The Hybrid Display: Smooth Transition between Virtual and Real Views for Location Unbound Observations in Telepresence Applications

Jan Leupold* and Stephan Behrendt**
Institute for Real-Time Computer Systems (RCS), Technische Universität München
D-80290 Munich, Germany
*Email: jan.leupold@rcs.ei.tum.de
**Email: stephan.behrendt@rcs.ei.tum.de

Abstract. The most common solution to display the remote scene of a telepresence scenario is to transmit images captured by cameras located at the remote scene and to display them to the operator. Thus only observations from a set of concrete positions are available. In order to generate views from arbitrary view points, a texture mapped 3D representation is used to generate photo-realistic views. Textures are extracted directly from received camera images, while the 3D model is reconstructed by stereo vision algorithms. These two processes introduce inevitable delays which decrease the perceived quality of the system. As an improvement the display of (location bound, up-to-date) camera images is combined with (location unbound, delayed) texture mapped 3D representations. Several strategies for switching between both view types are presented and are currently evaluated for acceptability by human operators.
Keywords. *Telepresence, Computer Graphics, Viewpoint Synthesis, Projection, Texture Mapping, Blending*

I. INTRODUCTION

In most telepresence systems a visual representation of the remote teleoperator scene is presented to the human operator. Immersion into the teleoperator environment increases if the operator's head movements directly lead to changes in the perspective of the presented view, as if the operator performs the task himself.

Image information for this representation is usually obtained by cameras located at the teleoperator side. The easiest approach to implement a visual display is to continuously move a camera to a location related to the current operator head position and to display the captured video stream. The main problem of this solution is that in most setups camera movements are restricted to less than six degrees of freedom (DOF). Nevertheless the presented video reflects the teleoperator environment with a minimal time lag introduced by the transmission of the video stream to the operator.

In a former research project [3] a predictive display was developed. A 3D polygon representation of the teleoperator side is reconstructed by means of stereo vision. This representation is rendered and photo-realistic display quality is achieved by texture mapping. The textures are extracted and updated directly from camera images captured at the teleoperator side. High immersion is achieved if the model reconstruction and texture extraction process is fast enough to follow changes in the teleoperator scene. For fast changing environments this approach tends to display delayed model and texture information to the operator.

In this paper a system is presented which combines up-to-date information from captured video streams with the capability to freely move around in the teleoperator workspace (IV). This is achieved by automatically switching and combining both display variants. Further contributions of this paper are to show how to:

- display video streams to the operator in 3D (V),
- decide when to switch (VI),
- arrange the transition between display variants (VII), and
- how to mark which display variant is used (VIII).

The following two sections discuss related work (II) and define some commonly used terms (III). A conclusion and future work is presented in section IX.

II. RELATED WORK

To simulate camera panning and zooming, 360-degree cylindrical panoramic images are well-known. These images are warped on-the-fly on the rectangular screen of the observer. Translations are restricted to a discrete set of camera positions [4]. In the context of telepresence, this approach is used in [2]. Only the visible image region is transmitted to provide higher refresh rates. Yamazawa et al. present a networked variation of this technique where multiple users can access the captured video streams [11].

The idea to mix up rendered geometry with captured images is discussed by Hitchner [7]. Either a six degree of freedom 3D environment mode or a two degree of freedom 2D panorama mode can be selected. The combined use of both modes is considered but not further developed.

Dai combines augmented 2D images and rendered 3D models, but on different displays [5]. While the 3D display is used for orientation in the scene, the augmented 2D image stream is the preferred display for executing telepresence tasks.

Image based rendering methods show promising results but are not always viable for the proposed purposes. Rademacher and Bishop [9] use a strip camera to construct a multiple-center-of-projection image that can be used to render a scene from arbitrary view points. As many images from registered locations are needed as input data and the scene must remain static for every update, this approach was not considered here. In [10] layered depth images are used to allow arbitrary view points within a certain range. Precise and dense depth information is elementary for layered depth images. Shade et al. suggest computer vision techniques for the use of real images, which lead to comparable delays as in [8].

III. TERMS AND DEFINITIONS

Two 6-DOF positions (extrinsic parameters) are regarded in the following: the *camera pose* P_c and the *operator pose* P_o. P_c describes the position of the camera in the teleoperator scene, while P_o describes the observation location requested by the operator. The generated images for the operator can be interpreted as images captured by a *virtual camera* located at the operator pose. For each camera the corresponding projection properties, i.e. their intrinsic parameters (pinhole camera model), are named p_c and p_o. The physical display device in front of the operator's eyes is called the *operator screen*.

Views that are generated using video frames captured by the cameras at the teleoperator side will be referred as *video views*. *Synthetic views* are those that are generated by rendering the textured 3D model of the teleoperator scene. *Hybrid views* combine video and synthetic views to display data on the operator screen.

IV. OVERALL SYSTEM DESCRIPTION

The overall system is designed for a typical telepresence scenario where a human operator controls a remote robotic manipulator (teleoperator) to execute a certain task, like e.g. a pick-and-place manipulation. Several cameras observe the robot's workspace.

The scene is displayed with a fixed frame rate on the operator screen, which in our scenario is a head-mounted display (HMD) with known intrinsic parameters p_o. The operator's head position is tracked in 6 DOF and head movements directly lead to movements of P_o. For each frame either a video, synthetic or hybrid view must be generated. In the latter case a synthetic view is used to fill those parts of the operator screen that are not covered by the video view.

Further data needed to generate views is received over the network: the video streams, information about camera poses P_c and their intrinsic parameters p_c.

Synthetic views are generated by drawing a polygonal 3D model of the remote teleoperator scene. The model contains a-priori known geometry of the teleoperator and is periodically completed using stereo vision algorithms. The rendering algorithm is a standard implementation drawing texture mapped polygons. To achieve photorealistic quality, the polygon textures are extracted directly from received camera images. Reconstruction and texture extraction cannot be executed fast enough to process every captured video frame, thus changes in the teleoperator scene are displayed with noticeable delays. Further details are given in [3, 8].

V. VIDEO VIEWS

A video view displays video frames that are captured by cameras at the teleoperator side and transmitted to the operator side. In order to consider p_c, P_c and p_o, P_o, which are in general different, the received video frames cannot be directly displayed on the operator screen.

The proposed rendering algorithm draws a quad with the video frame as texture. The quad lies in a plane parallel to the real camera image plane at distance d (projection plane). Parameter d is discussed later. The size of the quad is limited by the real camera viewing frustum.

In order to optimize runtime performance only the visible pixel data of the video frame is transferred from main memory to the graphics card for display. This region is calculated by first intersecting the four viewing frustum edges of the virtual camera with the projection plane (refer also to figure 1). The output, which is a new quad, is clipped to the viewing frustum of the real camera. The bounding box of the resulting polygon in real camera image coordinates determines which region of the texture has to be updated.

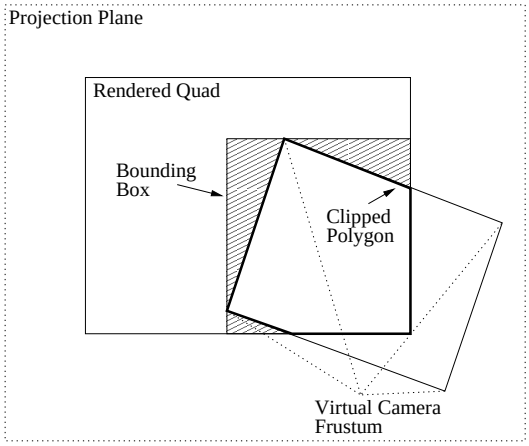

Fig.1: Update region calculation, seen from P_c

A. Rotation Only Pose Difference

If there is no translation difference between virtual and real camera pose, then a rotation of the virtual camera can be easily simulated. This is very similar to panorama image viewer applications, like e.g. the Apple QuickTime

VR Player [4]. The visual quality of the generated view is reduced by aliasing effects and incomplete coverage. Aliasing occurs if one video frame pixel is reprojected to many operator screen pixels. This effect increases e.g. for a small virtual camera field-of-view (FOV) in relation to the real camera FOV. Incomplete coverage is caused by high rotation angles that cause the rendered quad boundaries to become visible. As result only a part of the operator screen can be filled with data from the received video frame.

B. 6-DOF Pose Difference

If the pose difference also contains a translation, the visible region can still be calculated, but errors are introduced since depth information is not available for reprojection. This *reprojection error* $E(p_c, p_o, P_c, P_o, d, X)$ can be calculated for every 3D point X. As can be seen in figure 2, X is expected to be projected onto the virtual camera image plane at X'. But as the video frame is captured with the real camera, X gets projected to X'' and then to X'''. Therefore E can be calculated as the euclidean distance between X' and X''' (in pixels of the operator screen). Note that all points X that lie on the plane at distance d are projected correctly.

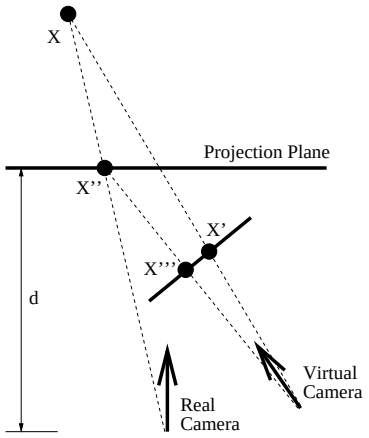

Fig.2: Error calculation for general pose difference.

Figure 3 shows an example error distribution for d=0.5m over an area of 1 by 1 meter. All points visible for the real camera are indicated by white pixels. The virtual camera is translated 0.02m and twisted 10 degrees relative to the real camera. All points visible for the virtual camera are drawn with colors representing E. Near the virtual camera errors raise up to 200 pixels (violet). At the projection distance d errors decrease to zero (red) and raise again up to 20 pixels (turquoise) at a distance of 1m. The real camera was chosen to have a high resolution of 1600 by 1200 pixels, the virtual camera has 800 by 600 pixels. This example shows that even small translations can lead to high values of E.

Aliasing is minimized for those values of d that result

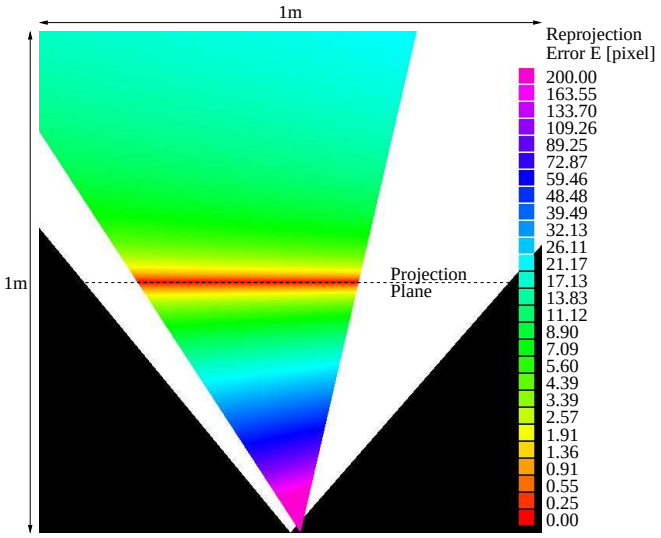

Fig.3: Exemplary distribution of E (top view).

in larger visible regions, but when regarding the time needed to transfer video frame data from main memory to the graphics card, small regions should be used. Further optimizations of d would include the dimension and location of the actual robot work space. For the sake of simplicity we set d to an experimentally found fixed value for our scenario.

VI. VIDEO USABILITY

For each frame that is displayed to the operator a decision has to be taken to either generate a video, synthetic or hybrid view. As a video view has a lower delay to display information, it is preferred. But as described in section V, several limitations occur that degrade the quality of a video view. Incomplete coverage can be compensated by using a hybrid view, but the reprojection error E can only be minimized along the projection plane at distance d. It noticeably decreases the perceived visual quality as the workspace is in general not a flat scenario. For usual intrinsic parameters, aliasing turned out to have almost no effect on the visual quality compared to E. Hence it is not further considered here.

In order to control the transition from video views to hybrid and then to synthetic views a measure is needed. This video error measure $m(p_c, p_o, P_c, P_o, d, W)$ is calculated as the ratio:

$$m = \frac{E_{max}}{C}$$

where C is the coverage and E_{max} is the maximum reprojection error in a volume W in front of the virtual camera. The observation space W is the virtual camera viewing frustum clipped to the teleoperator workspace (see figure 4). The maximum value of E within W is approximated as the maximum value E_{max} of all vertices W_i of W. Coverage C is the percentage of operator screen

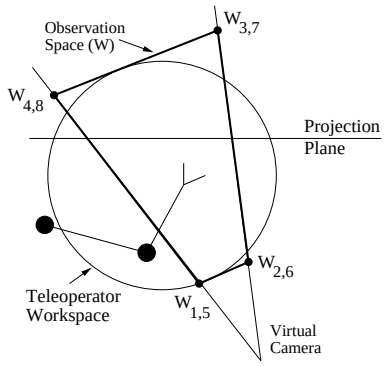

Fig.4: Teleoperator workspace and observation space.

pixels that are filled with data from the video view. By defining a threshold value m_t, m can be used to control the transition (see section VII). This threshold depends on human perception.

VII. DESIGN OF VIEW TRANSITION

When the video error measure exceeds threshold m_t, the presented view should switch from video view to synthetic view. This transition process influences the immersion perceived by the operator.

Three different approaches were implemented:

- Abrupt transition.
- Blending between views with a global blending parameter for each frame depending on m.
- Blending between views depending on m and on pixel position.

Obviously the simplest transition between the two views is the first method. If m_t is reached, the synthetic view is shown instead of the video view and vice versa. To avoid toggling near m_t, a hysteresis is implemented (see figure 5, dashed curve). The observer immediately recognizes the transition. As a positive effect, specific marking of the synthetic view does not need to be as intensive as in the following two cases (see also section VIII). On the other hand this method leads to a lowered immersion. In figure 6 left, discontinuities at the background objects are clearly visible, which distract even stronger during movements.

The second method tries to make the transition process smoother thus increasing the immersion. First the complete synthetic scene is drawn. Then if m is in an interval $\pm\epsilon$ around m_t the video view is blended over the synthetic view with a constant blending factor w_1.

$$w_1 = \frac{m - m_t + \epsilon}{2\epsilon};\ w_1 \in [0.0, 1.0]$$

Just like m_t, ϵ has to be determined considering human perception. The blended image is calculated as

$$I = w_1 I_{video} + (1 - w_1) I_{synth}$$

where I_{video} corresponds to the video frame and I_{synth} to the rendered synthetic image. Figure 5 illustrates the difference to the first method.

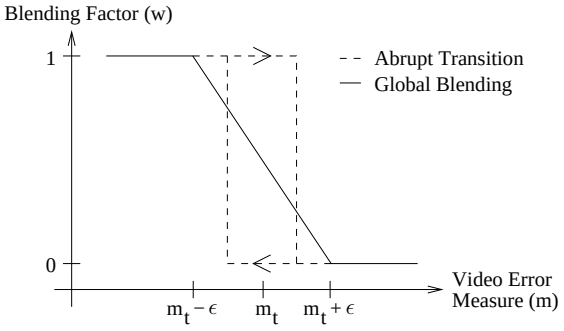

Fig.5: Transition options between views

Through blending the transition becomes smoother and the observer's immersion is improved. The problem is that differences between video and synthetic view result in clearly visible and possibly annoying artifacts (see figure 6 middle) over the whole video view. These differences result from reprojection errors E and unprecise model information. This negative effect is especially annoying as the appearance of these artifacts depends on P_o.

In order to minimize them a third method was developed. The same rendering algorithm as for the second method is used, but the blending weight w_2 now depends on the pixel position. Only pixels in a border region of the video frame are blended. Video frame pixels that are not in this border region completely replace synthetic view pixels.

To describe the border κ is introduced: $\kappa = 1 - w_1$. It represents the border width in relation to the video frame size (x_{max}, y_{max}). Then the border widths (in pixels) are:

$$x_b = \kappa \frac{x_{max}}{2}$$

$$y_b = \kappa \frac{y_{max}}{2}$$

A preliminary blending weight w_2' can be calculated for each video frame pixel (x,y) as:

$$w_2' = \begin{cases} 1.0 & : \kappa = 0 \\ & x' \in [0, 1-\kappa] \ \wedge y' \in [0, 1-\kappa] \\ \frac{1-x'}{\kappa} & : y' \in [0, 1-\kappa[\ \wedge x' \in]1-\kappa, 1] \\ \frac{1-y'}{\kappa} & : x' \in [0, 1-\kappa[\ \wedge y' \in]1-\kappa, 1] \\ w_{edge} & : x' \in]1-\kappa, 1] \ \wedge y' \in]1-\kappa, 1] \end{cases}$$

$$w_{edge} = 1 - \frac{1}{\kappa}\sqrt{(\kappa - 1 - x')^2 + (\kappa - 1 - y')^2}$$

$$x' = |1 - \frac{2x}{x_{max}}|$$

$$y' = |1 - \frac{2y}{y_{max}}|$$

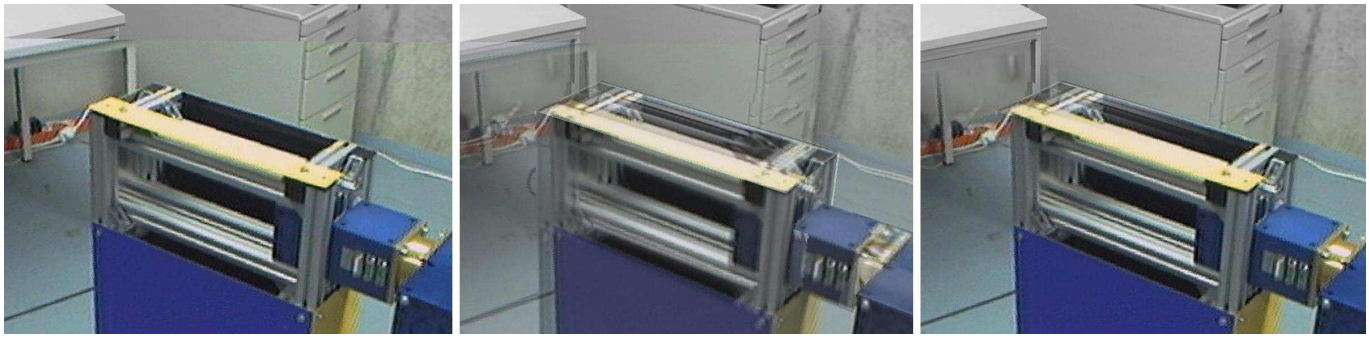

Fig.6: Hybrid View, synthetic part in grayscale. Left: Abrupt transition, Middle: Global blended images with artifacts, Right: Pixel-based blending

To avoid a perceptional phenomena called Mach-Band effect, w_2 should not be calculated as a linear function (w_2'). In this case the inner border and parts of the border edges appear brighter for the human observer. The resulting effect is illustrated in figure 7 left. Black pixels correspond to the video view while white regions illustrate the synthetic view. The effect occurs when a spatial ramp in luminance abruptly changes slope [1]. Therefore a sinusoidal changeover is more suitable for human perception. Additionally to avoid this effect at the edges of the border region, they are rounded (w_{edge}). Finally w_2 is calculated as:

$$w_2 = \frac{sin((w_2' - 0.5)\pi) + 1}{2}$$

Fig. 7: Pixel-based blending between views, Left: linear changeover, Right: sinusoidal changeover with rounded edges

This can be effectively calculated in a fragment-shader program, where alpha masks are adapted according to w_2. The blending capability of OpenGL enables this functionality at almost no additional computational cost.

The resulting hybrid view leads to a higher immersion as it displays fewer blending artifacts (see figure 6 right). Figure 8 illustrates a complete transition from video over hybrid to a synthetic view.

VIII. VIEW VARIANT MARKING

Video view and synthetic view have different time lags. The operator should be aware which variant is currently used to judge the reliability of the presented image data. This situation awareness is an important factor to ensure safe teleoperation [6]. In order to perceive this information, each view variant has to be marked clearly.

An easy solution for this task (using the visual modality) is to fill the border pixels of the operator screen with a colored frame. The color indicates which view type is used thus providing a strong cue to the operator. Another possibility is to apply a global modification on all pixels of one view variant, e.g. by changing color saturation. Alternatively checker patterns can be superimposed.

It is important to evaluate which practice achieves a clearly visible but not annoying marking. The current implementation reduces color saturation of the synthetic view to a grayscale display (figure 6) or applies a checker pattern (figure 8).

IX. CONCLUSION AND FUTURE WORK

The presented solution is capable to generate location unbound views of a remote teleoperator scene. Three view variants were implemented: video, hybrid and synthetic views. Video views present the most up-to-date information for a set of restricted locations. Synthetic views override this restriction with the disadvantage of delay. Hybrid views combine both variants in one display. It is described how to decide which variant to use, based on a video error measurement. Two blending possibilities for the transition between view variants are presented. Attention is payed to the Mach-Band effect. By view marking the operator always knows which view variant is currently used.

No technical optimum can be found for m_t and ϵ. By now experimentally found values are used. An evaluation to optimize them for human perception is necessary and planned.

To enable video views the operator has to move close to P_c. This is a problem if no visual indication is available. Augmentation can support the operator to find P_c

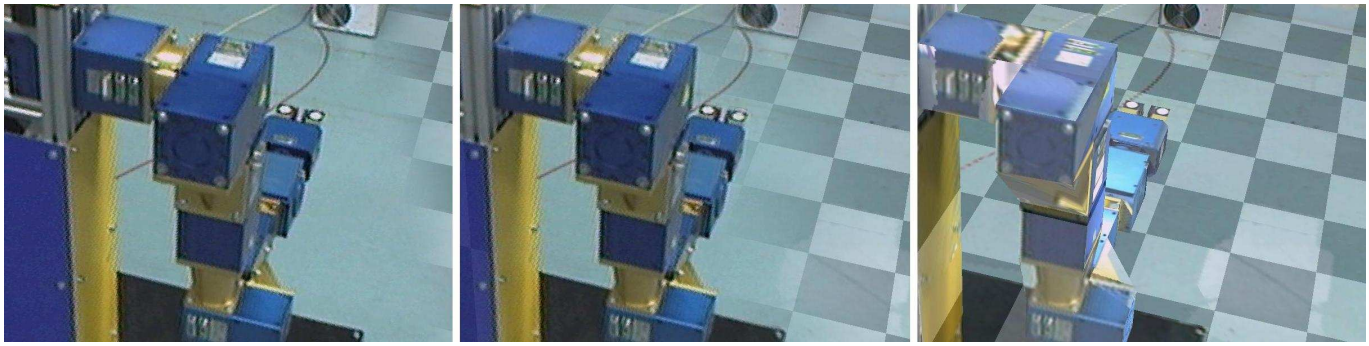

Fig.8: Hybrid View, synthetic part with checker pattern, pixel-based transition with increasing rotation angle

intuitively. If P_c can be controlled at least for some DOF according to P_o, video view usability can be increased. Regarding inevitable communication delays, prediction of P_o seems to be useful. In [12] a Kalman filter was successfully used for the prediction.

Until now the calculation of blending weights for transitions is based only on the video error measure. Additional consideration of time could further increase immersion: when detecting a tendency towards one view variant (based on m), the blending weight could be calculated as a function of time.

X. ACKNOWLEDGMENTS

This work is supported in part by the German Research Foundation (DFG) within the Collaborative Research Center SFB 453 on "High-Fidelity Telepresence and Teleaction".

REFERENCES

[1] Edward H. Adelson. *Lightness Perception and Lightness Illusions*, chapter 24. M. Gazzaniga, Cambridge, MA, MIT Press, 2000.

[2] Jonathan Baldwin, Anup Basu, and Hong Zhang. Panoramic Video with Predictive Windows for Telepresence Applications. In *ICRA*, pages 1922–1927, 1999.

[3] Tim Burkert, Jan Leupold, and Georg Passig. Hardware accelerated texture extraction for a photo-realistic predictive display. In T. Ertl, B. Girod, H. Niemann, H.-P. Seidel, E. Steinbach, and R. Westermann, editors, *Vision, Modeling, and Visualization 2003*, pages 11–18, Technische Universität München, November 2003.

[4] Shenchang Eric Chen. Quicktime VR: an image-based approach to virtual environment navigation. In *SIGGRAPH '95*, pages 29–38, New York, NY, USA, 1995. ACM Press.

[5] Fan Dai. Remote Supervision and Task-level Control of Industrial Robot Systems Using Augmented Vision. *Automatisierungstechnik*, 49(7):329–335, 2001.

[6] Mica R. Endsley and D. J. Garland, editors. *Theoretical underpinnings of situation awareness: A critical review.* Mahwah, NJ: Lawrence Erlbaum Associates, 2000.

[7] Lewis E. Hitchner. Virtual Planetary Exploration: A Very Large Virtual Environment. In *ACM SIGGRAPH Course Notes*, volume 9, pages 6.1–6.161, July 1992.

[8] Georg Passig, Tim Burkert, and Jan Leupold. Scene model acquisition for a photo-realistic predictive display and its application to endoscopic surgery. In *MIRAGE 2005: International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, pages 89–97. INRIA Rocquencourt, France, March 2005.

[9] Paul Rademacher and Gary Bishop. Multiple-center-of-projection images. In *SIGGRAPH '98*, pages 199–206, New York, NY, USA, 1998. ACM Press.

[10] Jonathan Shade, Steven Gortler, Li wei He, and Richard Szeliski. Layered depth images. In *SIGGRAPH '98*, pages 231–242, New York, NY, USA, 1998. ACM Press.

[11] Kazumasa Yamazawa, Tomoya Ishikawa, Tomokazu Sato, Sei Ikeda, Yutaka Nakamura, Kazutoshi Fujikawa, Hideki Sunahara, and Naokazu Yokoya. Web-Based Telepresence System Using Omnidirectional Video Streams. In *IEEE Pacific-Rim Conference on Multimedia*, pages 45–52, 2004.

[12] Yi Zhang, Huosheng Hu, and Huiyu Zhou. Study on Adaptive Kalman Filtering Algorithms in Human Movement Tracking. In *Int. Conf. on Information Acquisition.* IEEE, 2005.