VisualPen: A Physical Interface for natural human-computer interaction

Francesco La Rosa
Department of Mathematics,
University of Messina
C.da Papardo, Salita Sperone
98166, Messina - Italy.
flarosa@ingegneria.unime.it

Carlo Costanzo
Faculty of Engineering,
University of Messina
C.da Papardo, Salita Sperone
98166, Messina - Italy.
ccostanzo@ingegneria.unime.it

Giancarlo Iannizzotto
Department of Mathematics,
University of Messina
C.da Papardo, Salita Sperone
98166, Messina - Italy.
ianni@ingegneria.unime.it

ABSTRACT

In this paper we describe a physical user interface system for easy and natural user-computer interaction. VisualPen is a vision-based system for real-time detection and tracking of a stylus that completely replaces mouse and keyboard, thus providing a valid input device for mobile computers, and its low computational complexity renders it suitable also for PDAs. The system can be operated from a wide range of distances (either from a desk or from a wall-mounted projection panel) and is able to work with all lightning conditions. The architecture of the system is here described, and experimental results in several tests are presented and commented.

Keywords

Perceptual User Interfaces, Vision-based User Interfaces, stylus, IR

1. INTRODUCTION

Human-computer interaction has not changed its basic paradigm for nearly two decades: mouse, keyboard and icons are still the foundations of almost any computer interface. However in the last years an increasing number of researchers in various areas of computer science is developing new technologies to add perceptual capabilities such speech and vision to human-computer interfaces: such *perceptual user interfaces* are likely to be the next major paradigm in human-computer interaction. In particular, computer vision and other direct sensing technologies have progressed to the point where it is possible to detect several aspects of a user's activity, reliably and in real time, thus producing an increasing interest for vision based human-computer interaction: a technology which exploits a camera to sense the user's intentional actions and responds in real time.

This paper was presented at "Physical Interaction (PI03) - Workshop on Real World User Interfaces", a workshop at the Mobile HCI Conference 2003 in Udine (Italy). September 8, 2003. Further information and online proceedings are available at http://www.medien.informatik.uni-muenchen.de/en/events/pi03/ .

Several classes of such reactive systems can be found in literature [10, 4, 9, 14, 13], including among others, those exploiting facial pointing and other head gestures [8], facial expressions, finger pointing and selection [15, 12, 5], full-body gestures [2, 1] and even more complex interactions such as overall user behaviour (mainly used for surveillance and elder/impaired people care). Most of these approaches appear promising and quite simple to implement with off-the-shelf devices such as webcams; yet, unfortunately, most algorithms heavily depend on lightning constancy, so, very often, when illumination cannot be controlled they became unreliable. Moreover, when CPU power consumption is a major issue (mobile applications running on wearable computers, PDAs, and similar devices) the high computational complexity of most adopted image processing algorithms makes them mostly inapplicable.

Another issue, largely neglected by several researchers, is related to the real naturalness of the tracked gestures and to ergonomics. As regards hand gesture - based visual interaction, for example, most of researchers initially concentrated on bare [11, 7, 6, 3] (or even gloved) hand gesture recognition, regardless of what kind of gestures was more natural for what applications. After the initial enthusiasm, which led to extremely interesting, accurate and complex solutions, some researchers realised that in several cases the main issue is *how to make easier and painless the interaction for the user*, instead of *how to astonish him with special effects*. Our Group did not escape this destiny : after having developed a system for visual human-computer interaction based on finger pointing and bare hand gesture recognition [5], we realized that most "gesture units" associated to human-computer interaction are better performed by the user when holding in hand a physical device, such as a stylus or a pen. This is probably related to the way we learn to write and draw, some kind of "legacy" very hard to change: pen, pencils, pieces of chalk, remain undoubtely the more "human" approach to writing and drawing. Experience with PDAs confirms that millions of everyday-users do not require any input device but a stylus and a LCD touch screen.

Following this intuition, we developed the device described in this paper: an optical stylus - based system that completely replaces mouse and keyboard. The system, that uses a camera to track in real time an IR emitting stylus, is able to work with all lightning conditions and can be used on whatever (if any) surface (e.g. walls, writing desks, pro-

jection screens, a notepad...). The necessary feedback to the user can be provided by a video projector, a traditional CRT or LCD screen, or by more innovative devices such as head-mounted displays. The computational complexity of the exploited algorithms is so low that the system can be easily ported to a PDA without heavily affecting its power consumption.

The following two sections give a detailed description of our approach and present some experimental results. The final section draws our conclusions and outlines further research developments.

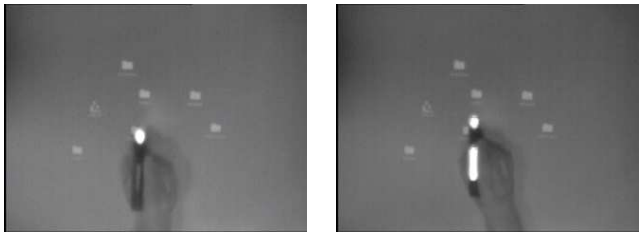
2. VISUALPEN

VisualPen is a vision-based system for real-time detection and tracking of a pen that allows to a user to interact with a kind of "virtual screen" projected on a flat surface without mouse or other pointing or keying devices (e.g. mouse, keyboard, etc.). The user can use the pen as a complete substitute for the mouse: it's possible to control the position of the cursor by moving the pen over the screen, to generate the events click and double-click and therefore to select and drag an icon, to open any folder, to draw and write. VisualPen is a system as simple to use as the mouse, but at the same time it is much more natural than the devices normally used to write or to draw. We all have experienced the difficulty to draw using the mouse and the trouble to use keyboard and mouse in order to write a text. Visual Pen puts togheter the naturalness of use of an everyday-life object, a pen, with the versatility of a personal computer and the the possibility of a distance interaction and collaborative work (it is possible to have more than a VirtualPen working at the same time). The system comprises a multimedia video projector, a gray-level video camera to acquire the scene and a pen with two IR emitting led. The scheme in Fig. 3 describes the main operational phases of VisualPen: the first phase consists of acquiring the image to be processed; the decision to implement a low-cost system and thus to use entry-level hardware means limiting the acquisition resolution to 320x240 so as to reduce the computational cost while meeting the real-time constraints. The decision to acquire in gray levels is due to the poor lighting of the environment in which VisualPen is likely to be used. A direct consequence of the poor lighting is the impossibility of distinguishing between colors. In addition, the projected images alter the scene being filmed even further. These considerations led us to exclude the use of color for the segmentation of the images acquired in these conditions.

Poor lighting and the need to make the system robust to abrupt background changes due to variations in the image being projected onto the screen make it necessary to have an additional lighting system for the projection surface or, as in our work, to add to the pointing device (the stylus) a visually detectable beacon to facilitate detection and tracking.

Two IR LEDs are mounted on the device: the first (Fig. 1a), of circular shape, is used to track the pen and is switched on for the whole period of activity; the second (of rectangular shape) is switched on by the user to generate a click (Fig. 1b).

We adopted IR leds because the IR radiation leaves the scene unaltered to the human eye therefore does not affect the projection itself. By filtering out the visible component of light while capturing the image, we then obtain an image from which it is very simple to detect and to track the led and the pen. We therefore placed a low-cost infrared filter in front of the videocamera lenses: the effect is to eliminate most of the visible light component, which is mainly represented by the projected images. Segmentation of the scene is performed by means of thresholding(Fig. 2), search of connected components and edge extraction.



(a) Pointing (b) Click event

Figure 1: acquired IR Images

(a) Pointing (b) Click event

Figure 2: Thresholded images

The edges resulting from the segmentation are then processed by the Classification algorithm which returns the position of the pen and the type of event.

The Classification phase discriminates the click event using the number of active leds in the same frame: pointing is characterized by only one LED active, click by two LEDs active. To simplify the detection of the second led we use informations on the shape of leds because the two leds have different shape (circular shape and rectangular shape). Shape analisys of the retrivied contours in the image of segmentation is based on the equations 1, 2, the leds are discriminated by the different ratio (*factor*) between the principal axis of the leds. The measure of principal axis of the leds is calculated using some statistical moments (M_{ij}) (zero, first and second order).

$$\begin{aligned} M_{pq} &= \left(\sum_x \sum_y x^p y^q I(x,y)\right) \\ x_c &= \left(\frac{M_{10}}{M_{00}}\right) \\ y_c &= \left(\frac{M_{01}}{M_{00}}\right) \\ invm_{00} &= \left(\frac{1}{M_{00}}\right) \\ a &= M_{20} * invm_{00} \\ b &= M_{11} * invm_{00} \\ c &= M_{02} * invm_{00} \end{aligned} \quad (1)$$

$$
\begin{aligned}
square = &\quad \sqrt{4*b^2+(a-c)^2} \\
\theta = &\quad \arctan\tfrac{2*b}{a-c+square} \\
cs = &\quad \cos\theta \\
sn = &\quad \sin\theta \\
rotatea = &\quad cs^2*M_{20}+2*cs*sn*M_{11}+sn^2*M_{02} \\
rotatec = &\quad sn^2*M_{20}-2*cs*sn*M_{11}+cs^2*M_{02} \\
length = &\quad 4*\sqrt{rotatea*invm_{00}} \\
width = &\quad 4*\sqrt{rotatec*invm_{00}} \\
factor = &\quad \tfrac{length}{width}
\end{aligned} \tag{2}
$$

Before passing the recognized command to the operating system, the coordinates supplied by VisualPen need to be corrected because the multimedia video projector and the gray-level videocamera are not orthogonal to the projected surface and generate a trapezoidal distortion (see Figs. 6, 7). To do so, we must determine the correction parameters. A test image is projected during the initialization phase and the user is asked to touch four highlighted points with his pen in the sequence indicated.

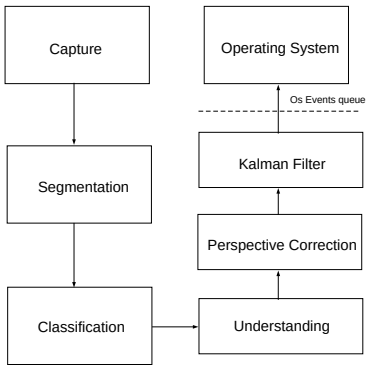


Figure 3: The main operational phases of VisualPen

In this way, once the positions of the pen in the image acquired by the camera and the four points in the test image are known, it is possible to determine the parameters of geometric transformation between the acquired image and the reference system.

The proportionality factors and the offset are given in equation 3, in which d_{12}, d_{23}, d_{34} and d_{14} represent the reciprocal distances between the four points in the test image, and D_{12}, D_{23}, D_{34} and D_{14} indicate the reciprocal distances between the four points in the acquired image.

$$
\begin{aligned}
w = &\quad (D_{12}-(D_{12}-D_{34})*\tfrac{Y-Y_1}{Y_4-Y_1}) \\
h = &\quad (\tfrac{D_{14}+D_{23}}{2}) \\
factorx = &\quad (\tfrac{d_{12}}{w}) \\
factory = &\quad (\tfrac{d_{14}}{h}) \\
offsetx = &\quad X_1*factor-x_1 \\
offsety = &\quad Y_1*factor-y_1
\end{aligned} \tag{3}
$$

Once *offsetx, offsety, factorx, factory* and the co-ordinates of a point (X,Y) in the acquired image are known, it is possible to determine the co-ordinates of the corresponding point on the screen (x,y) by means of the equation 4.

$$
\begin{aligned}
x = &\quad X*factorx-offsetx \\
y = &\quad Y*factory-offsety
\end{aligned} \tag{4}
$$

When a led has been detected, the information (after correction) about its position and velocity (interframe displacement) is passed to the tracker module. As shown in Fig. 5, this module comprises an estimator, a controller and a measure module connected in the conventional closed-loop fashion commonly adopted for visual object tracking. At each frame the Kalman Tracker, on the basis of the previous observations (measures), produces an estimate of the new status of the pen, the accuracy of which tends to improve at each iteration (in the ideal case, the error tends to zero) thanks to the information provided by each new measurement. Let us now define the status vector representing the status of the system to be tracked.

The status vector have a total of 4 elements, as expressed by the equation 5, it comprises 4 variables considered in the time instants i.

$$
X_i = \begin{pmatrix} x_i & y_i & \delta x_i & \delta y_i \end{pmatrix} \tag{5}
$$

In eqn.(5), (x_i, y_i) and $(\delta x_i, \delta y_i)$ are respectively the position and the velocity of the led (in screen co-ordinates).

The Prediction-Assimilation algorithm is outlined in Figure 4: Z is the vector of our measures, so it has the same composition as X_i in equation (5). The matrix G_i represents the linear relation between the measure and the status: in our case, $G_i = I$ (I is the Identity matrix). w_i and v_i represent the noise associated with the status and the observation process. We assume that they both have a Gaussian probability distribution, zero mean and variances, respectively: B_i and R_i. The variance of X_i is P_i. The model adopted for prediction is a linear and its parameters were determined experimentally.

Prediction-Assimilation paradigm

$\widetilde{X}_i = A_{i-1}X_{i-1} + w_{i-1}$	*Prediction*
$Z_{i-1} = G_{i-1}X_{i-1} + v_{i-1}$	*Observation model*
w_i and v_i have zero mean and variances: B_i and R_i	
X_i has a variance of: P_i	
$\widetilde{P}_i = AP_{i-1}A^T + BB^T$	*Riccati eqn.*
$K_i = \widetilde{P}_{i-1}G_i^T\left(G_i\widetilde{P}_iG^T + R_i\right)^{-1}$	*Kalman Gain*
$\widehat{X}_i = \widetilde{X}_i + K_i\left(Z_i - G_i\widetilde{X}_i\right)$	*Assimilation*
$P_i = (I - K_iG_i)\,\widetilde{P}_i$	

Figure 4: Prediction - Assimilation algorithm

The performance of the Kalman tracker described above is closely related to the hypothesis that both the noise vectors and the status vector have a Gaussian distribution. At this stage we will not address this issue, since the performance of the Kalman tracker is reasonable for our purposes; several different solutions do, however, exist for this problem.

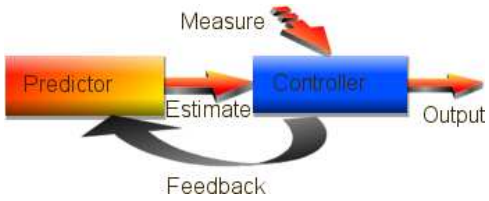


Figure 5: The prediction-measure-assimilation scheme.

3. EXPERIMENTAL RESULTS

The system was tested during and after development by several users for a considerable number of hours in numerous environments with different external lighting conditions. As VisualPen replaces the input devices in almost all their functions it was used to interact with the graphic interface of the operating system and most commonly used applications. For example, the system was used to open, select and drag icons, windows and other graphic objects on the desktop. The use of VisualPen is of particular interest in applications of free-hand interaction such as drawing in graphic processing applications (see Fig.6) and hand-writing in sign recognition software (e.g. PenReader) (see Fig. 7).

Figure 6: Drawing with Paint

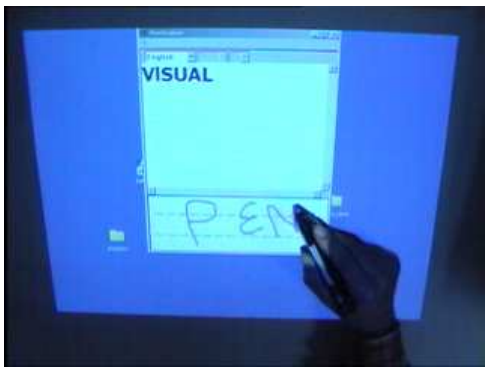


Figure 7: Use of a hand-writing

Tests were carried out on projections onto a desk, a wall and a projection screen to show the possibility of using VisualPen in different environments and situations. To evaluate the performance of the system in terms of accuracy and repeatability a considerable number of tests were carried out. To produce a quantitative evaluation we compared the output of VisualPen with a ground-truth reference. So we predisposed three classes of tests that can meaningfully characterize our system. At first we considered a segment of horizontal straight line that must be followed tracing it for its entire length with the pen. The measures have been realized asking 10 users to test 5 times the system following free hand the prefixed trajectories, that have been shown on the projection surface, and the system has stored during the tests the coordinates of the output points.

An estimation of the whole error (due both to the system and to the accuracy of the user) can be evaluated from the comparison between the acquired coordinates and those of the reference curve; carrying out then a statistical analysis on a considerable number of measures we obtained informations about the precision of the system calculating the standard deviations of the errors for each point along the reference trajectory; such errors are expressed in pixel or fractions of pixel.

For the second class of test we considered an arc of ellipse to be followed - again - free hand. Both these classes show, particularly in the second half of the abscissas, a defect of accuracy due to the uncertainty of the user. Nevertheless, the extreme naturalness of VisualPen allows to maintain the error under 3 pixels. We considered a third class of tests to try to render negligible eventual systematic errors unconsciously introduced from the users in order to estimate the intrinsic error of the system. This time the users must follow the same segment of horizontal straight line of the first class of tests, but with the pen constrained to slide on a fixed guide.

The analysis of 50 measures carried out for each class of tests shows that the standard deviation of the error is maintained always inferior to 3 pixels and that the total medium value on the three class of measures is approximately 1.5 pixels.

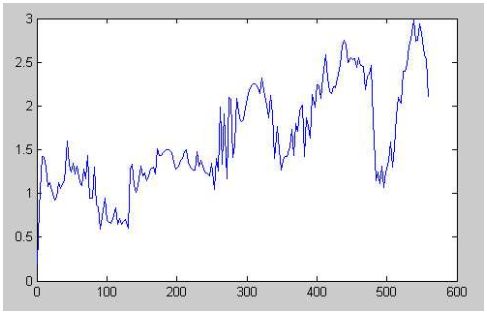


Figure 8: Standard deviation of the error made tracing free hand a segment of a straight line

Fig.8 shows results, along the 561 points of abscissa, of the standard deviation of the error made tracing free hand a segment of a straight line. The increment of the error in the second half of the segment is probably generated from a decay of the attention of users. Fig. 9 instead shows results, along the 466 points of the arc of ellipse, showing also in this time an increment of the error in the second half of the curve. Fig. 10 finally shows the obtained results along the previous segment constrained this time to a sliding guide. It is interesting to note that removing the error due to the users,

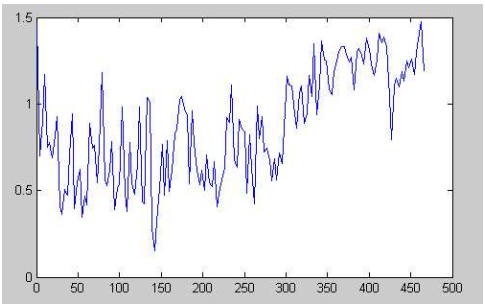


Figure 9: Standard deviation of the error made tracing free hand an arc of ellipse

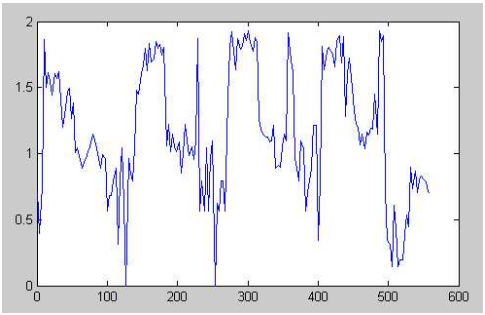


Figure 10: Standard deviation of the error made tracing constrained to a sliding guide a segment of a straight line

the system shows an intrinsic error that oscillates around 1 pixel. Such error is due to the different resolutions of the acquired image and of the projected image. To solve this problem we would need to use an algorithm that allows to obtain a sub-pixel accuracy. This kind of algoithm is usually very computationally intensive thus revealing unsuitable for our purpouses. We therefore decided to keep this error.

4. CONCLUSIONS

In this paper we presented a system for human-computer interaction that provides a more easy and suitable input device, we explained the insensitivity to lighting and the low computational complexity that permits a large number of application scenarios in several environments and with different types of devices like PDAs or other mobile device. We supplied measures of the accuracy in three classes of tests obtaining always good results that suggest the use of this system also in applications traditionally linked to mouse or keyboard. We are currently investigate the application of VisualPen to collaborative work sessions and to interact with Virtual and Augmented Reality Environments.

5. REFERENCES

[1] A. Bobick and J. Davis. The representation and recognition of action using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, March 2001.

[2] G. Bradski and J. Davis. Motion segmentation and pose recognition with motion history gradients. In *Proc. 5th IEEE Workshop on Applications of Computer Vision (WACV 2000)*, Palm Springs, CA, USA, December 2000.

[3] L. Bretzner, I. Laptev, T. Lindeberg, S. Lenman, and Y. Sundblad. A prototype system for computer vision based human computer interaction. Technical report, ISRN KTH/NA/P-01/09-SE, Stockholm, Sweden, 2001.

[4] R. Cipolla and A. Pentland. *Computer Vision for Human-Machine Interaction*. Cambridge University Press, 1998.

[5] C. Costanzo, G. Iannizzotto, and F. LaRosa. Virtualboard: Real-time visual gesture recognition for natural human-computer interaction. In *Proc. of the IEEE IPDPS'03*, Nice, France, 2003.

[6] C. Hardenberg and F. Bérard. Bare-hand human computer interaction. In *Proc. of PUI01*, Orlando, Florida, USA, November 2001.

[7] G. Iannizzotto, M. Villari, and L. Vita. Hand tracking for human-computer interaction with graylevel visualglove: Turning back to the simple way. In *Proc. PUI01*, Orlando, Florida, USA, November 2001.

[8] R. Kjeldsen. Head gestures for computer control. In *Proc. of the RATFG-RTS*, Vancouver, BC, Canada, 2001.

[9] M. Kohler and S. Schroter. A survey of video-based gesture recognition - stereo and mono systems. Technical report, Fachbereich Informatik, Dortmund University, 44221 Dortmund, Germany, 1998.

[10] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. *IEEE Trans. on PAMI*, 19(7):677–695, July 1997.

[11] M. C. R. Grzeszczuk, G. Bradski and J. Bouguet. Stereo based gesture recognition invariant to 3d pose and lighting. In *Proc. of the IEEE CVPR2000*, 2000.

[12] C. P. Rick Kjeldsen, Anthony Levas. Dynamically reconfigurable vision-based user interfaces. In *Proc. of the 3rd International Conference on Vision Systems (ICVS'03)*, Graz, Austria, April 2003.

[13] J. H. Rick Kjeldsen. Design issues for vision-based computer interaction systems. In *Proc. PUI01*, Orlando, Florida, USA, November 2001.

[14] Y. Wu and T. Huang. Vision-based gesture recognition: A review. In *Int. Gesture Workshop (GW99)*, Gif-sur-Yvette, France, 1999.

[15] Z. Zhang, Y. Wu, Y. Shan, and S. Shafer. Visual panel: Virtual mouse, keyboard and 3d controller with an ordinary piece of paper. In *Proc. PUI01*, Orlando, Florida, USA, November 2001.