

Reality Portals

Karl-Petter Åkesson
Swedish Institute of Computer Science
c/o Viktoriainstitutet, P.O. Box 620
405 30 Göteborg, Sweden
+46 31 773 5542
kalle@sics.se

Kristian Simsarian
Swedish Institute of Computer Science
P.O. Box 1263
164 29 Kista, Sweden
+46 8 633 1586
kristian@sics.se

ABSTRACT

Through interactive augmented virtuality we provide the ability to interactively explore a remote space inside a virtual environment. This paper presents a tool and technique that can be used to create such virtual worlds that are augmented by video textures taken of real world objects. The system constructs and updates, in near real-time, a representation of the user-defined salient and relevant features of the real world. This technique has the advantage of constructing a virtual world that contains the relevant video-data of the real world, while maintaining the flexibility of a virtual world. The virtual-real world representation is not dependent on physical location and can be manipulated in a way not subject to the temporal, spatial, and physical constraints found in the real world. Another advantage is that spatializing the video-data may afford more intuitive examination.

Keywords

Augmented Virtuality, Teleoperation, Collaborative Virtual Environments, Video Textures, Environment Visualization.

1. INTRODUCTION

Remote environment visualization and manipulation is a widely studied and practical area with applications in security, monitoring and exploration in distant or hazardous locations. The technique in this work creates an augmented virtual world that contains real world images as object textures that are created in an automatic way, these are called *Reality Portals*. Using Reality Portals allows a user to interactively explore a virtual representation of video from a real space. The textures are taken from video streams of the real world and which have dual (mirror) objects in the virtual world. This has one effect of making a virtual world appear, to a limited extent, as the real world, while maintaining the flexibility of the virtual world. In this respect, the augmented virtual world could be viewed as an instantiation of immersive 3D video photographs. Objects have a similar appearance to their real counterparts, but can be manipulated in a virtual setting.

To capture the video in a remote setting we use an active mobile robot with video camera that explores the remote physical environment. We also use a 3D graphical model of the remote environment. Segments of video are taken from the robot's video stream and 'smartly' placed in the 3D

environment as textures on virtual objects. This constructed 3D environment, a multi-user collaborative virtual environment (CVE), can be explored interactively concurrent to the texturing and exploration process. Though the discussion here focuses on robot-based video, the image source could also be a securely mounted fixed or pan-tilt camera.

The CVE is used as a control interface for manipulating objects in the real world. An advantage of the virtual world is that it is not dependent on physical location and can be manipulated in a way not subject to the temporal, spatial, and physical constraints of the real world. It also has the advantage that irrelevant parts of the real world can be left out. Thus the "interesting" parts of the world are extracted and made salient. In this way a custom view of the world is created, forming an instantiation of selective vision. This combination of real and virtual environments is called Augmented Virtuality, which is the converse of the more well-known technique of Augmented Reality.

The goals of using video images in the virtual world are plenty. The first is the same as the goal of using textures in a virtual environment; to give a richness of 'reality details' in the virtual world. Such photographic elements contain information and afford immediate access through user-visual memory for object identification and understanding. Another purpose is to furnish a 3D means of viewing visual information taken from the real world, for example to visually monitor physical spaces in near real-time via the CVE.

In this paper we first present related research and then a selection of visualization metaphors that lead us to Reality Portals. We then present Reality Portals in depth while offering an overview of basic methodology and system. We conclude with items for future work and applications.

2. RELATED WORK

This paper presents a methodology for performing remote environment visualization through the technique of Reality Portals. The work builds on a CVE-based telerobotic control platform already in existence[13]. With the robotic system, a human supervisor can control a remote robot assistant by issuing commands using the virtual environment as a medium. In this paper we examine different methods of using the video camera-equipped robot in a remote environment to visualize the remote physical space within the virtual environment. Through the different methods we show how using Reality Portals solves many limitations over other video visualization schemes.

It is not difficult to see how the capability of sending autonomous robots into hazardous and remote environments (e.g. space, sea, volcanic, mine exploration, nuclear/chemical hazards) can be useful. Robots can be built to stand higher environmental tolerance than humans, they can be built to perform specialized tasks efficiently and they are expendable.

To this end there has been much research on autonomous and teleoperated robots that can navigate into an area performing manipulations, and returning video imagery of the remote space. We are using an immersive virtual environment and a video-equipped robot as the interaction paradigm with the real world. Specifically, our work is an application in the SICS Distributed Interactive Virtual Environment (DIVE) system[3]. The basic function of Reality Portals is to incorporate and spatially display on-board video from the robot into the virtual world.

We first discuss the more well known technique of Augmented Reality. The overlaying and mixing of graphics into a video stream is referred to as Augmented Reality. Classic examples of Augmented Reality employ a head-mounted display that enables the bearer to see both the real world and a graphics display that is overlaid onto a semi-transparent screen. In this manner, a user of such a system would see the physical world augmented with 'appropriate' graphics. An application of such a system enables the guided repair of a laser printer[4]. Similar applications for repairing an automobile engine have also been demonstrated with visually annotated instructions appearing on the lens of the see-through glasses[7]. In the previous examples, the user was present in the physical environment. Similar operations can be performed remotely. An example of remote Augmented Reality is a telerobot system used to deliver video of a remote scene to a special display enhanced with interactive graphics and worn or observed by the operator. Such applications include virtual tape-measurement on real scenes as well as graphical vehicle guidance[11] and enhanced displays for teleoperated control[10].

In contrast to this standard notion of augmented reality, Reality Portals build on the complimentary operation, to embellish the virtual environment with real-world images. The laying of video onto and into graphics has been referred to as Augmented Virtuality. Milgram[11] attempted to map out a taxonomy of "mixed realities" that blend graphics and real world video. One axis of this taxonomy spans from Augmented virtuality to Augmented Reality, with a positioning of many of the possibilities in between. As compliments these two fields are closely related in technique, building on a model of the environment and imaging system.

One significant application of using photographic textures in a virtual model is that of reconstructing a crime scene in a VE with photographs taken of the physical space. Hubbard *et al* describe this crime-scene application and its implications, though the extraction process is not automatic[9]. In the field of entertainment, Benford *et al.* have applied these techniques in a CVE for inhabited and interactive television; a place where television streams, video conferencing, and traditional social virtual environment systems coincide[1]. The base techniques for all these systems are the same and include models of the real and virtual scenes, methods for locating views (camera and graphical) within those models, and methods for mixing the video and graphical streams. A popular application of Augmented Virtuality is the MarsMap system demonstrated as part of the Mars Rover environment visualization VR application in 1997[2]. In that work, the authors used techniques similar to the projection screens outlined in the next section, but did not take the next step toward Reality Portals.

The work in this paper could also be viewed as an instantiation of work in the general area of immersive 3D video, or 3DTV. In immersive video, a user is not restricted to the 2D plane for video but can interactively choose views and explore video regions. We have not yet gone so far as to create full live video

environments. Rather, for our purposes, we see the Reality Portal application as a means of filtering non-essential details from a potentially cluttered scene in the real world and displaying them spatially. Work in the field of 3D video has been done at UCSD by Ramesh Jain's group[6] and also at CMU by Takeo Kanade's group[5] among others. Both of those applications have concentrated more on creating a database that can be interactively accessed. In this work we have created a tool that can be used for near real-time remote investigation.

3. VIDEO IN VIRTUAL ENVIRONMENTS

The modeling fidelity of the virtual environment and the desire for a robot to explore dynamic remote environments suggests that a graphical virtual environment model can benefit by the addition of video images from the real scene. The rest of this paper concentrates on the situation where a virtual model exists for the environment and provides a base for the addition of textures from the remote environment.

How the video is treated and its source (e.g. camera placement) affects the different ways the user can visualize the environment. During our work with remote control interfaces for robots we have explored several techniques. Reality Portals, described in this paper, is the ultimate of these visualization techniques. Prior to Reality Portals we explored other techniques that we call the Monitor Metaphor and Projection Screens. We first describe these earlier techniques in the following sections. This discussion then offers some motivations for the use of Reality Portals. The Reality Portals technique is the most mature and provides the most elegant solution to the limitations encountered with other techniques.

3.1 Monitor Metaphor

We use *monitor metaphor* to describe the scenario where live video from the robot's working environment is presented through a virtual monitor on top of the robot model in the virtual environment (see figure 1).

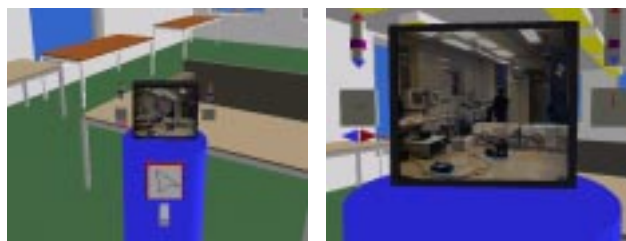


Figure 1. Monitor Metaphor. The left image shows an overview of the virtual environment with the robot and the monitor. The right image shows the operator's viewpoint when 'locked' to the robot. The operator then sees the video pan as the robot explores the space.

The live video presented on the monitor originates from a camera located on top of the physical robot. The monitor gives a view into the real world from the perspective of the robot's position. This way of displaying the robot's view of the world is an effective mapping in that it is reasonably and quickly clear to the operator what is being displayed. For example, the source point of the view and the image content are clear from the spatialized context in which they are presented. The dynamic behavior of the monitor is also clear. When the robot moves, the graphic robot object is updated and thus the monitor

containing the video is moved as well. The video stream on the monitor is also continuously updated from the new view. This is a consequence of the coupling between the robot's physical movements and those of its robot avatar in the virtual environment.

The operator has the choice to lock his/her movements to the robot's movements. When the user's view is locked to the robot's movements it is as if the operator uses the robot as a virtual vehicle to move around both the real and virtual environment. Here, the monitor is fixed to the operator's viewpoint, offering a view into the robot workspace. As the robot/operator pair rotate and translate around the room, the operator is given an interactive video scan of the remote environment. This can work to increase the impression of virtual presence. There is however a need to leave video images in the environment so they can be re-visited and re-examined. This leads to the next solution of using *projection screens*.

3.2 Projection screen

A large surface with a video image from the real world placed in the virtual environment is another way to place the video within the virtual environment. We call such a surface a *projection screen*.



Figure 2. Projection screen. A large surface with video is introduced into the virtual environment to display qualities of the physical space. Note that the virtual table intersects with the real one.

The projection screen can be updated with live video in a similar way as the monitor metaphor. Thus the projection screen becomes another monitoring source but not attached to the robot as the monitor metaphor described previously. Image stills taken by the robot are left in the virtual environment at the location they were taken. Alternatively, if the video source is from a secondary camera, the video stream could be incorporated within the virtual environment and viewed on the projection screen.

One key difference between the projection screen and the monitor metaphor is that the screen is statically placed within the virtual environment and does not move around as the monitor on top of the robot does. In this way, images previously taken can be stored spatially for later viewing. This is a function that the monitor metaphor did not readily afford.

The projection screens also afford a large viewing area within the virtual environment, offering the user an opportunity to view the real world content from a greater distance than the Monitor Metaphor allows. Typically the projection screens are 'wall-sized.' They can be compared to standard texturing techniques where a key difference is that a projection screen is

often a 2D object in 3D space within the virtual environment and displays 3D scene structure.

There is no obvious way to show the connection between the projection screen and the virtual objects whose physical counterpart is mirrored on the projection screen. One way that we have used is to let the projection screen intersect with the virtual objects (see figure 2). Such a positioning offers the user a clue to the connection between the virtual object and the real counterpart. However there are limitations related to the projection screen's 2D nature and the 3D spatial nature of the CVE.

3.3 Limitations

There are several limitations to the methods described above, related both to space and time. For the monitor metaphor we are limited in space to view what is currently viewable in the monitor. To obtain an accurate visual image of the remote workplace the operator would have to command the robot to pan around the room, even if the robot's camera had panned that space before. Thus, this limitation in space also relates to limitations in time (e.g. the time to move the robot). The monitor metaphor also restricts the user to be situated close to the robot if he/she does not want to lose the connection to the real world. Therefore he/she cannot roam the virtual environment freely without moving the robot and still be able to see the video in the real scene.

The second method employed is the projection screen. These are textures placed into the scene with a fixed flat view of the remote scene from a particular time instant and angle. There are two major limitations with the projection screen. First, it is only perspective correct from one position (the point where the image was taken)[†]. At other positions, the illusion of three-dimensionality is broken and the connection between the 3D model and the image is not evident. Secondly, with projection screens it is hard to make the associations explicit between the objects on the screen and the objects within the virtual environment.

Over time, there is no image history and, in addition, it costs time to re-acquire images of places previously visited even if those scenes remain unchanged. Though we could save video segments or images for recall, it is not clear how to present these images to the user. They could again be displayed through the monitor interface via a button-operated interface. In that way the user could flip through those images for reference. This problem of storing images was partially solved by using projection screens, but the scene quickly becomes cluttered. In the MarsMap system they attempt to remedy this cluttering by turning on and off the projection screens (called *billboards* in that system). However that work-around remedy does not seem adequate. In short, these solutions do not exploit the 3D nature of the virtual environment. These limitations are what led us to the concept of Reality Portals that offers solutions to these problems by making direct use of the 3D spatial nature of the CVE.

[†] Note this only holds if the camera and viewpoint characteristics, e.g. extrinsic and intrinsic parameters, are the same or similar.

4. REALITY PORTALS

The limitations suggested above pointed toward an implementation of another more general solution, that is based on applying the appropriate segments of video onto the actual corresponding virtual objects in the virtual model. We call these video-augmented objects *Reality Portals*, as they are viewpoints from the virtual world into the real world.

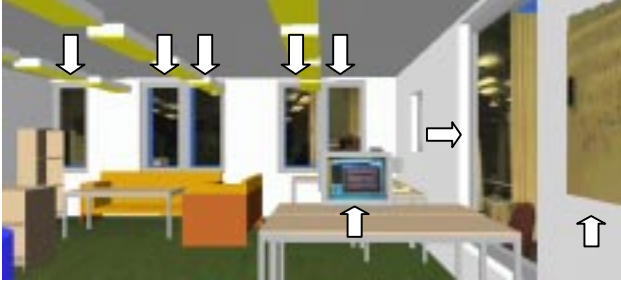


Figure 3. Reality Portals. The Virtual environment has been augmented with a number of textures from a video camera, Augmented objects are, the windows, the whiteboard and the computer monitor.

By using textures created from images of the actual physical objects we do not encounter the same problems conveying the associations from the real to the virtual world. Also the three-dimensionality of the virtual environment is used in a more sophisticated way as the whole virtual room is used to visualize the video. By laying textures in space it is also possible to have an image history located around the room. If the Reality Portals can supplant some of the uses of the monitor metaphor there is also an added benefit in a great reduction in the amount of data distributed between multi-user nodes of the CVEs.

To demonstrate this technique, a Reality Portals prototype has been developed. The prototype can, with proper mathematical camera model and a reasonable model of the environment, apply pieces of the extracted video images to corresponding objects in the virtual environment. This works by first specifying special flat objects in the virtual model, the Reality Portals. When the view-cone of the camera intersects one of these objects, an event is generated. This event can trigger the request of the appropriate piece of the video image, for that Reality Portal, as a texture. This texture is then applied to the requesting Reality Portal.

Through this process, textures are applied automatically in near real-time in the virtual world. We say 'near real-time' since our prototype currently only manage to generate 1-3 frames per second. But as discussed in the Future Work chapter there is much room for optimization. As the robot explores the world, these textures are automatically added to the virtual objects and stored in the virtual world database. Thus the time-history limitations mentioned before are partly solved in that old images are placed in the virtual space in their corresponding positions. The virtual world model offers an intuitive spatial sense of how to view these video images and their source. It is much like having a 3D video to navigate around (see figure 3).

Some of the space limitations are also solved by the process because now the operator is not limited to just look through the monitor at the video of the remote scene, but can instead navigate through the virtual world and see the video spatially displayed. Because highly structured images are split up, many of the video images applied on the Reality Portal are portraits of flat surfaces. Thus there are fewer problems in losing the illusion of three-dimensionality.

Note that the textures are extracted from the video image and applied only to the requesting surface. For example, to cover an entire 3D object in the virtual environment, e.g. a cube on a table, Reality Portals could be placed on the five potentially visible sides. As the robot navigates around the space and the camera's view-cone intersects with those real surfaces, the textures will be extracted and laid onto the virtual object covering it with textures from the video of the real space.

These different forms of using video in the virtual environment can also be used together. The operator can view the real-time video from the monitor interface while also viewing the history of video images via the textures in the virtual environment.

5. TEXTURE EXTRACTION

Our method to extract the textures for the Reality Portals from video is based on a basic camera model, a virtual model of the real scene and basic image processing.

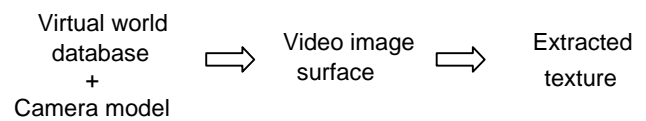


Figure 4. Method. A calculation based on the coordinates from the virtual world database and the camera model parameters gives the image coordinates. From the image, the region of the relevant surface is extracted as a texture.

The camera model together with the virtual model make it possible to predict where in the video image different objects appear. The database, which stores the definition of the virtual environment, provides the coordinates of surfaces within the virtual environment. These coordinates are transformed through the mathematical camera model, which gives the coordinates of the virtual surfaces in the image. Through image processing it is then possible to extract textures from those areas in the video image. This technique to predict objects within the video image is rather similar to the ones used in Augmented Reality.

5.1 Camera Calibration

Every image taken of a scene through a camera is a distorted version of the real scene. With a mathematical camera model it is possible to describe these distortions to some approximation. The most obvious effect comes from the perspective projection, the 3D to 2D transformation. There are also distortions from the camera lens, the CCD array and the video frame-grabber (these are the most significant factors). Point projection is the fundamental model for the perspective transformation wrought by imaging systems such as our eye, or a camera and numerous other devices. To a first order approximation, these systems behave as a pin hole camera, i.e. the scene is projected through one single point onto an image plane (the same model used for 3D image rendering). The camera model we use is based on the pinhole camera model but also takes care of some lens properties and effects from the frame grabber.

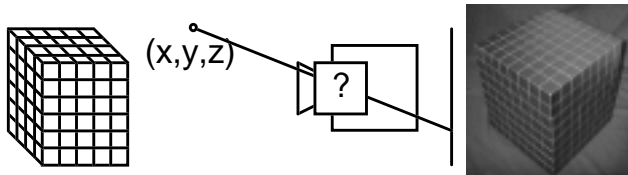


Figure 5. Camera Calibration. The extrinsic and intrinsic parameters of the camera model need to be discovered through a calibration process. This process is facilitated by capturing an image of a precisely known object.

The parameters in the camera model are not readily available but can be discovered through a camera calibration process[8]. The process of camera calibration has been studied intensively during the last decades in both the photogrammetry and computer vision communities. The parameters to be discovered can be divided into two different sorts. These are extrinsic and intrinsic parameters. The extrinsic parameters belong to the setup of the camera, e.g. the estimation of camera position, rotation and translation. Extrinsic parameters represent the relationship between the coordinate system of the camera and the global coordinate system. The intrinsic parameters include the optical and electronic properties of the camera such as focal length, principal point, lens distortion, the aspect ratio of the pixel array and other CCD effects.

The extrinsic and intrinsic parameters must be known in order to use the camera model and predict where 3D-coordinates will be mapped onto the 2D-image plane. To do real time measurements of the parameters is almost impossible, as the calibration process is computationally demanding. If the intrinsic parameters do not change during and between runs it is enough to calibrate them only once. By turning off features like auto white-balance and auto-gain and locking lens properties such as zoom and focus, the electrical and optical properties of the camera will mostly not alter during or between each run. Correspondingly neither will the intrinsic parameters change and it is enough to run the calibration process once.

For a camera which does not move, the extrinsic parameters can also be calibrated once. If however the camera moves it is necessary to update these parameters, which is difficult to perform in real-time through camera calibration. Though rotation and position are easily measured units from our robot, due to drift and sensor inaccuracies we can only get accurate real-time measurements of relative changes from a known start position for a limited amount of time. We add the robot starting configuration (coordinates plus orientation and height) to the camera start position discovered through the calibration process. Thus we have start values for all the parameters needed for the camera model.

The actual calibration process is begun by capturing an image of a scene with known 3D coordinates. The parameters are then calculated based on a comparison of where these coordinates appear in the image and where they should have been. In our case we use a cube with a grid of vertices painted on the surfaces and through edge detection these are found and the parameters can be calculated, see figure 5.

5.2 Texture generation

The actual process to generate the textures for a Reality Portal is based on standard image processing and transformation algorithms. In the virtual model the 3D-coordinates of different objects and definition of object surfaces within the virtual world are given.

As the virtual model is an approximation of the real environment, these virtual surfaces are the same as the ones for the real objects. The camera model, with calibrated camera parameters and continuous updates of the extrinsic parameters, is used to do a standard transformation from the 3D-coordinates in the virtual world to 2D-coordinates in the video image plane.

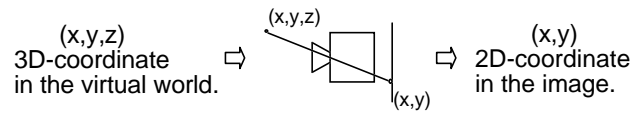


Figure 6. Transformation. The 3D coordinates in the virtual world are transformed into 2D image coordinates using the camera model.

We can now predict which parts of the video image belong to which different surfaces in the 3D virtual environment. Depending on the graphics system used to render the virtual world, different methods are employed to extract the texture. One the most common graphics system, OpenGL, only supports rectangle images as textures and therefore a non-rectangular texture for the Reality Portal has to be re-sampled using bilinear interpolation (a textbook image processing algorithm). This sampling is used to make a non-rectangular Reality Portal rectangular by adding pixels. Such problems are also encountered when the desired image segment is in perspective in the image. Such a segment will be extracted as a non-rectangle and warped to fit as a texture on the Reality Portal. Note that pixels have to be added, and thus parts of an image texture may have different clarity as a result of the sampling.

6. THE PROTOTYPE

The prototype is based on the SICS Distributed Interactive Virtual Environments (DIVE) platform. The DIVE platform is an internet-based multi-user VR system where participants navigate in 3D space and see, meet and interact with other users and applications. The DIVE software is a research prototype which supports the development of virtual environments, user interfaces and applications based on shared 3D synthetic environments. DIVE is especially tuned to multi-user applications, where several networked participants interact over a network.

The Reality Portals prototype has been implemented as an application that joins a virtual world and augments it with video textures. As our main interest has been remote robot control we have built upon a previous existing application for robot control within a virtual environment [14] and extended it with the Reality Portals prototype. Also, as we are dependent on continuous update of the extrinsic parameters for the algorithm, the prototype had to be integrated with the robot control application to get the relative changes of these parameters. As a result of the multi-user and networking functionality of the DIVE software it was easy to do this. The work with the monitor metaphor was also based on the same application and can therefore also be integrated easily with the use of Reality Portals.

In the DIVE virtual world database there are objects with special properties that signify to the application that they are Reality Portals. The properties also tell the application how to

treat each Reality Portal, which includes partial transparency and freezing (which are described later).



Figure 7. Texture Extraction. The left image highlights the Reality Portal areas to be extracted (table and whiteboard). In the right image, the whiteboard image has been extracted and placed as a texture in the virtual environment.

The Reality Portal textures are extracted using the algorithm described in the previous chapter. Then the distribution of the DIVE database is used to distribute the textures to different peers in the network. The prototype lets the user freely explore the augmented virtual environment. He/she can walk up to a Reality Portal, like the whiteboard in figure 7, and take a closer look. The user can also create an environment with a particular glimpse in time by freezing Reality Portals.

6.1 Partial Transparency

The Reality Portals extract the textures from one camera. If several cameras are used, or if an image is taken from a different viewpoint, more than one camera images the physical object. It is not obvious how the texture for the Reality Portal should then be generated, e.g. from which source? One solution is to use image processing to mix the parts from each camera image into one texture. Partial Transparency is a method we have explored that gives a good result and is much more efficient as it does not use any image processing to achieve the result. Each Reality Portal is associated with only one camera so it will only receive textures from that camera. Parts of the texture for an object that are outside the view of the camera are made transparent.

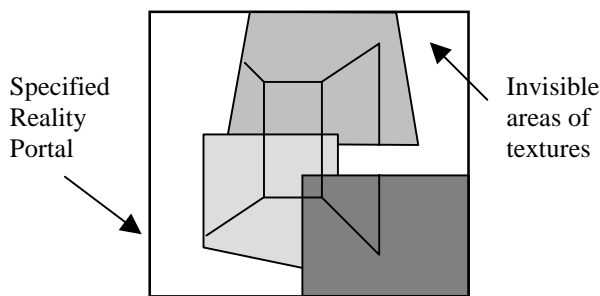


Figure 8. Partial Transparency. A specified Reality Portal might contain surfaces not visible in the camera image. Portions outside the known camera view are textured transparent. Stacking several image segments can, over time, complete the Portal.

By stacking Reality Portal objects on top of each other it is possible in a straightforward manner to generate an object that has a texture with an image of the whole physical object even if it is not completely imaged by any camera. Parts of the texture that are transparent, i.e. that are not imaged by a particular

camera, allow textures from Reality Portal objects beneath, to be viewed.

6.2 Freezing the Reality Portal in time

In the current version of the prototype it is possible for the user to freeze each Reality Portal, i.e. instruct it not to take any new textures. In this way it is possible for the operator to save snapshots of the environment. The operator can look back to previously visited areas without needing to steer the robot to that specific location or pan the environment with the camera. In the current prototype the operator clicks with the mouse on each Reality Portal in order to freeze it.

7. EXAMPLE OF APPLICATION

Beyond the use of Reality Portals for remote robot control, where we have concentrated our work, we see a number of other promising applications of this technique. One application would be for Security monitoring. Common practice of security professionals is to observe a suite of monitors and watch for abnormalities and security situations. The main limitations in this system are the great extent of space the guard must observe. Often to overcome this situation, the guards use a number of monitors, where each monitor rotates through a sequence of video inputs. The complimentary practice of security professionals is to perform walking tours of the monitored space.

An alternative is to employ a virtual model of the real space which is then used in combination with the application described in this paper. Cameras situated around the space could monitor the security space and apply textures to the virtual model. Thus the security guard could instead monitor a space by navigating through a virtual world instead of observing 2D video cameras. In current implementations of remote surveillance, the 3D model of the environment is inside the head of the security guard. Here the model is instantiated offering both a clear view of the structure of the environment and of the permissible navigation. By adding intelligent cameras and simple image processing, changes in the scene could be flagged and brought to the attention of the security guard. The main goal is to set up an intuitive interface where the choices and scene visualization and structure become more obvious by appealing to the human spatial senses.

Another application is to use it as a learning environment for robot control. As DIVE is a multi-user platform it would require very little work to implement a training application where both the teacher and the student interact and communicate within the virtual environment. While the student steers the robot and trains on the task at hand the teacher can both observe the student actions as well as the progress of the task and checking for possible failures. If the task is critical the teacher could first perform the task with the robot. Video from the real environment is collected during this session and then the student can train the task with the robot disconnected but still see the same things as he would if working with the robot.

8. CONCLUSION AND FUTURE WORK

In this paper we have presented a description of a system that implements a concept we are calling Reality Portals, which is an instance of Augmented Reality. While a robot roams around a real space, texture updates based on video are applied in a virtual world model of the physical space. Using the virtual model, a user can perform an off-line tour of the remote space in a form of tele-exploration. Using this application the

operator can de-couple his/her actions from the actions of the robot in the remote space to view video of the remote scene as the robot collects it.

Future work centers on improving the quality of the automatic extraction by improving the initial calibration of the camera as well as introducing dynamic calibration that refines the camera and robot model as it explores. With edge detection, a more robust application could be created and even if the virtual and real worlds do not synchronize, a good result could be achieved. Image processing hardware should be investigated to achieve frame-rate processing.

Another area of Reality Portal use is to explore unknown, unstructured environments. This would require the addition of methods for automatic detection of objects and methods for adding models of these encountered objects in the virtual world. These would allow the technique to be used in locations where no model has been supplied. An example of how to do this would be to use range sensors to sense solid objects. When such obstacles are detected an object could be instantiated in the location with a Reality Portal object that then extracts the texture. This information would prove useful to the operator for identifying the object that triggered the response. A more sophisticated sensor system could also be constructed with a basic image processing techniques coupled to the camera and calibration process. In this case, some structure, e.g. edges, could be determined from the scene, possibly with the user's assistance and then placed in the scene.

The application, the result of this work, is a good platform for further projects. The image warping performed can be implemented in hardware on machines that have support for these operations. Such libraries exist for Silicon Graphics computers. This would improve the run-time of the application.

The application does not take into account the orientation of a surface and therefore it cannot have textures on just one side. If the normal to the surface defines which side the texture should be on, the application could be extended to not generate textures if the object is seen from behind. If an object obscures the Reality Portal, the application is not aware of the obstruction. Thus, any artifact in between the target object and the camera will be included in the Reality Portal texture. This could be changed and the obscured part of the texture could be made transparent. The direct method to solve this is computationally expensive and will slow down the application. Such a check would require, for each pixel in the texture, projecting a ray from the camera to the Reality Portal to discover if it is obscured or not.

To date we have done no formal user studies with the system. We have, however, plans to perform such a study to evaluate a number of the control and visualization features of the robot system. In the process we hope to discover further ways to improve the general approach of adding video to virtual environments.

9. ACKNOWLEDGMENTS

We would like to thank our colleges that have contributed to the development of the DIVE platform and the robot control application. In particular we would like to thank Emmanuel Frécon and Olov Ståhl. without whom this work would not have been possible.

10. REFERENCES

- [1] Steve Benford and C. Greenhalgh, G. Reynard, C. Brown, B. and Koleva, "Understanding and Constructing Shared Spaces with Mixed Reality Boundaries," Transactions on Computer Human Interactions, to appear 1999, ACM Press.
- [2] Ted Blackmon, "MarsMap - VR for Mars Pathfinder", http://img.arc.nasa.gov/~blackmon/MarsMap/www_main.html.
- [3] Christer Carlsson, Olof Hagsand, "DIVE - A Platform for Multi-User Virtual Environments," Computers and Graphics, vol. 17, no 6, 1993.
- [4] Steven Feiner, Blair MacIntyre, and Doree Seligmann, "Knowledge-based augmented reality," Communications of the ACM, vol. 36, no 7, pp. 52-62, July 1993.
- [5] T. Kanade, P.J. Narayanan, and P. Rander, "Virtualized reality: concepts and early results," IEEE Workshop on the Representation of Visual Scenes, Boston, MA, June 1995.
- [6] Arun Katkere, Saied Moezzi, Don Kuramura, Patrick Kelly, and Ramesh Jain, "Towards video-based immersive environments," Multimedia Systems Journal: Special Issue on Multimedia and Multisensory Virtual Worlds, ACM-Springer 1996.
- [7] Gudrun J. Klinker, Klaus H. Ahlers, David Breen, Pierre-Yves Chevalier, Chris Compton, Douglas Greer, Dieter Koller, Andres Kramer, Eric Rose, Mihran Tuceryan, and Ross T. Whitaker, "Confluence of computer vision and interactive graphics for augmented reality," Presence, August, vol. 6, no 4, 1997.
- [8] M. Li. "Camera calibration of the KTH head-eye system." Technical report, Computational Vision and Active Perception Lab., Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology (KTH), March 1994.
- [9] A.D. Murta, S. Gibson, T.L.J. Howard, R.J. Hubbard, A.J. West, "Modelling and Rendering for Scene of Crime Reconstruction: A Case Study," Proceedings Eurographics UK, Leeds, March 1998, pp.169-173.
- [10] M. Malle, F. Chavand, and E. Colle, "Computer-assisted visual perception," Teleoperated Robotics. Robotica 10, pages 99-103, 1992.
- [11] Paul Milgram and David Drascic, "Enhancement of 3-d video displays by means of superimposed stereo-graphics," Proceedings of the Human Factors Society 35th Annual Meeting, pages 1457-1461, 1991.
- [12] Paul Milgram, Anu Rastogi, and Julius Grodski, "Telerobotic control using augmented reality," Proceedings 4th IEEE International Workshop on Robot and Human Communication, Tokyo, July 1995.
- [13] Kristian T. Simsarian, Jussi Karlgren, L. Fahlen, Emmanuel Frecon, Ivan Bretan, Niklas Frost, Lars Jonsson, and Tomas Axling, "Achieving virtual presence with a semi-autonomous robot through a multi-reality and speech control interface," In M. Goebel, J. David, P. Slavik, and J.J. van Wijk, editors, Virtual Environments and Scientific Visualization '96, SpringerCS, 1996.