2D HAND GESTURE RECOGNITION METHODS FOR INTERACTIVE BOARD GAME APPLICATIONS

A. C. Kalpakas, K. N. Stampoulis, N. A. Zikos and S. K. Zaharos
Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki
54 124 Thessaloniki, Greece

Keywords: Human-computer Interaction, Board Games, Hand gestures, Gestures Recognition, Image processing, Usability, Entertainment and games.

Abstract: The purpose of the current project is to demonstrate a complete interactive application capable of recognizing 2D hand gestures in order to interact with computer-based board games without the use of a special input devices, such as pointer, mouse or keyboard. A web camera is placed at the top of the platform and captures in real-time player's hand gestures and then recognizes the position of his fingertip on the board. The user is able to choose a piece, select a destination spot and move a piece just by simply placing and moving his/her index finger onto the board. Therefore an interactive, compact platform was developed, containing a light-wood construction, a printed chess board and a conventional webcam in order to test the effectiveness of the system. The suggested interactive system is fully compatible with the latest software technologies, uses a custom GUI, real-time 2D hand gesture recognizer and earcons.

1 INTRODUCTION

Interactive application is the application or the interface that allows people to communicate with machines (Human-Machine Interaction). It is generally acceptable that these kinds of applications must be designed in such way so that they are user friendly and support multiple accessibility options for the disabled. This is the main condition that these systems must fulfil in order to successfully incorporate in productive processes or processes of recreation so as to be widely accepted by the users (Tzovaras 2001).

At this point we have to point out that we do not refer to the widely known assumption "friendly to the user of systems" but to an application of theory and rules that implicates the interaction of human-computer with the use of laborious processes for analysis and planning of interactive systems. This difficulty arises by the fact that a lot of cognitive sciences are involved in the study of Human – Machine Interaction, such as information technology, psychology and cognitive psychology ,social psychology, ergonomy, linguistics, philosophy, anthropology, industrial planning etc. State-of-the-art systems include virtual hand gesture interfaces using special devices (i.e. gloves, stereoscopic glasses) (Technisches Museum Wien 2004, Zhang 2001) and speech recognition tools which emulate keystrokes (FORTH-ICS 2004).

Several techniques and models were combined to create a fully functional application, such as:

- Image recognition of the board and hand gesture recognition without using special equipment such as gloves, colored nails etc.
- Use of earcons in every critical action so as to inform the user
- A construction which is usable, fits in any place, is fully adjustable and compatible to different hardware setup

The primary goal of the project is to develop an interactive system which in conjunction with a software interface can boost the HCI board-based applications in a more natural interaction, enhance the usability of computer-based board games, suggest new interactive experiences and approaches aiming primary at people with handicap (i.e. low vision, hand-motor impaired) and secondary to public kiosks and exhibitions (i.e. museums, technology parks, schools).

C. Kalpakas A., N. Stampoulis K., A. Zikos N. and K. Zaharos S. (2008).
2D HAND GESTURE RECOGNITION METHODS FOR INTERACTIVE BOARD GAME APPLICATIONS.
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 325-331
DOI: 10.5220/0001939103250331

2 MATERIALS AND METHODS

2.1 Board Recognition

The first step in designing the software, is to develop an algorithm which will be able to recognize the board. The recognition will be visual. One of the most popular techniques for edge detection is based on the Laplacian operator which is defined with the second class partial derivative of the function f(x,y) depending to x and y as it is shown below (R. Gonzalez & R. Woods, 2002).

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (1)$$

The first class derivative of the above function gives local maximums and minimums at the edges of the image (a grayscale copy of the captured image is used) due to the big changes in brightness. Consequently, equation (1) is equal to zero at the edges of the processing image.

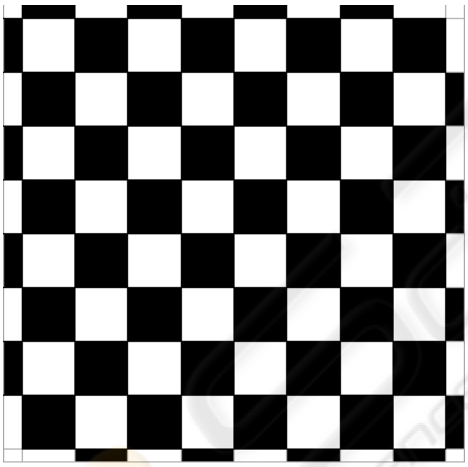

Figure 1: A chess board setup.

The above method is used in normal image processing. Taking advantage of the fact that the board has discrete edges and these edges are located at places where color changes rapidly from white to black and vice-versa, a much more flexible special-points recognition method is used, not dealing with second partial derivatives of the image edge functions; thus reducing the calculations and minimizing the execution time. The discrete approach of the Laplacian operator is shown in equation (2):

$$\nabla^2 f(x,y) \simeq f(x,y) - \frac{1}{4}\left[f(x,y+1) + f(x,y-1) + f(x+1,y) + f(x-1,y)\right] \quad (2)$$

A typical chess board was put to the test so as to verify the precision of the algorithm (Figure 1). Let us have the pixel A with coordinates (x,y), and f(x,y) as the color content of the specific pixel as shown in Figure 2. We also consider the neighbor pixels of pixel A. The aggregated P is calculated as follows:

$$P = \left[f(x,y+1) - f(x,y-1)\right] + \left[f(x+1,y) - f(x-1,y)\right] \quad (3)$$

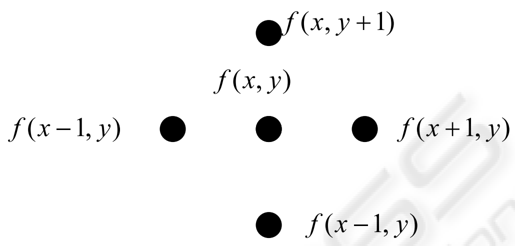

Figure 2: Pixels used for calculating P.

If the pixel is inside a random square of the chess board then, as it can be easily deduced, P equals to zero. This happens because all the neighbor pixels have the same value and the two parts of the aggregation are equal to zero. If the chosen pixel happens to be placed onto an edge then one of the two parts of the aggregation differs from zero. The result of applying the function on the image is illustrated in Figure 3.

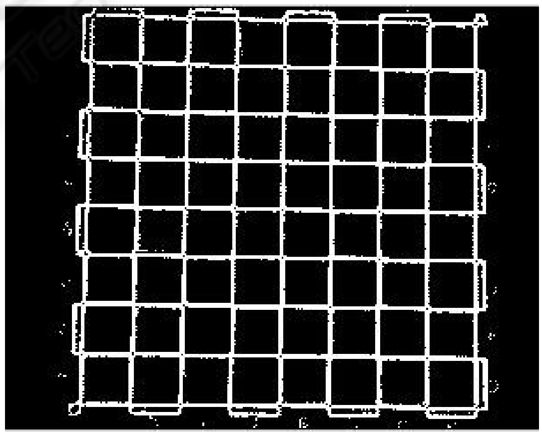

Figure 3: Recognizing the edges of the chess board.

Further more, it is easier to calculate the intersection of the edges (crosses) that will define the chess squares, as fewer calculations are needed. The focus is on the white lines in order to locate the desired pixels (crosses).

As shown in Figure 4, we are currently seeking the A1 type pixels (center of the cross). The distinctive difference between pixels A1 and A2 is the surrounding color (black or white).

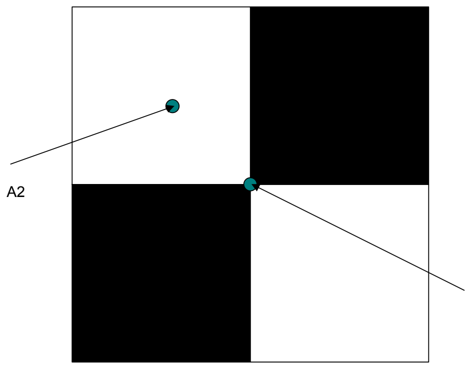

Figure 4: Board marks detection.

In order to achieve the aforementioned, a 5x5 mask is applied to the recognized edges (board marks). The parameters α and β of the mask may vary and depend on the type of the board (squares, circles, polygons etc.) :

$$M = \begin{matrix} 1 & \alpha & 0 & -\alpha & 1 \\ \alpha & \beta & 0 & -\beta & -\alpha \\ 0 & 0 & 0 & 0 & 0 \\ -\alpha & -\beta & 0 & \beta & \alpha \\ -1 & -\alpha & 0 & \alpha & 1 \end{matrix} \quad (4)$$

For example, for the chess board case the applied mask has α=β=0,1. As M(x,y) we define the matrix element (3,3) of matrix M. In order to investigate whether a random pixel f(x,y) belongs to the cross or not, we used the neighbor pixels. We define the matrix below.

$$f = \begin{matrix} f(x-2,y-2) & f(x-1,y-2) & f(x,y-2) & f(x+1,y-2) & f(x+2,y-2) \\ f(x-2,y-1) & f(x-1,y-1) & f(x,y-1) & f(x+1,y-1) & f(x+2,y-1) \\ f(x-2,y) & f(x-1,y) & f(x,y) & f(x+1,y+1) & f(x+2,y) \\ f(x-2,y+1) & f(x-1,y+1) & f(x,y+1) & f(x+1,y+1) & f(x+2,y+1) \\ f(x-2,y+2) & f(x-1,y+2) & f(x,y+2) & f(x+1,y+2) & f(x+2,y+2) \end{matrix} \quad (5)$$

From functions (4) and (5) we calculate the sum which is described in function (6).

$$p = \sum_{x=-2}^{x=2} \sum_{y=-2}^{y=2} M(x,y) * f(x,y) \quad (6)$$

Applying (6) to all image edges, all A1 pixels (Figure 4) will return either a very large or a very small integer. These results are shown in Figure 5.

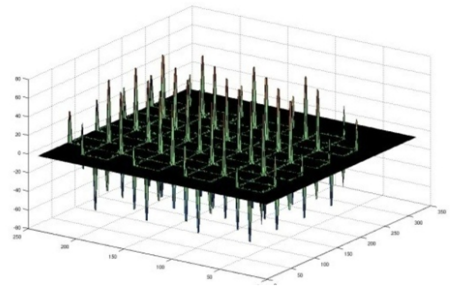

Figure 5: Identifying the board marks after mask results.

If the mask is applied to a pixel of A2 type (Figure 4) the result is close to zero due to the effect of mask M. This method returns more pixels than expected, since the mask outputs higher values for a group of neighboring pixels due to noise effects of the captured image (although 81 board markers were expected initially the actual result was more than 500 board markers). In order to solve this, a clustering process is performed using small clustering radius (8-15 pixels) and then the centers of the clusters are extracted (Figure 6). The clustering radius depends mainly on the resolution and noise effects of the imaging device (webcam).

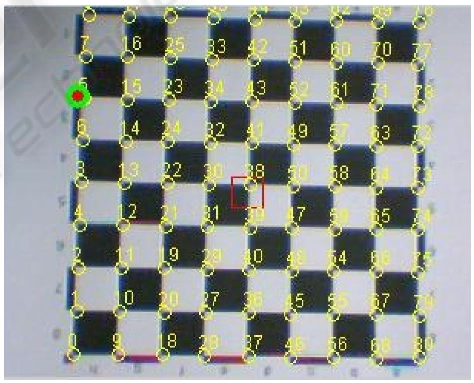

Figure 6: Board marks recognition and numbering.

The aforementioned approach poses the limitation that the angle between the horizontal and the chess board has to be less than 45 degrees (which is almost always the case due to the construction of the base and the physical user access). If for some reason the board has to be rotated at 45 degrees in relation to the camera axis, then mask M (4) has to be rotated accordingly. In either case, the algorithm returns the correct mapping of the board marks including the desired 81 pixels.

2.2 Gesture Recognition

After recognizing and mapping the board the next step is to recognize the fingertip that defines the moves. There are several approaches regarding this task: image processing, use of an artificial supplement such us a glove or a colored finger shield. The approach used in this project for gesture recognition is based on image processing techniques in order to reduce the complexity of the system, increase usability, avoid the use of extra equipment and reduce the overall cost (Imagawa 1998).

When the hand enters the board it changes the color context of the captured image. One important aspect of the algorithm is that the form of the hand should be as close as possible (but not necessarily exactly the same) to the one shown at Figure 7. This means closed grasp and extended index finger which is very close to the natural, human hand-driven selection pattern.

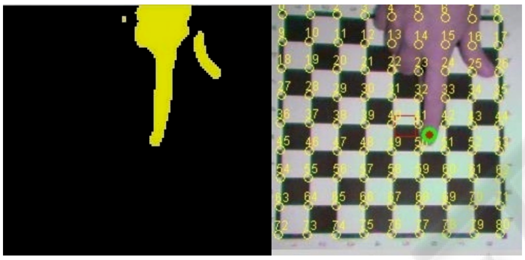

Figure 7: Successful gesture recognition.

The only requirement is that a definite pointing outline must be present. The hand gesture must not form an irregular shape, like multiple fingers pointing at the same time etc. The algorithm scans the captured image taken from the web-camera and distinguishes the pixels that have a special color information that range between a lower and an upper color bound (thresholds). These two thresholds are pre-stored through an actual hand color sampling (two independent frame grabs by the web-camera during the initialization of the application and they represent the variations of skin color). A low pass filter is applied in order to remove the noise. The points retrieved by the algorithm do not represent the whole hand but they are enough for the recognition of the tip of the finger.

In Figure 7 we notice that the algorithm successfully identifies a chess board along with the fingertip that enters the camera's view field. We can also see both the board marks (numbered circles) and the fingertip (colored circle) recognized by the algorithm.

2.3 Hot Spots Recognition

After we have recognized the position of the fingertip each one of the recognized board marks is given a specific numbering. The numbering is associated with a set of coordinates (x,y) for each board mark.

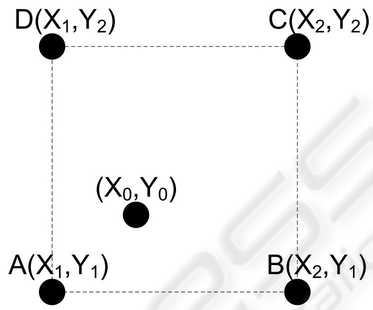

Figure 8: Example of a chess hot spot.

Generally, the hot spots identification is based on the shape of the board pattern (i.e. diamonds, oval, circle, polygons). Assuming the ABCD square in our chess board example (Figure 8) and a fingertip at point (X_0,Y_0), the following conditions must apply at the same time:

- $X_2>X_0>X_1$
- $Y_2>Y_0>Y_1$

The algorithm compares the coordinates (X_0,Y_0) to the coordinates of every square of the board. As the aforementioned conditions are satisfied, the active square (that is the one in which the fingertip rests for 2 seconds) is selected.

3 IMPLEMENTATION

A complete, versatile and fully functional board game system with embedded optical recognition capabilities was implemented aiming at the testing of the precision and overall efficiency of the methods analyzed above. A typical chess board was selected for the trials. The implementation plan includes both software and hardware development.

3.1 Software Implementation

The applications built to support the board recognition, hand gesture recognition and graphical user interfaces were based on object oriented programming using Java so as to achieve full compatibility with all hardware and software platforms.

The chess game engine was based on a customized version of “JChessBoard”, a java based chess game under the GNU General Public License (GPL) armoured with the necessary additions and improvements, so as to fully co-operate with the image processing algorithms and special application needs (such as audio and visual notification messages).

The development plan used the spiral model of software evolution. Thus, a prototype of the platform was developed, trials were carried out and usability assessments in a numerous aspects were completed. Based on users remarks the software as well as the rest of the construction were upgraded, so as to become more user friendly and more efficient short after several development cycles.

3.2 Hardware Implementation

In order to achieve the aforementioned goals, a light construction was made as the base which hosted the chess board. A satisfactory solution was to put a camera on the top of the chess board. The specified image resolution has been chosen so that the algorithm would perform the lowest possible number of calculations; thus boosting the execution speed. A low-end, off-the-shelf web camera was used for this purpose. The base of the application was made of plain chipboard and included illumination placeholders and a laptop opening (accepts almost every type of laptop up to 17 inch monitor). The drawing layout of the hardware platform is shown in Figure 9.

In more detail, a web camera (horizontally and vertically adjustable) was installed in order to capture the user’s fingertip as it moves on the chess board. The image is captured in a resolution of 320x240 pixels capable of accurately recognizing details but not high enough so as to avoid excessive processing load. The typical chess board was printed in black and white on a A3 paper (29,7x42 cm) firmly placed on top of the construction base. The chess board is slightly modified with the addition of peripheral edges. This ensures the efficient recognition of the peripheral squares. The typical chess board setup used is shown in Figure 1.

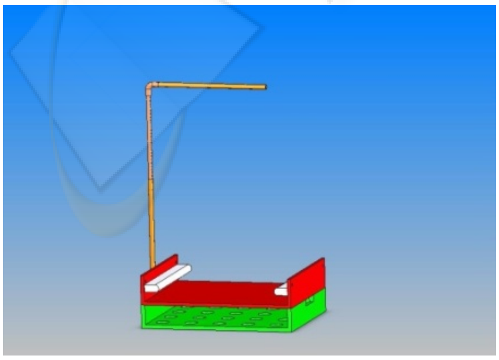

Figure 9: Drawing of the base (final prototype).

For better performance, the boards can be painted on the chipboard so that displacement problems are vanished. At the lower part of the construction one can see the opening (pocket) were the laptop is placed. On the upper part of the construction, an adjustable (lengthwise and in height) telescopic arm has been placed, in order to host the web camera and adjust it to the proper position. Another issue was the illumination of the chess board. Two energy-efficient fluorescent lights were placed at opposite sides (as shown in Figure 10) in order to have proper lighting in less illuminated environments as well as to eliminate shadowing effects caused by the moving hand.

Figure 10: A view of the actual implemented system.

3.3 Commands and Actions

After having finished with the software and hardware implementation, an interaction scheme is designed considering the specific requirements and specifications posed by the type of the board game including a complete command list, interrupts and external actions. For example, in our implemented system one is ready to play chess using the interactive chess board (Figure 10) using only his hand and no other input device such as mouse or keyboard. The hand gesture commands that apply during this game are as follows:

1. The player positions his/ her index fingertip to the desired square where the chosen-to-move chess piece resides.

2. Leaving the fingertip at this position for 2 seconds is the sign for the system to recognize the chosen piece (followed by an earcon / text message confirmation).
3. Afterwards the player's fingertip has to move to the desired destination square.
4. Resting the fingertip at this destination square for another 2 seconds is enough for the system to recognize the destination square. If the movement is allowed, an earcon confirmation is reproduced and the game continues. Otherwise, the chosen chess piece returns to the originating square and the system returns to step 1 for the same player, again followed by a failure-like earcon.

4 USABILITY ISSUES

Assessing usability is a complex and multidimensional procedure due to the fact that different type of assessments have to carried out at several implementation stages. Different types of assessment include: usage of mathematical models without human participation, experimental methods using human participation as well as inquiring methods in conjunction with interviews and questionnaires.

The suggested HCI system had to prove its efficiency so an actual usability evaluation was performed on the implemented chess application. Experimental methods with human participation were the chosen including: observing the players and recording their moves, execution times, number of errors, facial expressions etc. and giving merit ratings, questionnaires, interviews. There were several usability indicators examined: required time for a successful interaction with the system, minimum time for the execution of a command or action, successful actions to failures ratio etc. The goal is to investigate the user degree of satisfaction, the ease of use, common bugs etc. A series of custom-made questionnaires have been developed and they were given to 16 volunteers (two of them had a hand/grip handicap) of various ages (between 20 to 40 years old) and various educational levels (high school graduates, university graduates, Phd holders) with low, medium and high computer and/or gaming skills (Lewis, 1995).

Figure 11: Users interact with the system and developers record their responses to certain tasks during the evaluation procedure.

After a cycle of evaluation and additions – changes to the original plan, the first prototype was redesigned and a final prototype was developed. The evaluation of the prototype by the representative sample of targeted users (Figure 11) lasted about two weeks, during which quantitative data (timings, error measuring etc) as well as qualitative data (general mood and satisfaction, facial expressions, body language, emotions etc) had been recorded and analyzed. An illustrative part of the analysis of the qualitative data is shown in Figure 12. The evaluators rewarded the system with a total average of 3,91 out of 5 in terms of user-friendly environment, general feeling, ease of use, well-organized information on screen, recovery time upon failure, quality of multimedia, help on demand and variety of features.

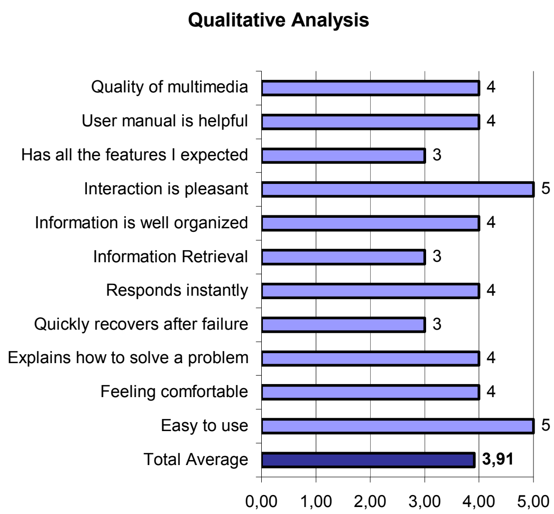

Figure 12: Overall score of the evaluation.

5 CONCLUSIONS

Human – Machine Interaction is a very sophisticated as well as demanding field. According to our opinion, it is the field that will play a dominant role in the near future. Innovative technical ideas are not sufficient if they are not accompanied with user friendliness and user satisfaction in general.

The implemented recognition technique was proven quite successful with only minor bugs and restrictions (i.e. the angle of the hand). This was proven by both experts and non-experts quantitative and qualitative usability evaluation reports. It was to our surprise the wide acceptance of the implemented project (3.97 / 5) not only by people who had never played chess before or people who were not open to computer games in general but also by people with special handicap in arms and hands. Mean time for a successful interaction was less than 1 minute. Successful interactions to failures ratio was acceptably high (9:1). The use of conventional materials and off-the-shelf hardware kept the total cost very low; thus affordable to almost anyone.

The suggested recognition approach, which is the most user friendly one, is not to use any supplements but to take advantage of the different color context of the hand compared to the board. The use of alternative recognition methods such as gloves and colored nails have certain disadvantages. Despite the fact that the recognition algorithm can be less complex, the user doesn't have to use any extra equipment (such as gloves or other special equipment) in order to play the game. Also, the production cost of the system is minimized. The users that evaluated both suggested solutions in early development stage strongly recommended the second approach.

Future development of the project, which actually turns to be a very challenging goal, is to use actual pieces and play board games in real-time with the computer as our live opponent. This requires advanced image processing algorithms for decoding the position of the pieces as well miniature robotic arm (with 6 degrees of freedom) utilizing kinematics models for moving the opponents pieces.

REFERENCES

Avouris N., 2000, *Introduction to Human-Computer Interaction.* Athens: Diavlos.

FORTH-ICS 2004, *UA Chess*. Retrieved March 3, 2008, from http://www.ics.forth.gr/uachess/

Gonzalez R. & Woods R., 2002, *Digital Image Processing*. 3rd edition. USA: Prentice Hall.

Imagawa, K., Lu, S., Igi, S.: Color-Based Hand Tracking System for Sign Language Recognition, IEEE Int. Conf. on Automatic Face and Gesture Recognition, Japan. (1998)

JChessBoard, 2002, Retrieved June 21, 2007, from http://jchessboard.sourceforge.net/

Lewis, J. R., 1995, IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. International Journal of Human-Computer Interaction, p.7:1, 57-78. (Online). Available at: http://www.hcirn.com/ref/refl/lewj95.php (accessed 12 April 2008).

Papamarkos N., 2005, *Digital Image Processing & Analysis*. Athens: V. Giourdas.

Tzovaras, Dimitrios, 2001, University Notes on Human-Machine Interaction.(ebook).Thessaloniki. Available at: http://alexander.ee.auth.gr:8083/eMASTER/cms.downloadCategory.data.do?method=jsplist&PRMID=4 (accessed 12 April 2008)

Technisches Museum Wien, 2004. *"The Virtual Turkish Chess Player"*, Retrieved January 21, 2006, from http://www.ims.tuwien.ac.at/~flo/ vs/chessplayer.html

Zhang, Z., et al. Visual Panel: Virtual Mouse, Keyboard, and 3D Controller with an Ordinary Piece of Paper. in Proc. ACM Perceptual/Perceptive User Interfaces Workshop (PUI'01),. 2001. Florida, USA.