A Vision-Driven Gesture-Based Pointer For Computer-Augmented Environments

Gabriele Baggiani [(*)], *Carlo Colombo* [(*)]
and Alberto Del Bimbo [(*)]

[(*)] Dipartimento di Sistemi e Informatica, Via Santa Marta 3, I-50139 Firenze, ITALY
E-mail: {baggiani,colombo,delbimbo}@dsi.unifi.it

ABSTRACT
This paper describes a system for advanced man-machine interaction based on computer vision technology allowing users to manipulate information displayed on large wall panels by using their own hands as pointing devices. Graphic interface operation is the same as with standard computer mice; the main points of innovation concern naturality of interaction, low intrusiveness and a priori training, and low equipment cost. An experimental version of the system is currently being employed to provide museum visitors with advanced interaction services.

KEYWORDS: advanced human-computer interaction, computer vision, graphic computer interfaces.

INTRODUCTION

Computer interpretation of human action is the basis for advanced man-machine interface design. Mice and keyboards are gradually being complemented with new interaction devices that can monitor people as they act in everyday life [1, 2]. In particular, the possibility to capture the position and movements of user body parts in real-time and in a non intrusive way is of key importance for developing new applications in fields such as virtual and augmented reality, computer-supported cooperative work, telepresence and robotics. Current advanced interaction devices (e.g. 3D pointers or data helmets) do not appear to be immune to serious problems such as high intrusiveness, high cost, and low performance/cost ratio. Being intrinsically non intrusive and quite inexpensive, real-time computer vision is a technology that perfectly meets the basic requirements of next generation human-computer interaction applications [3]. Several vision-based interaction approaches have been presented so far [4]. In [5, 6, 7] a number of examples are presented of semantic interpretation of human hand gestures and facial expressions. Another side of applications is that exploting continuous user action to develop advanced pointing devices based on hand, head or eye motion [8, 9].

This paper illustrates the main features of a new vision-based interaction system capturing user actions in a three-dimensional environment. The system allows users to explore and select information displayed on large wall panels using their own hands as pointing devices. Graphic interface operation is the same as supported by standard interaction tools such as the computer mouse; the main points of innovation concern instead interaction style, and in particular naturality and usability issues. First of all, unlike both traditional and more advanced yet intrusive human-computer communication contexts, users do not have to wear or otherwise use any extra hardware device to carry out the interaction: they just have to point to an area of interest in the wall panel with their own hands. Secondly, while a traditional mouse must lay and be operated on a flat 2D surface such as a table, the system allows users to move freely in a 3D, real-life environment such as a room. Last, but not least, pointing in 3D is an everyday life operation, which therefore does not require a priori skills or training.

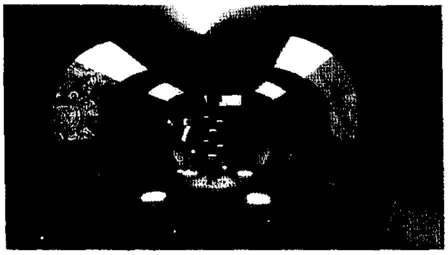

Figure 1: A computer-augmented museum.

A prototype of the system is being used to support interaction within an "augmented museum" room (Fig. 1). This is a museum environment provided with computer facilities, in which users can ask the system to display on the museum walls additional information about the paintings they are currently looking at by activating interface links associated to specific wall locations.

SYSTEM DESIGN

Overview

The main system components are shown in Fig. 2. The system gets its input from a pair of color cameras (placed so as to have the user in view); a personal computer performs image analysis and updates interaction parameters. Such parameters, which reflect current user status, are then transformed into graphic interface commands and output to the screen through a beamer.

Interface operation is based on both spatial and temporal characteristics of user action. On the spatial side, the screen location *P* currently pointed to by the user is continuously evaluated as the intersection of the pointing line *L* with the screen plane. To this end, user's hands and face are located and tracked from both images, and then used as the input of an affine stereo triangulation algorithm. On the temporal side, the system monitors pointing persistency: as point *P* remains inside a limited portion of the screen for an appropriate amount of time (i.e. about two seconds), a discrete event similar to a mouse click is generated for the interface. In conclusion, the overall interaction system behavior is that of a one-button mouse, whose "drags" and "clicks" reflect respectively changes and fixations of interest as communicated by the user through natural pointing actions.

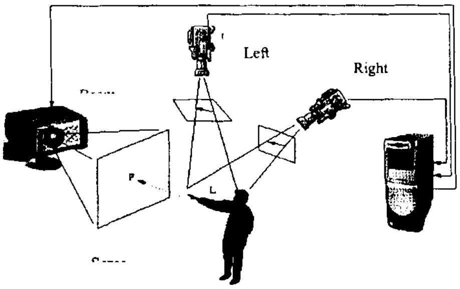

Figure 2: The main interaction components.

Interaction Geometry

A careful modeling of interaction geometry is essential to system design, since it heavily affects both localization accuracy and speed of operation. For an interaction scenario similar to the one depicted in Fig. 2, it is reasonable to assume that the cameras are enough far away from the user to neglect any nonlinear (perspective) imaging effect. The resulting affine camera model characterizes image projection as a linear mapping with six degrees of freedom $\{a_k\}$, $k = 1 \ldots 6$ [8].

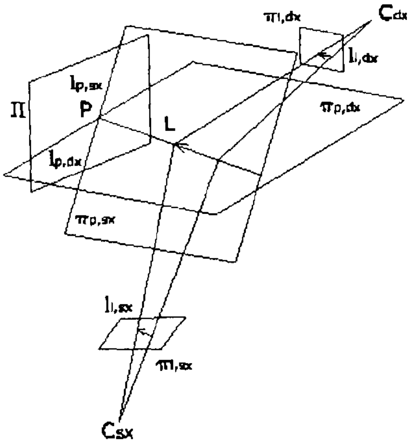

Figure 3: Interaction geometry. The subscripts 'sx' and 'dx' refer to entities associated respectively to the left and right camera.

The basic geometric problem to be solved is how to compute the location of interest P from image observations. Referring to Fig. 3, we can write $P = L \cap \Pi$, where Π is the screen plane. For each camera, an observable entity is the image line l_i, i.e. the projection of the pointing line L onto the image plane π_i. The projection plane π_p through L and l_i intersects Π in the screen line l_p; since P must belong to this line, a linear constraint of the type $P \in l_p$ can be associated to each image line l_i, whose parameters are continuously observed and updated. During normal operation (screen point remapping), two cameras are used to compute P from left and right screen line intersection. The affine stereo algorithm described here does not constrain the flat surface pointed to by the user to be visible by the cameras; as such, it can be thought of as an extension of the triangulation method proposed in [8].

In order to be used later for remapping, the affine projection parameters of each camera must be evaluated at system startup by a calibration procedure, which can be run simultaneously for the two cameras. Camera calibration requires a minimum of six known screen points, which are used together with image line observations in order to compute the projection coefficients $\{a_k\}$. In the current system implementation, nine points regularly sampled on the screen are used to calibrate, and the resulting overdetermined linear system is solved by singular value decomposition.

Image Analysis

Image analysis algorithms are used to monitor user action, and compute in real time image line parameters to be used during both calibration and remapping. Fig. 4 shows that image lines are computed as the lines through the head and pointing hand centroids.

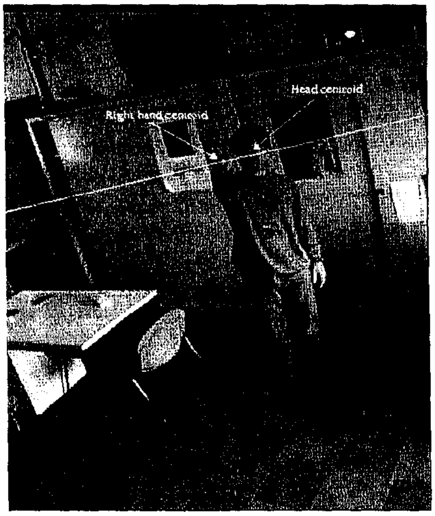

Figure 4: Run time extraction of an image line.

Before being tracked, the user is first located in the image by background subtraction. For this purpose, the background image is analyzed at a very low resolution (7×5 pixel for a 160×120 pixel image) in search of foreground information; once this is found, a recursive region growing algorithm starts, operating at different resolution levels. The multi-resolution nature of the algorithm allows a significant processing speed-up, as more than 90 percent of the raw image data are not analyzed at all. It is sufficient to grow the foreground region from a single point since the body is a connected shape. The growing algorithm expansion stops when the pixel encountered is very similar to the acquired background pixel in that position. Once a relevant foreground area is found, the user communicates his intention to start interacting with the system by setting himself in the reference body configuration shown in Fig. 5. A "locking" phase starts then, in which the anthopomorphic characteristics of the reference configuration are used by the system to capture a number of user features such as skin color, head and hand size, which are used later for user tracking.

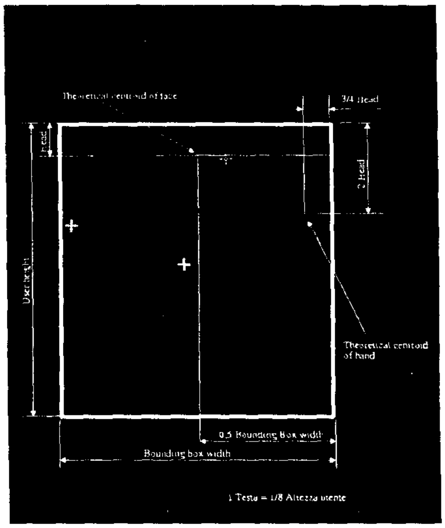
Theoretical centroid of face
3/4 Head
Head
2 Head
User height
Theoretical centroid of hand
0.5 Bounding Box width
Bounding box width
1 Testa = 1/8 Altezza utente

Figure 5: Anthropometric heuristics used during the locking phase.

Human body tracking at the image level is carried out by combining color and motion information. To check whether the pixel belongs to foreground or background, or whether it has or not a skin color, two different conditions are checked in two color spaces, *(Y,U,V)* and *(U/Y,V/Y)*, the latter being very useful in dark image portions. A region growing algorithm expanding from suitable foreground pixel "seeds" is used to evaluate the current position and shape of the hands and face regions; since these regions are very smooth and regular, skin pixel search only takes place in three directions (N, S-E, S-W) out of the 8-neighborhood canonical directions. Current region measurements are combined with previous ones using temporal low-pass filters, both to reduce noise and smoothen the tracking behavior; a constant velocity predictive filter is also used, whose effect is to produce growing seeds of better quality, thus speeding up the system response by significantly reducing the search range. Another important use of the prediction filter is the management of critical tracking situations, such as the detection of and the recovery from occlusions (hand-hand or hand-face).

EXPERIMENTAL RESULTS

The system has been implemented in C++ and runs in real time on a Pentium III PC with Linux operating system and GLUT/OpenGL graphic libraries. Two Sony EVI-D30 color cameras and a dual channel Linux Media Labs LML33 frame grabber board are used for image acquisition. As shown in Fig. 6, the system is structured into three different layers (physical: image acquisition and display, logical: image analysis and graphic synthesis, application: human-computer interfacing), communicating through a message-passing protocol. Videos showing examples of interaction with the system can be found at http://viplab.dsi.unifi.it/HCI.

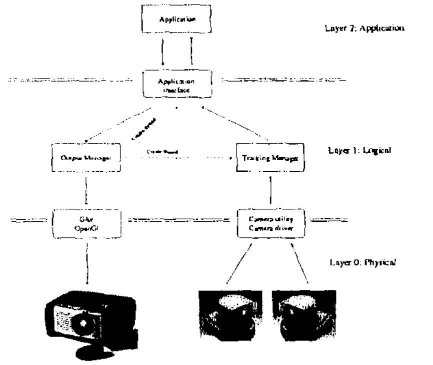

Figure 6: System architecture.

Several experiments were made to assess system performance both in terms of tracking performance and remapping accuracy. Concerning user tracking, the system has proved to be robust enough to support even long (i.e. several minutes) interaction sessions without missing the target. The main causes of tracking failure are occlusions; the system is very tolerant instead with respect to fast user movements. Concerning remapping error (i.e. the mismatch between the estimated and actual screen point locations), this depends on a number of factors, such as tracking and calibration accuracy, camera-user relative position and lighting conditions.

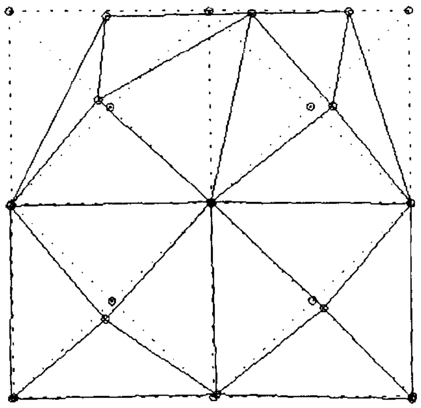

Figure 7: Remapping error: graphic data.

Results of remapping accuracy are given in Fig. 7. The user was asked to point in sequence at a number of reference locations arranged as a 13-point regular grid (dotted lines), while the corresponding locations remapped by the system were being recorded (solid-line grid). The mismatch between the two grids is generally very small, save for the three upper screen locations. Indeed, as for this experiment the two cameras were placed on the wall exactly at the same height and on either sides of these three points, the resulting image centroids of hand and head were so close to each other to cause ill-conditioning in image line measurement. This suggests that camera layout is a factor that dramatically affects system accuracy. More generally, remapping performance should increase by increasing the separation between image lines when pointing at nearby screen locations. However, such separation should be obtained with larger screens and not with wider camera angles, as in this last case the perspective effect would be so relevant to make the affine camera model unrealistic.

	$\Delta\varepsilon(X)$	$\Delta\varepsilon(Y)$	$\|\Delta\varepsilon\|$
P_1	0.241915	0.013934	0.242315
P_2	0.106528	0.006241	0.106711
P_3	-0.150202	0.002524	0.150223
P_4	-0.002958	0.004211	0.005146
P_5	-0.000634	-0.003825	0.003877
P_6	0.000961	0.000932	0.001339
P_7	-0.005624	0.002830	0.006296
P_8	0.007297	-0.007223	0.010267
P_9	-0.001437	0.005160	0.005356
P_{10}	-0.029769	-0.019459	0.035565
P_{11}	0.056718	-0.001845	0.056748
P_{12}	-0.017756	0.047626	0.050828
P_{13}	0.028386	0.019119	0.034224
Average error: 0.05906			

Table 1. Remapping error: numerical data (%).

Tab. 1 reports the numeric results (percentage of screen size) for the same experiment. Of the thirteen reference points, nine (P_1 through P_9, corresponding to the eight outer grid points, plus the grid center) had been used to calibrate. Apart from the bigger errors of the first line (P_1 through P_3), the error on the calibration points is very low (typically less than 1%). However, even far from calibration points, the error keeps limited to 5-10% of the screen size, thus making the system appropriate for use in a number of application interfaces and interaction scenarios.

REFERENCES

1. J. Nielsen Noncommand user interfaces. In *Communication of the ACM*, 36(4), 83-99.

2. R.J.K. Jacob Human-computer interaction: Input devices. In *ACM Computing Surveys*, 28(1), 177-179.

3. J.L. Crowley, J. Coutaz, F. Bèrard Things that see. In *Communications of the ACM*, 43(3), 2000, 54-64.

4. R. Cipolla and A.P. Pentland, eds. Computer vision for human-machine interaction. In *Cambridge University Press*, 1998.

5. A.P. Pentland Smart rooms: Machine understanding of human behavior. In [4], chapter 1, pp. 3-21.

6. C. Maggioni, B. Kämmerer Gesture computer - history, design and applications. In [4], chapter 2, pp. 23-51.

7. K. Mase Human reader: A vision-based man-machine interface. In [4], chapter 3, pp. 53-81.

8. R. Cipolla, N.J. Hollinghurst A human-robot interface using pointing with uncalibrated stereo vision. In [4], chapter 5, pp. 97-110.

9. C. Colombo, A. Del Bimbo, S. De Magistris Interfacing through visual pointers. In [4], chapter 8, pp. 135-153.

ABOUT THE AUTHORS

Gabriele Baggiani graduated in 2001 in Computer Engineering at the University of Florence. His main research interests are in computer vision and advanced human-machine interaction.
E-mail: *baggiani@dsi.unifi.it.*

Carlo Colombo is an Assistant Professor at the Department of Sistemi e Informatica of the University of Florence. His research work is focussed on computer vision and its applications to semi-autonomous robotics, advanced human-machine interaction and multimedia systems technology.
E-mail: *www.dsi.unifi.it/~columbus.*

Alberto Del Bimbo is Full Professor of Computer Engineering at the Department of Sistemi e Informatica of the University of Florence, the Director of the Master in Multimedia, and Deputy Rector for Research and Innovation Transfer at the same University. His scientific interests address the subject of Image Technology and Multimedia, with particular reference to object recognition and image sequence analysis, content-based retrieval for image and video databases, visual languages and advanced man-machine interaction.
E-mail: *www.dsi.unifi.it/~delbimbo.*