A Multimodal Perceptual User Interface for Collaborative Environments

Giancarlo Iannizzotto, Francesco La Rosa, Carlo Costanzo, and Pietro Lanzafame

VisiLab, Faculty of Engineering, University of Messina
{ianni, flarosa, ccostanzo, planzafame}@visilab.unime.it
http://visilab.unime.it

Abstract. In this paper a 3D graphics-based remote collaborative environment is introduced. This system is able to provide multiclient and multimedia communication, and exploits a novel multimodal user interaction paradigm based on hand gesture and perceptual user interfaces. The use of machine vision technologies and a user-centered approach produce a highly usable and natural human-computer interface, allowing even untrained users a realistic and relaxing experience for long and demanding tasks. We then note and motivate that such an application can be considered as an Augmented Reality application; according to this view, we describe our platform in terms of long-term usability and comfort of use. The proposed system is expected to be useful in remote interaction with dynamic environments. To illustrate our work, we introduce a proof-of-concept multimodal, bare-hand application and discuss its implementation and the obtained experimental results.

1 Introduction

The field of multiuser, networked Virtual Reality applications is wide and etherogeneous. A number of applications have been developed, each one of them being focused on some specific aspect of graphics [1] [2], communication or collaboration [3], or even portability on most widespread OSes and hardware platforms [4]. Among others, most popular issues in literature have been integrability with well-established Internet applications (such as the Web) [5], mobility of the user [6], portability, and low-traffic communication. In the era of multimodal and multimedia communication, though, the new frontier is represented by the user interaction interface. Perceptual User Interfaces (PUIs) use alternate sensing modalities to replace or complement traditional mouse, keyboard, trackball and joystick input: specialized devices are exploited to provide the user with alternate interaction channels, such as speech, hand gesture, etc. Three classes of technologies can be exploited for PUIs. User-obtrusive technologies are based on sensorized devices such as gloves, jackets, finger- or wrist- mounted sensors [7], which the user must wear before initiating an interaction session. Environment -obtrusive technologies rely on a series of sensors or sensorized devices connected (usually physically attached) to common life objects, such as touch-sensitive flat panels attached to the usual whiteboard or on a desk, which communicate to the computer the needed information about the user interaction. For example, the NTII virtual reality-based communication system [8] is composed by two or more individual stations, located in different

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 115–122, 2005.

places, connected by Internet and each one having its own personal PUI. An almost straightforward alternative to obtrusive technologies is using computer vision to process and analyse the video stream produced by a video camera pointing at the user. Computer vision has been exploited to produce several "demonstration systems", as they are named in [9], for each of the classes listed above.

Indeed, in the past those studies have mainly involved the computer vision and image processing communities, thus producing a plethora of different and often very effective machine vision applications, which in most cases did not address some very important issues related to traditional HCI research, such as usability, ergonomics, compatibility and integration with current applications. On the other hand, in recent years, studies on Augmented Reality (AR) have closely focused on the problems related to human factors such as comfort, long-term usability, user interfaces and perceptual problems, suggesting applications related to advanced visualization for scientific, medical and industrial purposes, entertainment, and soldier augmentation [10].

A Virtual Reality application exploiting vision-based PUI technologies involves combining real and virtual objects in a real environment, running interactively and in real time, registering real and virtual objects with each other. *Such a system matches exactly with the definition of Augmented Reality system* [11]. In this paper, thus, we propose the use of vision-based PUIs to enable advanced interaction with Virtual Reality environments as an Augmented Reality application. We then introduce an Augmented Reality-based communication and collaboration environment, able to provide multiclient and multimedia communication, which exploits a novel multimodal user interaction paradigm based on hand gesture and other perceptual user interfaces. The use of machine vision technologies and a user-centered approach produce a highly usable and natural human-computer interface, allowing even untrained users a realistic and relaxing experience for long and demanding tasks.

The paper is structured as follows: Section 2 describes the presented architecture, motivating the main design options and technology issues; Section 3 illustrates the experimental results obtained from our proof-of-concept application and discusses its implementation details; Section 4 resumes our concluding remarks.

2 System

The system is composed by a number of remote identical units connected each other by a network infrastructure. The users, by means of these units, collaborate in design tasks. Each unit (see Fig. 1) is composed by an entry-level PC, a video projector, a low-intensity infrared spotlight, a network interface, a headset, two videograbbers and two videocameras. Two video streams are thus acquired: one containing the user's face, the other the user's hand gestures and the panel. A metallic support has been built to hold a transparent plexiglas panel coated with a special semi-transparent film for rear projection (see fig. 1), on which the graphical interface is projected through the video projector.

The graphical interface of each unit is splitted in two sections, the first for rendering (and interaction with) 3D Graphical objects and the second to show the remote collaborators we are working with. Each user can *seize* a shared object for editing: when the

Fig. 1. The System

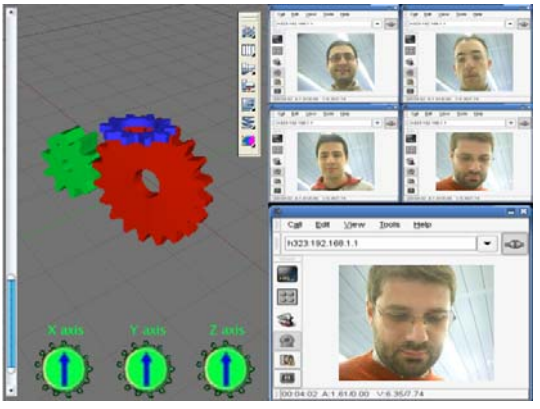

Fig. 2. The Interface

object is seized, the other users can only visualize it and cannot edit it. If an user wants to edit a seized object, he/she can reserve it and will be granted its control as soon as it is again available. Users can create and save a project, add objects to it and "manipulate" the objects as they would do in a real environment with a real object. Once a new project has been created, all the remote users share a common virtual workspace in which different users can integrate or modify some components. A semaphore based policy has been implemented to manage the access and the interaction within the workspace. The second section of panel shows a video-conference environment with its common tools in which each user working on the project is present. All the users at the same time are showed in small windows, while a megnified window shows the user "active" on a specified (clicked) object. Thus, at any time, each user knows who is editing which object.

2.1 Human Computer Interface

The conventional gestures to be used for human-computer interaction should be few, simple and natural.

After a careful evaluation, we choose some of the most common gestures in every day life. For example, the gesture choosen for the common task of *drag&drop* is composecd of the sequence "hand open-close-open" as we would do with real objects to pick them up and release them in the required position. Another functionality, for a simple interaction, should permit to easily select an object or a part of it: for this task we choose the gesture that we use when pointing an object. We obtain the *click event* (the selection) just holding the fingertip upon the object for a few seconds. The *double-click* event is obtained with the sequence " hand open-close-open". *Object resizing* is obtained by selecting an object, extending the thumb and the forefinger and then moving the hand to reduce or magnify the selection. The context menu is opened by extending the thumb. A complete gallery of the gestures we use is shown in Fig. 3.

To get more functionalities, we *augment* our panel with some other graphical tools as a *virtual scrollbar*, placed on the side of the "screen" which we use to zoom in and out the entire workspace. Also, we introduce three *virtual knobs* that allow the user to rotate the workspace around its axes, thus allowing a full 3D vision of the project; the rotation speed is proportional to the rotation angle we virtually impress to the knob.

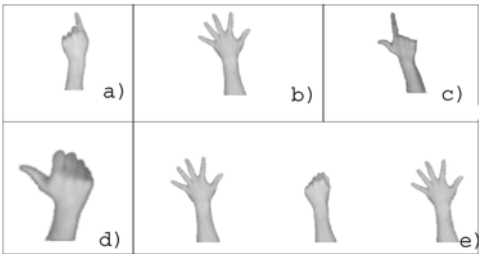

Fig. 3. The complete interaction gestures list: a) point and click, b) rotate, c) resize, d) open menu, e) double-click (or drag & drop)

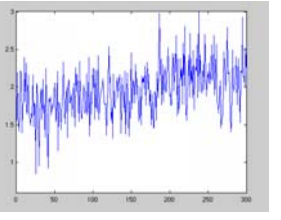

Fig. 4. Standard deviation of the error made tracing free hand an arc of ellipse

The Vision System. The video stream of the user's hand is acquired through a videocamera located under the panel, while the acquisition of the user face is obtained through a videocamera placed in front of her/him. The decision to implement an economical system and thus to use entry-level hardware requires limiting the acquisition resolution to 320x240 for the "panel" and to 176x144 for the "user face" so as to reduce the computational cost while meeting the realtime constraints. The *panel side* of the vision system works in particularly critical lighting conditions: the user hand is lit up mainly by the light beam from the video projector. This introduces a great degree of variability and unpredictability in the lighting of the target (the user hand) of the tracking system. Poor lighting and the need to make the system robust to abrupt background changes due to variations in the image being projected onto the panel make it necessary to have an additional lighting system for the projection surface. This is achieved by using a low-intensity infrared spotlight pointing towards the panel, which increases the overall luminosity but does not affect the projection itself. In this way the user hand appears white in the acquired image against an almost uniform black background. Also, a low-cost infrared filter is placed in front of the videocamera lenses (*panel side*): the effect is to eliminate most of the visible light component, which is mainly represented by the projected images. The overall result is therefore a sufficiently luminous and contrasted image in which the user hand can be seen against a dark, almost uniform background.

The segmentation of the scene, with the user hand moving against a variable background, is performed on each frame acquired by carrying out the following operations in sequence: background subtraction, thresholding, morphological closing and extraction of the connected components.

2.2 Fingers Extraction

Several finger detection systems, available in literature, use signature (an unidimensional representation of an object) analysis with respect to a fixed reference point. Our system analyses the contours of the objects, found in the frame, to check whether an hand is present. We use a contour analysis to understand if the frame contains fingers: for each point $A(x, y)$, the distance from another point B of the contour laying on the perpendicular through A to the contour itself is calculated (see fig. 7). The distances calculated and stored, for each point of the contour, produce an array. For an open hand, the signature is like in Figure 5. Each part of the hand shows some peculiar features that are similar for all users of the application. The figure 5 helps to understand the methods used for the contour analysis. Red section refers to the palm of the hand where the peaks are undoubtedly larger than those related to the fingers, this section is not interesting for fingers detection. Instead the section of array related to a finger (see figure 6) shows some typical features that do not repeat in any other point of the array. In other words, the fingers are distinguished by the presence of a central peak, due to the distances calculated close to the fingertip, and of two flat regions on both sides, due to the distances calculated along the sides of the finger. The green section in Figure 5 refers to the thumb: the peak is lower than in the other fingers but the flat regions on both sides are a little higher than in the other fingers. The other fingers, blue in the figure, are a

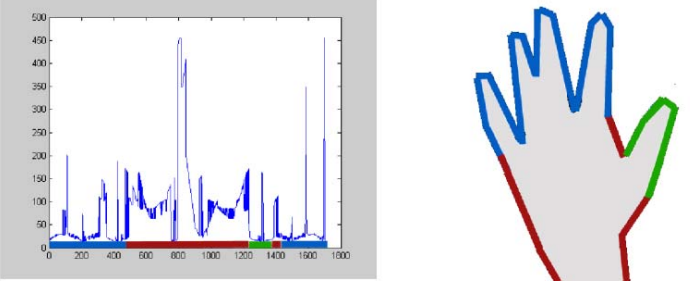

Fig. 5. An example of user hand signature

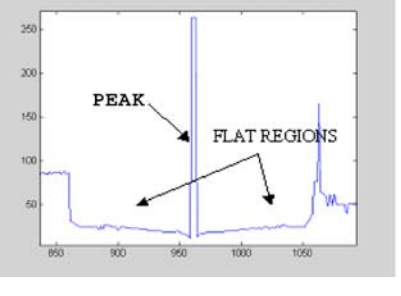

Fig. 6. A Finger in the Signature

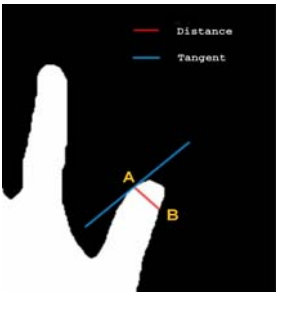

Fig. 7. Contour Analysis Method

little narrower and their peaks are higher. Figure 5 shows that sometimes some peaks can be ambiguous: in these situations a detection based only on the features described above is unadequate or not fully reliable. Therefore, to improve the robustness of the detection, some variations have been introduced. The first approach is to design an algorithm able to select those points candidate to belong to a finger. In this way the range of analysis is reduced to a smaller number of points and consequently the computational cost is lower. The contour of fingers consist of points for which the direction coefficient of the straight line, tangent to the contour and passing for them, changes abruptly. The values of the direction coefficient are stored in an array and those points for which the difference:

$$|arc[i] - arc[i-1]|$$

exceed a fixed threshold are selected. The points for which the direction coefficient changes abruptly can be the fingertips or the *valleys* of the hand. A *valley* is a region of the hand between two fingers: the shape and size of this space make it look like an upside-down finger. To discriminate a valley from a fingertip we determine the sign of the curvature of the contour. Following the contour in the clockwise direction, variations in the sign of the curvature make it possible to distinguish between convex curves (fingers) and concave curves (valleys), whose signs are respectively positive and negative. To do so, we used the method described in [12]. Once the valleys of the hand have been found, the portion of the contour between two of these could be a finger. Finally the signature of the regions classified as fingers are tested to verify the typical configuration of the fingers, i.e. central peak with flat regions on both sides.

Gesture Classification. To discriminate between the active gestures, useful for the interaction, and the neutral postures a classification phase is executed. The distances between the fingertips detected (see section 2.2) and a fixed point (center of gravity of the hand palm) are computed. To obtain the coordinates of this center of gravity a binary image, result of the pre-processing phase, is filtered by a morphological operator (erosion of *FingerWidth* size): we obtain an image in which only the hand palm is present. The center of gravity is a quite stable reference point from which to compute the distance from the fingertips; then, these distances are sorted and used to recognize the fingers extended on the panel. The computed distances are then compared with the values collected during the training phase. Before passing the recognized command to the system, the computed coordinates (fingertip position) need to be corrected because the multimedia video projector and the gray-level videocamera are not orthogonal to the projected surface (panel) and generate a trapezoidal distortion. To do so, we determine the correction parameters using the method described in [12].

3 Experimental Results

The system was tested during and after development by several users for a considerable number of hours with different external lighting conditions. To evaluate the performance of the system a considerable number of tests were carried out and repeated for each of the primitive gestures listed below:***click, double-click, resize, rotate, open menu, drag & drop***.

Table 1. Experimental Results

Command	# Tests	Hits	Near Hits
Click	100	96	//
Double-Click	100	93	//
Resize	100	95	//
Rotate	100	91	//
Open Menu	100	94	//
Drag & Drop	100	97	99

The results obtained are given in Table 1. The percentage of hits in Table 1 indicates the number of times the action was performed correctly, i.e. according to the intention of the user, while the column referring to the percentage of near hits in drag & drop operations comprises both real hits and the number of times the object was released in the wrong place. To produce a quantitative evaluation of the tracker accuracy we compared the output of the system with a ground-truth reference. So we predisposed a test that can meaningfully characterize our system. For this test, we considered an arc of ellipse projected onto the control panel, that must be followed tracing it for its entire length with the fingertip. The measures have been realized asking 10 users to test 5 times the system following free hand the prefixed trajectory, that has 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 curve points coordinates (computed by the system) and the real coordinates of the curve points; 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. The measures 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 the system allows to maintain the error under 3 pixels and the analysis of 50 measures carried out shows a medium value of 2 pixels. Fig.4 shows results, along the 300 points of abscissa, of the standard deviation of the error made tracing free hand an arc of ellipse for the test. The increment of the error in the second half of the segment is probably generated from a decay of the attention of the users. 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 algorithm is usually very computationally intensive thus revealing unsuitable for our purpouses. We therefore decided to keep this error.

4 Conclusions

In this paper we introduced a 3D graphics communication and collaboration environment, able to provide multiclient and multimedia communication, which exploits a novel multimodal user interaction paradigm based on hand gesture and perceptual user interfaces. The use of machine vision technologies and a user-centered approach pro-

duce a highly usable and natural human-computer interface, allowing even untrained users a realistic and relaxing experience for long and demanding tasks. The system has been carefully characterized in terms of accuracy thus allowing for an estimate of its uncertainty and of its usability. Experimental results are shown and discussed.

References

1. Leung, W.H., Goudeaux, K., Panichpapiboon, S., Wang, S.B., Chen, T.: Networked intelligent collaborative environment (netice). In: Proc. of the IEEE Intl. Conf. on Multimedia and Expo., New York (2000)
2. Rich, C., Waters, R.C., Strohecker, C., Schabes, Y., T.Freeman, W., Torrance, M.C., Golding, A.R., Roth, M.: Demonstration of an interactive multimedia environment. IEEE Computer **27** (1994)
3. A.Fersha, J.: Distributed interaction in virtual spaces. In: Proc. Of the 3rd International WS on Distributed Interactive Simulation and Real Time Applications, IEEE CS-Press (1999)
4. Carlsson, C., Hagsand, O.: Dive - a platform for multi-user virtual environments. IEEE Computers and Graphics **17(6)** (1993)
5. : (Smartverse web page) `http://www.smartvr.com`.
6. Ferscha, A.: Workspace awareness in mobile virtual teams. In: WETICE 2000, IEEE Computer Society (2000)
7. Rekimoto, J.: Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In: Proceedings of ISWC'01, Zurich (2001)
8. Towles, H., Chen, W.C., Yang, R., Kum, S.U., Fuchs, H., Kelshikar, N., Mulligan, J., Daniilidis, K., Holden, L., Zeleznik, B., Sadagic, A., Lanier, J.: 3d tele-collaboration over internet2. In: Proceedings of International Workshop on Immersive Telepresence (ITP2002), Juan Les Pins, France (2002)
9. Ye, G., Corso, J., Burschka, D., Hager, D.: Vics: A modular vision-based hci framework. In: Proceedings of ICVS 2003, Graz, Austria (2003)
10. Azuma, R.: A survey of augmented reality. Presence: Teleoperators and Virtual Environments **6** (1997) 355–385
11. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. IEEE Computer Graphics and Applications **21** (2001) 34–47 `http://citeseer.csail.mit.edu/azuma01recent.html`.
12. Costanzo, C., Iannizzotto, G., LaRosa, F.: Virtualboard: Real-time visual gesture recognition for natural human-computer interaction. In: Proc. of the IEEE IPDPS'03, Nice, France (2003)