Gesture Recognition for Improved User Experience in a Smart Environment

S. Gaglio, G. Lo Re, M. Morana, M. Ortolani

In Proceedings of the thirteenth International Conference on Advances in Artificial Intelligence

Gesture Recognition for Improved User Experience in a Smart Environment

Salvatore Gaglio, Giuseppe Lo Re, Marco Morana and Marco Ortolani

DICGIM, University of Palermo – Italy
{salvatore.gaglio,giuseppe.lore, marco.morana, marco.ortolani}@unipa.it

Abstract. Ambient Intelligence (AmI) is a new paradigm that specifically aims at exploiting sensory and context information in order to adapt the environment to the user's preferences; one of its key features is the attempt to consider common devices as an integral part of the system in order to support users in carrying out their everyday life activities without affecting their normal behavior.
Our proposal consists in the definition of a gesture recognition module allowing users to interact as naturally as possible with the actuators available in a smart office, by controlling their operation mode and by querying them about their current state. To this end, readings obtained from a state-of-the-art motion sensor device are classified according to a supervised approach based on a probabilistic support vector machine, and fed into a stochastic syntactic classifier which will interpret them as the basic symbols of a probabilistic gesture language. We will show how this approach is suitable to cope with the intrinsic imprecision in source data, while still providing sufficient expressivity and ease of use.

1 Introduction

Our daily-life activities involve spending an increasingly high amount of time indoor, whether at home or at work, so improving the perceived quality-of-life without intruding into the users' habits is of great importance; this is the purpose of the novel discipline of Ambient Intelligence (AmI) [1], which combines the use of pervasively deployed sensing devices and actuators with advanced techniques from Artificial Intelligence. Due the primary role of the end user, the intrinsic prerequisite of any AmI system (i.e. the presence of "ubiquitous" monitoring tools) must be coupled to the additional requirement of providing the system with efficient functionalities for interaction with the system; moreover, considering the high level of pervasiveness obtainable through currently available devices, the use of equally unobtrusive interfaces is mandatory. An interesting scenario is represented by *smart offices*, where typical applications include energy efficiency control, ambient-assisted living (AAL) and human-building interaction (HBI).

In this work, we build up on our previous experience about a testbed for designing and experimenting with WSN-based AmI applications [2] and we describe a module aimed at assisting users in order to ease their interaction with the system itself. In our previous work, we described the deployment of the AmI infrastructure in our department premises; the sensory component is implemented through a Wireless Sensor and Actuator Network (WSAN), whose nodes are equipped with off-the-shelf sensors for measuring such quantities as indoor and outdoor temperature, relative humidity, ambient light exposure and noise level. WSANs extend the functionalities of traditional sensor networks by adding control devices to modify the environment state. The present proposal suggests the use of a motion sensor device as the primary interface between the user and the AmI system. The device we used for sensing motion within the environment is Microsoft Kinect, which, in our vision, plays the dual role of a sensor (since it may be used for some monitoring tasks, i.e. people detection and people counting) and a controller for the actuators. In particular, the latter is performed by training a probabilistic classifier for recognizing some simple hand gestures in order to produce a basic set of commands; since the output of such a classifier is noisy due to the non perfect recognition of the hand shape, we chose to reinforce the classification process by means of a probabilistic syntactic recognizer, realized as a stochastic parser for an ad-hoc gesture language. The use of a grammar for the comprehension of the visual commands is significant as it helps both in dealing with the intrinsically noisy input, and in providing a smoother interface for users.

The paper is organized as follows: after presenting some related work in Section 2, we describe the main modules of the proposed system architecture in Section 3. The assessment of the system in a deployment scenario realized in our department will be discussed in Section 4. Conclusions will follow in Section 5.

2 Related Work

Several works have focused on hand gesture recognition using depth image; a survey is presented in [3], whereas in [4] an approach for hand gesture recognition using Kinect is proposed. The authors preliminary detect the hand by thresholding the depth image provided by the Kinect, then the finger identification task is performed by applying a contour tracing algorithm on the detected hand image. Even if the obtained results are promising, the whole system is based on a number of parameters and thresholds that make the approach unreliable while varying the application scenario. In [5] a novel dissimilarity distance metric based on Earth Mover's Distance (EMD) [6], i.e., Finger-Earth Mover's Distance (FEMD), is presented. The authors showed that FEMD based methods for hand gesture recognition outperform traditional shape matching algorithm both in speed and accuracy.

Besides processing visual input, we are interested in classifying the different hand shapes and effectively translate them into commands for the AmI system. Our system is based on a structural approach to pattern recognition of vectors of features. Most commonly used methods (e.g. nearest-neighbor classifiers, or neural networks) resort to machine-learning techniques for addressing this problem when the features can be represented in a metric space. Our choice, instead, was to work with symbolic input and to rather address a discrete problem by a syntactic pattern recognition method. More specifically, we assume that if user hand gestures are to represent an intuitive language for interacting with the system, then rules of some sort must be used to generate the strings of such language; hence, their inherent structure should not be disregarded, and might turn into a useful support when the system needs to recognize and interpret such strings. This approach is not new and in fact its first proposals date back in time, even though they were referred to different contexts, such as image processing or grammatical inference [7]; recognition based on parsing has been successfully employed also to automatic natural language of handwriting recognition [8].

Such methods have been later expanded also to allow for stochastic grammars, where there are probabilities associated with productions [9], and it has been shown that a grammar may be considered a specification of a prior probability for a class. Error-correcting parsers have been used when random variations occur in an underlying stochastic grammar [10]. Finally probability theory has been applied to languages in order to define the probabilities of each word in a language [11].

3 Providing Smooth Human-System Interaction

Our aim is to provide users of a smart environment with the possibility to interact with the available actuators as naturally as possible, by controlling their operation mode and by querying them about their current state. For instance, they might want to act upon some of the actuators (e.g. air conditioning system, or lighting) by providing a set of consecutive commands resulting in complex configurations, such as turn on the air conditioning system, set the temperature to a certain degree, set the fan speed to a particular value, set the air flow to a specified angle, and so on.

This section describes our proposal for an interface whose use, consistently with AmI philosophy, will not impact on the users' normal behavior; to this aim, we adapt the functionality of a flexible motion sensing input device, namely Microsoft Kinect, to detect simple gestures of the user's hand and to translate them into commands for the actuators. This peripheral has attracted a number of researchers due to the availability of open-source and multi-platform libraries that reduce the cost of developing new algorithms; a survey of the sensor and corresponding libraries is presented in [12, 13]. Direct use of the motion sensor is however not viable for our purposes, due to its non negligible intrinsic imprecision; moreover, naively mapping hand gestures into commands would be very awkward for users, and would likely result into them refusing to use the interface altogether.

Our perspective is to consider Kinect as a sensor to transparently gather observations about users' behavior [14], higher-level information may be extracted by such sensed data in order to produce actions for adapting the environment to the users requirements, by acting upon the available actuators. A preliminary version of this approach was presented in [15] with reference to the architecture

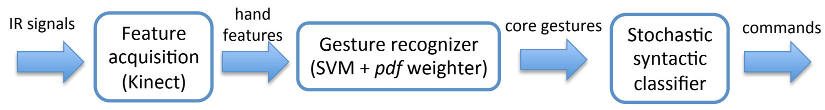

IR signals
Feature acquisition (Kinect)
hand features
Gesture recognizer (SVM + *pdf* weighter)
core gestures
Stochastic syntactic classifier
commands

Fig. 1. The main functional blocks of the proposed gesture recognition algorithm.

described in [16]. In order to turn the Kinect sensor into an effective human-computer interface, we adopt a probabilistic approach based on recognizing simple hand gestures (i.e., the number of extended fingers) in order to produce a set of commands; however, the output of such a classifier will be affected by noise due to the imperfect recognition of the shape of hand, therefore we chose to reinforce the classification process by means of a grammar. The use of an error-correcting parser for such grammar will add significant improvement to the recognition of visual commands and will make the use of the whole system more natural for the end user.

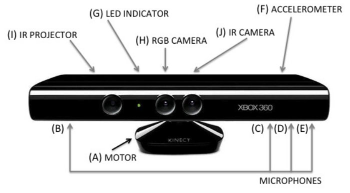
The overall scheme for our gesture recognition system is shown in Figure 1.

3.1 Capturing and Classifying Hand Gestures

Several vision-based systems have been proposed during the last 40 years for simple gesture detection and recognition. However, the main challenge of any computer vision approach is to obtain satisfactory results not only in a controlled testing environment, but also in complex scenarios with unconstrained lighting conditions, e.g., a home environment or an office. For this reason, image data acquired by multiple devices are usually merged in order to increase the system reliability. In particular, range images, i.e., 2-D images in which each pixel contains the distance between the sensor and a point in the scene, provide very useful information about the elements of the scene, e.g., a moving person, but range sensors used to obtain them are very expensive.

We selected Kinect for our target scenario, thanks to its flexibility, and limited cost; it is equipped with 10 input/output components, as depicted in Figure 2, which make it possible to sense the users and their interactions with the surrounding environment. Its projector shines a grid of infrared dots over the scene, and the embedded IR camera captures the infrared light, then its factory calibration allows to compute the exact position of each projected dot against a surface at a known distance from the camera. Such information is finally used to create depth images of the observed scene, with pixel values representing distances, in order to capture the object position in a three-dimensional space.

An example of hand tracking using Kinect is provided by the OpenNI/NITE packages, whose hand detection algorithm is based on the five gesture detectors listed on the right-hand side of Figure 2, and a hand point listener; however, those APIs are based on a global skeleton detection method, so the hand is defined just as the termination of the arm and no specific information about the hand state (e.g., an open hand vs a fist or the number of extended digits) is



Gesture	Description
`wave`	hand movement as left-right waving, carried out at least four times in a row.
`push`	hand movement as pushes towards and away from the sensor.
`swipe`	hand movement, either up, down, left or right, followed by the hand resting momentarily.
`circle`	hand that make a full circular motion in either direction.
`steady`	hand that hasn't moved for some time.

Fig. 2. Kinect components, and the included OpenNI/NITE gesture detectors.

provided. For this reason, we only consider this method as the first step of our gesture recognition algorithm, and we add further processing to extract more precise information based on the image representing the area where the hand is located.

The depth images are segmented in order to detect the *hand plane*, i.e. the set of image points that are at the same distance z from the Kinect sensor, where z is the value of the depth image $DI(x, y)$, with x and y indicating the coordinates of the hand as detected by the APIs. Each hand mask is then normalized with respect to scale, and represented as a time-series curve [17]. Such shape representation techniques is also proposed in [5] and is one of the most reliable method for the classification and clustering of generic shapes. A time-series representation allows to capture the appearance of the hand shape in terms of distances and angles between each point along the hand border and the center of mass of the hand region; in particular, as suggested by [5], a time time-series representation can be plotted as a curve where the horizontal axis denotes the angle between each contour vertex, the center point and a reference point along the hand border and the vertical axis denotes the Euclidean distance between the contour vertices and the center point. Figure 3 shows three examples of hand masks, as depth images capture by Kinect (top row); the centroid of the hand region (red cross) is used as center point for the time-series computation, while the lowest point along the hand perimeter (red circle) is used as reference point for computing the angles between the center and the contour points. The time-series curve representation (bottom row) is obtained by plotting the angles in the horizontal axis and the distances on the vertical axis.

The time-series describing the hand shape represents the feature we will analyze in order to discriminate between the set of considered hand gestures; in particular we need to classify each captured hand image according to one of six possible classes, with each hand gesture characterized by the number of extended fingers (from zero in the case of a fist, to five if the hand is open).

The gesture classification has been performed by means of a multi-class Support Vector Machine (SVM) classifier based on a RBF kernel. SVMs are supervised learning models used for binary classification and regression, and multi-class SVMs are usually implemented by combining several binary SVMs according to three main strategies: one-versus-all, one-versus-one, Directed Acyclic

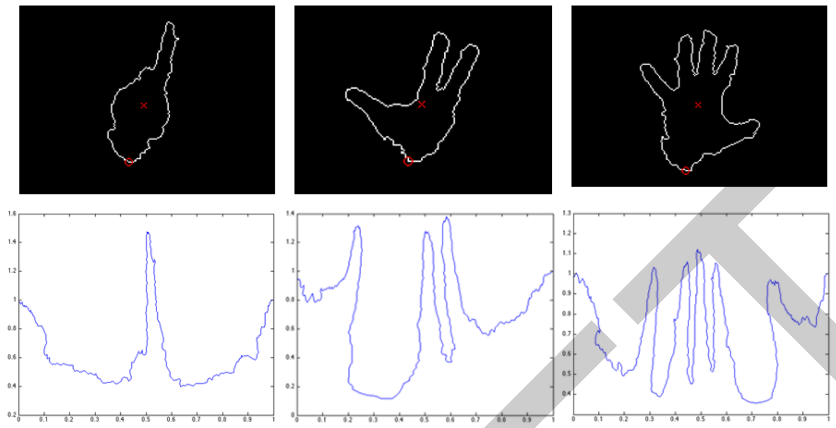


Fig. 3. Gesture representation by means of time-series features.

Graphs SVM (DAGSVM). Several studies addressed the issue of evaluating which is the best multi-class SVM method and most works (e.g., [18, 19]) claimed that the one-versus-one approach is preferable to other methods. However, what we want to obtain is a membership degree of each detected gesture in the whole set of gestures. Since the output of an SVM is not a probabilistic value, we chose to apply the method proposed by [20] to map the SVM output into a positive class posterior probability by means of a sigmoid function.

3.2 Gesture Language Description

Our Kinect sensor provides an effective way of capturing the input from the user, but although in principle hand gestures could be directly translated into commands, the mere recognition of very basic ones is likely inadequate to cover the broad spectrum of possible instructions with sufficient detail; moreover, a 1-to-1 correspondence between gesture and command would result into an awkward interaction with the system. An additional challenge is represented by the intrinsic imprecision in data obtained through Kinect, which is represented by the probabilistic output of the SVM classifier. In order to obtain an overall satisfying behavior, and to allow users to express a broad set of commands in a natural way starting from elementary and customary gestures, we regard the set of possible commands and queries as strings of a language, such as for instance: "**query** id_1 **status**", or "**set** id_1, id_2, id_3 **start**" for getting the operating status of actuator id_1, or activating actuators id_1, id_2, id_3, respectively.

In this perspective, the language can be precisely defined with the notation borrowed from formal language theory, and hand gestures can be regarded as the symbols of the underlying alphabet of core gestures, assuming we can sample the images acquired by the Kinect with a pre-fixed frequency (i.e., we can identify repetitions of the same symbol); the Kinect sensor only allows to roughly assess the number of extended fingers, with no finer detail; moreover, we will consider an additional symbol representing a separator, corresponding to the case when no gesture is made. The following alphabet will thus constitute the basis for the subsequent discussion:

$$\Sigma = \{\circ, \bullet^1, \bullet^2, \ldots, \bullet^5, \textvisiblespace\};$$

with $\circ$ indicating the fist (i.e. no recognized finger), $\bullet^n$ indicating that n fingers were detected, and $\textvisiblespace$ as the separator, when the hand is out of the field of view.

We used this alphabet to code the basic keywords of our language, and we devised a basic grammar capturing a gesture language expressing simple queries and commands to the actuators. So for instance, the proper sequence of gestures by the user (i.e. "$\bullet^1 \textvisiblespace$") will be understood as the **query** keyword, representing the beginning of the corresponding statement, and similarly for other "visual lexemes".

Even a straightforward language will be able to capture an acceptable range of instructions given by the user in a natural way, and may be easily defined in terms of a formal grammar; its core symbols, however, will inevitably be affected by noise, due to the imprecision in the hand shape classification, and such uncertainty will be expressed by the posterior probabilities attached by the SVM classifier to each instance of the hand shape. In order to interpret instructions for actuators we will thus need to infer the correct gesture pattern to which the incoming symbol sequence belongs.

Following the approach initially described by [21], we chose to regard this problem as a variant of stochastic decoding: we formalized the structure of proper queries and commands by means of context-free grammars expressed in Backus-Naur Form (BNF), and we consequently expect our instructions to conform to one of the pre-defined gesture patterns. In particular, our reference grammars for the language of queries $\mathcal{L}_q$, and the language of commands $\mathcal{L}_c$ are respectively defined as follows:

$$\begin{aligned} S' &\rightarrow \textit{stat}' \mid \textit{stat}'\ S' \\ \textit{stat}' &\rightarrow \textbf{query}\ \textit{idlist}\ \textbf{status} \\ &\mid \textbf{query}\ \textit{idlist}\ \textbf{compare}\ \textit{id} \end{aligned}$$

$$\begin{aligned} S'' &\rightarrow \textit{stat}'' \mid \textit{stat}''\ S'' \\ \textit{stat}'' &\rightarrow \textbf{set}\ \textit{idlist}\ \textit{cmd} \\ \textit{cmd} &\rightarrow \textbf{start} \mid \textbf{stop} \\ &\mid \textit{modif}\ [\textbf{increase} \mid \textbf{decrease}] \\ \textit{modif} &\rightarrow [\textbf{low} \mid \textbf{med} \mid \textbf{high}] \end{aligned}$$

where the sets of non-terminal symbols, and the start symbol of each grammar are implicitly defined by the productions themselves, and the terminal ones are actually coded in terms of the gesture alphabet[1].

Interpreting gestures may thus be regarded as an instance of a two-class classification problem in which each class of patterns is assumed to be described by the language generated by either of the two grammars. The symbols of alphabet are produced by a noisy sensor, so we must take into account the presence of erroneous symbols; however our noise deformation model will assume that the

[1] For the sake of brevity the productions for *idlist* were not included; unsurprisingly, the list of *id*'s can be described by a regular expression over the available symbols. The regular expression $(\bullet^1 \circ)^+$ was used to define a valid *id* in our prototypal system.

deformation process does not affect the length of the original string. The only information needed to characterize the process is the probability of corruption for each of the input symbols; more specifically, if $a, b \in \Sigma$ are two terminal symbols (i.e. hand gestures), we take advantage of the training process for the SVM classifier also to compute the conditional probability $r(a|b)$ that a is mistakenly interpreted as b; the probability that an entire string x is deformed to turn into a different y (in other words, that a query may be mistaken for a command, for instance) is given by $p(x|y) = r(a_{1)}|b_1)r(a_{2)}|b_2)\ldots r(a_{n)}|b_n)$.

As pointed out in [22], this kind of stochastic decoding basically corresponds to maximum-likelihood classification of deformed patterns. In our model, the grammars for both $\mathcal{L}_q$ and $\mathcal{L}_c$ are context-free, so each of them may be rewritten in Chomsky normal form; this allows us to use the probabilistic version of the traditional Cocke-Younger-Kasami parsing method as described in [21] to reliably recognize gesture patterns and translate them into instructions for the actuators.

At the end of the process, we obtain a reliable translation of a whole sequence of gestures into the corresponding command or query for the actuators. Such approach gives us also the opportunity to exploit the potentialities of the parser used to process the visual language; the underlying structure provided by our formulation in terms of grammars results into a smoother interaction for the user, as well as in greater precision for the overall gesture sequences recognition.

4 Performance Assessment

The proposed method is part of a system aiming for timely and ubiquitous observations of an office environment, namely a department building, in order to fulfill constraints deriving both from the specific user preferences and from considerations on the overall energy consumption.

The system will handle high-level concepts as "air quality", "lighting conditions", "room occupancy level", each one referring to a physical measurement captured by a physical layer. Since the system must be able to learn the user preferences, ad-hoc sensors for capturing the interaction between users and actuators are needed similarly to what is described in [23]. The plan of one office, giving an example of the adopted solutions, is showed in Figure 4.

The sensing infrastructure was realized by means of a WSAN, whose nodes are able to measure temperature, relative humidity, ambient light exposure and noise level. Actuators were available to change the state of the environment by acting on some quantities of interest; in particular the air-conditioning system, the curtain and rolling shutter controllers, and the lighting regulator address this task by modifying the office temperature and lighting conditions. The users' interaction with actuators was captured via the Kinect sensor (Fig. 4-**H**) that was also responsible for detecting the presence and count the number of people on the inside of the office. The Kinect was connected to a miniature fanless PC (i.e., fit-PC2i) with Intel Atom Z530 1.6GHz CPU and Linux Mint OS, that

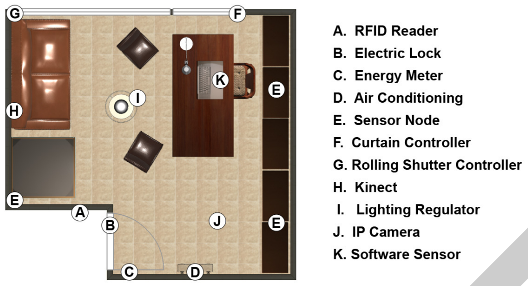


Fig. 4. Monitored office.

guarantees real-time processing of the observed scene with minimum levels of obtrusiveness and power consumptions (i.e., 6W).

We evaluated the performance of our system by separately assessing the multi-class SVM classifier and the interpreter for the gesture language. The basic hand gesture classifier was tested under varying lighting conditions and poses showing an acceptable level of robustness, thanks to the fact that the input sensor makes use of depth information in detecting the hand mask, while compensating for the lower quality of the RGB data. Results showed that about 75% of the gestures are correctly classified when the user acts in a range of 1.5 to 3.5 meters from the Kinect; however, greater distances have a negative impact on performance due to the physical limits of the infrared sensor. Moreover we noticed that even when a gesture is misclassified, the correct gesture is the second choice (i.e., rated with the second most probable value) in approximately 70% of the cases. This observation shows that the information provided by the SVM classifier can provide reliable input for the gesture sequence interpreter.

The interpreter functionalities were preliminarily verified by means of a synthetic generator of gestures allowing for the validation of both the alphabet and the grammar we chose. The overall system has been tested by conducting a set of experiments involving 8 different individuals. Each person was positioned in front of Kinect at a distance within the sensor range and was asked to interact with the device by performing a random sequence of 20 gestures chosen from a predefined set of 10 commands, and 10 queries referred to the actuators depicted in Figure 4. Examples of commands and queries, expressed in terms of the grammars described in the previous section, were:

set *light* **low increase**	:	increases lighting by a small amount by acting on the dimmer
set *heater* **high increase**	:	increases HVAC temperature setpoint by 3°C
set *heater* **stop**	:	stops HVAC
query *light, heater* **status**	:	gets current status of light and HVAC
query *energymeter*	:	gets current value read from energy meter

where *light*, *heater*, *energymeter* indicate the id's of the corresponding actuators.

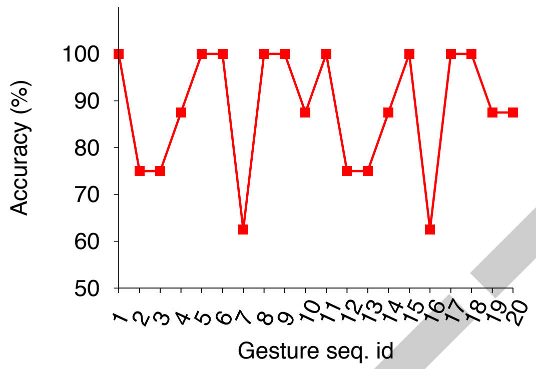


Fig. 5. Accuracy of gesture recognition.

The proposed system was able to correctly classify the input gestures sequences in 88.125% of the cases, corresponding to 141 positives out of 160 inputs; Figure 5 shows a plot of the outcome for each trial. While performing translation of the visual commands, the gesture interpreter had to classify the sequences into one of the two classes represented by the languages for queries and commands respectively, thus giving insight about the "structure" of the sentence recognized so far. In a future development, this could be used to provide feedback to the user about the system comprehension of the interaction, resulting in an overall smoother user experience.

5 Conclusions

This work described an approach to the design of a user-friendly interface based on gesture recognition as part of an AmI system; our application scenario is the management of an office environment by means of Kinect, an unobtrusive sensing tool equipped with input/output devices that make it possible to sense the user and their interactions with the surrounding environment. The control of the actuators of the AmI system (e.g., air-conditioning, curtain and rolling shutter) is performed by capturing some simple gestures via the Kinect and recognizing opportunely structured sequences by means of a symbolic probabilistic approach. Besides allowing for a smoother interaction between users and the system, our approach is able to cope with imprecision in basic gesture acquisition thanks to the use of stochastic decoding, which basically corresponds to maximum-likelihood classification of deformed patterns. The system was tested on an actual prototype of a smart office, which we built in our department premises as part of a research project investigating the use of Ambient Intelligence for energy efficiency.

Acknowledgment

This work was partially supported by the SMARTBUILDINGS project, funded by PO-FESR SICILIA 2007-2013.

References

1. E. Aarts and S. Marzano, *The new everyday: views on ambient intelligence.* 010 Publishers, 2003.
2. A. De Paola, S. Gaglio, G. Lo Re, and M. Ortolani, "Sensor9k: A testbed for designing and experimenting with WSN-based ambient intelligence applications," *Pervasive and Mobile Computing. Elsevier*, vol. 8, no. 3, pp. 448–466, 2012.
3. J. Suarez and R. Murphy, "Hand gesture recognition with depth images: A review," in *RO-MAN, 2012 IEEE*, sept. 2012, pp. 411 –417.
4. Y. Li, "Hand gesture recognition using kinect," in *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, june 2012, pp. 196 –199.
5. Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of the 19th ACM international conference on Multimedia*, ser. MM '11. New York, NY, USA: ACM, 2011, pp. 1093–1096.
6. Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vision*, vol. 40, no. 2, pp. 99–121, Nov. 2000.
7. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation, 3/E.* Pearson Education, 2008.
8. K.-S. Fu and J. E. Albus, *Syntactic pattern recognition and applications.* Prentice-Hall Englewood Cliffs, NJ, 1982, vol. 4.
9. M. Y. Liberman, "The trend towards statistical models in natural language processing," in *Natural Language and Speech.* Springer, 1991, pp. 1–7.
10. S.-Y. Lu and K.-S. Fu, "Stochastic error-correcting syntax analysis for recognition of noisy patterns," *Computers, IEEE Transactions on*, vol. 100, no. 12, pp. 1268–1276, 1977.
11. T. L. Booth and R. A. Thompson, "Applying probability measures to abstract languages," *Computers, IEEE Transactions on*, vol. 100, no. 5, pp. 442–450, 1973.
12. S. Kean, J. Hall, and P. Perry, *Meet the Kinect: An Introduction to Programming Natural User Interfaces*, 1st ed. Berkely, CA, USA: Apress, 2011.
13. G. Borenstein, *Making Things See: 3D Vision With Kinect, Processing, Arduino, and MakerBot*, ser. Make: Books. O'Reilly Media, Incorporated, 2012.
14. P. Cottone, G. Lo Re, G. Maida, and M. Morana, "Motion sensors for activity recognition in an ambient-intelligence scenario," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, 2013, pp. 646–651.
15. G. Lo Re, M. Morana, and M. Ortolani, "Improving user experience via motion sensors in an ambient intelligence scenario," in *PECCS 2013 - Proceedings of the 3rd International Conference on Pervasive Embedded Computing and Communication Systems, Barcelona, Spain, 19-21 February, 2013*, 2013, pp. 29–34.
16. A. De Paola, M. La Cascia, G. Lo Re, M. Morana, and M. Ortolani, "User detection through multi-sensor fusion in an ami scenario," in *Information Fusion (FUSION), 2012 15th International Conference on*, july 2012, pp. 2502 –2509.

17. E. Keogh, L. Wei, X. Xi, S.-H. Lee, and M. Vlachos, “Lb keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures,” in *Proceedings of the 32nd international conference on Very large data bases*, ser. VLDB ’06. VLDB Endowment, 2006, pp. 882–893.
18. C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415 –425, mar 2002.
19. K.-B. Duan and S. Keerthi, “Which is the best multiclass svm method? an empirical study,” in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, N. Oza, R. Polikar, J. Kittler, and F. Roli, Eds. Springer Berlin Heidelberg, 2005, vol. 3541, pp. 278–285.
20. J. C. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.
21. L.-W. Fung and K.-S. Fu, “Stochastic syntactic decoding for pattern classification,” *Computers, IEEE Transactions on*, vol. 100, no. 6, pp. 662–667, 1975.
22. R. C. Gonzalez and M. G. Thomason, *Syntactic pattern recognition: An introduction.* Addison-Wesley Publishing Company, Reading, MA, 1978.
23. M. Morana, A. De Paola, G. Lo Re, and M. Ortolani, “An Intelligent System for Energy Efficiency in a Complex of Buildings,” in *Proc. of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability*, 2012.