Detecting and Tracking Multiple Users in the Proximity of Interactive Tabletops

Cristian Andy TĂNASE, Radu-Daniel VATAVU, Ştefan-Gheorghe PENTIUC, Adrian GRAUR
"Stefan cel Mare" University of Suceava
str.Universitatii nr.13, RO-720229 Suceava
andy_t@eed.usv.ro

***Abstract*—Interactive tabletops allow various opportunities for collaborative work as multiple users may interact simultaneously with the same multimedia content. Despite the fact that the interaction is shared and work is carried out collaboratively by multiple users, it is not always clear who interacts and where as there is no way of knowing where users position themselves at the table. We describe in this paper a method for detecting and tracking multiple users around the physical table using a simple solution implemented with a few short-range distance optical sensors. We are thus able to properly address the following questions: How many users are there? Where are they located around the table? Has a new user approached the tabletop or has someone just left? Are two users close to each other? We believe that answering these questions provides new interaction opportunities enriching thus the overall tabletop experience. We discuss possible applications and describe several implementations.**

***Index Terms*—sensors, proximity, tracking, detection, tabletop, human-computer interaction**

I. INTRODUCTION

Tabletops and interactive surfaces provide great potential for achieving collaborative work due to multiple users able to interact simultaneously with the same information. Content may be shared and manipulated among a group of people distributed around the physical table which triggers many similarities with the real-world practice. One connected problem occurs when users change locations frequently around the table. In this case, the content they own, i.e. windows of opened applications, graphical items, pictures, etc. must be manually dragged to the new location in order to be displayed correctly (i.e. so they would be displayed correctly up-front for users at their new location).

Several approaches have been proposed for solving the orientation problem. The most common solution mimics the real-world: users engage in manual control and re-position themselves the items they own [1]. This can be achieved out directly using gestures [3, 4] or via special designed widgets [6]. The manual reorientation approach is familiar in tabletop installations due to the direct paper or photo manipulation analogy from the real-world. Manipulation may be achieved using touch gestures [8, 9, 10], stylus pens [7] or dedicated widgets such as the DiamondSpin rotation handle [6] or the corner widget from the Photohelix system [2]. There is also the option of automatic self-positioning of objects according to the closest-edge [6], to the most recent user who touched or interacted with them [5] or as influenced by the presence of "magnets" [6]. Every approach comes however with its drawbacks and disadvantages and may not work every time and in every scenario.

It must be noted that the above orientation issue actually hides a more complex research problem: how to identify users at their exact locations around the table and even more, how to perform tracking of users while they move in the proximity of the tabletop installation.

The paper is organized as follows: section II describes our prototype consisting of installation details of short-range distance sensors around the table and hardware aspects of implementation; section III discusses the tracking mechanism that was put in place; section IV describes several implementations that benefit from detecting and tracking users in the proximity of the tabletop: login/greeting, items with auto-tracking features, a discussion on tabletop games; we conclude with discussions and present future work ideas. The main contributions of this paper are represented by our innovatory idea for the hardware prototype which addresses the orientation problem, tracking algorithm and the new interactions types which enrich the tabletop experience.

II. HARDWARE DESIGN

A prototype was built in our laboratory consisting of 12 short-range optical sensors which were installed on the 4 sides of a 75 x 55 cm^2 table (3 sensors for each side of the table). The placement of sensors makes that at least one and at most two sensors trigger valid detection when a person (with average thigh) stands close to the table hence only few sensors manage to cover all the table sides. A schematic view of our prototype is described in Figure 1 with various details for the sensors installation while actual implementations are illustrated by Figures 3 and 4. The following equipment was used for the development of the electronic circuit around the sensors:

- 12 SHARP GP2D12 general purpose distance measuring sensors were used with an equal number of 3 sensors per each side. The sensors consist in a PSD (Position Sensitive Detector), an infrared emitting diode and a signal and timing logic processing circuit. We chose them due to their high immunity to ambient light without any influence on the color of reflective objects. Reading of the output distance is continuous and they return a corresponding analog voltage. The effective detection range is in the interval 10-80 cm where the diameter of the actual detection area is 6 cm at maximum distance (80 cm). Figure 2 illustrates several characteristics of the short-distance sensors.
- A processing board FPGA Xilinx XC2V1000-4FG456C Altium NanoBoard NB1 was used in order to implement the program logic. The board contains of an

Digital Object Identifier 10.4316/AECE.2008.02011

I8052 microcontroller, an I2C controller and a RS232 interface. The output voltages as delivered by the short distance sensors were digitally converted by a MAX1037 A/D converter and then forwarded to the I8052 microcontroller. Data was filtered against noise and sent to the main desktop computer through the serial connection.

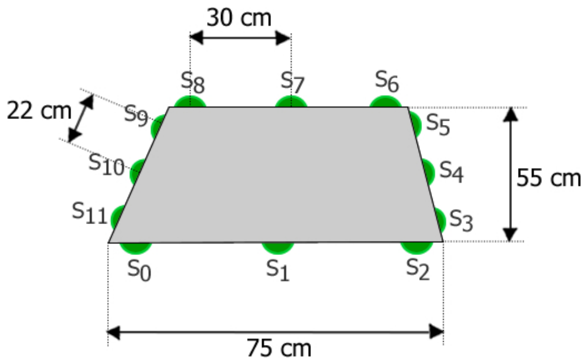

Figure 1. Prototype tabletop with locations of the sensors highlighted. For each side, one sensor was installed at the middle while the others were positioned next to the corners. At least one and at most two neighbor sensors trigger detection when a person approaches the table.

Every short-range IR sensor is fed into an intelligent sensor module for processing and filtering the streamed data. The information from all modules is collected into the main module (μC I8052) for analyzing information and getting decisions of tabletop user's position. Then, the command is send to the PC via serial port to display the properly tabletop windows. SW Counter is a counter address generator used for scan all sensors. In the same time the address is decoded and the corresponding sensor module is activated.

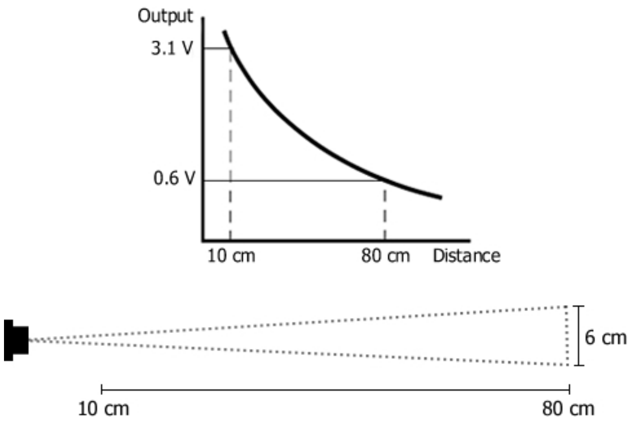

Figure 2. Short-range distance measuring sensors. Top: output response vs. distance to target. Bottom: field of detection with 6 cm diameter at maximum distance (80 cm).

The solution we chose for the hardware design on the Altium NanoBoard NB1 is presented in Figure 5. Each IR sensor is connected to an intelligent unit (implemented on the FPGA device as well). These intelligent units generate the proper signals to access and read data from the sensors. The data is filtered against noise and converted into useful information.

The sensors were installed on each side of the table at approximate 25 cm one from another with a total of 3 sensors per side: one at the middle and two close to corners.

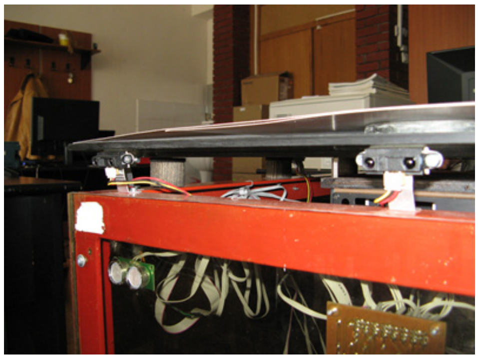

Figure 3. Tabletop prototype: installation of short-range sensors around the table (two sensors in view: middle and corner installed on the same edge).

Figure 4. Prototype installation of equipments: video projector and tabletop with installed sensors. The projector images are automatically computed by a desktop PC in accordance with the data received from the tabletop via the serial connection.

Although only few sensors are used, their placement makes that at least one and at most two of them be active and trigger a valid detection when a person stands close to the table. This means that the actual user's location can be computed at a finer resolution than given by the number of sensors: if neighbor sensors S_i and S_{i+1} of installed locations (x_i, y_i) and (x_{i+1}, y_{i+1}) output detection simultaneously then the user's actual location may be inferred in their middle as follows: $user\_location = ((x_i + x_{i+1})/2, (y_i + y_{i+1})/2)$. Besides increasing the resolution sensing, this approach particularly proves useful around the corners of the table: when users stand at a corner, both sensors from each sides trigger a response which averages into the correct 45 degrees shift in orientation.

Several application levels were designed. At the basis we find the sensor level whose main goal is to read the various distances between users and the tabletop. In the middle we find the intelligent sensor units which are connected to the IR sensors designed to retrieve data and to turn it into useful information. Level three is represented by communication and is implemented with the use of a switch which collects the information packages from the intelligent sensor units and sends them to the main module. The inputs and the output of the switch are buffered in order to avoid bottle-neck situations. Packages of information are taken from every sensor and inputs that are not enabled are ignored. At the last level there is the main processing unit which takes decisions with respect to the user's locations around the tabletop. The hardware architecture of our implementation is illustrated by Figure 5.

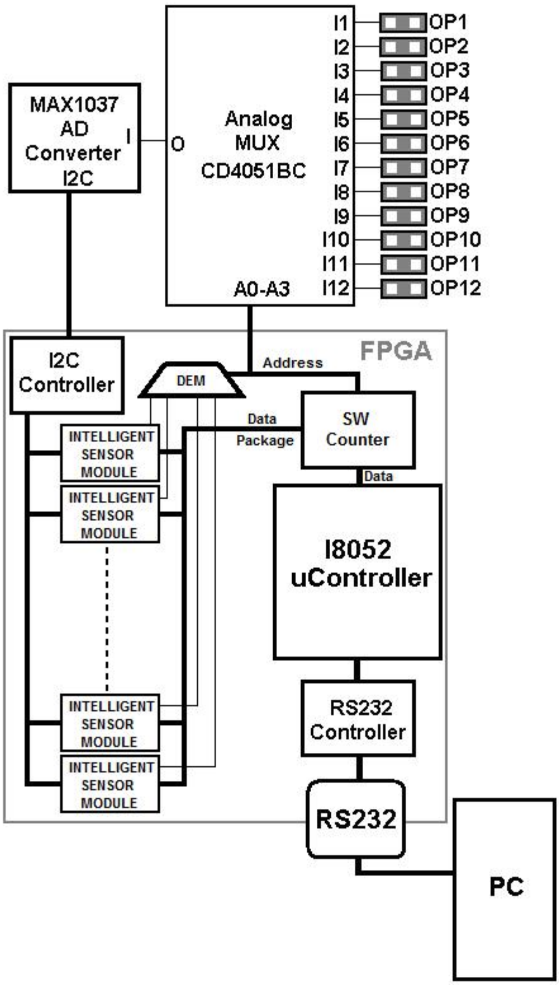

Figure 5. Hardware design (schematic diagram).

III. TRACKING

Tracking is achieved by detecting connected groups of one or two neighbor sensors which trigger valid detection. Considering the physical distance at which sensors are installed one from each other, a person is detected by at least one and at most two sensors simultaneously. This constraint that comes from physical details of our installation allows inferring the users' locations from the current and previous sensors readings by simply monitoring shifts in groups of neighboring sensors. We present below the pseudo code for tracking users while they move around the tabletop as well as for detecting users that left or just approached the table. In the pseudo code procedure, *prevPos* and *newPos* represent arrays storing the *(x, y)* positions of users at the last (previous) and current (new) sensors data reading; GET-CLOSEST*(prevPos[i], newPos)* returns the closest position as well as the actual distance from *prevPos[i]* to the *newPos* array with respect to the Euclidean metric.

```
PERFORM-USERS-TRACKING(prevPos)
sensors ← READ-SENSORS-DATA()
n ← length(sensors)
► Detect new users positions from
► current sensors data
newPos ← nil
for i = 1, n-1
   if sensors[i] == ON then
      if sensors[i+1] == ON then
         x' ← (x[i] + x[i+1])/2
         y' ← (y[i] + y[i+1])/2
         i ← i + 1
      else x' ← x[i], y' ← y[i]
   newPos ← newPos + (x',y')
► Cater for the last sensor, if needed
if sensors[n] == ON and
   sensors[n-1] == OFF then
   ► Deal with cases of sensor[1]
   ► and sensor[2] being ON/OFF due to
   ► circular installation
► Process existing users and identify
► the ones that left
for i=1, length[prevPos]
   (j,dist) ← GET-CLOSEST(
               prevPos[i], newPos)
   if dist < Threshold then
      PROCESS-EXISTING-USER(
         prevPos[i], newPos[j])
      newPos ← newPos - {j}
   else PROCESS-USER-LEFT(prevPos[i])
► Detect and process new users that
► just approached the table
for i=1, length[newPos]
   PROCESS-NEW-USER(newPos[i])
```

IV. INTERACTION POSSIBILITIES

Monitoring the sensors around the table allows for multiple possibilities for novel interactions which range from simple presence sensing to complex tracking mechanisms. Not only that a solution is provided for the correct orientation problem of windows, text and graphics but several new interactions may be developed. One or multiple users may be detected and tracked and, according to the current application scenario running, application dependant decisions may be taken. We further describe several implementations that were developed in our laboratory:

- Auto-greeting and auto-hibernate option. We enhanced the tabletop system with friendly automatic greetings that were displayed each time a user approached the installation. Equally important, the greetings were oriented towards the direction from which the user approached the table giving a nice feel of welcome as well as letting the user know that it was sensed and that the system is ready for interaction. Also, when the last user leaves, the system can turn into hibernate or idle mode, limiting thus consummation of additional resources.
- Auto-orientation of text and graphics. When the tabletop is running a complex application scenario, there may be the need of one or multiple users to walk around the tabletop in order to access the various options. However, when users change their positions

and table sides in traditional tabletop installations, the items they were working with (files, windows, applications) remain unchanged hence users are forced to drag them around and re-orient accordingly for up-front reading. We implemented an automatic tracking mechanism that updates the position as well as the orientation of user-owned items when users change locations around the tabletop. This option reduces the time necessary to fetch windows to new locations. Figure 7 illustrates the auto-tracking feature running.

- Games. Sensing the locations of users around the table can be effectively exploited for computer games. We developed two implementations: a chess board that automatically re-orients itself when users change locations (Figure 6 left) and a water simulation effect that concentrates the actions at the exact users' locations (Figure 6 right).

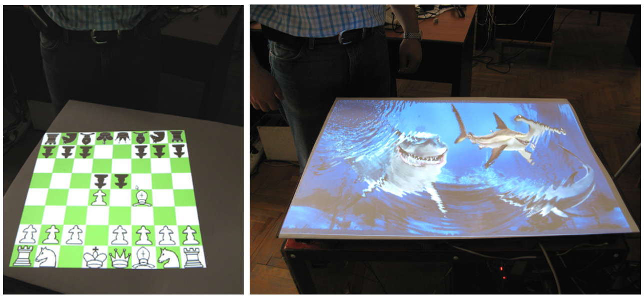

Figure 6. Left: Simply sensing the users may be explored for gaming effects or passive style interaction. Right: Relative positioning of users exploited for automatic reorientation of a chess board.

V. CONCLUSIONS

We presented in this paper a simple solution that allows detecting and tracking of multiple users around a tabletop installation. We achieved this by monitoring and correlating the output provided by a few short-range distance sensors installed on each side of the table. Despite the simple solution, the benefits are multiple: not only we provide a solution for the correct orientation problem but we also describe how new interactions may be achieved through simple user tracking.

As for future development, the main unit could retrieve information from the intelligent sensor modules and determine the path followed by the tabletop users using a software technique to calculate the speed of the user and his direction of movement. Using a fuzzy algorithm we could determine and estimate their future locations in accordance with their current direction and speed. Also it would be interesting testing a greater number of sensors for larger tabletop installations. This would lead to a finer resolution in detection and could be implemented using a structure of on-chip networks.

ACKNOWLEDGEMENTS

The research reported in this paper was supported by the national research grant Research of Excellence, INTEROB, CEEX 131/2006, grant TD – 161 / 01.10.2007 - Contributions to the Development of Computer Aided Design Techniques of Integrated Structures, Program PNII – RU and grant METAPRO, 58 CEEX–II 03/2006 - 6166/26.07.2006.

Figure 7. User interacting with a virtual keyboard and window which were enhanced with auto-tracking features.

REFERENCES

[1] Hancock, M.S., Vernier, F.D., Wigdor, D., Carpendale, S., Shen, C. Rotation and Translation Mechanisms for Tabletop Interaction, In Proceedings of TableTop2006: the 1st IEEE International Workshop on Horizontal Interactive Human-Computer Systems (2006), 79-86

[2] Hilliges, O., Baur, D. and Butz, A. Photohelix: Browsing, sorting and sharing digital photo collections. In Proceedings of TABLETOP '07: the 2nd IEEE Int. Workshop on Horizontal Interactive Human-Computer Systems (2007), 87–94.

[3] Kruger, R., Carpendale, S., Scott, S. D., and Tang, A. Fluid integration of rotation and translation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Portland, Oregon, USA, April 02 - 07, 2005). CHI '05. ACM, New York, NY, 601-610.

[4] Liu, J., Pinelle, D., Sallam, S., Subramanian, S., and Gutwin, C. TNT: improved rotation and translation on digital tables. In Proceedings of Graphics interface 2006 (Quebec, Canada, June 07 - 09, 2006). ACM International Conference Proceeding Series, vol. 137. Canadian Information Processing Society, Toronto, Ont., Canada, 25-32

[5] Morris, M., Morris, D., and Winograd, T. Individual audio channels with single display groupware: Effects on communication and task strategy. In Proceedings of CSCW '04: Computer Supported Cooperative Work (2004), ACM Press, 242–251

[6] Shen, C., Vernier, F. D., Forlines, C., and Ringel, M.. Diamondspin: an extensible toolkit for around-the-table interaction. In Proceedings of CHI '04, ACM Press,

[7] Tandler, P., Prante, T., M¨uller-Tomfelde, C., Streitz, N. and Steinmetz, R. Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. In Proceedings of UIST '01: the 14th ACM symposium on User interface software and technology (2001), ACM Press, 11–20

[8] Wilson, A. Playanywhere: A compact tabletop computer vision system. In Proceedings of UIST '05: the 18th ACM symposium on User interface software and technology (2005), ACM Press, 83–92.

[9] Wu, M. and Balakrishnan, R.. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In Proceedings of UIST '03: the 16th ACM symposium on User interface software and technology (2003), ACM Press, 193–202.

[10] Wu, M., Shen, C., Ryall, K., Forlines, C., and Balakrishnan, R. Gesture registration, relaxation, and reuse for multipoint direct-touch surfaces. In Proceedings of TABLETOP '06: the 1st IEEE Int. Workshop on Horizontal Interactive Human-Computer Systems (2006), 185–192.