Improving User Experience via Motion Sensors in an Ambient Intelligence Scenario

Giuseppe Lo Re, Marco Morana and Marco Ortolani
DICGIM, University of Palermo, Viale delle Scienze, ed. 6, Palermo, Italy

Keywords: HCI, Sensor Networks, Ambient Intelligence, Embedded Systems.

Abstract: Ambient Intelligence (AmI) is a new paradigm in Artificial Intelligence that aims at exploiting the information about the environment state in order to adapt it to the user preferences. AmI systems are usually based on several cheap and unobtrusive sensing devices that allow for continuous monitoring in different scenarios. In this work we present a gesture recognition module for the management of an office environment using a motion sensor device, namely Microsoft Kinect, as the primary interface between the user and the AmI system. The proposed gesture recognition method is based on both RGB and depth information for detecting the hand of the user and a fuzzy rule for determining the state of the detected hand. The shape of the hand is interpreted as one of the basic symbols of a grammar expressing a set of commands for the actuators of the AmI system. In order to maintain a high level of pervasiveness, the Kinect sensor is connected to a miniature computer capable of real-time processing.

1 INTRODUCTION

With the widespread diffusion of cheap and unobtrusive sensing devices, nowadays it is possible to perform continuous monitoring of a wide range of different environments. The availability of an ever-increasing amount of data acquired by such sensor devices, has piqued the interest of the scientific community in producing novel methods for combining raw measurements in order to understand what is happening in the monitored scenario.

Many works have been proposed in literature that address the problem of heterogeneous data analysis for obtaining a unitary representation of the observed scene. In particular, Ambient Intelligence (AmI) is a new paradigm in Artificial Intelligence that aims at exploiting the information about the environment state in order to adapt it to the user preferences. Thus, the intrinsic requirement of any AmI system is the presence of pervasive sensory devices; moreover, due the primary role of the end user, an additional requirement is to provide the system with efficient HCI functionalities. Considering the high level of pervasiveness obtained through the use of the nowadays available sensory and actuating devices, the use of equally unobtrusive interfaces is mandatory.

In this work we present a system for the management of an office environment, namely the rooms of a university department, using a motion sensor device, i.e. Microsoft Kinect, as the primary interface between the user and the AmI system. In our architecture, 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. Such networks (De Paola et al., 2012b) do not only passively monitor the environment, but represent the tool allowing the system to interact with the surrounding world. WSANs are the active part of the system and allow to modify the environment according to the observed data, high-level goals (e.g., energy efficiency) and user preferences.

In our vision, Kinect represents both a sensor (since it is used for some monitoring tasks, i.e. people counting) and a controller for the actuators. In particular, the people counter algorithm we developed is based on an optimized version of the method natively implemented by the Kinect libraries, taking into account the limited computational resources of our target device. The actuators control is performed by training a fuzzy classifier for recognizing some simple gestures (i.e., open/closed hands) in order to produce a set of commands opportunely structured by means of a grammar. The use of a grammar for the comprehension of the visual commands is also exploited

Lo Re G., Morana M. and Ortolani M..
Improving User Experience via Motion Sensors in an Ambient Intelligence Scenario.
DOI: 10.5220/0004306000290034
In *Proceedings of the 3rd International Conference on Pervasive Embedded Computing and Communication Systems* (PECCS-2013), pages 29-34
ISBN: 978-989-8565-43-3

since we make use of the parser in order to fine tune the behavior of the fuzzy recognizer.

The paper is organized as follows: related works are presented in Section 2, while the proposed system architecture is described in Section 3. An experimental deployment realized in our department will be discussed in Section 4. Conclusions will follow in Section 5.

2 RELATED WORK

The core of our proposal involves the use of a reliable gesture detection device, and we selected Microsoft Kinect as a promising candidate to this aim. Kinect is based on the hardware reference design and the structured-light decoding chip provided by PrimeSense, an Israeli company whose also provides a framework, OpenNI, that supplies a set of APIs to be implemented by the sensor devices, and another set of APIs, NITE, to be implemented by the middleware components. Moreover, PrimeSense ha recently released a proprietary Kinect-based sensor, called PrimeSense 3D Sensor.

Even if Kinect has been on the market for a couple of years, it 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 (Kean et al., 2011; Borenstein, 2012).

In (Xia et al., 2011) a method for human bodies detection using depth information taken by the Kinect is presented. The authors perform the detection task by applying some state of the art computer vision techniques, however their system is developed in a traditional PC so that computationally intensive tasks (i.e., 3D modeling) cannot be implemented in a low-power device. The problem of segmenting humans by using the Kinect is also addressed in (Gulshan et al., 2011), while the authors of (Raheja et al., 2011) focused on hand tracking. The authors presented an intuitive solution by detecting the palm and then the fingers, however, in order to obtain high-resolution hand images that can be successfully processed, the user is forced to stay close to the Kinect.

In our vision, the Kinect sensor represents an "eye" that observes the user acting freely in the environment, collects information and forwards user request to a reasoner according to the architecture described in (De Paola et al., 2012a). The remote devices act as the termination of a centralized sentient reasoner that is responsible of intelligent processing. Higher-level information is extracted by sensed data

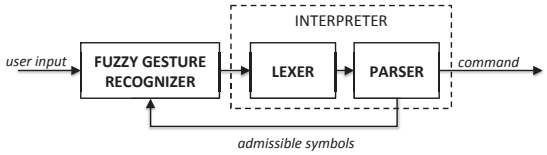

Figure 1: Block diagram of the proposed gesture recognition algorithm.

in order to produce the necessary actions to adapt the environment to the users requirements. A set of actuators finally takes care of putting the planned modifications to the environment state into practice.

3 SYSTEM OVERVIEW

In the context of Ambient Intelligence, a key requirement is that the presence of the monitoring and control infrastructure is hidden from the users, so as to provide a smoother way for them to interact with the system. For the purpose of the present discussion, we will specifically consider the possibility for the users 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, the user can control some actuators (e.g. air conditioning system, or lighting) by providing a set of subsequent commands for obtaining complex configurations, e.g., 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.

Figure 1 shows a block diagram for the HCI module of our system; namely it depicts the core components of the proposed gesture recognition algorithm. The actions of the users are captured by Kinect and analyzed by the Fuzzy Gesture Recognizer. The recognized input symbols are then processed by our interpreter and, at each step (i.e. for every recognized gesture), a set of the next admissible input symbols is provided as feedback to the fuzzy classifier.

3.1 Fuzzy Gesture Recognition

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., 2D images

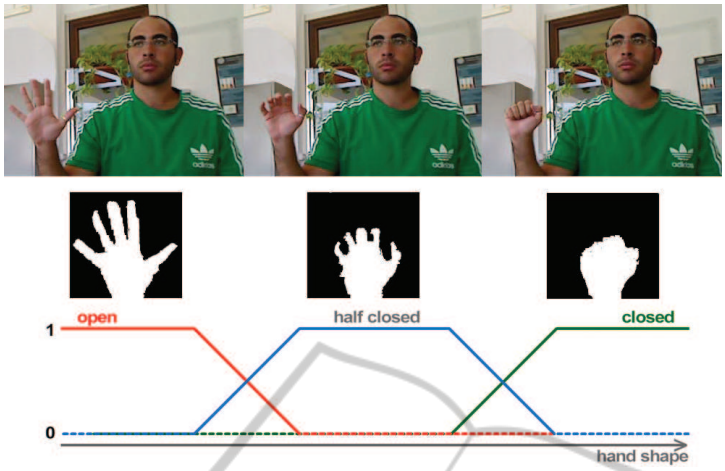

Figure 2: The gesture recognition method. For each frame (first row), the hand is detected by means of OpenNI / NITE APIs. The coordinates of the detected hands are used to define a search window whose size is proportionally to the distance from the Kinect. RGB-D information allow to obtain an hand mask (second row) that is described in terms of its roundness. A fuzzy technique (third row) is finally applied to express the uncertainty about the hand shape.

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.

According to the considered scenario, we found that Kinect represents the most suitable device both in terms of cost and functionalities since it is equipped with ten input/output components that make it possible to sense the users and their interaction with the surrounding environment. The Kinect sensor rests upon a base which contains a motor that allows for controlling the tilt angle of the cameras (30 degrees up or down). Starting from the bottom of the device, you can find three adjacent microphones on the right side, while a fourth microphone is placed on the left side. A 3- axis accelerometer can be used for measuring the position of the sensor, while a led indicator shows its state. However, the core of the Kinect is represented by the vision system composed of: an RGB camera with VGA standard resolution (i.e., 640x480 pixels); an IR projector that shines a grid of infrared dots over the scene; an IR camera that captures the infrared light. The factory calibration of the Kinect make it possible to know the exact position of each projected dot against a surface at a known distance from the camera. Such information is then used to create depth images of the observed scene (i.e., pixel values represent distances) that capture the object position in a three-dimensional space.

An example of hand tracking using Kinect comes with the OpenNI/NITE packages.

However, the APIs are based on a global skeleton detection method, so that 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) is provided. For this reason, such approach is useful just as first step of our detection procedure since it allows us to define the image area where the hand is located.

The coordinates of the detected hand are used to define a search window, whose size is chosen according to a heuristic rule based on the distance z from the Kinect. In our system, the depth information is combined with data acquired by the RGB camera and a classification algorithm (Lai et al., 2011) is applied in order to define the hand mask within the search window. Each hand mask is then normalized with respect to scale and the final binary region is described in terms of its roundness. In particular, the roundness of the hand is efficiently computed as the variance of the set of distances between each point along the hand border and the center of mass of the hand region:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n} \quad (1)$$

where μ are the coordinates of the center of mass and x are the coordinates of the n points along the border.

This feature provides useful information for discriminating between open/half open hands that have a low-level of roundness and closed hands that result almost round. Moreover, in order to better deal with the uncertainty of visual features, the concept of hand shape is modeled through a fuzzy logic rule based on the roundness values:

```
IF roundness IS low THEN hand is open
IF roundness IS normal THEN hand is half closed
```

```
IF roundness IS high THEN hand is closed
```

Fig. 2 shows the steps of the recognition procedure. The high-level reasoning for recognizing the hand gesture is performed by using the output of the fuzzy rule in conjunction with a set of grammar rules.

3.2 Gesture Language Description

As already mentioned, Kinect provides an effective way of capturing the input from the user, which in principle could be directly translated into commands; however, the mere recognition of hand gestures may prove inadequate to cover with sufficient detail the broad spectrum of possible instructions. The fuzzy recognizer described above, for instance, is able to tell only two hand gestures apart and, while those could be sufficient to translate a set of commands according to a binary alphabet, such coding would produce lengthy words, and would be too cumbersome to be of any practical use.

Our goal consists in providing the users with a tool able to let them express a (relatively) broad set of commands, starting from elementary and customary gestures; to this aim, we regard the set of possible commands and queries as a language, which can be precisely defined with the notation borrowed from formal language theory.

For our purposes, we define such language by specifying a simple grammar, expressed in the usual BNF notation (Aho et al., 2007); from this point of view, the hand gestures can be regarded as the symbols of the underlying alphabet, assuming we can sample the images acquired by the Kinect with a prefixed frequency (i.e., we can identify repetitions of the same symbol); 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:

$$\Sigma = \{\circ, \bullet, \text{␣}\};$$

with $\circ$ indicating the open hand, $\bullet$ the fist, and ␣ the separator; it is clear, however, that the alphabet can be easily extended by acting on the fuzzy recognizer.

Such alphabet is used to code the basic keywords, such as those for identifying the beginning of a statement; upon this, 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. "$\bullet$ ␣") will be understood as the **query** keyword, representing the beginning of the corresponding statement, and similarly for other "visual lexemes".

The grammar we used is a context-free grammar, which is completely defined in Backus-Naur Form (BNF) by the following productions[1]:

$$\begin{aligned}
P &\rightarrow Slist \\
Slist &\rightarrow stat \mid stat\ Slist \\
stat &\rightarrow query \mid cmd \\
query &\rightarrow \textbf{query}\ \ \textbf{id}\ \ [\textbf{status} \mid \textbf{value}] \\
cmd &\rightarrow \textbf{act_on}\ \ \textbf{id}\ \ \textbf{start}\ cmdLoop\ \textbf{stop} \\
cmdLoop &\rightarrow [\textbf{increase} \mid \textbf{decrease}] \\
&\quad \mid [\textbf{increase} \mid \textbf{decrease}]\ cmdLoop
\end{aligned}$$

Despite the simplicity of the devised language, its grammar is able to capture an acceptable range of instructions given by the user in a natural and unobtrusive way; the software running on the motion detection sensor provides the input symbols which are then processed by our interpreter and translated into commands/queries.

Such structured approach gives us also the opportunity to exploit the potentialities of the parser used to process the visual language; in particular, as is customary practice, our interpreter performs the recognition of an input sequence as a word of the language by building an internal data structure which matches the syntax of the sentence recognized so far; moreover, in order to keep the process computationally manageable, a set of the next admissible input symbols is constructed at each step (i.e. for every input symbol).

In our system, we exploit this information in order to tune the fuzzy recognizer tied to the motion sensor by means of a weighting mechanism (Cho and Park, 2000; Alcalá et al., 2003). Namely, the fact that at a given time instant only some of the possible input symbols (i.e. gestures) are expected provides an invaluable feedback which may be exploited by the fuzzy recognizer from avoiding useless computations. Although the effect of such feedback might appear almost negligible when only two different gestures are considered, the addition of more symbols is straightforward in our system, and such a feedback can heavily improve the efficiency of the fuzzy classifier by preliminarily discarding inadmissible symbols. For instance, we may conceivably consider some easily recognizable gestures involving the use of both hands and their relative position in order to allow the definition of a more complex grammar.

[1]The sets of terminal, and non-terminal symbols, and the start symbol are implicitly defined by the productions themselves.

4 CASE STUDY

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 reason on 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 (Morana et al., 2012). The plan of one office, giving an example of the adopted solutions, is showed in Figure 3. The devices labelled as A and B are used as a first-level, rough, control of the user's presence.

The sensing infrastructure is realized by means of a WSAN, whose nodes (Fig. 3-**E**) are able to measure temperature, relative humidity, ambient light exposure and noise level. Sensor nodes can be located close to several points of interest, e.g., door, window, user's desk; moreover, nodes equipped with outdoor sensors can also be installed on the building facade, close to the office windows, in order to monitor outdoor temperature, relative humidity, and light exposure. Actuators are able to change the state of the environment by acting on some measures of interest. The air-conditioning system (Fig. 3-**D**), the curtain and rolling shutter controllers (Fig. 3-**F, G**), and the lighting regulator (Fig. 3-**I**) address this task by modifying the office temperature and lighting conditions. Both temperature and light management are fundamental for the energy efficiency of the office, but in order to achieve better results a more accurate analysis of the power consumption is required. For this reason, we used energy monitoring units for each device (e.g, air-conditioner, lights, PCs) and an energy meter (Fig. 3-**C**) for the overall monitoring of each office. The users' interaction with actuators is captured via the Kinect sensor (Fig. 3-**H**) that is also responsible for detecting the presence and count the number of people on the inside of the office. An additional contribution for detecting user's presence is given by video sensors integrated with wireless sensor nodes (Fig. 3-**J**), that can be used to perceive high-level features such as who is in the office.

The monitoring infrastructure is based on the IRIS Mote produced by Crossbow, equipped with a number of sensors (i.e., temperature, humidity, light intensity, noise level, CO2). The IRIS is a 2.4 GHz Mote module designed specifically for deeply embedded sensor networks. Other ad-hoc sensors and actuators (e.g., curtain reader and controller) can be connected with the WSAN by means of standard protocols (e.g., ZigBee, EIA RS-485).

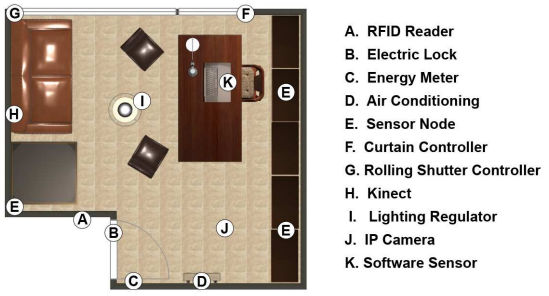

Figure 3: Monitored office.

The Kinect is connected to a miniature fanless PC (i.e., fit-PC2i) with Intel Atom Z530 1.6GHz CPU and Linux Mint OS, that guarantees real-time processing of the observed scene with minimum levels of obtrusiveness and power consumptions (i.e., 6W).

Several tests have been performed in order to separately evaluate both the fuzzy gesture recognizer and the interpreter. The former has been tested under varying lighting conditions and poses showing an acceptable level of robustness. This is mainly due to the primary role of the depth information in detecting the hand mask, while compensating for the lower quality of the RGB data. Results showed that about 70% of the gestures (i.e., masks of the hands) are correctly classified when the user acts in a range of 1.5 to 3.5 meters from the Kinect. Greater distances make performances worse due to the physical limits of the infrared sensor.

The interpreter functionalities have been 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 (i.e., turn on the light, turn off the light, turn on HVAC, turn off HVAC, set the temperature, set the airflow angle, open the door, lock the door, open the curtains, close the curtains) and 10 queries (i.e., get the instantaneous energy consumption, get the monthly energy consumption, get the list of active appliances, get the temperature, get the humidity level, get the position of the sensors, get the position of the actuators, get the state of the sensors, get the state of the actuators, get the state of the system).

The proposed system was able to correctly classify the input gestures in 83.75% of the cases, corresponding to 134 positives out of 160 inputs. Such a result shows that compared to the standalone fuzzy recognizer, representing the set of possible commands and queries as a language increases the performance of the system.

5 CONCLUSIONS

In this work we presented a system for the management of an office environments by means of an unobtrusive sensing device, i.e., the Kinect. Such a sensor is equipped with a number of input/output devices that make it possible to sense the user and its interaction with the surrounding environment. We considered a scenario where the whole environment is permeated with small pervasive sensor devices, for this reason the Kinect is coherently connected to a miniature fanless computer with reduced computation capabilities.

The control of the actuators of the AmI system (e.g., air-conditioning, curtain and rolling shutter) is performed by the Kinect by recognizing some simple gestures (i.e., open/closed hands) opportunely structured by means of a grammar.

Once the hand of the user has been detected, some local processing is done using RGB-D data and the obtained hand region is described according to its roundness. Such a descriptor is verified by means of a fuzzy procedure that predicts the state of the hand with a certain level of accuracy. Each state (i.e., open or closed) represents a symbol of a grammar that defines the corresponding commands for the actuators.

The construction of a real prototype of the monitoring and controlling system allowed for exhaustive testing of the proposed method. Experimental results showed that the system is able to perform efficiently on a miniature computer while maintaining a high level of accuracy both in terms of image analysis and gesture recognition.

Although the effectiveness of the system has been evaluated considering only two different gestures, the addition of more symbols is straightforward. As future work we may conceivably consider some easily recognizable gestures involving the use of both hands and their relative position in order to allow the definition of a more complex grammar.

ACKNOWLEDGEMENTS

This work is supported by the SMARTBUILDINGS project, funded by POR FESR SICILIA 2007-2013.

REFERENCES

Aho, A., Lam, M., Sethi, R., and Ullman, J. (2007). *Compilers: principles, techniques, and tools*, volume 1009. Pearson/Addison Wesley.

Alcalá, R., Casillas, J., Cordón, O., and Herrera, F. (2003). Linguistic modeling with weighted double-consequent fuzzy rules based on cooperative coevolutionary learning. *Integr. Comput.-Aided Eng.*, 10(4):343–355.

Borenstein, G. (2012). *Making Things See: 3D Vision With Kinect, Processing, Arduino, and MakerBot*. Make: Books. O'Reilly Media, Incorporated.

Cho, J.-S. and Park, D.-J. (2000). Novel fuzzy logic control based on weighting of partially inconsistent rules using neural network. *Journal of Intelligent Fuzzy Systems*, 8(2):99–110.

De Paola, A., Cascia, M., Lo Re, G., Morana, M., and Ortolani, M. (2012a). User detection through multisensor fusion in an ami scenario. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2502 –2509.

De Paola, A., Gaglio, S., Lo Re, G., and Ortolani, M. (2012b). Sensor9k: A testbed for designing and experimenting with WSN-based ambient intelligence applications. *Pervasive and Mobile Computing. Elsevier*, 8(3):448–466.

Gulshan, V., Lempitsky, V., and Zisserman, A. (2011). Humanising grabcut: Learning to segment humans using the kinect. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1127 –1133.

Kean, S., Hall, J., and Perry, P. (2011). *Meet the Kinect: An Introduction to Programming Natural User Interfaces*. Apress, Berkely, CA, USA, 1st edition.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011). Sparse distance learning for object recognition combining rgb and depth information. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4007 –4013.

Morana, M., De Paola, A., Lo Re, G., and Ortolani, M. (2012). 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*.

Raheja, J., Chaudhary, A., and Singal, K. (2011). Tracking of fingertips and centers of palm using kinect. In *Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on*, pages 248 –252.

Xia, L., Chen, C.-C., and Aggarwal, J. (2011). Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 15 –22.