of degrees of freedom is to consider that for the index, middle, ring, and little fingers, in order to bend the DIP joints the correspondent PIP joints must also

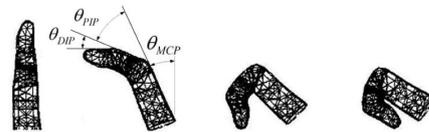


Fig. 4. Constrains between PIP and DIP joints.

bend, as seen in Fig. 4. According to [4] the relationship can be approximately presented as:

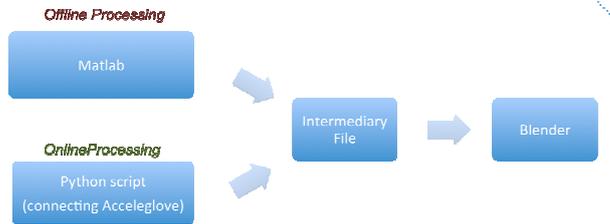
$$\theta_{DIP} = \frac{2}{3} \times \theta_{PIP} \tag{3}$$

**Software**

3D graphics is the grandchild of Euclids Elements, a geometric construction of the Universe as a mesh of points connected by measurable lines [7]. OpenGL is a natural choice for drawing those primitives. But to allow more control and fast implementation, a higher level software layer is needed. Blender is a free open source 3D content creation suite that allows several different approaches for 3D representation. Moreover, it allows scripting control of the software. Also, choosing Blender allowed the use of this visualization tool for datasets of the European project HANDLE<sup>†</sup> [8].

Blender allows Python scripting language to control all the environment of the software. This means it is possible to dynamically use one’s own equations and algorithms to set up the environment. This resulted in reading the output of the processed information from other external sources (such as Matlab or any other program) and have this information resulting in a user-controlled rendering. This workflow can be seen in Fig. 5.

As indicated in Fig. 5 we use Matlab for the offline processing for gesture recognition and Python for online processing the accelerometer data and in both we update a file that is continuously monitored by Blender to update the rendered model.



**Fig. 5** Workflow for the visualization of the hand angular pose.

**5 Results**

**5.1 Experimental Setup**

To implement the methods in section 4, Matlab and Python were used. Matlab for the processing and Python to interface with Blender.

For the distributed accelerometer system the Anthrotronix Aceleglove [9] was used. In the Aceleglove the accelerometer used is the MMA7260QT from Freescale Semi-conductor [5]. This device is a low-cost capacitive micromachined accelerometer and can measure acceleration up to 6G.

<sup>†</sup> HANDLE is a collaborative project is funded by the European Commission within the Seventh Framework Programme FP7, as part of theme 2: Cognitive Systems, Interaction, Robotics, under grant agreement 231640.

### 5.1 Recognition

From Fig. 6 it is possible to see the motion level classifier working. According to the deviation in the modulus of each sensor a classification of the level of movement is made, meaning Level-1 to be fairly static, while Level-2 refers to a smooth movement and Level-3 refers to a sudden movement that eventually could be used to signal the start and stop of a stream of gestures.

Each contiguous subset of Level-1 samples vectors are converted into a single vector, called frame. All the frames are then used to find the quaternion of rotation between each finger and the palm. After calculating the quaternion of rotation a reprojection error is measured for each of the frames. This reprojection error is shown in Fig. 7.

The values found on Fig. 7 allow saying reprojection error is very small, with all the values standing before 7 degrees. This correspond to 3 distinct positions (frames) for which the user had to maintain the same gesture, and a few degrees will not significantly alter the gesture so this result indicates that the estimated relative

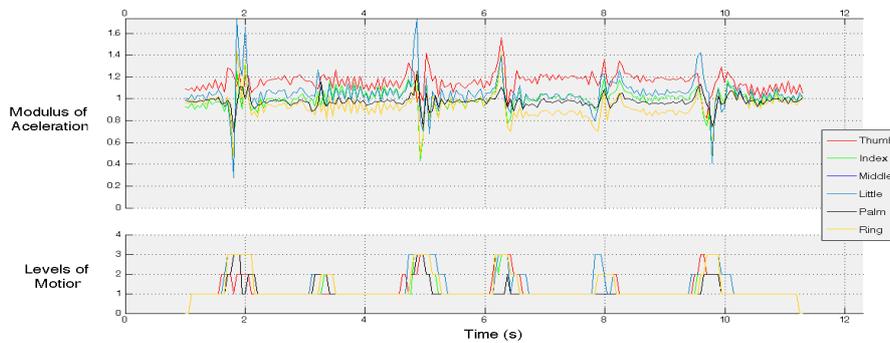


Fig. 6 Example of the motion classification after a data acquisition.

angular pose values are suitable for gesture recognition.

The gesture that it is being analyzed is compared against a library. In this library each entry relates all the information of a single gesture. That includes the name of gesture, the roll, pitch and yaw angles, the quaternion values and weight that define how much impact should roll, pitch or yaw values have in the recognition. Comparing a gesture against the library produced the results like the ones shown in Table 1.

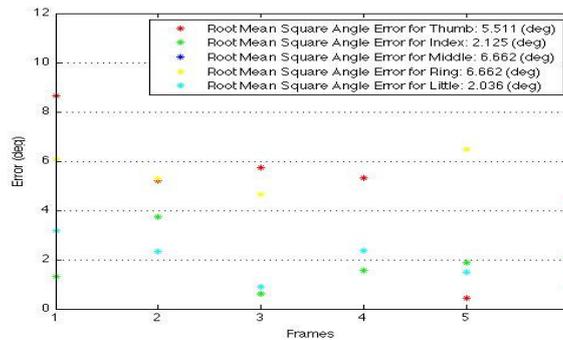


Fig. 7 Rotation reprojection error after the calculation of the quaternions of rotation.

**Table 1 - Manhattan distances from the current gesture to the ones in the library.**

Gesture	G	H	K	L	O	Q	R	S	T	U	V	W
Distance	6.0	<b>3.1</b>	4.2	3.9	4.2	5.2	5.3	5.6	4.3	4.2	7.0	4.4
Gesture	X	Y	Z	1	2	3	5	4	6	7	8	9
Distance	7.4	4.0	5.0	6.0	3.9	6.2	5.5	6.3	4.3	6.4	5.1	4.2

In Table 1 it is possible to see a list of all the gestures included in the library and the distance, in a Manhattan geometry of the current gesture to the ones in the library.

The gesture performed that resulted in the comparison shown in Table 1, was indeed the “H” gesture. The visual perspective of the gesture is shown in Fig. 1.

## 5.2 Visualization

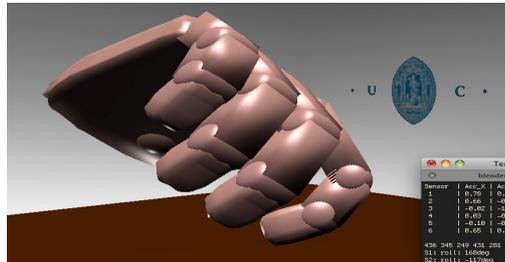
The visualization was structured to allow the representation of the hand pose processed in the recognition process and also directly from a real-time connection to the Aceleglove [9]. The visualization of the results that came from Matlab was possible to represent because Matlab outputted the Roll Pitch and Yaw values for each sensor it processed. Having blender prepared to read those values and running the Python routines defined in blender a correct pose from blender was possible to represent. Fig. 8 shows a render of the pose. Again, the relatively small error in calculating the hand pose allowed the visual representation as seen in this figure.

When this visualization was performed in real time, an external (to Blender) script was running. This script would provide Blender the pose information for each sensor. At a framerate of about 29fps, Blender updates the pose of every joint in the hand and renders it. This performance was achieved in a Macbook Pro 2.5GHz from 2009.

## 6 Conclusions

The recognition of gestures is a vast area of research. In this work a novel approach using distributed accelerometers in the hand to recognize the gestures solely by the measurement of acceleration was presented.

By using an intelligent distribution of accelerometers on the hand it was possible to create an algorithm to recognize a hand gesture. Our approach relies on the vertical reference provided by gravity to infer angular pose, however in this way rotations about vertical axis are not observable. To overcome this limitation, a novel approach, based on the work of [6] on relative pose calibration between inertial sensors and cameras, allowed to find the exact 3D angular pose of the fingers in relation to the palm. The recognition is based on observation of multiple frames of



**Fig. 8:** A render from Blender showing the encountered angular pose of the hand.

static gestures. As future work this recognition should be extended to allow dynamic gesture to be recognized as well.

We also proposed a 3D visualization tool for the hand pose. By creating an intelligent structure this tool was already capable of fulfilling a broader project, like the HANDLE project [8]. The approach presented in this work can be improved in many ways, by addressing dynamic gestures, applying a probabilistic approach for learning the gestures, or even using more accelerometers on the intermediate joints.

## References

1. Wang, R. and Popovic, J.: Real-Time Hand-Tracking with a Color Glove. ACM Transaction on Graphics (2009)
2. Jones, L., Lederman, S.: Human Hand Function. Oxford University Press (2006)
3. Kapandji, I. A.: The physiology of the joints: Vol. 1. Upper limb (2d ed., L. H. Honore, Trans.). Edinburgh: E & S Livingstone (1970)
4. Kang Li, I-Ming Chen, Song Huat Yeo: Design and Validation of a Multi-finger Sensing Device Based on Optical Linear Encoder. IEEE International Conference on Robotics and Automation. Anchorage, Alaska, USA (2010)
5. Freescale Semiconductor MMA720QT webpage, [http://www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=MMA7260QT](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MMA7260QT).
6. Lobo, J., Dias, J.: Relative Pose Calibration Between Visual and Inertial Sensors, Int. J. Rob. Res., vol. 26, p.561-575 (2007)
7. Oliver, J.: Buffering Bergson: Matter and Memory in 3D Games (2008)
8. HANDLE European Project website, <http://www.handle-project.eu>
9. AnthroTronix Aceleglove, <http://www.aceleglove.com/>
10. Barbour, N.M.: Inertial Navigation Sensors, Sensors & Electronics Technology Panel. Draper Report no. P-4151 (2003)
11. T. Baudel and M. Beaudoin-Lafon: Charade: remote control of objects using free-hand gestures. Communications of the ACM, vol. 36, no. 7, pp. 28-35 (1993)
12. Fraden, J.: Handbook of Modern Sensors: Physics, designs, and applications, 3rd Edition. Springer (2003)
13. Ferreira, A. et al.: Gestuário - Língua Gestual Portuguesa. Secretariado Nacional para a Reabilitação e Integração das Pessoas com Deficiência (1997)
14. Horn, B.K.P.: Closed-Form Solution of Absolute Orientation Using Unit Quaternions. Journal of the Optical Society of America, 4(4):629-462 (1987)
15. Khan, S., Gupta, G., Bailey, D., Demidenko, S., Messom, C.: Sign Language Analysis and Recognition: A Preliminary Investigation. IVC New Zealand (2009)
16. Takayuki, H., Ozake, S., Shinoda, H.: Three-Dimensional Shape Capture Sheet Using Distributed Triaxial Accelerometers. 4th Int. Conf. on Networked Sens. Syst. (2007)
17. Taylor, C. L., & Schwarz, R. J.: The anatomy and mechanics of the human hand. In Selected Articles From Artificial Limbs. (pp. 49-62). Huntington, NY: Kreiger (1970)
18. Weinberg, M. S., Kourepenis, A.: Error Sources in In-Plane Silicon Tuning-Fork MEMS Gyroscopes. Journal of Microelectromechanical Systems, vol. 15, No. 3 (2006)