An Experimental Multimedia System Allowing 3-D Visualization and Eye-Controlled Interaction Without User-Worn Devices

Siegmund Pastoor, Jin Liu, and Sylvain Renault

***Abstract*—In this paper, a new kind of human–computer interface allowing three-dimensional (3-D) visualization of multimedia objects and eye-controlled interaction is proposed. In order to explore the advantages and limitations of the concept, a prototype system has been set up. The testbed includes a visual operating system for integrating novel forms of interaction with a 3-D graphic user interface, autostereoscopic (free-viewing) 3-D displays with close adaptation to the mechanisms of binocular vision, and solutions for nonintrusive eye-controlled interaction (video-based head and gaze tracking). The paper reviews the system's key components and outlines various applications implemented for user testing. Preliminary results show that most of the users are impressed by a 3-D graphic user interface and the possibility to communicate with a computer by simply looking at the object of interest. On the other hand, the results emphasize the need for a more intelligent interface agent to avoid misinterpretation of the user's eye-controlled input and to reset undesired activities.**

***Index Terms*— Eye-controlled interaction, gaze tracking, graphic user interface, head tracking, human–computer interaction, interface agent, 3-D display.**

I. Introduction

GRAPHICAL human–computer interfaces applying windows, icons, menus, and mouse pointers have considerably simplified the use of computer programs compared with the use of purely text-oriented input and output techniques. However, strictly two-dimensional (2-D) surfaces present obvious restrictions: when several applications run simultaneously, overlapping windows make it difficult to watch the screen and to follow its contents. In this context, three-dimensional (3-D) displays literally offer one more dimension for visualizing the data flow and the interplay of programs in complex multimedia applications in a very natural way.

Binocular vision allows humans to perceive and unambiguously interpret spatial structures without additional mental effort [1]. Thus, stepping into the third dimension could make it easier for users to perceive and understand complex information structures. In experiments on the understanding of abstract information networks presented in 2-D versus 3-D, Ware and Franck [2] found evidence that true 3-D viewing can increase the size of a graph that can be understood by a factor of three. Compelling examples of the more effective use of the limited computer-screen space by 3-D presentation were developed at Xerox Palo Alto Research Center (Xerox PARC). For instance, Card *et al.* [3] implemented an animated cone tree browser to three-dimensionally visualize the structure of hierarchical databases, and Robertson *et al.* [4] designed a perspective wall providing a detailed view of multimedia objects on the front part of the wall and context information on the perspective peripheral left and right wings (refer to [5] for an overview of approaches for coping with screen-space limitations).

To fully exploit the potential of 3-D visualization, a further step forward in the evolution of human–computer interaction has been suggested [6]: while current computer operating systems like MS Windows, MacOS and Unix implement command-based, direct-manipulation interfaces [7], next-generation user interfaces should supplement this concept by introducing nonconventional controls [8] and intelligent interface agents to support noncommand interactions [9]. The idea is to give the computer sensors so that it can constantly observe the user. Being aware of the user's activities and taking the situational context into account, the interface agent will be able to interpret the user's intentions and to infer how to optimally adapt interaction to the user's needs. Hence, the interface agent could relieve the user of "routine" actions, giving him or her the freedom to fully concentrate on the task at hand—rather than on operating the computer.

In previous studies, gaze tracking has proven to be a powerful approach when implementing noncommand interactions [10]. Knowing the user's current point of fixation on the display screen and the immediate history of eye movements allows the interface agent to distinguish whether the user is focusing attention on a particular item (e.g., an icon representing an application program) or whether he or she is unintentionally scanning the display screen [9]. The fundamental studies on vision-based noncommand interactions by Jacob [11] demonstrated that when the system responds quickly and accurately users will forget about the fact that the computer is reading their eye—they get the feeling that the system anticipates their intentions before they express them.

The approach outlined so far can be summarized as an attempt to merge the benefits of 3-D visualization techniques with computer vision in order to make interaction with a computer more user-friendly. Up to now, experiments concerning

Manuscript received August 26, 1998; revised November 23, 1998. This work was supported by grants from the German Ministry for Education, Science, Research and Technology (BMBF) under Contract 01BK410 and from the State of Berlin. The associate editor coordinating the review of this paper and approving it for publication was Prof. U. Neumann.

The authors are with the Heinrich-Hertz-Institut Berlin, 10587 Berlin, Germany (e-mail: pastoor@hhi.de; liu@hhi.de; renault@hhi.de).

Publisher Item Identifier S 1520-9210(99)01578-3.

the measurable advantages of this approach have been very sparse, and it is impossible to give reliable conclusions about the user's appreciation of the new concept and the usability of the interface as a whole.

Against this background the authors of this paper have developed a practical testbed in order to perform usability tests and to assess cost/benefit tradeoffs in an anticipated application scenario. A prototype system [12] was used for preliminary user testing and shown to the public at an international broadcasting exhibition. At present, the testbed's key components are free-viewing 3-D displays, a visual operating system providing a graphical 3-D user interface, a camera to sense the user's head position and motion (head tracker), and equipment to measure the user's point of fixation (gaze tracker). The 3-D displays eliminate the need for 3-D viewing glasses and provide the users with high-resolution images, including live stereo-images for videoconferencing. With the head tracker, a simple movement of the head is sufficient to open the view of a document hidden behind a visually overlapping foreground object (dynamic perspective). Simultaneously, the gaze tracker determines the current point of fixation, so that looking at the formerly hidden document will pull it closer to the user making it easier to read. Moreover, the gaze tracker adjusts the process of image rendering so that only the object being looked at appears in full focus—objects out of the user's gaze are temporarily considered unimportant and therefore shown out of focus, helping them to fade from perception (active accentuation). This effect mimics the limited depth of focus of the human eye [13], [14]. Commercial voice control software as well as a video-based hand tracker for direct manipulation of visualized 3-D objects [15] will be integrated in the testbed in order to offer a variety of multimodal interactions.

II. System Overview

Fig. 1 illustrates the basic arrangement of the system with the 3-D display and interface components. For noncontact interaction two video cameras are mounted at the front of the display. One camera is used for head tracking and video communication and the other one for sensing the gaze direction. The gaze camera and an array of infrared light-emitting diodes are mounted on a pan-tilt platform. This camera is aimed at one of the user's eyes and captures a zoomed infrared image which is evaluated in order to estimate the current line of sight. A novel operating system supports the 3-D representation of applications and media objects as well as the eye-controlled interactions. The interface also includes a keyboard and a mouse because we expect that, even when the computer supports noncommand interactions, several tasks are more naturally accomplished by explicit commands using conventional input devices. The hand tracker shown in Fig. 1 and voice control have not yet been implemented. The key components of the current setup are discussed in detail in the following sections.

III. Visual Operating System (VOS)

The core element of the proposed user interface is a new operating system based on the concepts of both object-oriented programming (in terms of adaptation and the use of already

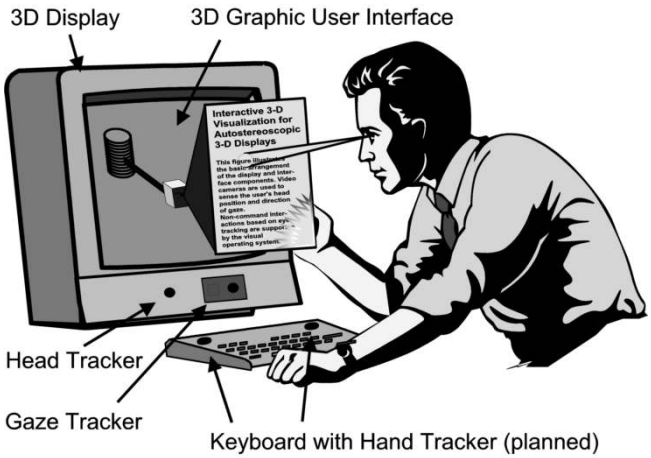


Fig. 1. Basic components of the proposed system. The interactions with three-dimensionally visualized multimedia objects include a natural change in perspective with head motion, accentuation of the currently fixated object, and direct manipulation of displayed objects by hand gestures.

available program functions) and visual programming (in terms of the display of software modules). This means that users can use a graphic editor to create and "program" their environment and applications by simply linking appropriate software modules together. Compared to conventional visual programming tools, the proposed system builds on the advantages of 3-D visualization, allowing a clearly structured representation of the interconnected program modules. The operating system runs on a Silicon Graphics Onyx computer and uses the virtual reality software dVS by Division Ltd. as a basis for generating 3-D graphics and for implementing user interactions.

The operating system subdivides all user-accessible software modules into three levels of complexity, comprising primitives as the smallest units (basic arithmetic or graphic functions), components, and large-scale applications (aggregations of lower-level modules), respectively. The graphic representations of primitives (called gadgets) can be used to add animation, sound, or database queries to a module, for example. The gadgets are stored in the library of the VOS and can easily be modified in order to change their visual appearance, without changing their particular functionality. Fig. 2 shows representations of a gadget used for scrolling the visible part of a text document. Several gadgets may be combined in order to form a component which can be used in various applications. In order to form larger software entities from lower-level modules, a module may have "docks" which can be visually connected via a pipeline in order to enable the exchange of information packages between the modules (Fig. 3). Thus a network of interconnected software modules can be created, tested, and modified step by step in order to ultimately build a complex application program.

In order to get a clearly structured representation of complex programs, users can apply a "zoom-out" function. This way, components can be compiled to form an aggregation where only the external docks are visible and the inner network is hidden (Fig. 4). It is also possible to zoom in in order to visualize the large number of primitives and components forming the network of software modules in an application program [6].

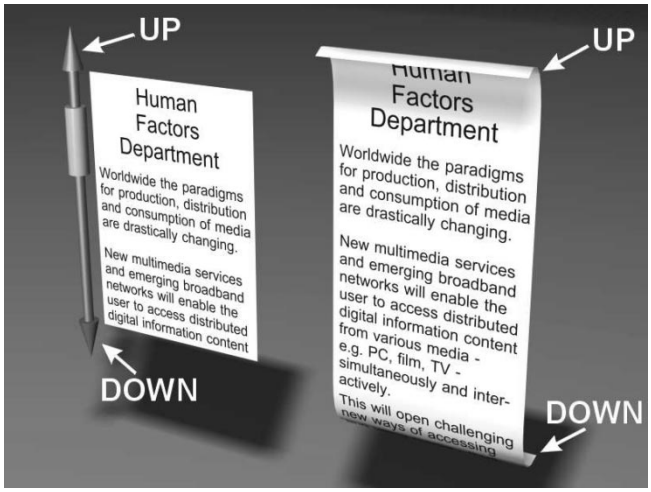


Fig. 2. Appearance of a basic gadget is easily adapted to the user's preference without changing its functionality (scrolling the contents of a text document with a conventional scroll bar versus applying a papyrus-scroll metaphor).

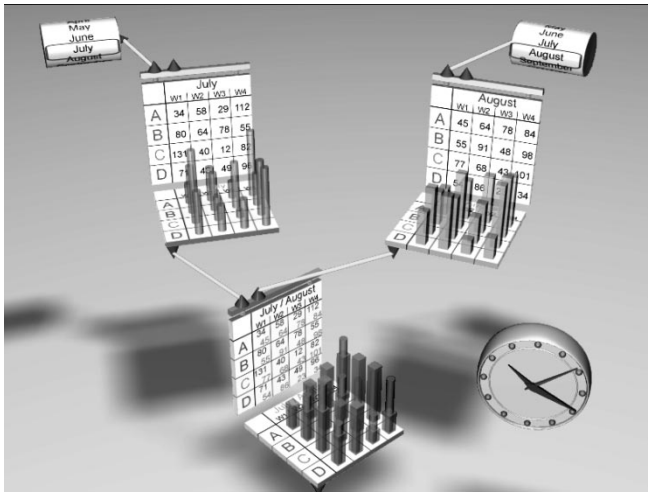


Fig. 3. The user-accessible software modules provide synapses (docks). Pipelines between the docks enable the transfer of information and allow the user to create complex applications (in this example a multidimensional spreadsheet) by visually connecting the docks of basic components.

When talking of "users" in the framework of the new operating system traditional differences between software developers and end users are intentionally diminishing. Future end users should have the possibility to set up their own individually tailored application programs. Instead of using a huge monolithic program with a multitude of possibly never used functions, end users should be allowed to combine software modules from different suppliers according to their particular needs. The VOS aims at making interchangeable software components applicable to end users whilst providing an easy-to-use, clearly structured visual interface in connection with intuitive manipulation techniques.

IV. Application Scenario

The boxes in Fig. 5 show icons of frequently used application programs or tools mapped onto their surfaces. By changing the viewing position (head movement), the user will see the boxes from different perspectives. The boxes change their perspectives overproportionally so that it is possible to make each side visible by small head movements. Now, imagine that the user is looking at one of the application icons (e.g., a spreadsheet application). Since the computer knows the gaze point, an "event" is triggered, starting an animation primitive which magnifies the fixated spreadsheet icon for feedback and improved visibility. If the user continues to look at the icon for a certain period of time (e.g., for 150 ms, as proposed in [11]), thus signaling an increased interest, the interface agent will instance the spreadsheet program. A sequence of visual interactions will allow the user to select and visualize numerical data in a 3-D bar diagram. In the example shown in Fig. 3, some data related to the months of July and August are displayed on a weekly basis. Looking at the corresponding input and output docks of the spreadsheets will create pipelines and produce a combined presentation of the data. Meanwhile, the operating system will automatically reposition the applications within the limited 3-D display volume to warrant optimal visibility. The position and size of an object can also be manually relocated by drag-and-drop operations with a conventional 2-D input device, such as a mouse. In this case, the user's gaze selects the object to be manipulated in a predefined mode (movements in a plane parallel to the screen or along the z-axis, respectively).

Another scenario requires the user to find the "music box" icon located on the surface of one of the tool boxes and to launch it by eye-controlled interaction. After selection, the interface agent will automatically position the application in the displayed 3-D volume. When the user looks at the music box, it comes closer and opens its cover. Next, the user can visually select a title and start playing it by glancing at the virtual play button. After playing and when no further interactions happen, the interface agent will remove the music box from the display.

Figs. 6 and 7 show the implementation of a local file browser and a Web browser, respectively. In Fig. 6, the object in the background represents a file system which is connected to a file-type filter (the cylindrical object with buttons to select the type of file desired). When the user looks at a particular file, the corresponding viewer (in this example a text viewer) will be launched. The text automatically scrolls when the user's gaze point has reached the top or bottom of the text page. Glancing at a hypertext item on a Web page (Fig. 7) for a certain dwell time automatically downloads the hyperlinked document. The previously loaded documents will move backward, thus indicating the search path to the current document. Looking at a background document will in turn move it closer. Changing the viewing position by moving the head discloses occluded documents and thus helps the user keep track when browsing the Web.

V. Autostereoscopic Display

Because the computer evaluates the user's gaze, the proposed system requires a 3-D display without polarizing glasses or any other head gear occluding the user's eyes. Autostereoscopic 3-D displays [16] are based on the concept of direc-

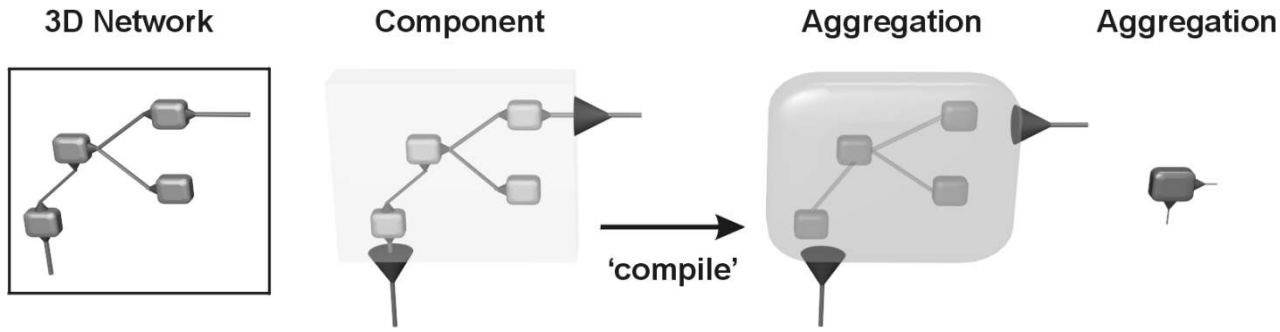


Fig. 4. Network of software modules forming an application can be visualized at different levels of detail using a particular zoom function. At each level, the accessible external docks are accentuated. Obscuring the inner network improves the clearness of the presentation.

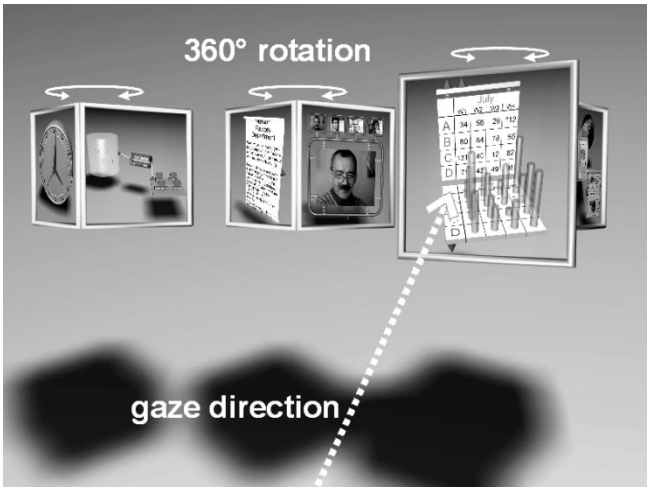


Fig. 5. Tool boxes showing icons of applications or documents change perspective in response to the user's head movements and react when the user looks at a particular icon by, e.g., launching the relevant application program.

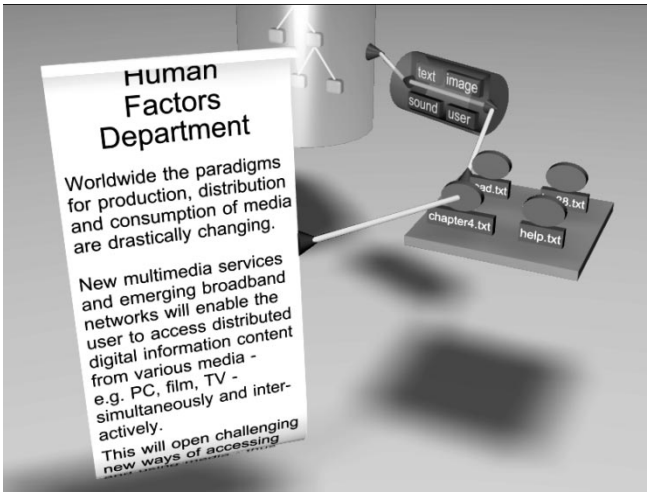


Fig. 6. Implementation of a local file browser.

tional multiplexing, which means that the different perspective views are visible only from a limited number of fixed viewing positions. When the user positions his or her head so that the eyes are within the viewing zone, both views are immediately fused to create the illusion of a 3-D space. For practical reasons, such displays must have a tracking mechanism in order to optically address the eyes, both at a fixed head position and also when the user moves.

Fig. 7. Implementation of a Web browser.

Fig. 8 shows the optical principle of a 3-D display which we made in cooperation with Carl Zeiss (Germany). This display uses a movable lenticular screen in order to optically address the eyes over an extended viewing zone. The lenticular screen (made by Philips Optics, The Netherlands) is placed in front of an LCD screen. The left and right image contents are simultaneously displayed in columns side by side. As a result, columns 1, 3, 5, 7, etc., (labeled "R" in Fig. 8) display the information for the right eye, while columns 2, 4, 6, 8, etc., (labeled "L") display the information for the left eye. Since the lenticular screen has a directional selectivity in the horizontal plane, the color primitives of the LCD panel have to be aligned vertically one above the other in order to avoid color separation of the RGB components. Since the color primitives in commercial LCD panels are aligned horizontally, the LCD panel is rotated by 90°. The lenticular plate separates the two stereo pictures for the viewer's eyes. Depending on head movements, the lens plate is mechanically adjusted to the left and right as well as in the frontal direction. The maximal tracking range is mechanically limited to about 30 cm for lateral and frontal head movements with a nominal viewing distance of 67 cm (this translates to a lens shift of 0.042 mm per 1 cm of frontal or horizontal head movement at the nominal distance; for more details, refer to [17]). As the

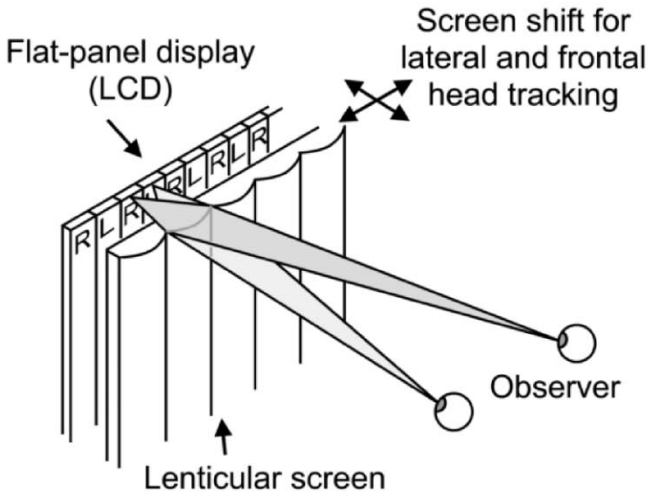
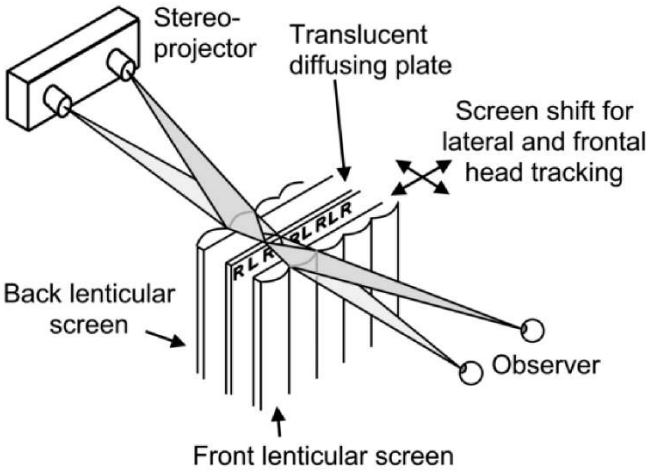


Fig. 8. Lenticular screen allows the separation of viewed 3-D images created in a raster-scan mode. The photo shows a 14-in XGA resolution prototype 3-D display developed for desktop applications.



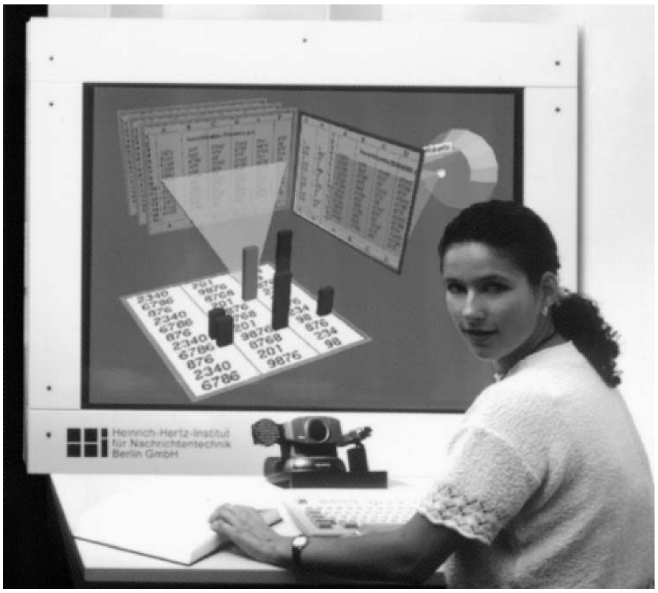


Fig. 9. Principle and prototype of a 50-in projection-type dual lenticular screen 3-D display.

screen is composed of vertically oriented cylindrical lenses, vertical head movements do not require lens shifting; they are only limited by the tracking range of the head position sensor.

A 50-in, projection-type 3-D display has also been developed in cooperation with Philips Optics and Cybertron (Germany). This display uses a dual lenticular screen with 1000 lenses and two LCD projectors, each with XGA resolution (Fig. 9). The cylindrical lenses of the back lenticular screen are used to form an array of left-right image stripes on an intermediate diffusing plate. The front lenticular screen has the same lens pitch as the back screen, which causes the two stereo half images to be channeled to the left and right eyes. Again, the front lenticular screen is shifted mechanically in the lateral and frontal directions in order to follow the head movement (0.064 mm per 1 cm of head movement). The tracking range and the nominal viewing distance are about twice as large as for the desktop display. Both displays are single-user displays and provide changes in perspective (generated by the computer) in response to horizontal, frontal, and vertical head movements within the tracking range.

The 3-D displays developed so far have an inherent shortcoming which strains the eyes: the user must focus on a fixed viewing distance (the screen distance), although the stereo objects may appear close to the eyes or far behind the screen. By contrast, in natural viewing conditions the accommodation distance changes in accordance with the distance of the object observed (convergence distance of the lines of sight). To make display viewing more comfortable, we propose the "depth-of-

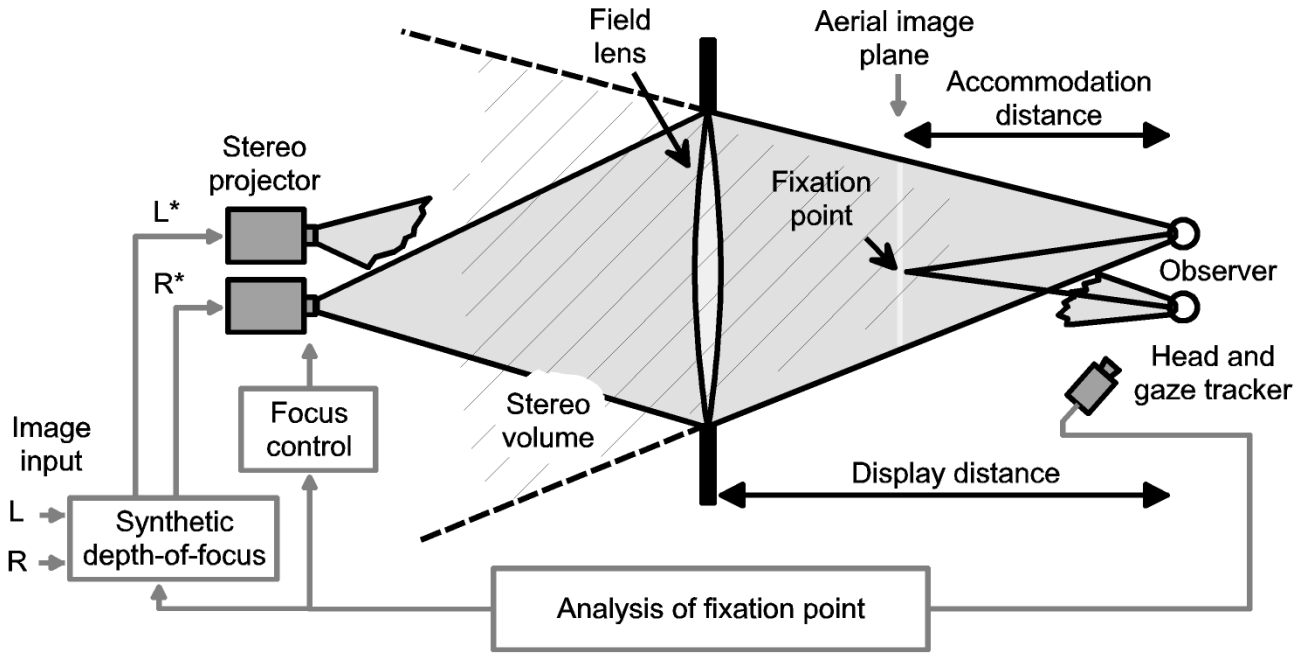


Fig. 10. Components of the proposed depth-of-interest display. The position of the aerial image of the observed object corresponds to its apparent location within the stereo volume.

interest" display [18] shown in Fig. 10. Note that the stereo images are not projected onto a physical display screen at a static location; instead, the display provides a movable image plane enabling fixation-point dependent accommodation. The stereo images appear as aerial images floating in front of or behind a large Fresnel-type field lens. The location of the image plane is controlled by motorized adjustments of the projection optics (focus) in such a way that the aerial image appears at a distance corresponding to the stereoscopic distance of the object the user is looking at. This type of display needs to "know" both the user's head position as well as the stereoscopic depth of the currently fixated object (as estimated from the head position and gaze direction). As the viewer accommodates on the aerial image plane, accommodation distance and convergence distance coincide like in natural vision. Additionally, the display concept encompasses a natural depth-of-focus effect by depth-selective, spatial low-pass filtering of the projected images. As opposed to the lenticular screen displays, the "depth-of-interest" display has not yet been built; however, a proof-of-concept test using static stereo images was successful.

VI. Head Tracker

Much effort has gone into implementing real-time video-based methods for detection and tracking of the user's head (or more precisely, of the 3-D locations of both eyes) [19]. The tracking algorithm developed had to overcome difficulties such as variable orientations, sizes and partial occlusions of the face in the camera image, as well as noise and poor camera resolution (a special high-resolution video camera was not foreseen in our system). The two essential steps in the tracking process are face detection and eye localization.

A. Face Detection

Although it is very easy for humans to locate, recognize, and identify faces, there is still no image processing system available that is capable of solving this task comparably well [20]. To implement a real-time system, skin-color based approaches have several advantages compared to other methods [21]. The processing of color information has proven to be much faster than processing of other facial features. Under constant lighting conditions, color is almost invariant against changes in size, orientation, and partial occlusion of the face. For distinguishing the face color from other image regions, either a general color model (based on the distribution of skin colors in a population) or a user-specific model can be used. We decided to adapt the color model to the individual user, as several studies (e.g., [22]) showed that this approach is more reliable and robust than the use of general models.

During initialization, the user's image is analyzed to determine the individual skin color distribution. An adaptive threshold technique is applied in order to extract image regions with high probability of face color when compared with a generalized skin-color model. In order to speed up the segmentation process, a look-up table is generated which relates each color sample with its corresponding area inside the defined RGB color space. However, as shown in Fig. 11, color information alone is not sufficient for face segmentation, since the background behind the user may also contain skin-colored objects that could mistakenly be considered to be part of the facial region.

In order to improve the classification process, we make use of a reference image containing only background objects. Such a reference image is conveniently captured before the user sits down. By detecting the luminance differences between the current image and the reference image, the user is easily

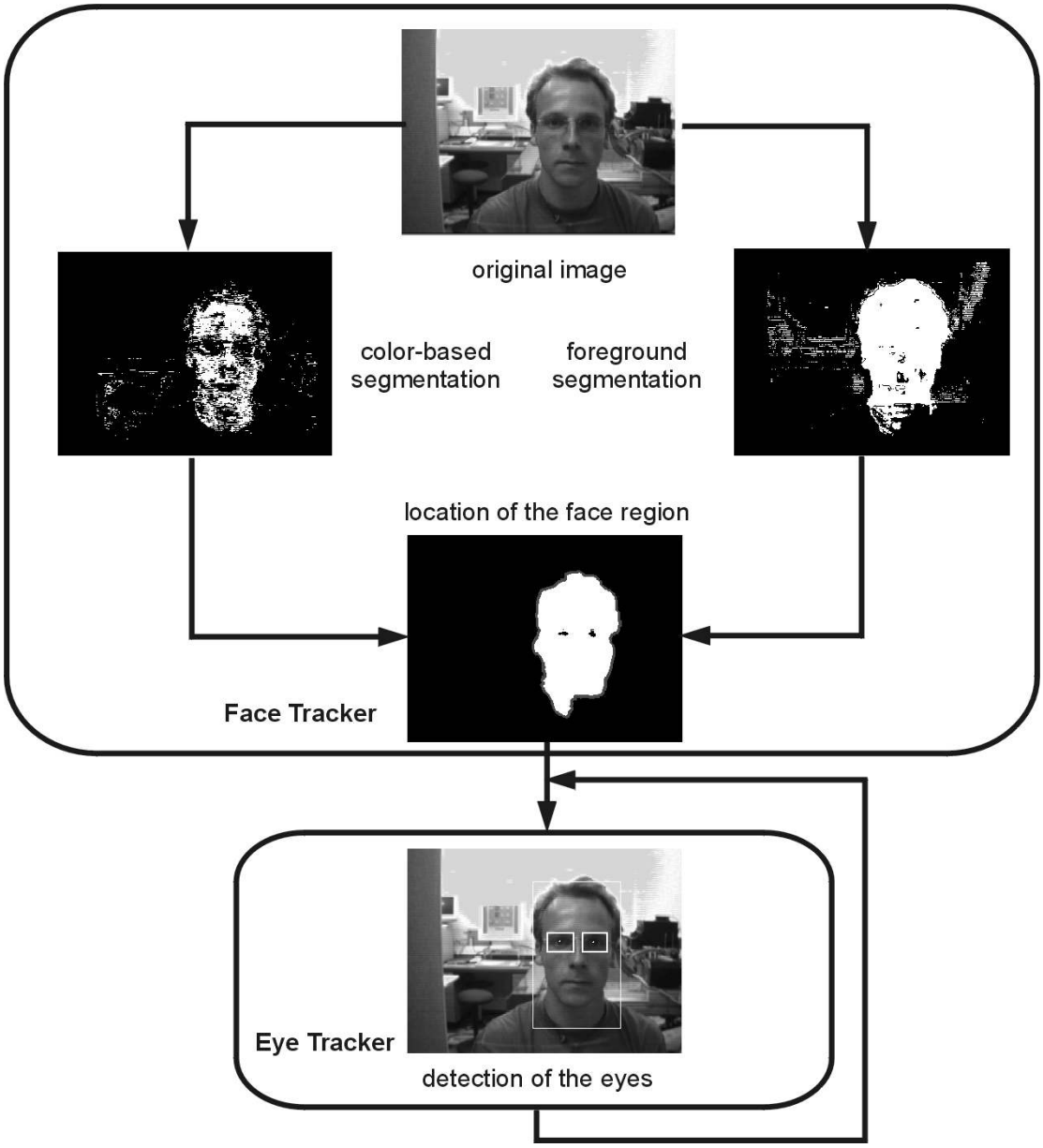


Fig. 11. Basic image processing steps of the video-based head tracker.

located as the foreground object. The reference image must be updated during the entire tracking process by an adaptive algorithm which takes changes within the background region into account. Regions classified both as skin and foreground are registered as facial regions. The segmentation result is further improved by applying nonlinear filters (e.g., dilation and erosion) in order to fill holes in large connected regions and to remove small regions. After the facial region is located, the color, foreground/background, and motion information are combined and evaluated to keep track of the face when the user moves.

B. Eye Detection and Tracking

In the second step, the eyes must be found within the defined facial region. As humans periodically blink to lubricate their eyes, the closing of the eyelids is analyzed to locate the eyes. The fact that both eyes blink at the same time provides useful information for distinguishing blinking from other motions. Eye blinking can be detected by analyzing the luminance differences in successive video images. Within the facial region, three (instead of two) equal-sized, nonoverlapping blocks with the maximal difference between every two subsequent image frames are detected. This is carried out by calculating the values of the squared frame differences (SFD) of the blocks. Two blocks are registered as eye regions

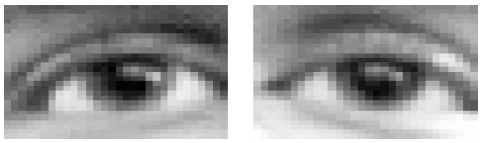


Fig. 12. Eye patterns stored and used to find the eye positions in images.

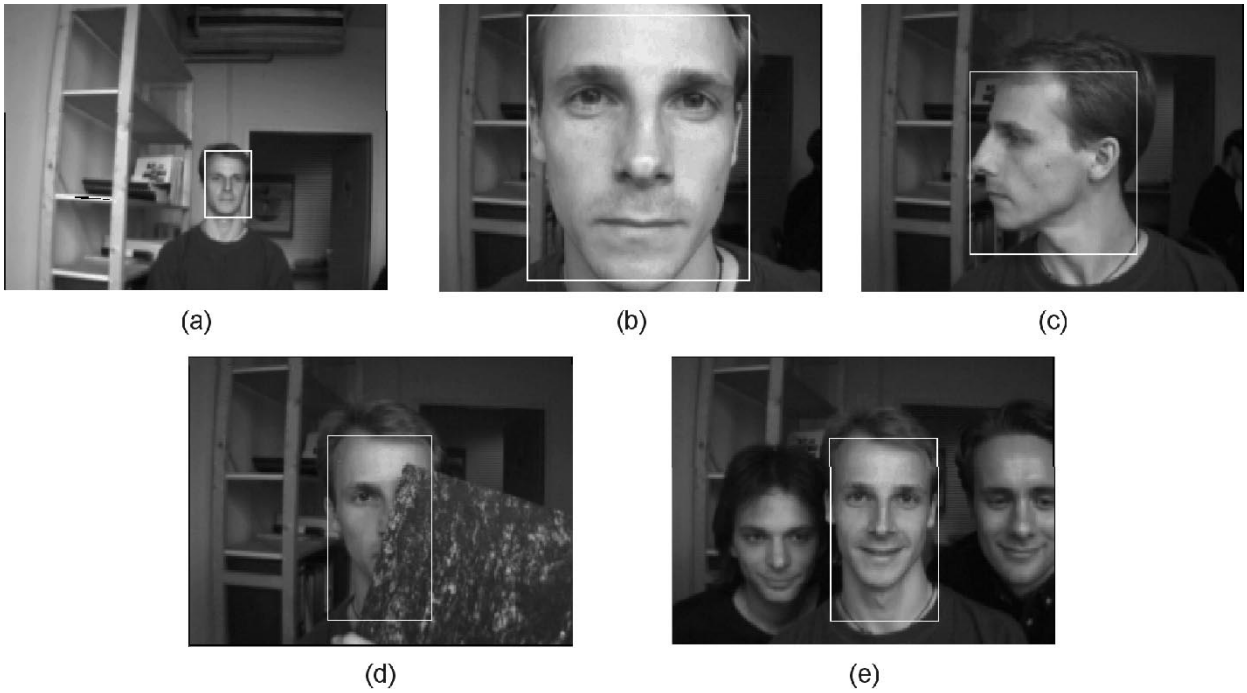


Fig. 13. Video-based head tracker is able to cope with a range of difficult imaging conditions.

if their SFD values are very similar (since the eyes blink simultaneously) and significantly larger than the SFD value of the third block. Additionally, the algorithm takes account of geometric relationships of the eyes. The eye regions are registered and stored as reference patterns for tracking the eye positions when the user moves.

The tracking algorithm is based on a luminance-adapted block matching technique. In order to compensate for temporal changes in luminance and size of the eye pattern, both the reference eye patterns and the extracted eye regions in the previous image frame (temporal patterns) are used for matching the eye regions in the current image frame. A block with the same size as the reference pattern is shifted in the defined facial region, and the squared block differences between the reference pattern and the current block, as well as a weighted difference between the temporal pattern and the current block, are calculated and summed up. The block with the minimal weighted difference is considered to be the best estimate of the eye region. After successful matching, the luminance of each reference eye pattern is adjusted by the difference of the average luminance values of the detected eye region in the current image and in the reference pattern, respectively, in order to compensate for temporal changes in lighting. Fig. 12 shows an example of the stored eye patterns. The darkest circular part in the eye pattern marks the position of the pupils.

The performance of the head tracker is demonstrated in Fig. 13. Fig. 13(a)–(c) shows some extreme conditions with strong variations in size and orientation of the head. Fig. 13(d) shows that the position of the head is correctly detected even when partially occluded, and Fig. 13(e) illustrates that the user's face is still correctly found when other people enter the scene.

Fig. 14. Eyes are detected and tracked in a sequence of video images.

In a workplace with near-constant lighting conditions, the face detection process can be terminated after both eyes have been found, and the information contained in the two eye patterns is sufficient for tracking. The automatically detected eye regions are marked in the video image in Fig. 14. After localization of both pupils in an image, their positions in 3-D space can be derived if the interocular distance of the user and the camera parameters are known. The formula for computing the 3-D locations of the eyes assumes that the plane of the user's eyes is parallel to the plane of the display.

The accuracy of the head tracking algorithm depends on the resolution of the head camera and the tracking range (camera viewing angle). The permissible tolerances for the autostereoscopic display in order to correctly address the user's

eyes are particularly small. The accuracy of our system setup is about 2–3 mm in the lateral direction and about 1 cm in the frontal direction, which meets the requirements of the autostereoscopic displays. The accuracy numbers were measured by direct comparison with the Origin Instruments DynaSight sensor (which has a measurement resolution of 0.1 mm in three axes) and by evaluating the variation of the measured pupil locations in the 2-D head images. The accuracy could be further improved by applying two head cameras in connection with stereo analysis techniques. The head tracker has been implemented on a SGI O2 workstation using a commercial miniature camera and a standard video frame grabber. A measurement rate of 25 Hz is currently possible (limited by the grabber's frame rate). When the user turns away from the display (i.e., when no eyes are found in the camera image) and then turns back, it takes about three to four frames to "regain lock." The tracking process breaks down when illumination changes drastically. In this case, the initialization process will start again in order to update the user's eye pattern.

The user's 3-D eye positions must be known for three purposes: 1) for the autostereoscopic display in order to optically address the user's eyes; 2) to adapt the 3-D perspectives of the graphic output to the user's view point; and 3) to aim the gaze camera at one of the user's eyes in order to attain a zoomed image for precise gaze detection.

VII. Gaze Tracker

The gaze tracker measures eye movements and estimates the user's current point of fixation. The fixation point is defined as the intersection of the line of sight of one eye (gaze line) with the surface of the object being viewed in stereoscopic space. This information is used to interpret the user's intention for noncommand interactions and to enable (fixation-dependent) accommodation and dynamic depth of focus.

We have developed a novel algorithm for determining the gaze line with active compensation of head movements. The measurement of the gaze line is derived from the measurements of both the gaze direction (the unit vector of the gaze line) and the eye location (determined by the head tracker).

For the purpose of determining the gaze direction, we opted for the cornea reflex method [23] because of its high precision and stability and because it is nonintrusive (the measurements can be taken from a distance). The eye is illuminated with low-intensity infrared light. As a result, the pupil appears as a black elliptical region in the camera image. The center of the pupil and the reflection of the light from the cornea are determined by image processing. There is a monotonic relationship between the vector pointing from the center of the pupil to the light reflection (eye vector) and the user's gaze direction. After individual calibration, the gaze direction can be precisely derived from the eye vector.

A. Finding the Eye Vector

In order to speed up measurements, the generally elliptical image of the pupil is approximated by a circle that seems to be admissible in conditions where the optical axis of the camera is fairly parallel to the optical axis of the eye [24]. The resulting error in gaze measurements is smaller than the error caused by inhomogeneities of the surface of the cornea. First, a Hough transform-based technique (circle Hough transform) is used to roughly determine the position of the pupil. For edge detection, the image is filtered with a Sobel operator. All edge pixels are transformed into an accumulator array where the magnitudes of the luminance gradients are accumulated along their particular direction. The global maximum of this array is interpreted as the center of the estimated pupil circle (Fig. 15). The threshold value of the edge detector should be sufficiently low so that the global maximum is even found in a video image which is slightly blurred due to motion.

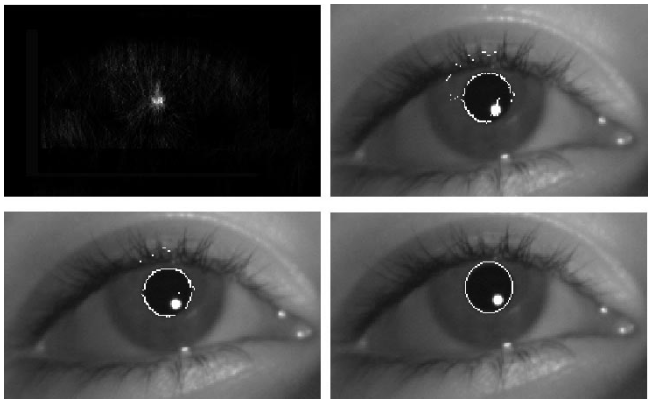


Fig. 15. Intermediate steps in matching a circle to the border of the pupil: accumulator array, detected contour points after the first iteration, detected contour points in the second iteration, and the final result.

A circular ring with predefined radii $R_{\max}$ and $R_{\min}$ is formed around the estimated center point. Starting from the outer bounds of the circular ring, the position of the maximal gradient (within the limits of $R_{\max}$ and $R_{\min}$) is searched along the radius of the circle. Only gradients which deviate from the radius of the circle by no more than 45° are considered in this computational step. The maximal gradients selected in this way roughly outline the contour of the pupil. The center of the circle is subsequently redefined by applying a least square fitting technique to the determined contour points. Due to image noise, the estimated circle usually deviates from the true position of the pupil. Further improvements are achieved in an iterative process, where the search area for the contour points is repeatedly repositioned (according to the estimated center) and narrowed down by reducing the difference between $R_{\max}$ and $R_{\min}$. The final result is shown in Fig. 15.

Having located the pupil, the reflection of the light is easily detected since it forms a very bright region close to the pupil. Based on a simple threshold technique, the center of the largest connected, bright region is determined. The eye vector aiming from the center of the pupil at the center of the reflected light can now be calculated. After an individual calibration process, where the user will fixate a sequence of five calibration points shown on the display, the eye vector is scaled to determine the user's current gaze direction.

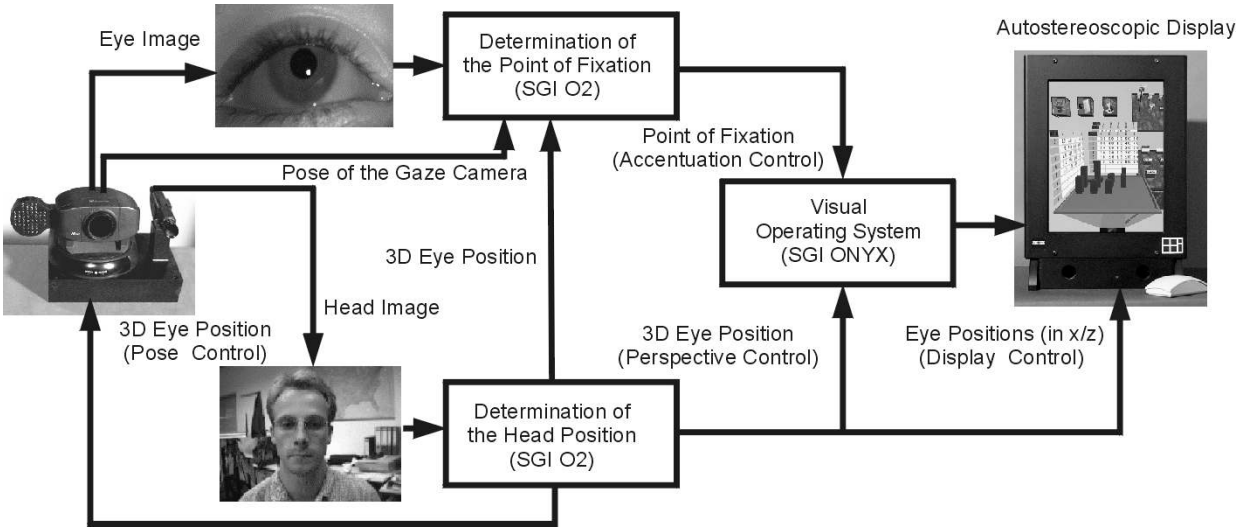


Fig. 16. Overall system diagram of the implemented testbed.

B. Compensation for Head Movements

The cameras used for head and gaze tracking in our prototype are based on commercial Sony video (conference) cameras and connected to the analog video inputs of separate computers connected via Ethernet. When the user moves, the gaze camera (which has an autofocus system) is tilted and panned in order to keep the eye at the center of the image. These movements cause changes in the eye vector and the calculated gaze direction, even when the absolute point of fixation remains unchanged. In order to ensure exact measurements at any head location, we have introduced a novel transformation technique using a head-fixed coordinate system that compensates for measurement deviations caused by head movements [25]. This way, the measurement accuracy is not affected by head movements and the pose changes of the gaze camera.

The point of fixation can be estimated with a precision of about 0.4° (at best). The accuracy was tested using still images of a human eye with simulated pupil and highlight positions. Additionally, we presented small targets on a 2-D monitor and asked subjects to fixate these targets as precisely as possible. In this case, measurement precision depended on the accuracy of the calibration process. The tracking speed of the gaze camera is limited by the speed of the pan-and-tilt unit and the autofocus system. After complete loss of the eye image, it takes less than 1 s to regain lock. A measurement rate of 25 Hz is achieved on a standard SGI O2 workstation without special DSP's.

VIII. Preliminary Results of User Testing

Ideally, the visual operating system and the interaction processes described in this paper should run on a single machine. In order to keep delays due to limitations in computing power to a minimum, the various processes were shared among a cluster of separate computers connected via Ethernet. The overall system diagram in Fig. 16 shows the basic components used in our setup.

Preliminary testing in the lab (using the applications described in Section IV) and at an exhibition indicated that most of the users were impressed by the 3-D presentation and the possibility to communicate with the computer by simply looking at the object of current interest or by changing the viewing position. Critical remarks concerned the design of the graphic elements (some were too small for easy gaze interaction in our initial setup) and the delay due to the various interprocess communications. Moreover, users frequently asked for a "park button" on the keyboard in order to anchor an application at a fixed position on the display. (Normally, the interface agent moves an application to the background in order to tidy up the display when the user visually scans the screen.)

Another point was the way noncommand actions were launched by the interface agent depending on how long the user was looking at a particular graphic object. In order to avoid visual stress immediate feedback of initiated actions seems to be indispensable. A fixed dwell time, however, does not seem to be the optimal solution since it appeared to be too long for certain applications and users and too short for others. Moreover, any action launched by the agent but not desired by the user (due to misinterpretation or when the user changed his/her mind) should preferably be reset by the agent. Such trivial undo could be based on monitoring the user's activities that follow immediately. (In our current system the interface agent acts on the basis of the sequence and duration of looking at particular objects or items, such as the docks of connectable applications.) Sensitive actions like the deletion of a file should rely on unmistakable confirmation by the user.

A critical issue in the concept of eye-controlled interactions is the fact that human eyes are normally used as input organs and not for manipulation tasks [10]. It was generally stated that intolerable eye strain occurred when the eyes were used for manipulating graphical objects, e.g., in a drag-and-drop operation. On the other hand, pointing operations were very easily performed by looking at the corresponding icons. If the icons are sufficiently large, eye pointing seems to be

significantly faster and easier than mouse pointing. Obviously, it would be preferable to combine gaze-controlled pointing with another modality (such as speech input) in order to specify the action desired [10].

Three-dimensional displays allow image presentation at a distance in front of the screen where the displayed objects are within reach and where stereo information (binocular disparities) outperforms any other depth cue [26]. Therefore, direct manipulation of a virtual 3-D object through hand gestures appears to be another useful interaction modality for a range of applications. On the other hand, the discrepancy between accommodation and convergence requires the positioning of the stereoscopic depth volume close to the screen plane when conventional 3-D displays are used. The depth-of-interest display outlined in this paper could help to overcome this problem, thus making direct hand manipulation of stereo objects a useful interaction modality.

IX. Conclusions

We have created an experimental multimedia system including a visual operating system running on a high-end graphic workstation that supports 3-D display and eye-controlled interactions without the user having to wear glasses or other encumbering devices. The system enables noncommand (visually controlled) interactions and will serve as a testbed for studying and evaluating new concepts for user interaction with a computer. The first prototype was recently demonstrated to the public. Our future work will focus on optimizing the overall system performance and on extending the system's functionality. Further interaction modalities, such as natural hand gestures and voice control, should be integrated into the user interface. On the other hand, customized and application-specific low-cost versions (e.g., a down-scaled version using eye-controlled interaction with traditional 2-D displays) should be regarded as a means of speeding up commercialization of the proposed system.

Acknowledgment

The authors gratefully acknowledge the cooperation of the 3-D Displays Group at HHI. The authors are exclusively responsible for the contents of this paper.

References

[1] A. Arditi, "Binocular vision," in *Handbook of Perception an Human Performance,* K. E. Boff, L. Kaufman, I. P. Thomas, Eds. New York: Wiley, 1986.

[2] C. Ware and G. Franck, "Evaluating stereo and motion cues for visualizing information nets in three dimensions," Univ. New Brunswick, Fredericton, N.B., Canada, Tech. Rep. TR94-082, Feb. 1994.

[3] S. K. Card, G. G. Robertson, and J. D. Mackinley, "The information visualizer, an information workspace," in *SIG CHI'91 Conf. Proc.,* 1991, pp. 181.

[4] G. G. Robertson, J. D. Mackinley, and S. K. Card, "Cone trees: Animated 3-D visualizations of hierarchical information," in *SIG CHI'91 Conf. Proc.,* 1991, pp. 189–194.

[5] R. A. Reaux and J. M. Carroll, "Human factors in information access of distributed systems," in *Handbook of Human Factors and Ergonomics,* 2nd ed., G. Salvendy, Ed. New York: Wiley, 1997.

[6] R. Skerjanc and S. Pastoor, "New generation of 3-D desktop computer interfaces, in *Stereoscopic Displays and Virtual Reality Systems, Proc. IS&T/SPIE EI'97 Conf.,* San Jose, CA, Feb. 8–14, 1997.

[7] B. Shneiderman, *Designing the User Interface—Strategies for Effective Human-Computer Interaction,* 2nd ed. Reading, MA: Addison-Wesley, 1992.

[8] G. R. McMillan, R. G. Eggleston, and T. R. Anderson, "Nonconventional controls," in *Handbook of Human Factors and Ergonomics,* 2nd ed., G. Salvendy, Ed. New York: Wiley, 1997.

[9] J. Nielsen, "Noncommand user interfaces," *Commun. ACM,* vol. 36, no. 4, pp. 83–99, 1993.

[10] A. J. Glenstrup and T. Engell-Nielsen, "Eye controlled media: Present and future state," Thesis, Univ. Copenhagen, Denmark, 1995.

[11] R. J. K. Jacob, "Eye movement-based human-computer interaction techniques: Toward noncommand interfaces," in *Advances in Human-Computer Interaction,* vol. 4. Norwood, NJ: Ablex, 1993.

[12] S. Pastoor and R. Skerjanc, "Autostereoscopic user-computer interface with visually controlled interactions," in *Dig. Tech. Papers, SID'97 Int. Symp.,* Boston, MA, 1997, pp. 277–280.

[13] M. Wöpking, "Viewing comfort with stereoscopic pictures: An experimental study on the subjective effects of disparity magnitude and depth of focus," *J. Soc. Inf. Displ.,* vol. 3, no. 3, pp. 101–103, 1995.

[14] W. Blohm, I. P. Beldie, K. Schenke, K. Fazel, and S. Pastoor, "Stereoscopic image representation with synthetic depth of field," *J. Soc. Inf. Displ.,* vol. 5, no. 3, pp. 307–313, 1997.

[15] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human computer interaction: A review," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 19, pp. 677–695, July 1997.

[16] S. Pastoor and M. Wöpking, "3-D displays: A review of current technologies," *Displays,* vol. 17, pp. 100–110, 1997.

[17] R. Börner, B. Duckstein, O. Machui, R. Röder, T. Sinnig, and T. Sikora, "A family of single-user autostereoscopic displays with head-tracking capabilities," submitted for publication.

[18] G. Boerger and S. Pastoor, "Autostereoskopisches Bildwiedergabegerät mit natürlicher Kopplung von Akkommodationsentfernung und Konvergenzentfernung und adaptiver Schärfentiefe (Autostereoscopic display providing a natural link of accommodation distance and convergence distance and adaptive depth-of-focus)," DE 195 37 499, patent pending, 1997.

[19] L.-P. Bala, K. Talmi, and J. Liu, "Automatic detection and tracking of faces and facial features in video sequences," in *Picture Coding Symp. 1997,* Berlin, Germany, Sept. 10–12, 1997.

[20] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey,"*Proc. IEEE,* vol. 83, 1995.

[21] H. Sako and A. V. W. Smith, "Real-time facial expression recognition based on features' positions and dimensions," in *Proc. ICPR,* 1996.

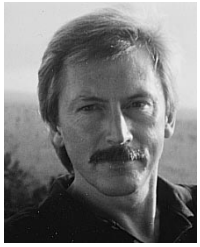
[22] N. Oliver and A. Pentland, "LAFTER: Lips and face real time tracker," in *IEEE Conf. Computer Vision Pattern Recognition,* San Juan, PR, June 17–19, 1997.

[23] L. Young and D. Sheena, "Methods & designs: Survey of eye movement recording methods," *Behav. Res. Meth. Instrum.,* vol. 7, no. 5, pp. 397–429, 1975.

[24] K. Talmi and J. Liu, "Eye and gaze tracking for visually controlled interactive stereoscopic displays," *Image Commun.,* 1998.

[25] J. Liu, "Determination of the point of fixation in a head-fixed coordinate system," in *14th Int. Conf. Pattern Recognition,* Brisbane, Australia, Aug. 16–20, 1998.

[26] S. Nagata, "How to reinforce perception of depth in single two-dimensional pictures," in *Proc. Soc. Inf. Displ.,* 1984, vol. 25, no. 3, pp. 239–246.



Siegmund Pastoor received the diploma and Ph.D. degrees in aerospace technology from the Technical University of Berlin, Berlin, Germany, in 1975 and 1980, respectively.

He was a Lecturer of courses on man–machine systems from 1979 to 1983. In 1980, he joined the Heinrich-Hertz-Institut Berlin (HHI). As a Leader of the Human-Factors Research Group, he has initiated and directed research activities at HHI in the fields of information display (character design, user guidance, use of color) and 3-D imaging (psycho-optical foundations of 3-DTV, autostereoscopic displays, image processing for multiview systems). He was a Visiting Scientist at NHK Labs, Tokyo, Japan, and has been involved in several European research activities. His present research interests focus on novel free-viewing 3-D display technologies and on nonconventional user interfaces for advanced multimedia applications.

Jin Liu received the B.S. degree from the University of Science and Technology of China in 1982, and the diploma and Ph.D. degrees from the Technical University of Berlin, Berlin, Germany, in 1985 and 1989, respectively.

She then joined the Heinrich-Hertz-Institut Berlin, where she has been involved in research projects related to 3-D techniques for television and video communications (3-D image analysis and synthesis, psycho-optical foundations of stereo image coding), and interactive multimedia services (image processing for multiview systems, autostereoscopic displays). Her current research interests focus on the development of an advanced interactive user interface for 3-D multimedia terminals (computer vision, image processing, and gaze-controlled interactive user interface).

Sylvain Renault received the diploma in computer science in 1997 from the Technical University of Berlin, Berlin, Germany.

He then joined the Human Factors Department of the Heinrich-Hertz-Institut Berlin, where he has developed and implemented an advanced interface concept for 3-D multimedia computers. His research interests include the integration of agents, graphic animations, and sounds in VR components and approaches to multimodal interaction. He has supervised the development of a new visual operating system (a full object-oriented API). He is privately interested in the whole domain of computer graphics, including digital image processing, web design and multimedia applications.