COMPENSATION OF TIME DELAYS IN TELEPRESENCE APPLICATIONS BY PHOTO-REALISTIC SCENE PREDICTION OF PARTIALLY UNKNOWN ENVIRONMENTS

C. EBERST, N. O. STÖFFLER, M. BARTH, and G. FÄRBER
Institute for Real-Time Computer Systems
Technische Universität München
D-80333 Munich, Germany
stoffler@rcs.ei.tum.de

ABSTRACT

A time delay in the visual feedback given to an operator, who is controlling a remote manipulator, noticeably impairs his performance. The use of predictive displays is an approach that has proven its suitability to compensate this effect. Most current implementations employ simple computer-graphics for the prediction and require a well known remote environment.

The presented work aims at increasing the overall immersion of the operator on the one hand and the flexibility of the system on the other hand. The applications should include maintenance and repair of machines located in remote plant- or office-type environments. These kind of environments are typically non-static and a priori only partially known. Communication media are high- and medium-bandwidth LANs and WANs inducing round trip delays varying from 0.2 to 2 seconds. Also a sufficient bandwidth cannot be assured during the entire remote operation.

The paper describes the concept of a predictive display system achieving photorealism based on autonomously explored and updated scene models. The model acquisition task is performed with a binocular video system. Photorealism is obtained by using computer graphics techniques, especially the mapping of original textures. Besides the concepts, an early stage of implementation and first experimental results are presented.

keywords: teleoperation, environmental modeling, predictive displays, surface reconstruction, computer graphics, texture mapping, image based rendering

1 INTRODUCTION

An old but still demanding problem in [8] are the delays between commanding an action and receiving the feedback via a communication channel. A delay on the visual feedback of 250 ms is recognized by a human operator, delays of about 1000 ms impair his performance tremendously. Beyond this delay time only the "move and wait" strategy [3] remains, complex manipulations become impossible. For coping with delays in the haptic feedback, the classical position-control can be replaced for example by shared compliance control [2, 13]. For the improvement of the visual feedback, several variations of predictive displays have been suggested, which superimpose simple [18] or complex [3, 14, 15] graphics on the camera images (augmented reality), or completely replace it by a synthetic image (virtual reality)[12]. Those predictions are only possible, if models of the geometry, kinematics and dynamics of the remote scene are locally available.

The efficiency of the operator also increases with the realism of the visual feedback. Simple visualizations still require training and do not support an easy estimation of depth. Stereo displays contribute a depth information for close objects. To make the estimation of longer distances possible, additional depth cues are required, such as illumination, shading and motion parallax.

One of the computer-graphics' key-technology to the synthesis of photo-realistic images is texture mapping[7, 11]. Real-time capabilities are obtained by efficient methods for the preprocessing of textures such as *summed area tables* [6] and *mip maps* [21], optimized architectures for texture memories [20] and hardware-based trilinear filters [1].

Autonomous, vision-based acquisition of scene-models is also a very active field of research. Related work includes the video-based recognition and registration of known objects [16, 19] and the reconstruction of an unknown background [4].

The presented work combines the concept of predictive displays with the texture mapping technique to obtain photorealism and with computer vision methods to acquire the scene models. The system is designed for partially unknown environments incorporating a priori knowledge with on-line exploration.

Section 2 introduces the overall concept and the system structure. The acquisition of the scene-models is explained in section 3, section 4 describes the prediction part. The paper closes with first results in section

5 and a conclusion in section 6.

2 CONCEPT AND SYSTEM OVERVIEW

Figure 1 depicts the structure of a teleoperation scenario incorporating the predictive display. The actions of the human operator, such as motions of its hand and head are measured by pose-tracking devices and a data-glove.

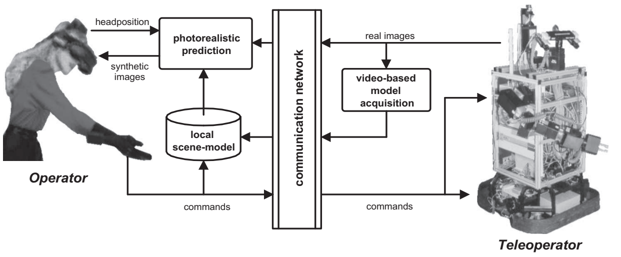

Figure 1: Teleoperation scenario including a predictive display.

These actions are mapped on commands that are transmitted over the communication network and executed by the robot. This first transmission induces a delay between the generation of a command and its execution due to the latency of the network. The transmission time of the command itself due to bandwidth limitations can be neglected, as commands typically consist of very small sets of data. The video images showing the execution of the commands then are sent back over the network to the operator. This second transmission again adds a delay time, consisting of the latency of the network and the transmission time of the image data.

In order to have immediate visual feedback the local scene model is continuously modified according to the commands. Based on this model a synthetic image is generated and presented to the operator via a head-mounted display (HMD) with virtually no time delay. In parallel, the scene model is updated by the sensor information acquired by the teleoperator.

Photorealism is accomplished by using the delayed real images for the generation of the textures in the predicted images. To keep the prediction as accurate as possible and cope with changes of illumination in the remote scene, the textures also have to be continuously updated and verified.

3 ACQUISITION OF THE SCENE MODEL

For the generation and update of the scene model the teleoperator is equipped with a binocular video system which is also employed to send the images to the operator. The model generation is divided into the recognition and registration of known objects and the reconstruction of a description of the unknown background. Typically, known objects include the object that has to be manipulated and the manipulator itself. While the localization of the object must be precise enough for manipulation, the description of the background is only employed to support the operators overall orientation.

Both image interpretation tasks are currently using preprocessed images, i.e. lists of line segments. Line segmentation is accomplished by a fast contour following algorithm [17], which is cost efficient by focusing all processing on regions in the images that contains significant grey level gradients.

3.1 OBJECT RECOGNITION

The recognition and registration of known objects is performed by modules which were developed in former projects concerning autonomous robots. Line-based recognition and localization of objects uses the system *MORAL* (*Munich object Recognition and Localization*) [16]. It closely interacts with the modeling software *GEM* (*Generalized Environmental model*) [10]. The descriptions of a priori known object are provided by the class layer of GEM. Recognized objects are fed back to the object layer as an instance of their class. Also articulated objects are recognized and their state can be determined. Details can be found in [9, 10, 16].

An approach for the reconstruction of three dimensional polygons, which can be fed to the background layer of GEM, is presented in the next chapter.

3.2 BACKGROUND RECONSTRUCTION

The reconstruction of the background is based on a multi stage surface recognition from sequences of stereo video images (see figure 2).

First, line segments are extracted by an edge detector. Detected line segments are corrected for radial distortions and deviations of the focal point of each camera. In order to form a virtually parallel oriented camera pair, the extracted line segments are shifted and rotated according to the deviations in the external camera parameters pan, tilt, and roll, and according to the influence of these parameters on the non-centric focal point of the image [5]. Figure 2 (left) shows these corrections. The resulting line segment description is normalized by the pixel size and the focal length. This correction simplifies further application of matching constraints and the reconstruction. Very short line segments are then rejected and cluttered ones are merged before processing.

Second, polygon chains in the images are determined by searching vertices, T-junctions and elongations in line segment constellations. Therefore line segments are compared whether they are lying close in the image plane. If several of these line couples have one segment in common, the most convex combination is chosen, in order to focus on simple bordered planar

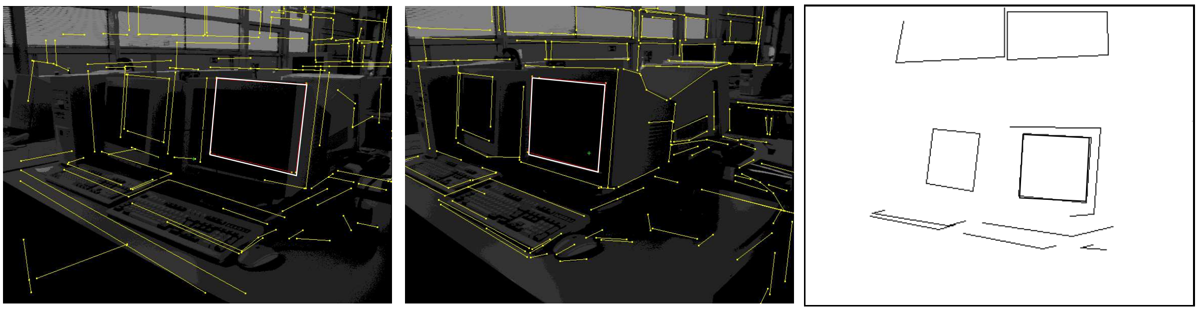

Figure 2: Polygon-chain-based surface reconstruction. The left and middle images show the binocular sensor view and the preselected line segments. The right image shows the reconstructed surfaces after one sensor reading. A double of matching 2D polygons and the boundary description of the reconstructed 3D surface are highlighted

surfaces. The remaining couples are then successively combined to more complex polygon chains and sorted counterclockwise. This procedure simplifies the following matching of the resulting convex polygon chains of both images. Next, the virtual intersection points of the polygons' line segments are computed. All types of 2D intersection points are termed *junctions* in the following.

Third, the polygon chains of both images are matched against each other. The determination of correspondences applies multiple cues in order to increase the robustness and to restrict the influence of threshold tuning. Line-segment-based and junction-based cues are thereby applied. As there may be differences of how much of the boundary of a surface is seen in the left and right image, the number of junctions that can be assigned depends on the completeness of the two polygon chains. To determine correct assignments, a feature with approximately equal length and angle in the left and right polygon chain is searched. To evaluate the correspondence between two preselected polygon chains, the following cues are applied. The first constraint to be fulfilled by a pair of possibly matching junctions is that the horizontal disparity is higher than a minimum positive value which corresponds to the maximum range that the sensor system can reliably resolve. Then, the slopes of the two line segments at each possibly matching junction of both polygon chains are compared in two ways: the deviation of the slope must be below threshold ($(\zeta_l - \zeta_r) \leq \epsilon$), and the slopes must be consistent with the epipolar constraints, i.e. the disparity must be positive also for the distant parts of the line segments. The later test is only applied if the slope itself is above threshold, to avoid that noisy line segmentation leads to erroneous rejection of matching couples. Next, the epipolar constraint is evaluated for possibly matching junctions. Due to the precorrection of line segments according to a virtually parallel camera arrangement, the test simply evaluates the vertical position of the polygons' junctions in the images. If two polygon chains are not completely matching, its subparts are tested for correspondences. This allows the determination of surfaces in the presence of noise, failed detection, and occlusion. In this case the boundary lines of the resulting surface are not completely specified.

Fourth, for all matching junctions of the two polygons, the 3D information is recovered. The face parameters are reconstructed in two redundant ways. The first method calculates the coordinates of the vertices (X, Y, Z) of a face using equation 1 from [5] (with $x_{l,r}, y_{l,r}$ being the coordinates in the image planes and $B_l - B_r$ being the baseline's length of the stereo system).

$$Z = \frac{B_l + B_r}{x_l - x_r}; \quad Y = B_l - (x_l \cdot Z); \quad X = -y_l \cdot Z; \quad (1)$$

By connecting the reconstructed 3D points, the nonplanar boundary description of the polygon is obtained. The face orientation is calculated for each enclosed vertex by the cross product of the connecting 3D lines. All edges that show a consistent depth and orientation are fused into one 3D polygon hypothesis. Please note that this method cannot identify apparent polygons that arise from occluding constellations of independent faces.

Redundant to this surface determination, the orientations of the 3D boundary lines $(\Delta X, \Delta Y, \Delta Z)$ are determined from the orientation of the 2D line segments ($\delta x_{l,r}, \delta y_{l,r}$) in the image for each vertex of the face according to equation 2. Application of this method is limited on line segments that are not horizontal, i.e. not parallel to the epipolar lines.

$$\begin{aligned} \Delta Z &= \tfrac{(B_l+B_r)\cdot(\delta x_r - \delta x_l)}{(x_l - x_r)\cdot((x_l+\delta x_l)-(x_r+\delta x_r))}; \\ \Delta Y &= -(\delta x_l \cdot Z) - (x_l \cdot \Delta Z) - (\delta x_l \cdot \Delta Z); \\ \Delta X &= -(\Delta Z \cdot \delta y_l) - (Z \cdot \delta y_l) - (y_l \cdot \Delta Z); \end{aligned} \quad (2)$$

Due to the higher accuracy of the virtual intersection points, the point to point method usually yields more precise results than the line-based method, but is only applicable on restricted aspects of non-closed polygon chains. In case that both methods can be applied, the redundancy is exploited to detect apparent polygons. Deviation in the depth estimation indicates that a surface is hidden by another one. If the hiding face and the hidden face can be identified by depth cues of adjacent 3D polygons or of the 3D polygon itself, the two resulting 3D polygons are reconstructed separately. Otherwise, the polygon is split into two or more overlapping ones. The plausibility of these polygons is set below a threshold. The faces that are reconstructed based on these polygons are eventually confirmed or ruled out by fusion with faces that are reconstructed from a different point of view, thus reducing the influence of occlusions.

Next, planar faces are obtained by fusing depth and orientation cues of both reconstruction methods for each enclosed, non-contradictive vertex and line segment of the 3D polygons. The planar boundary description is obtained by projecting the polygon onto the newly reconstructed face.

Finally, surface hypotheses are fused versus image sequences in order to improve the accuracy of the surfaces' description and to remove incorrect plane hypotheses from ambiguous or wrong correspondences by applying rule-out mechanisms on inconsistent arrangements of surfaces. Faces that are to be fused are identified by evaluating the normal vectors of the faces, their normal distance, and their overlapping regions.

4 SCENE PREDICTION

For optimal performance of the operator the prediction of synthetic images should run continuously at video frame rate. For the rendering of the next predictive image I_p the current commands issued by the operator and the delayed real image I_r can be taken into account.

In a first step, the state of the local scene model at time t_p has to be estimated. This is achieved by continuously modifying the model according to the issued commands: Movements of the hand-controller change the position of the manipulator, motion of the head affects the point of view on the model.

In the next step the image I_p is rendered by feeding the polygons of the scene model into a standard computer graphics pipeline, which does the coordinate system transformations, visibility calculations and scan line conversions.

Photorealism is accomplished by texture mapping.

4.1 TEXTURE MAPPING

A texture can be regarded an image, which is orthogonally projected onto a polygon. The pixels of this image are called texels. Each vertex (X, Y, Z) of a polygon is augmented by the texture coordinates (u, v) that define its position in the texture image, thus its position in texels. In the case of a rectangular polygon and a texture image of the size $w \times h$ that is covering the polygon completely, the four texture coordinates would be

$$(u, v)_{i, i \in \{0 \ldots 3\}} \in \{(0,0), (0,h), (w,h), (0,h)\} .$$

During the rendering of the polygon its vertices are projected onto the image plane. The resulting image coordinates for each vertex are $(x, y)_i$. For all positions (x, y) inside the polygon, the corresponding texel coordinates

$$(u, v) \rightarrow (x, y)$$

can be calculated, either by inverting the projection or as an approximation, by linear interpolation. For the actual rendering we use OpenGL and a hardware accelerator to achieve satisfying frame rates. Polygons are handed to the GL rendering engine together with their texture coordinates and a texture image. Current implementations vary in how many texels around (u, v) are taken into consideration for the calculation of the color of pixel (x, y) and what kind of filtering is done. All techniques supported by todays hardware are approximations to the actual, perspective mapping of the texels, but in practice, the results are sufficient.

4.2 TEXTURE EXTRACTION

In standard computer graphic applications, the texture images are assigned to their polygons during the building of the model, thus are known a priori. In our case, those textures have to be reconstructed from the delayed real images I_r.

For this purpose, a second model prediction, reflecting the state at t_r is generated. The coordinates of the projected vertices then match the corners of the corresponding faces in the image I_r. By inverting the texture mapping to

$$(x, y) \rightarrow (u, v)$$

the color of each texel could be calculated by filtering the pixels around (x, y). When doing this texture extraction continuously for each transmitted real image, not only spatial filtering but also temporal filtering has to be applied. A strategy for replacing old texture information by more current one is under development. At the moment, we only consider the most recent real image for texture extraction. The drawback of this memory-less approach is of course, that no texture information is available for polygons, which were hidden in this single image.

But as a big advantage on the other hand, the two projection steps

$$(x, y)_r \rightarrow (u, v) \rightarrow (x, y)_p$$

can be combined to one single projection. The real image itself is used as texture image and $(x, y)_r \equiv (u, v)$. No further storage or management of the textures is necessary, the single image is just passed to the renderer together with all polygons and their texture coordinates.

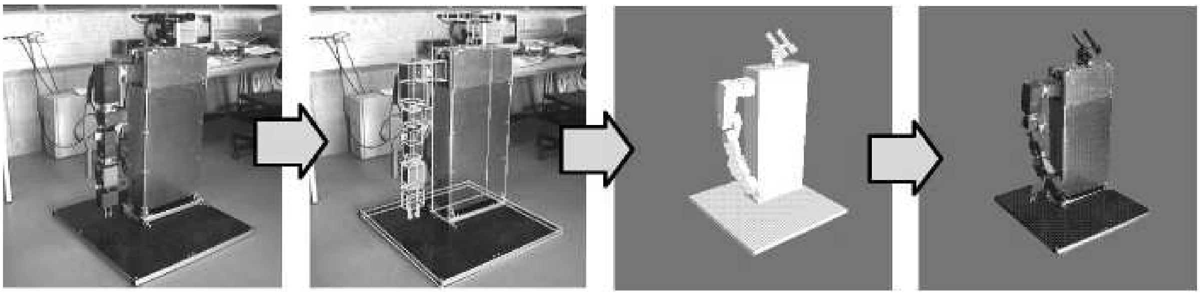

Figure 3: Model-update by object recognition. From left to right: image, superimposed (recognized) object, predicted scene based on the movements of the operator, predicted scene with mapped textures.

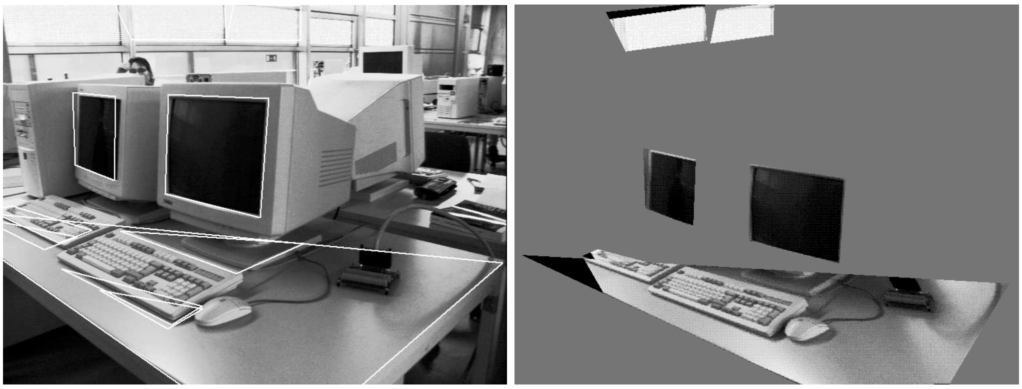

Figure 4: Model-update by background reconstruction. Left: real image with overlayed polygons for the extraction of textures. Right: synthetic image from different viewpoint (note the black parts of some faces where no textures from the real image were available).

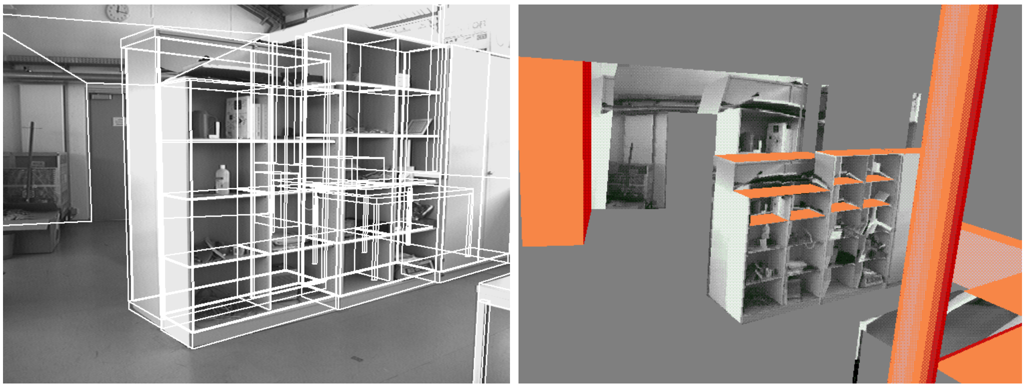

Figure 5: Model-update by registration of an a priory known background model. Left: real image with overlayed polygons for the extraction of textures. Right: synthetic image

5 RESULTS

Up to now, the system has been tested without HMD and without magnetic trackers. For the experiments motions of head and arm have been controlled by a space-mouse.

The exemplaric scene predictions in figure 3 - 5 are already showing the feasibility of the approach and the gain of realism by the use of texture mapping.

While the model acquisition is still tested off-line, the prediction part has already a first closed-loop implementation using a lowest-cost "Voodoo"-graphics accelerator with very limited texture resolution. As the results look promising, we are currently porting the system to high-end graphics hardware.

A major advantage of the texture mapping besides providing the details of the scene that are required for efficient teleoperation, is the effective concealment of inaccuracies. In the above experiments, the pose estimation yielded an average error in the magnitude of $1-3cm$. This accuracy would not permit autonomous manipulation tasks. But it showed to be sufficient for realistic image generation, as long as the perspective of the synthetic image is similar to that of a real one.

6 CONCLUSION AND FUTURE WORK

This paper described the concepts and early experiments for the compensation of time delays on the visual channel by *photo-realistic* predictions of *partially unknown* scenes.

Future work will focus on improvements in the scene reconstruction to generate dense background descriptions by integration of complementary image processing techniques and incorporation of longer image sequences.

In addition, a more elaborate implementation for extraction and storage of textures is under development, which will be able to handle occluded and distorted textures appropriately. Further experiments will include the installation of a HMD and magnetic pose trackers. The manipulator that is currently connected to a steady base will be mounted on a mobile platform.

ACKNOWLEDGMENT

The work presented in this paper was supported by the, *Deutsche Forschungsgemeinschaft* as a part of the interdisciplinary research project “High-Fidelity Telepresence and Teleaction” (SFB453). Special thanks deserves Darius Burschka for providing a framework of a line-based video-stereo reconstruction on which the surface-detection is based on.

References

[1] K. Akeley. RealityEngine Graphics. In *SIGGRAPH 93 Conf. Proc.*, volume 27, pages 109–116, August 1993.

[2] Robert J. Anderson and Mark W. Spong. Bilateral Control of Teleoperators with Time Delay. *IEEE Transactions on Automatic Control*, 34(5):494–501, May 1989.

[3] Antal K. Bejczy, Won S. Kim, and Steven C. Venema. The Phantom Robot: Predictive Displays for Teleoperation with Time Delay. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'90)*, pages 546–551, Cincinatti, Ohio, 1990.

[4] C. G. Bräutigam, J. Gaerding, and J-O. Eklundh. Seeing the obvious. Technical report, Computational Vison and Active Perception Lab (CVAP), KTH, 1996.

[5] Darius Burschka. *Videobasierte Umgebungsexploration am Beispiel eines binokularen Stereo-Kamerasystems.* PhD thesis, TU München, Fakultät für Elektro- und Informationstechnik, 1998.

[6] F. C. Crow. Summed-Area Tables for Texture Mapping. In *SIGGRAPH 84 Conf. Proc.*, volume 18, pages 207–212, July 1984.

[7] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and Rendering Architecture from Photographs. In *SIGGRAPH 96 Conf. Proc.*, August 1996.

[8] W. R. Ferrel. Remote Manipulation with Transmission Delay. *IEEE Trans. on Human Factors in Electronics*, 6:24–32, September 1965.

[9] A. Hauck, S. Lanser, and C. Zierl. Hierarchical Recognition of Articulated Objects from Single Perspective Views. In *Proc. Computer Vision and Pattern Recognition (CVPR'97)*, pages 870–883. IEEE Computer Society Press, 1997.

[10] A. Hauck and N. O. Stöffler. A Hierarchical World Model with Sensor- and Task-Specific Features. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'96)*, pages 1614–1621, 1996.

[11] Paul S. Heckbert. Survey of Texture Mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, November 1986.

[12] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. ROTEX - The First Remotely Controlled Robot in Space. In Edna Straub and Regina Spencer Sipple, editors, *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'94)*, volume 3, pages 2604–2611, Los Alamitos, CA, USA, May 1994. IEEE Computer Society Press.

[13] W. S. Kim, B. Hannaford, and A. K. Bejczy. Shared Compliance Control for Time-Delayed Telemanipulation. In *1st Int. Symp. on Measurement and Control in Robotics*, NASA JSC, Houston, Texas, June 1990.

[14] Won S. Kim. Virtual Reality Calibration and Preview/Predictive Displays for Telerobotics. *Presence*, 5(2):173–189, 1996.

[15] Won S. Kim, Donald B. Gennery, Eugene C. Chalfant, Lucien Q. Junkin, Ivan M. Spain, and Suzie B. Rogers. Calibrated Synthetic Viewing. In *Proc. of the ANS 7th Topical Meetin on Robotics and Remote Systems*, volume 1, pages 596–602, Augusta, Georgia, April 1997. American Nuclear Society.

[16] S. Lanser, C. Zierl, O. Munkelt, and B. Radig. MORAL – A Vision-based Object Recogntion System for Autonomous Mobile Systems. In *Computer Analysis of Images and Patterns (CAIP)*, number 1296 in Lecture Notes in Computer Science, pages 33–41. Springer-Verlag, 1997.

[17] G. Magin and C. Robl. A Single Processor Real-Time Edge-Line Extraction System for Feature Tracking. In *IAPR Workshop on Machine Vision Applications (IAPR MVA'96)*, 1996.

[18] M. V. Noyes and T. B. Sheridan. A Novel Predictor for Telemanipulation Through a Time Delay. In *Proc. of the 20th Annual Conf. on Manual Control*, NASA Ames Research Center, Moffet Field, CA, 1984.

[19] A.R. Pope. Model-Based Object Recognition - A Survey of Recent Research. Technical Report TR-94-04, University of British Columbia, January 1994.

[20] Andreas Schilling, Günter Knittel, and Wolfgang Strasser. Texram: A Smart Memory for Texturing. *IEEE Computer Graphics and Applications*, 16(3):32–41, May 1996.

[21] L. Williams. Pyramidal Parametrics. In *SIGGRAPH 83 Conf. Proc.*, volume 17, pages 1–11, July 1983.