ChAirGest – A Challenge for Multimodal Mid-Air Gesture Recognition for Close HCI

Simon Ruffieux
University of Applied Sciences of Western Switzerland
1705 Fribourg, Switzerland
simon.ruffieux@hes-so.ch

Denis Lalanne
University of Fribourg
1705 Fribourg, Switzerland
denis.lalanne@unifr.ch

Elena Mugellini
University of Applied Sciences of Western Switzerland
1705 Fribourg, Switzerland
elena.mugellini@hes-so.ch

ABSTRACT
In this paper, we present a research oriented open challenge focusing on multimodal gesture spotting and recognition from continuous sequences in the context of close human-computer interaction**.** We contextually outline the added value of the proposed challenge by presenting most recent and popular challenges and corpora available in the field. Then we present the procedures for data collection, corpus creation and the tools that have been developed for participants. Finally we introduce a novel single performance metric that has been developed to quantitatively evaluate the spotting and recognition task with multiple sensors.

Categories and Subject Descriptors
H.1.2 [**Information Systems**]: User/Machine Systems – *human information processing;* H.2.4 [**Information Systems**]: Systems – Multimedia databases; H.5.2 [**Information Systems**]: User Interfaces – *Benchmarking, Evaluation/methodology, Natural language.*

General Terms
Algorithms; Performance.

Keywords
Open Challenge; Gesture Spotting; Gesture Recognition; Corpus; Algorithms; Performance Evaluation; HCI.

1. INTRODUCTION
The ChAirGest challenge is a research oriented open challenge designed to encourage researchers to take advantage of data recorded from multiple sensors to optimize and evaluate methods for gesture spotting and recognition. The developed dataset provides researchers a common benchmark tool which enables quantitative comparison of algorithms under strictly comparable conditions. Those algorithms can use any combination of the data types available in the corpus.

The provided data come from a Kinect camera and four inertial motion units (IMU) attached to the right arm and the neck of the subject. The corpus contains 10 different gestures, started from 3 different resting postures and recorded in 2 different lighting conditions by 10 different subjects. Thus, the total corpus contains 1200 annotated gestures split in continuous video sequences containing a variable number of gestures. The goal of the challenge is to promote research on methods using multimodal data to spot and recognize gestures in the context of close human-computer interaction. Note that several other research paths may be explored with the provided corpus such as fusion of sensors data, influence of the gesture spotting accuracy in improving the gesture recognition task, enhancement of a single sensor recognition from multi-sensory data or even automatic mapping between sensor modalities [4].

ICMI'13, December 9–13, 2013, Sydney, Australia.

http://dx.doi.org/10.1145/2522848.2532590

2. STATE OF THE ART
These last years, with the introduction of widely affordable sensors and efficient SDKs, natural gesture-based interfaces have gained a huge increase in popularity for the public. Body-movements and touch less gestures are topics of particular interest for researchers and developers and more and more applications are being developed.

When developing and training machine learning algorithms, in order to efficiently recognize gestures and discriminate them from gesticulation, the developers require datasets containing annotated examples of the gestures to recognize. The creation of a synchronized multi-sensors dataset is a time-consuming and complex task but it is mandatory in order to provide means to develop and test algorithms or to objectively compare them; in particular when algorithms are based on different sets of sensors.

Different challenges and datasets have been publicly or commercially released these last years. The first available datasets were focusing on full-body movements to improve motion capture [14], pose estimation [27] or activity recognition [16][22][20] using expansive synchronized motion capture systems and video cameras. Datasets of sign language have also been frequent topics of research [21][8], mostly based on webcams to record hands or upper body of subjects. Hands and arms oriented datasets appeared more recently with different research axes: shape and motion estimation from webcams [1][13] or object manipulation for robotic applications [6]. Multi-sensors corpora focusing on activity recognition have also been released recently; the CMU-MMAC dataset [7] is set in a kitchen context, while the Opportunity challenge [26] focused on the recognition of everyday activities, both using wearable and environmental sensors. A dataset created recently contains data from multiple sensors attached to the arm to precisely record its movements in order to study manipulative virtual interfaces [12]. The "ChaLearn" challenge provides multimodal data from a Kinect sensor to recognize partial and full-body gestures; several workshops have been organized in conferences with success [15]. The HASC challenge takes advantage of crowd-sourcing and smartphones accelerometers to create a large dataset containing real-life activities [18]. The Intel Perceptual Computing Challenge is being organized by Intel and has recently been launched with hundred thousands of dollars in awards and promotions [17].

However it focuses on best demonstration applications rather than best algorithms. It should still help improving gesture recognition techniques with RGB and depth information for close human-computer interaction.

3. OPEN CHALLENGE

3.1 Introduction

The proposed open challenge provides a framework for researchers to develop, optimize and compare their algorithms for gesture spotting (temporal segmentation) and/or gesture recognition. It also provides an elegant solution to objectively compare different algorithms based on different combinations or sets of sensors. Indeed, the two types of sensors (Kinect and Xsens IMUs) provide partially redundant information but may also be seen as complementary sensors. Furthermore RGB-D sensors such as Kinect are environmental sensors which tends to be more and more present in our daily life [10] and IMUs are cheap, wearable and widely available sensors being embedded in most new technology driven objects for a wide range of applications [2, 3]. Therefore the combination of those two types of sensors seems a meaningful path of investigation for future algorithms focusing on human computer interaction. The previously mentioned datasets and challenges tackle interesting research problems, but none of them provides a multi-sensory dataset for the task of spotting and recognizing gestures in the context of close HCI. Indeed most of the challenges focus on activity recognition or full-body gestures; furthermore the classes are pre-segmented to reduce the size of the data when working with video and/or to reduce the task complexity for the participants. This pre-segmentation prevents addressing the problem of gesture spotting. Therefore we have developed the ChAirGest challenge which introduces the four following novelties:

1. Strong focus on hand/arm gestures for close-HCI
2. Multiple synchronized sensors
3. Continuous recording sequences
4. A single performance metric for spotting & recognition

ChAirGest is organized as an open challenge: participants are allowed to work on the data for an unlimited time and are encouraged to submit their results for evaluation when ready. A dedicated website[1] has been created to host the information about the corpus, the rules and ranking information of the challenge.

3.2 Goals

The proposed challenge targets two main goals: optimize the temporal segmentation and optimize the recognition of one hand/arm gestures. The process of temporal gesture segmentation (also called gesture spotting) still is an important and challenging task in the domain; it consists in segmenting the start and end points of a gesture in a continuous sequence. The process of gesture recognition has been much more studied and consists in correctly identifying a segmented gesture within a known vocabulary. Therefore the task of the challenge can be resumed as follows:

Spot and recognize the gestures independently of subjects using any set of sensor(s)

[1] https://project.eia-fr.ch/chairgest

3.3 Corpus

3.3.1 Gestures

A vocabulary of 10 one-hand/arm gestures focusing on the specific context of close human-computer-interaction has been chosen. The vocabulary has been chosen to span over different potential difficulties for each recognition modality and thus promote algorithms using fusion from multiple sensors. In Table 1, the gestures are characterized by their motion in the space according to the cardinal planes and axes of the human body as described by Neumann [24]. The transverse axis represents moving leftward/rightward, the vertical axis represents moving upward/downward while the sagittal axis represents forward/backward movements. The rotation parameter represents rotation of the subject's arm along its axis. Motion along a particular axis may be more difficult to recognize depending on the modality used. For example, gestures 5-6 should be harder to recognize using only an RGB stream but trivial using the information from IMUs. However, using video streams may help distinguishing overlapping gestures by identifying intermediary hand/fingers postures and positions; notably when motion parameters are similar, such as in gesture 3-4 or 9-10. The vocabulary has also been chosen for the prior existence of the gestures in literature with different sensors: accelerometers [19], Kinect [15] or other hardware [23]. No semantic has been defined for the gestures as it is out of the topic of this work.

Two levels of complexity can be assessed for the gestures: basic gestures (1-6) and complex gestures (7-10). The motion involved in complex gestures partially overlaps the motion of basic gestures; therefore their temporal segmentation and recognition are more prone to error.

In order to better reflect reality of users performing gestures, we forced the subjects to have a same resting posture before and after each performed gesture. Three different resting postures have been defined accordingly to classical user position when sitting in front of a computer: "Hands on the table" when working/typing, "Elbows on the table, hands joined under chin" when thinking and "Hands on the belly button" when watching a movie.

Table 1: List of gestures recorded in the corpus. For each gesture, a description of the main motion characteristics of the arm are defined (Transverse & Vertical, Sagittal or Rotational) and the last column illustrates the fact that the motion of a gesture Overlaps another gesture.

#	Name of gesture	Tr& V.	Sa.	Rot.	Ov.
1	Swipe left	X			
2	Swipe right	X			
3	Push to screen		X		X
4	Take from screen		X		X
5	Palm-up rotation			X	
6	Palm-down rotation			X	
7	Draw a circle I (longitudinally)	X	X		X
8	Draw a circle II (hand rotation as 5)	X		X	X
9	Wave hello	X			X
10	Shake hand (LR)	X			X

The repartition of the gesture occurrences in the corpus is described by the formalism below.

*10S * (2L * [2O * 10G * 3R]) = 1200 gesture occurrences*

Where S = subject, L = lighting condition, O = gesture occurrence, G = unique gesture, R = resting posture.

The dataset contains 10 subjects, each doing 4 recording sessions with 2 different lighting conditions (dark and normal). In a recording session, the subject performs once each gesture class for each of the resting postures. To resume, in a single recording session, a subject performs 3 times each of the 10 gesture classes. The full corpus contains 1200 gesture occurrences for an approximate total of 6 hours of continuous recording.

3.3.2 Sensors

Two different types of sensors are used: a Kinect for Windows from Microsoft and Xsens MTw Inertial Motion Units (IMU). The Kinect records the subject at a frame rate of 30Hz. It provides 3 different streams of information: RGB stream, Depth stream and the approximate 3D position of the upper-body skeleton joints as provided by the official Microsoft Kinect SDK. Each IMU provides linear acceleration, angular acceleration, magnetometer, Euler orientation and orientation quaternion at a frame rate of 50Hz. The main advantages of the combination of these two sensors are that they have overlapping and complementary features while both being widely used in research and embedded in many innovative commercial devices [2, 3] [10]. The data provided by both sensors partially overlap: the Kinect skeleton and the position inferred from IMUs can be related to each other [4]; an interesting difference is that the skeleton data from Kinect is prone to occlusion and the quality of accuracy highly dependent on the position of the user relatively to the sensor; contrary to the IMUs sensors which are occlusion free and position independent. The Skeleton data can rapidly recover from temporary data loss, while IMUs may experience drift over time and loss of data for a short period of time may be critical when inferring position [29]. While Kinect often requires complex processing to extract features from video streams, data from IMUs can be used with less complex processing. Finally both types of sensors are complementary in their practical usage: Kinect is an environmental sensor requiring to be installed at a fixed position while IMUs are wearable sensors that can be embedded in a watch or a bracelet and thus can be always available for the user. The two sensors may be used as completely distinct sources, completely merged into a single enhanced source of information or could be opportunistically fused according to context [5]; The combination of the two sensors provides multiple potential research paths.

3.3.3 Recording setup and procedure

The recording setup involves a subject sitting on a chair in front of a desk as if working on a computer. A Kinect for Windows records the scene from the top of a computer screen point-of-view with a 30° downward angle. The subject wears 4 Xsens IMU attached under his clothes on his shoulder, arm, fore-arm and hand as depicted in Figure 1.

The acquisition procedure involves the subject mimicking pre-recorded RGB videos of gestures displayed on a computer screen facing the subject. According to Fothergill et al [11], a dataset should aim for correctness and coverage to optimize learning performances of algorithms. High correctness implies that all the recordings of a same gesture class should have similar features while coverage imposes some variance of those features. Our video-mimicking method theoretically promotes correctness [11], while coverage is promoted through the various imposed resting postures and the natural heterogeneity between subjects.

Figure 1. Visuals of the recording setup. On the left, an image captured from the Kinect RGB stream. On the right, an external view of a subject performing a gesture in pseudo-recording conditions. Note that the accelerometers were hidden under the subject's clothes to avoid visual clues.

The standard procedure for the recording of each subject is the following: once the subject enters the recording room, he is placed in front of a computer and has to read carefully the instructions on the tasks that he will perform during the experiment. Then he fills a form containing detailed information about him. (Name will not be disclosed). The IMUs are attached to his right arm, under his long sleeve clothing in order to avoid visual clues on the video streams. A T-posture is imposed at the beginning of each session for potential calibration purposes. When the system is ready, a first trial session is performed, allowing the subject to understand the task and familiarize himself with the gestures. Then the subject completes four recording session with alternated lightning condition. Between each session, the user has a break of five minutes to relax, wander around and rest his arms. At the end, the subject filled a qualitative questionnaire to gather information such as fatigue. A complete recording procedure took about one hour per subject.

3.3.4 Data formats

The data has been recorded in a raw binary format at maximum frame rate and highest quality. The binaries contain the serialized data classes of the sensors with, for each frame, the absolute timestamp of acquisition. The raw data has also been converted to provide a lighter and more generic format which significantly reduces the size of the data and should simplify usage with different coding languages than original (C#). The Kinect binary data has been converted into "avi" files (RGB, Depth) and text files (skeletons). The Xsens data has been converted to text files without loss in quality. The biggest advantage of the full quality binary format is that it allows using most standard functions from the official SDKs to access and/or process the recorded data. This method facilitates switching from real sensors to recorded data and vice versa. However its main disadvantage is its size in memory; the complete dataset is approximately 1 teraoctet.

The Kinect binary format contains the RGB and Depth streams along with the 3D skeleton representation as acquired from the Kinect sensor using the official SDK with a frequency of 30Hz. The images in the video streams have a resolution of 640x480 pixels. The Xsens binary format contains the information about the sensors such as number of IMU, rate, absolute timestamps and a vector of Xsens objects. Each of these Xsens objects contains the information from a single IMU: linear acceleration, angular acceleration, magnetometer, Euler orientation and orientation quaternion as provided by the Xsens SDK with a frequency of 50Hz.

The ground truth data have been generated during the recording process. For each gesture, its name and four specific timestamps have been acquired automatically using a custom scenarized acquisition method. The four specific timestamps, illustrated in Figure 2, correspond to the three phases of a gesture as described in previous research on hand gesture [25].

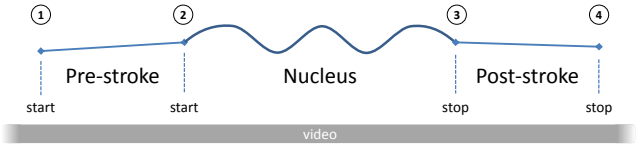

Figure 2. This figure illustrates the three phases format with four timestamps for the ground truth labeling of gestures.

The pre-stroke corresponds to the subject moving from the resting posture to the initial posture for gesture start, while the post-stroke corresponds to the motion from the end of gesture back to the resting posture. The middle part is often called nucleus and corresponds to the actual gesture. On Figure 3, an example of such labeling is shown on a real acquisition represented with accelerometers data over-time.

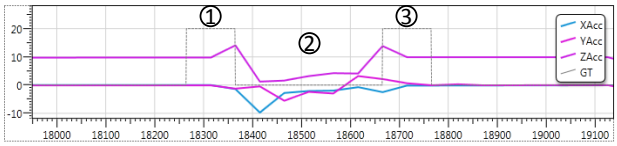

Figure 3. Example of plotted Xsens acceleration data with superposed ground truth timestamps in dotted-gray. (1: pre-stroke, 2: nucleus, 3: post-stroke)

These ground truth data have been stored in separate files to simplify the management of sensors and training/evaluation data. Having them in separate file per sensor and from the data allows developers to have only the information required according to the sensor(s) used or the task being performed.

3.4 Performance metric

The combination of *Detection Rate* (DR) and *False Positive* (FP) are standard performance metrics used to optimize and tune gesture recognition systems, mainly using ROC curves analysis [9]. A single measure to assess global performance of a continuous recognition system has not been strictly defined yet; instead a comprehensive set of frame and event-based metrics has been proposed for continuous activity recognition [28]. Using frame-based metrics would complicate our evaluation as the different sensors have different frame rates in our dataset. Therefore, for this challenge, a combination of existing event-based metrics and a novel time-based metric is proposed to assess the performances of the methods.

The two event-based metrics are *Precision* (P) and *Recall* (R) as described in [28]. The *Precision* metric corresponds to the number of correctly detected events divided by the number of returned events and the *Recall* metric represents the correctly detected events divided by the number of events in the ground truth (metric also known as *Detection Rate*). Those two metrics can be combined in the *F-score*, a standard measure of the accuracy of a test in statistics [28]. When the parameter $\beta = 1$, the *F-score* is said balanced and written F_1.

$$F_\beta = (1+\beta^2) * \frac{P*R}{\beta^2 P+R} \quad (1)$$

$$F_1 = 2 * \frac{P*R}{P+R} \quad (2)$$

To measure the gesture spotting performance of the algorithms, a novel time-based metric evaluates the accuracy of temporal segmentation. This metric has been named *Accurate Temporal Segmentation Rate* (ATSR) and represents a measure of the performance in terms of accurately detecting the start and stop events of all correctly detected gesture occurrences. The *ATSR* is computed as follows: for each correctly detected gesture occurrence, the *Absolute Temporal Segmentation Error* (ATSE) is computed by summing the absolute temporal error between the ground truth and the result of the algorithm for the start and stop event and dividing this sum by the total length of the gesture occurrence measured from the ground truth as formalized in Equation 3. Finally the *ATSR* metric is computed for a particular sequence by subtracting the average *ATSE* to 1 in order to obtain the accuracy rate as shown in Equation 4. A perfectly accurate segmentation produces an *ATSR* of 1.

$$ATSE = \frac{\|Start_{GT} - Start_{Alg}\| + \|Stop_{GT} - Stop_{Alg}\|}{Stop_{GT} - Start_{GT}} \quad (3)$$

$$ATSR = 1 - \frac{1}{n} * \sum_{i=1}^{n} ATSE(i) \quad (4)$$

In order to avoid small ground truth timing errors producing irrelevant penalities during the computation of the *ATSE*, a temporal error for the start or end event inferior to 10% of the gesture duration is considered as irrelevant and set to 0 in Equation 3.

The final single metric performance is evaluated using a combination of *F-score* and *ATSR*. Note that, in Equation 5, the F_1*-Score* has twice more importance than the *ATSR* as the main goal remains the correct recognition of the gestures being performed.

$$Perf = 5 * \frac{ATSR*F_1}{4*ATSR+F_1} = 10 * \frac{ATSR*P*R}{4*ATSR+P+R} \quad (5)$$

The example on Figure 4 illustrates the importance of the *ATSR* metric for the evaluation of performances. Algorithm A and B have the same *Precision* and *Recall* scores (P=3/4, R=3/3, F_1=86%). However, as visible on Figure 4, it is clear that algorithm B is more accurate at spotting gestures than algorithm A. This can be qualitatively measured using the *ATSR* metric: ATSR(A)=1-Avg(1/4, 2/6, 2/4) = 64% and ATSR(B)=1-Avg(0/4, 0/6, 1/4)=92% which shows that B performs better than A. Finally when combining the F_1*-score* and the *ATSR* metric to obtain a single performance measure, we obtain Perf(A)=80% and Perf(B)=87%.

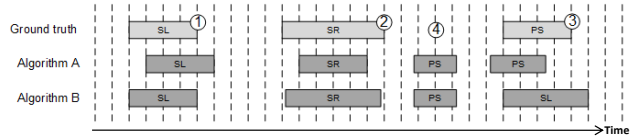

Figure 4. A practical example that demonstrates the importance of the *ATSR* performance metric when comparing two algorithms. The 'SL', 'SR' and 'PS' represent gesture classes. In this example Perf(A)=80% and Perf(B)=87%.

3.5 Tools

Several tools and libraries are provided to the participants in order to familiarize themselves with the dataset. All mentioned tools are provided as open-source code and can be modified or reused freely by participants. These tools are only based on the full quality binary data and cannot be used with the converted text files. The tools to access the data are also available as separate libraries and can be partially used with other compatible programming languages such as Matlab.

The list of tools:

1. Access Kinect binary data (RGB, depth and skeleton).
2. Access Xsens binary data.
3. Convert Xsens binary data to its equivalent in text format.
4. Convert Kinect binary in compressed video files using the FFMPEG library and the Kinect Skeleton data to text files.
5. Visualize Kinect and Xsens data.
6. Evaluate the performance of an algorithm.

The first four tools illustrate how to access and convert the data from Xsens and Kinect. The fifth tool is a custom visualizer which allows the visualization of the synchronized data from the IMUs and from the video stream of the Kinect simultaneously. The user can navigate in the stream of data as if using a traditional video player application. The sixth tool is based on the evaluation method described in section 3.4 and provides a rapid way for participants to evaluate the performance of their algorithms. It needs a text file containing the classification results as input and outputs the detailed performances.

3.6 Data

The data is split in two main sets: a development set and an evaluation set. Both sets exhibit the same folder and file hierarchy. The files are separated in folders, per subject and recording session. Each session has been split into 7 to 10 batches to simplify memory management. Indeed a whole session is approximately 25GB on disk and would be problematic to load in RAM during processing. A batch represents a different number of files according to the quality: raw or converted. A batch in raw quality is about 3GB on disk and contains four binary files: Kinect data, Kinect ground truth data, Xsens data and Xsens ground truth data. A batch in converted quality is about 10MB on disk and contains 6 files: 2 Kinect video "avi" files, 1 Kinect skeleton text file, 1 Kinect ground truth text file, 1 Xsens data text file and 1 Xsens ground truth text file.

The development set contains the data for participants to train and self-evaluate their algorithm(s). The set contains the sensor data and the ground truth data in separate files. The developers may split the data according to their wishes in order to train and self-evaluate their system. This set contains ¾ of all the data of the corpus as it is very important for the participants to have enough data to train and evaluate their algorithm efficiently. To obtain the data, participants must register on the website and fill an End User License Agreement (EULA) on the usage of the data due to image property of the subjects. The evaluation set contains ¼ of the data and is not released to participants. It is used by the organizers of the challenge to evaluate performances of the algorithms developed by the participants.

3.7 Evaluation

Once a participant has finished working on his algorithm, he must submit an executable program to the organizers for performance evaluation on the evaluation set. Specific instructions are provided in order for the participant to be aware of the exact input data format and the desired output format. With the consent of the participant, the results are shown on a webpage containing the ranking of the evaluated algorithms and some information for each entry.

4. CONCLUSION

This paper described the current challenges and available corpora for gesture recognition and demonstrated the need for a corpus providing the possibility to objectively compare gesture spotting algorithms. The ChAirGest Open Challenge has been introduced and its multimodal corpus has been precisely described presenting its advantages for the evaluation of the gesture spotting task and research based on multiple heterogeneous sensors. A novel performance metric has also been introduced to provide a single evaluation measure for the recognition and spotting algorithms.

5. ACKNOWLEDGMENTS

This research has been supported by HASLER foundation within the framework of "Living in Smart Environment project".

6. REFERENCES

[1] Alon, J. et al. 2009. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE transactions on pattern analysis and machine intelligence*. 31, 9 (Sep. 2009), 1685–99. DOI=10.1109/TPAMI.2008.203.

[2] Amft, O. and Tröster, G. 2008. Recognition of dietary activity events using on-body sensors. *Artificial intelligence in medicine*. 42, 2 (Feb. 2008), 121–36. DOI=10.1016/j.artmed.2007.11.007.

[3] Amma, C. et al. 2010. Airwriting recognition using wearable motion sensors. *Proceedings of the 1st Augmented Human International Conference* (2010), 1–8.

[4] Banos, O. et al. 2012. Kinect=IMU? Learning MIMO Signal Mappings to Automatically Translate Activity Recognition Systems across Sensor Modalities. *2012 16th International Symposium on Wearable Computers* (Jun. 2012), 92–99. DOI=10.1109/ISWC.2012.17.

[5] Carrino, S. et al. 2011. Gesture-based hybrid approach for HCI in ambient intelligent environmments. *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)* (Jun. 2011), 86–93. DOI=10.1109/FUZZY.2011.6007691.

[6] CVAP arm/hand activity database: *http://www.csc.kth.se/~danik/gesture_database/*. Accessed: 2012-11-23.

[7] De, F. and Jessica, T. 2009. Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database. July (2009).

[8] Dreuw, P. et al. 2008. Benchmark Databases for Video-Based Automatic Sign Language Recognition. *International Conference on Language Resources and Evaluation* (Marrakech, Morocco, 2008), 1–6.

[9] Fawcett, T. 2006. An introduction to ROC analysis. *Pattern Recognition Letters*. 27, 8 (Jun. 2006), 861–874. DOI=10.1016/j.patrec.2005.10.010.

[10] Firth, N. 2013. Kinect sensor leaps into every day life. *New Scientist*. 217, 2900 (Jan. 2013), 22. DOI=10.1016/S0262-4079(13)60159-1.

[11] Fothergill, S. et al. 2012. Instructing people for training gestural interactive systems. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. (2012), 1737. DOI=10.1145/2207676.2208303.

[12] Glomb, P. et al. 2011. Choosing and Modeling Hand Gesture Database for Natural User Interface. *Proceedings of the 9th International Gesture Workshop* (Athens, Greece, 2011), 72–75.

[13] Gross, R. and Shi, J. 2001. *The CMU motion of body (MoBo) database*. Citeseer. DOI=10.1.1.16.5015.

[14] Guerra-Filho, G. and Biswas, A. 2012. The human motion database: A cognitive and parametric sampling of human

motion. *Image and Vision Computing*. 30, 3 (Mar. 2012), 251–261. DOI=http://dx.doi.org/10.1016/j.imavis.2011.12.002.

[15] Guyon, I. et al. 2012. ChaLearn Gesture Challenge: Design and First Results. *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (Jun. 2012), 1–6. DOI=10.1109/CVPRW.2012.6239178.

[16] Hwang, B. et al. 2006. A Full-Body Gesture Database for Automatic Gesture Recognition. *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. (2006), 243–248. DOI=10.1109/FGR.2006.8.

[17] Intel Perceptual Computing Challenge: *http://software.intel.com/en-us/vcsource/tools/perceptual-computing-sdk*.

[18] Kawaguchi, N. et al. 2011. HASC Challenge: gathering large scale human activity corpus for the real-world activity understandings. *Augmented Human 2011* (2011), 27–27.

[19] Kela, J. et al. 2006. Accelerometer-based gesture control for a design environment. *Personal Ubiquitous Comput.* 10, 5 (2006), 285–299. DOI=10.1007/s00779-005-0033-8.

[20] Kuehne, H. et al. 2011. HMDB: A large video database for human motion recognition. *2011 International Conference on Computer Vision*. (Nov. 2011), 2556–2563. DOI=10.1109/ICCV.2011.6126543.

[21] Martinez, AM; Shay, R;Kak, A. 2002. Purdue RVL-SLLL ASL Database for Automatic Recognition of Amercian Sign Language. *Proceedings of the 4th IEEE International Conference*. (2002).

[22] Meinard, M. et al. 2007. *Documentation Mocap Database HDM05*.

[23] Nancel, M. et al. 2011. Mid-air pan-and-zoom on wall-sized displays. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), 177–186. DOI=10.1145/1978942.1978969.

[24] Neumann, D.A. 2002. *Kinesiology of the Musculoskeletal System*.

[25] Pavlovic, V.I. et al. 1997. Visual interpretation of hand gestures for human-computer interaction: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 19, 7 (1997), 677–695.

[26] Roggen, D. et al. 2011. OPPORTUNITY : Towards opportunistic activity and context recognition systems. *Networks*. February 2009 (2011).

[27] Sigal, L. et al. 2009. HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. *International Journal of Computer Vision*. 87, 1-2 (Aug. 2009), 4–27. DOI=10.1007/s11263-009-0273-6.

[28] Ward, J.A. et al. 2011. Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology*. 2, 1 (Jan. 2011), 1–23. DOI=10.1145/1889681.1889687.

[29] Zhou, H. et al. 2008. Use of multiple wearable inertial sensors in upper limb motion tracking. *Medical Engineering & Physics*. 30, 1 (2008), 123–133.