Real-Time Photorealistic Virtualized Reality Interface For Remote Mobile Robot Control

Alonzo Kelly, Erin Capstick, Daniel Huber, Herman Herman, Pete Rander and Randy Warner

Abstract The task of teleoperating a robot over a wireless video link is known to be very difficult. Teleoperation becomes even more difficult when the robot is surrounded by dense obstacles, or speed requirements are high, or video quality is poor, or wireless links are subject to latency. Due to high quality lidar data, and improvements in computing and video compression, virtualized reality has the capacity to dramatically improve teleoperation performance — even in high speed situations that were formerly impossible. In this paper, we demonstrate the conversion of dense geometry and appearance data, generated on-the-move by a mobile robot, into a photorealistic rendering database that gives the user a synthetic exterior line-of-sight view of the robot including the context of its surrounding terrain. This technique converts remote teleoperation into line-of-sight remote control. The underlying metrically consistent environment model also introduces the capacity to remove latency and enhance video compression. Display quality is sufficiently high that the user experience is similar to driving a video game where the surfaces used are textured with live video.

1 Introduction

Effective operation of any mobile platform without direct line-of-sight is intrinsically difficult to achieve. In video-based teleoperation, the loss of peripheral vision caused by viewing the world through the soda straw of a video camera reduces driving performance, and it increases the operator's frustration and workload. Wireless

Alonzo Kelly, Dan Huber, Herman Herman, Pete Rander, Randy Warner
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA 15213, e-mail: name@rec.ri.cmu.edu

Erin Capstick
DCS Corporation. 1330 Braddock Place, Alexandria, VA, USA, 22314 e-mail: ecapstick@dcscorp.com

communication links are also subject to dropouts and high levels of latency. Their bandwidth limitations typically cause a large reduction in image quality relative to the fidelity of the underlying video cameras. When the robot undergoes significant or abrupt attitude changes, the operator response may range from disorientation, to induced nausea, to dangerous mistakes. The need for high attention levels also deprives operators of the capacity to pay attention to their surroundings. Wireless communications issues and difficulty controlling the robot also increase time on task, and they increase the time required to become a skilled operator.

1.1 Motivation & Technical Approach

The term virtualized reality [7] refers to the production of views of a rendering database where the geometry and appearance content is derived from measurements of a real scene. For visualization, such a database makes it possible to render synthetic views of the scene from arbitrary perspectives that may never have been the site of any real sensor. Such techniques represent an extreme on a spectrum of real data content with augmented or mixed reality somewhere in between and virtual reality at the other extreme. Virtualized reality enables a new capacity to address many of the problems described above by providing a photorealistic, synthetic, line of sight view to the robot based on the content of geometry-augmented real-time video feeds.

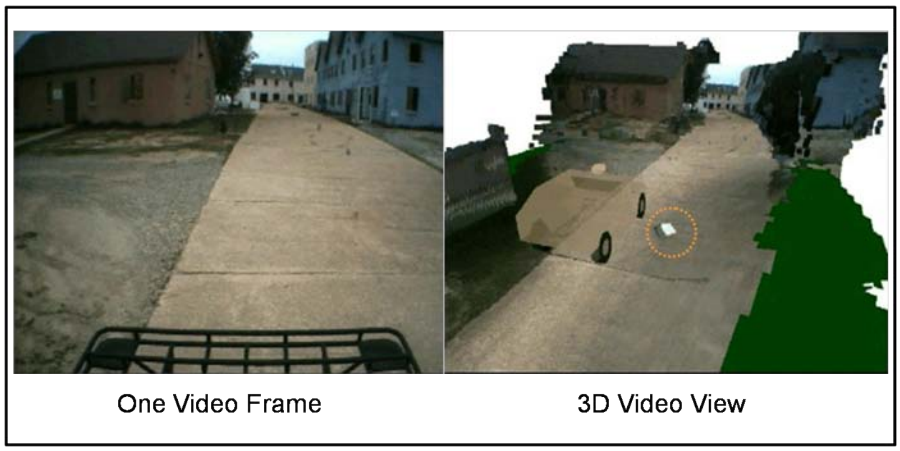

Fig. 1 3D Video View of a Mobile Robot. Left: A video frame produced from a camera on a moving vehicle. Right: The 3D Video view produced from all of the video that has been received in the last few seconds by the vehicle. The operator can look at this database from any angle, at any zoom, while it continues to be updated in real time. The vehicle is synthetic since no sensor has imaged it, but the rest of the content is generated from live video produced by the forward looking sensor mounted on the vehicle roof.

If the processing is performed in real-time, a kind of hybrid 3D Video (Figure 1) is produced that can be viewed from arbitrary perspectives while exhibiting the photorealism and dynamics of live video. The operator experience is equivalent to following the robot in a virtual helicopter that provides arbitrary viewpoints including an overhead viewpoint and the over-the-shoulder view that is popular in video games.

A large number of benefits can be realized with this approach to user interfaces.

- The operator can see any part of the hemisphere around the vehicle at any time.
- The display provides a natural mechanism to introduce augmented reality operator aids.
- The viewpoint is stabilized regardless of the attitude (pitch and roll) changes of the real vehicle.
- Viewpoints can be customized and switched for each task. For example, parallel parking is much easier when the vehicle is viewed from above, and objects can be examined closely by zooming in on them.
- Multiple viewpoints can be shown at once, and multiple operators or observers can view the scene independently.
- The frame rate of the display can be adjusted independently from that of the underlying video feed.
- Dropped frames can be easily tolerated by rendering the most recent model.
- Deliberate dropping of frames or many other schemes can be used to compress the data transmission.
- Even with a vehicle moving in the display, latency can be essentially eliminated by rendering the predicted vehicle state.
- A photorealistic map of the area traversed is produced as a byproduct of system operation.

The engineering difficulty of producing and processing such data is considerable, and it is even more difficult if it must be photorealistic and produced in part from data feeds of scanning lidar sensors derived from a continuously moving vehicle in natural terrain. Nonetheless, the rewards of such efforts are also considerable as we hope to show in the sequel.

1.2 Related Work

The technique of view interpolation was introduced in computer graphics [2] as a mechanism to exploit depth information in order to efficiently produce synthetic imagery while bypassing many of the computationally expensive geometric aspects of rendering. View interpolation is one of several approaches to image-based rendering. Such techniques achieve remarkable realism through the use of natural imagery to texture surfaces. Given depth, a purely synthetic view of a real scene can be produced by projecting the pixels of an image to their proper 3D locations and reprojecting them onto a new image plane. Kanade coined the term virtualized reality

[7] to emphasize that the image data was natural rather than the synthetic imagery used in virtual reality. Initial virtualized reality work was based on off-line stereo ranging and stationary sensors that surrounded a localized dynamic scene. More recently real-time image-based rendering has been accomplished for a single discrete object and fixed cameras based on a visual hull method for computing depth [11]. While computer vision and computer graphics evolved toward virtualized reality, telerobotics was simultaneously developing augmented reality displays for more effective remote control. Numerous techniques for supervisory control and teleoperation of manipulators, and even telepresence, were clearly outlined as early as the mid 1980s [16]. The same concepts were considered early for legged vehicles [12] and wheeled Mars rovers [1]. Given the sensor data needed, the earliest approaches to teleoperation simply displayed the raw sensor data or showed the robot in a 2D overhead view in the context of its surrounding perceived objects. Applications like space exploration generated a strong impetus to develop more realistic virtual displays as early as 1991 [5]. The potential of augmented reality environments has been explored in both nuclear servicing [13] and space [10] contexts. In these cases, a small amount of virtual information was rendered over natural video. Innovations included registration of the virtual model to reality using vision, and latency compensation using motion preview and predictive displays.

These techniques can be applied with more effort to robot vehicles. One sustained research effort in the use of virtual environments for the control of robot vehicles was the Virtual Environment Vehicle Interface (VEVI) described in [6]. This system was tested terrestrially [4], and derivatives were ultimately used on the Mars Pathfinder mission. Contemporary developments include more emphasis on sensor fusion [3] as well as efforts that display both forms of data (appearance and geometry) in a less integrated but more useable way [18]. The VEVI system is a clear landmark in related work and it is closest to the work we present here. Our work is distinct from VEVI in that VEVI did not perform image-based rendering and hence did not use virtualized reality. VEVI did render false color terrain maps produced by on-board lidar sensing for a slow moving legged vehicle and this achievement was unprecedented using the technology of that period. VEVI used a classical form of latency compensation based on vehicle autonomy and supervisory control interfaces but it did not perform the kind of high fidelity continuous motion prediction in virtualized reality that we will present here. We also achieve results in data compression and unprecedented vehicle speeds that derive respectively from the commitment to virtualize the entire scene and the use of custom photogeometric sensing as described below.

2 Hardware and Architecture

Virtualized reality constructs a computer graphics model of a real scene. The set of geometrically consistent graphics primitives to be displayed will be referred to as the model. For teleoperation, a key design decision is the location of the model build-

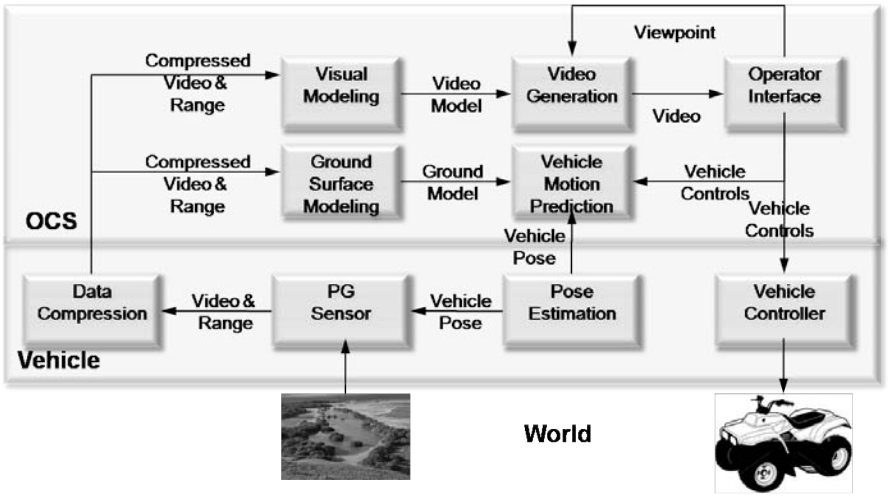

Fig. 2 System Architecture. The system includes an operator control station (OCS) and a remote-control retrofit of a standard all terrain vehicle.

ing process. If modeling is performed on the vehicle processor, then model updates can be communicated to the remote operator control station and communications bandwidth requirements can presumably be reduced. Reduction is possible because it takes less bandwidth to transmit a fused model that lacks the redundant content of video. If modeling is performed on the remote operator control station, raw sensor data must be communicated, and bandwidth requirements are higher. Despite this bandwidth cost, we chose the second option (Figure 2) in our implementation, in part, because the latency compensation process, discussed later, is more straightforward. In this case, operator commands to the robot can be observed (at the operator control station) without significant latency.

The basic hardware setup at the system level involves an operator control station (OCS) that communicates over wireless to a remotely located mobile robot equipped with photogeometric sensing.

2.1 Sensing

The term appearance will be used to refer to sensing modalities that are sensitive to the intensity of incident radiation including visible color, visible intensity, and visible or invisible infrared modalities. Conversely, geometry will be used to refer to modalities that register any of depth, range, shape, disparity, parallax, etc. The term photogeometric (PG) sensor will refer to a sensing device that produces both kinds of data in a deeply integrated manner. For our purpose in this paper, the data is deeply integrated if the spatial correspondences of the data are known. Ideally, as

shown in Figure 3, the resolutions are matched as well so that a one-to-one mapping exists between geometry and appearance pixels.

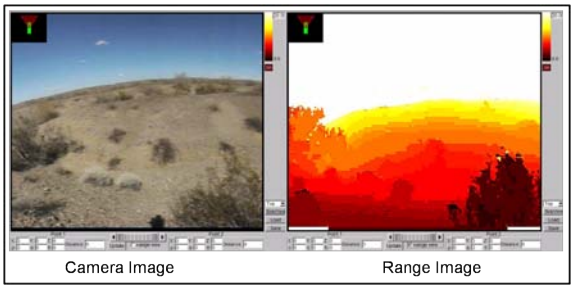

Fig. 3 Photogeometric Data Set. Every color pixel in the left image has an associated range pixel in the right image. Sensors that produce such data do not exist on the market today but they can be constructed by integrating more basic components.

Computational stereo vision is a natural and classical approach to photogeometric sensing because range is produced for every pixel in the reference appearance image. However, its utility in our application is limited due to the relatively poor quality of the range data. Our implementation approach therefore produces an integrated data set of appearance and geometry data from two distinct sensors: lidar and video.

Due to many considerations including the numerous robotic platforms that we construct annually and the desire to standardize solutions across programs, we have been continuously refining our photogeometric sensor concept for many years. A recent sensor design is shown in Figure 4. For scanning lidars, we typically purchase an off the shelf scanning lidar that scans in one degree of freedom (called the fast axis), and then we actuate the housing in a second degree of freedom (called the slow axis) in order to produce a scanning pattern that spans a large angle in both azimuth and elevation.

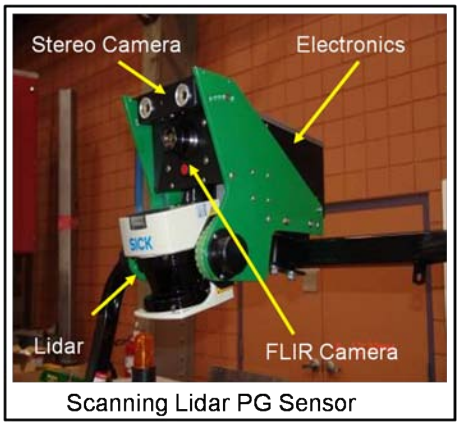

Fig. 4 Custom Photogeometric Sensor. The device fuses data from a commercial scanning lidar by SICK, stereo cameras, and a forward looking infrared (FLIR) camera. The interface to the composite device is a combination of fast Ethernet (used for high bandwidth data) and CAN Bus (used for low latency control).

The lidar pointing control system provides precisely timed feedback on the angle of rotation. This data stream is merged with the range and angle data coming from the lidar to form a 2D scanning lidar data stream. This stream is then optionally merged with any camera data and transmitted to the host computer system. In our autonomy systems, it is useful to merge the data at the level of individual imagery. However, for visualization, we instead merge the data later in the model building process at the level of an integrated geometric model of the scene.

2.2 Vehicle

Our latest vehicle test bed (Figure 5) is a custom retrofitted LandTamer® amphibious remote access vehicle. We chose this vehicle for its terrainability, ease of transport, and (deliberately) for the difficulty of modeling its skid steering.

Fig. 5 Robot Vehicle. A LandTamer® vehicle was retrofitted for remote control. Three custom colorized range (CR) sensors with a total field of view of 160° are mounted high on the vehicle looking forward. The lidars are manufactured by SICK providing range points at 2 KHz separated by ½ degree of angle over 180° of field of view. The cameras are the Firefly® by Pt. Grey Research Inc., and they provide color imagery at 720 X 500 resolution over a 60° field of view.

A custom field programmable gate array (FPGA) board is used to implement the servos that control the nodding motion of the lidars. It is also used to integrate the data into time tagged colorized range data sets and to provide the results over an ethernet link to the main vehicle computer.

A Novatel SPAN INS-GPS system is used for pose estimation, including the vendor's Kalman filter. The system is augmented by a portable real-time kinematic (RTK) differential base station. Under favorable satellite viewing conditions 2cm accuracies are achievable. A small computing cage houses the sensor control and data acquisition FPGA board and two Intel® Core™ Duo processors. These processors concentrate the data from all sensors and send it to the OCS over 802.11g wireless. They also receive the OCS commands over the same wireless link and pass them to the vehicle controller.

2.3 Operator Control Station (OCS)

The OCS (Figure 6) incorporates a steering wheel and throttle and brake pedals as well as a large LCD monitor. The OCS also contains a processor capable of both communicating with the robot and rendering the data on the display.

Fig. 6 OCS. The Operator Control Station includes a steering wheel equipped with selection buttons, foot pedals, and a large LCD display. The display provides selectable views including the raw video, over-the-shoulder, and birds-eye (direct overhead). The selected view is enlarged and the others are reduced in size and shown to the right.

3 Modeling and Visualization Algorithms

In order to achieve photorealism, we aspire to produce geometry for every camera pixel (rangefied color). Rather than do this in the image, however, results are improved if geometry is interpolated in the scene by fitting surfaces to the lidar (geometry) data and projecting the camera (appearance) data onto those surfaces.

Numerous effects give rise to situations where the color of a scene point is known, though its range is not. For many reasons, lidar data produced on a ground vehicle ceases to be reliable beyond a range on the order of 30 meters. Let the region beyond this range be known as the far field, and let that inside this range be known as the near field. Even in the near field, the reduced angular resolution of lidar relative to cameras implies that the vast majority of near field color pixels in a camera image will not have a lidar pixel that corresponds directly.

A second important issue is range shadows. It is necessary in general to depth buffer the range data from the camera perspective in order to ascertain which ranged points are occluded by others and therefore have unknown color. When the viewpoint differs significantly from that of the lidar sensor, substantial missing parts in the model become possible. For our purposes, the required precision of geometry depends on the offset of the viewpoint from the original sensor viewpoint. When the offset is small, substantially incorrect geometry will still look highly realistic.

When the offset is large, differences in parallax of scene points from their correct parallax will result in distortion that is noticeable to the operator.

In general, four classes of points can be distinguished. The system uses several strategies described below to process them.

- **Surface and texture known**. This is the easiest case where the texture is projected onto the surface.
- **Texture only known**. In this case, the geometry has to be assumed or the data rejected. Two cases of practical interest are under-sampled smooth surfaces, and regions beyond the lidar maximum range.
- **Geometry only known**. Enough cameras can be used to ensure that this case does not occur with two exceptions. First, the vehicle does not normally produce a lidar image of itself, but its geometry can be measured or coded offline. Second, regions occluded by other surfaces can be drawn in an unusual color to identify them to the operator, and both sensor separations in space and image separations in time can be minimized to the degree possible to mitigate this effect.
- **Nothing known**. Once the possibility exists to place a viewpoint anywhere, scenes with complex geometry will often show holes in the model that correspond to regions that no sensor was able to see for occlusion reasons. This is the cost of arbitrary viewpoints applied to data imagery from a specific viewpoint. There is no way in general to generate the missing data, but the advantages of arbitrary viewpoints can outweigh this imperfection.

Of course, regions of the scene may become unknown over the passage of time when the scene is dynamically changing. In such cases, omnidirectional lidars and cameras may be used to continuously update the view in all directions.

3.1 Near Field Surface Modeling

The application to ground vehicles justifies the assumption that the environment around the vehicle includes a ground surface and optional objects that may lie on it. In many environments, lidar data is sufficiently dense, out to 20 to 30 meters, to sample the terrain surface adequately for its reproduction. For this reason, the implementation segments all lidar points into those that lie on the ground and those that lie above it.

In forested environments, situations like overhanging branches invalidate the assumption that height is a single-valued function of position. Therefore, all lidar data is initially accumulated in a 3D voxelized, gravity-aligned data structure, called the point cube, before it is segmented. Each voxel counts the number of lidar beams that have terminated in the voxel (called hits), and the number that have passed through (called pass-throughs) to terminate in voxels further from the sensor. After each full sweep of the lidar beam, the point cube is analyzed to determine the lowest cell in each vertical column with enough hits to determine a ground surface. The average height of these hits is used to define the ground height at the position of the voxel in

a new data structure called the terrain map. This structure is a horizontal 2D array arranged into cells (20 cm on a side) that store the ground height at the center of each cell. The terrain map accumulates all data from multiple lidar scans. Its spatial extent, like the point cube, is limited to some adjustable region around the present vehicle position defined in terms of 3D space, distance, or time.

3.2 Projective Texture Mapping

Each cell in the terrain map is converted to two untextured triangles that must then be textured from the camera imagery. Densely populated voxels are those containing a large number of lidar hits. Those that are above the ground surface, but are not dense enough to define a surface, have their geometry hallucinated as the sides of the voxel. The points in sparsely populated voxels are enclosed in very small cubes to create geometry onto which to render their textures.

The baseline separation between camera and lidar can unfortunately be enlarged significantly due to asynchrony of the camera and lidar during periods of vehicle motion. Also, camera imagery may overlap due to multiple overlapping fields of view or multiple frames captured over time. Unless depth buffering is performed, the same textures will be painted onto foreground objects as well as those background objects that are occluded by them. This would not be a problem if the terrain map was the only surface in the scene, but there are others above it. Therefore, we instead use projective texture mapping [15] implemented in the OCS graphics processing unit (GPU) to paint the video onto the geometry. The system maintains a list of the most recent images from all cameras. Each image is used to apply texture to the geometry in the scene in temporal order so that cells that fall outside the field of view of more recent images will retain the last texture painted onto them.

3.3 Far Field Modeling and Visualization

Often, the far field region of space corresponds to the higher parts of the images, and it extends to the horizon. For such data, we erect a temporary surface (a billboard) that is normal to each camera's optical axis. The camera data is then projectively textured onto the surface.

The billboards move with the vehicle (Figure 7). Provided the viewpoint is not significantly different from the camera, the parallax error is tolerable, and operators overwhelmingly prefer their use. In the case of an overhead view, the billboards become normal to the viewing axis, and they mostly disappear.

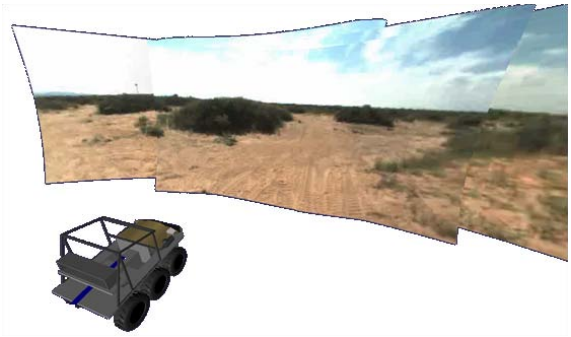

Fig. 7 Billboards Used to Display Far Field Video. This view shows the geometry of the three billboards and how video frames are pasted onto them. The technique of using billboards for complex scenes has been used for many years [14].

4 Teleoperation Algorithms

So far, we have described the basic mechanisms for producing and rendering a photorealistic model that surrounds a moving vehicle. This section describes how this basic mechanism is augmented to produce a teleoperation system.

4.1 Latency Compensation and Simulated Leader-Follower

Any real wireless communications system will introduce bidirectional latency into the telemetry passing between the vehicle and the OCS. It is well known that such latency is one factor that makes teleoperation difficult. The capacity to render the vehicle from an external viewpoint not only provides hemispherical exterior context to the operator but it also provides the opportunity to remove latency using prediction (Figure 8). Our vehicle display is virtual anyway so it is straightforward to draw the vehicle in any position. We render the vehicle at its predicted position at the time in the future when commands issued by the operator will arrive at, and be acted upon, by the vehicle. This technique produces a continuously predictive display that appears to respond with no latency. Given the capacity to predict the future, a potentially more useful technique is to predict slightly beyond the command arrival time to produce a display of a vehicle slightly more into the future. In this case, some of the prediction error has not happened yet, and the real robot can be given the responsibility to reduce the error before it occurs. This is accomplished by considering the simulated vehicle to be a lead vehicle, which the real one is obliged to follow. The simulated lead vehicle is rendered in the context of the 3D Video feed that is updated to include new information as the real (but usually not displayed) vehicle moves. Hence, the operator has the sensation that a real vehicle is being controlled. In this case, the path followed by the leader is passed to the real vehicle as the command input.

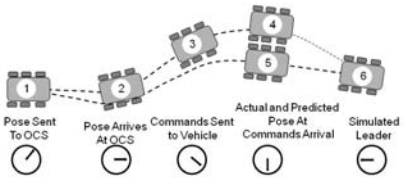

Fig. 8 Prediction for Latency Compensation. Prediction of station 5 can be used for latency-free rendering. Errors in the initial state and prediction process may place the vehicle at station 5 when it will really arrive at station 4. Such errors can be treated as following errors if a simulated leader is rendered at station 6.

4.2 Telemetry Compression

Virtualized reality creates an opportunity to implement effective data compression in remote control applications. The fact that the rendering of the model is decoupled from the frame rate of the cameras means that the update of the display, showing a moving vehicle, can proceed at high rates regardless of the camera rates and the pose update rates. This capacity also implies a high degree of robustness to dropped frames of video or pose data since each appears to be a momentary increase in input latency for which the system is already configured to compensate. The input pose data and output control signals are small enough to be irrelevant to the compression issue. Lidar data is converted from floating point to 16 bit integers. For the video, we use an Xvid (MPEG-4) encoder on the vehicle and a decoder on the OCS to perform compression. Visible parts of the vehicle are cropped. The three streams from the sensor pods are reduced to 10Hz frame rate before they are compressed. Based on just these measures, we are able to produce very high quality 3D Video displays that compete with full frame video using only 1 Mbit /sec of communication data rates.

4.3 Augmented Reality and Mixed Initiative Interactions

The capacity to produce metrically accurate photorealistic maps follows from a decade of work in using colorized range data for robot autonomy systems. Many of the data structures involved are identical or similar to those used to represent the scene for autonomy purposes. In particular, both the point cube and the terrain map are standard components of our autonomy systems, which are produced for the purpose of obstacle and hazard detection [8]. Given such algorithms, it is natural to wonder how they can be used to help the operator drive. Figure 9 shows a simple augmented reality display where the classifications of simple slope based obstacle detection algorithms are used to partially color the terrain. The colors are blended with the video such that reddish areas are to be avoided, and greenish ones are safe for driving. In benign terrain, in broad daylight, this augmented reality facility may not add much value. However, when terrain is difficult or lighting or visibility is poor, such an autonomy system could add value if the human interface were con-

figured correctly. Lidar works better at night due to reduced solar interference, and infrared appearance data can be processed like daytime video to produce a system that permits an operator to drive in total darkness. The 3D Video system also uses many of the same data structures for rendering and autonomy, so the operator and the autonomy system can interact more readily through the display; augmented reality is but one such mechanism. It is possible to have autonomy veto operator commands or bring the vehicle to stop, and the display can likely provide the operator with the reason for the robot initiative.

Fig. 9 Augmented Reality Display for Autonomy Assisted Hazard Avoidance. The photorealistic display is augmented with false color obstacle annotations. Billboards are turned off.

5 User Study Results

The 3D Video system has been under continuous testing for the last two years including three week-long tests in the northeast and southwest of the US in both winter and summer conditions. In one test, a formal user study was conducted over a week in Pittsburgh in December of 2006. The details are described more fully in [9]. Five operators of different skill levels were tested on an obstacle course designed to elicit errors known to occur commonly in teleoperation. The participants averaged 20 years of automobile driving experience. Three subjects had prior experience teleoperating a live vehicle, including one with a 3D video system. One subject had never played a driving based video game. Course features included slaloms, decision gates, discrete obstacles, and loose and tight turns. The course was difficult enough to induce errors even when driving a vehicle manually from the driver's seat. Each operator drove the course in 4 different ways including sitting in the ve-

hicle using standard controls, basic teleoperation with live video, and 3D video with and without latency. Each driving mode was assigned in random sequence to remove bias associated with learning the course. Driving performance was measured in terms of completion time, speed, and error rates.

Table 1 User Study Results

Metric	Live Video	3D Video with Latency	3D Video without Latency	Manual Drive
Completion Time (min)	9.3	7.2	6.4	2.2
Average Speed (m/s)	1.0	1.3	1.6	4.2
Errors	9.6	5.0	7.9	2.4

In all cases, 3D video produced driving performance that was significantly (30% to 60%) better than standard teleoperation but not as good as manual driving. Furthermore, operators uniformly preferred 3D video to standard video teleoperation. A bug in the latency compensation process resulted in poor performance of this aspect in the test. However, the latency compensation techniques have subsequenctly proven to be very valuable in practice after this bug was removed. In a second test conducted over a one week period in south Texas in November of 2008, 50 novice operators drove the system (based on no training) in natural terrain using augmented reality waypoint guidance. There were no mishaps or incidents but we did notice that many operators with no teleoperation experience tended to forget that a real vehicle was being controlled. During this test, one of our skilled staff also tried to drive the system at high speed for a kilometer on dirt roads while avoiding discrete obstacles. Figure 10 indicates how we achieved unprecedented speeds in this test using our latency compensation while driving from up to a kilometer away from the real vehicle.

6 Conclusion

This paper has proposed a method to expend significant engineering effort in order to convert the task of robot teleoperation into a synthetic form of line-of-sight remote control. User studies have verified substantial gains in the effectiveness of the man-machine system. Along the way, we have produced improved solutions to problems like latency compensation and data compression for which there has been little hope of significant progress for some time. While many component technologies have made this possible, the most novel is photogeometric sensing applied to virtualized reality. Photogeometric sensing has the capacity to produce displays with both the photorealism of video and the interactivity of a video game. We expect that as sensors, rendering, guidance, and communications technologies continue to evolve, such displays and their derivatives will become a standard part of our tool-

Fig. 10 High Speed Obstacle Avoidance. Latency compensation is most valuable during high speed driving. Here, the operator avoids an obstacle to the right, fitting the vehicle into a narrow space. A custom fly-behind view was used. The speedometer reads 24.55 km/hr. The operator control station is about 1 km further down the road.

box. Technologies like flash lidar with bore-sighted video for ground vehicles will hopefully come on-line and reduce the engineering effort significantly. Even in the meantime, we find the effort is worthwhile in those very difficult applications for which robots and remote control are justified in the first place. Although we have not developed the idea here in detail, our 3D Video system is basically a projective texturing engine added to visualize colorized range data sets that were already being produced for the purposes of autonomy. The mental model used by both operator and robot is virtually identical in our system and this suggests many more derived advantages will be possible in contexts where autonomy shares more of the load and human and robot cooperate more fully.

Acknowledgements

This work was funded by the US Army Tank Automotive Research, Development, and Engineering Command (TARDEC). The colorized ranging sensor concept described here was based on a concept originally developed under funding from John Deere Corporation.

References

1. R. Chatila, S. Lacroix, T. Simion, M Herrb, Planetary exploration by a mobile robot: mission teleprogramming and autonomous navigation Autonomous Robots, 1995.
2. S. E. Chen and L. Williams, View interpolation for image synthesis, in Proc. SIGGRAPH 93, pp. 279-288, 1993.
3. T. Fong, C. Thorpe, and C. Baur, "Advanced Interfaces for Vehicle Teleoperation: Collaborative Control, Sensor Fusion Displays, and Remote Driving Tools. Autonomous Robots 11, pp 77–85, 2001.
4. T. Fong, H. Pangels, D. Wettergreen , Operator Interfaces and Network-Based Participation for Dante II, SAE 25th International Conference on Environmental Systems, San Diego, CA, July 1995.
5. B. Hine, C. Stoker, M. Sims, D. Rasmussen, and P. Hontalas, The Application of Telepresence and Virtual Reality to Subsea Exploration, the 2nd Workshop on Mobile Robots for Subsea Environments, Proceedings ROV 94, Monterey, CA, May 1994.
6. B. Hine, P. Hontalas,, T. Fong, L. Piguet, E. Nygren, A. Kline, VEVI: A Virtual Environment Teleoperations Interface for Planetary Exploration, SAE 25th International Conference on Environmental Systems, San Diego, CA, July 1995.
7. T. Kanade, P. Rander, and P. J. Narayanan, Virtualized Reality: Constructing Virtual Worlds from Real Scenes, IEEE MultiMedia, Vol. 4. No. 1, 1997, IEEE Computer Society Press.
8. A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, R. Warner, Toward Reliable Off-Road Autonomous Vehicles Operating in Challenging Environments, The International Journal of Robotics Research, Vol. 25, No. 56, pp. 449-483, June 2006.
9. A. Kelly, D. Anderson, E. Capstick, H. Herman, P. Rander, Photogeometric Sensing for Mobile Robot Control and Visualization Tasks. In symposium on New Frontiers in Human-Robot Interaction, Edinburgh, Scotland, April 8-9, 2009.
10. W. S. Kim. "Virtual Reality Calibration and Preview / Predictive Displays for Telerobotics." Presence: Teleoperators and Virtual Environments 5, 2 (Spring 1996), 173-190.
11. W. Matusik, C. Beuhler, R. Raskar, S. Gortler, L. McMillan, Image-Based Visual Hulls, SIGGRAPH 2000 (July 2000), pp369-374.
12. D. A. Messuri and C. A. Klein, Automatic Body Regulation for Maintaining Stability of a Legged Vehicle during Terrain Locomotion, IEEE Journal of Robotics and Automation, RA-1, pp 132-141, Sept 1985.
13. P. Milgram, S. Yin, J. Grodski, An Augmented Reality Based Teleoperation Interface For Unstructured Environments., In Proc. American Nuclear Society (ANS) 7th Topical Meeting on Robotics and Remote Systems Augusta, Georgia, USA, April 27-May 1, 1997. pp. 966-973.
14. J. Rohlf, J. Helman, "IRIS Performer: A high performance multiprocessing toolkit for real-time 3D graphics", In Proceedings of SIGGRAPH 94 (1994), Glassner A., (Ed.), pp. 381 394.
15. M Segal, C. Korobkin, R. van Widenfelt, J. Foran, Paul Haeberli, "Fast shadows and lighting effects using texture mapping", In Proceedings of SIGGRAPH 92, pages 249-252, 1992.
16. T. Sheridan, Human Supervisory Control of Robot Systems, Proceedings of the 1986 IEEE International Conference on Robotics and Automation, Apr 1986.
17. B. Stewart, J. Ko, D. Fox, K. Konolige, The Revisiting Problem in Mobile Robot Map Building: A Hierarchical Bayesian Approach, In Proc of the Conference on Uncertainty in Artificial Intelligence, 2003.
18. G. Terrien, T. Fong, C. Thorpe, and C. Baur. "Remote driving with a multisensor user interface". In Proceedings of the SAE ICES, Toulouse, France. 2000.