IMPROVING THE VISUAL EXPERIENCE FOR MOBILE ROBOTICS

Geb Thomas, William D. Robinson, Steven Dow
The University of Iowa Department of Industrial Engineering

ABSTRACT

This paper describes how a panospheric camera improves an operator's understanding of a mobile robot's remote environment. The first section describes two of the principle challenges faced by traditional camera systems in mobile robot applications: time delay and the difficulty of finding landmarks with a narrow field of view. By imaging the bottom surface of a reflective sphere, panospheric cameras provide a 360 degree, ground to sky view of the remote environment. This highly distorted view is graphically dewarped and presented to the operator by texture-mapping the image to the surface of a specially constructed hemisphere in a three-dimensional operator interface. The technique was applied to the Nomad robot during the summer of 1997. Panospheric imaging allow novice drivers in Pittsburgh, Pennsylvania to control the robot during its 200 km traverse of the Atacama desert in Chile, South America.

INTRODUCTION

Just as drivers rely on their eyes to navigate their car down the road, robot operators use images received from on-board cameras to navigate their vehicle through a remote terrain. In many robot systems, the operator examines the images taken by the robot, decides where to direct the robot, and executes the directions through a joystick or some other input devices [1,4,8]. In contrast, some researchers have built mobile robot systems based on a model of a scene. In these systems the robot is directed through the model, and only indirectly from to images from the robot's cameras [7,8]. Other researchers have integrated both the modeling and immediate image approaches into a single system [2,9,10,12]. In these systems the operator uses both the model and the live images to navigate the robot through its remote environment.

In systems where the images are the principle source of information, their quality, along with vehicle responsiveness and ability to surmount obstacles, are the key aspects of a robot's driveability. The images help the operator see the upcoming obstacles such as rocks or people, find distant landmarks for navigation, and determine how far the robot has traveled. Often the power of a robot to deliver high quality imagery is subverted by designers' interest in mechanical systems. In applications where the robot is primarily a moving eye, however, it may be appropriate to design the robot around the image acquisition system. For example, the Dante II and Marsokhod robots both had geological missions to accomplish – one to retrieve volcanic gas samples and one to find interesting rocks in an extra-terrestrial environment. In both these cases, robot operation was dominated by the task of looking at images and deciding where to move. For these systems, the imaging system was a crucial part of the robot's performance.

This paper describes a camera and a user interface vision system that fundamentally improves the way images from mobile robots are acquired and displayed. It begins by describing the problems of time delay and landmark finding encountered by the operators of the Dante II and Marsokhod robots. Next, the paper describes the panospheric camera and display system developed by the authors and used for the Nomad robot. The Nomad robot was operated by novice drivers as it traversed 200 km across the Atacama desert in Chile, South America during the summer of 1997. This experiment illustrated the powerful synergy that can exist between the robot and the operator when the quantity and quality of information between the human and machine is improved.

CHALLENGES FACED BY MANY MOBILE ROBOTS

In July 1994 Dante II, an 8-legged robot, was used to descend into Mt. Spurr to gather data. The robot took stereo images using cameras with pan/tilt capability. These cameras were used to check for obstacles in their path, or to confirm the robot's position. Many times the images were limited by the field of view, causing the operator to be disoriented, and forcing him or her to focus on finding the context of the image rather than a safe position to move to. With the fixed mounted cameras, the operator has trouble seeing beside him without backing up. In February 1995 a planetary rover, Marsokhod, was deployed on Kilauea Volcano HI. Marsokhod was equipped with a set of pan/tilt cameras, which allowed the operator to look in new directions.

Time Delay

Another issue when combining mobile robotics with a vision based system is time delay. After operator sends the command to tilt the camera, he/she receives the new image moments later. The operator must mentally verify that the correct change of position happened. This cognitive load is what has caused many robots/operators to become lost. One way an operator can overcome this setback is by using landmarks to remind him/her of where the robot was. There are limitations to this, though.

Landmarks

Even with its impressive arsenal of cameras, Marsokhod encountered many problems regarding losing its sense of direction. The robot did not have any video cameras rather still images would be taken and sent back to the operator (common in mobile robotics). In one image the operator would see a landmark and send a command to move forward. A new image would be taken and sent to the operator. Many times the landmark was not in the image. The operator would send a command to pan the robot's camera right and take another image. The operator would repeat this process until the landmark was found. Many times the landmark would be right next to the robot, but the field of view would distort the image to point the landmark was undistinguishable. If the landmark was not located, the robot would ultimately have to back up. This leads to much frustration by the scientists. As the illustration shows, a robot has only 90 degrees of freedom of which to move without losing a landmark and thus risking becoming lost (Figure 1).

SOLUTION: PANOSPHERIC IMAGING

A panospheric camera may be the next generation of vision based control in mobile robotics. The panospheric camera displayed in illustration 2000 has 360 degrees of vision. Unlike the pan and tilt cameras used on Marsokhod, the panospheric camera simultaneously captures information on all sides of the robot without forcing the operator to send a movement command to the camera gimble.

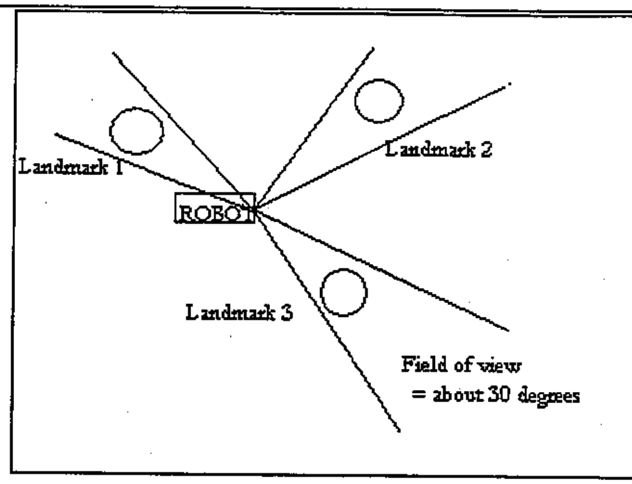

Figure 1: A robot with normal cameras can see approximately 30° in any particular direction. If there are only three landmarks, the cameras only allow 90 degrees of freedom.

The panospheric camera is also superior to the pan/tilt camera by giving the operator more cognitive freedom. The camera acts as a kind of predictive display. The operator can see where the robot will be positioned before he actually sends the signal to move the robot. In addition, when the operator moves the robot in a certain direction, he does not need to remember where he has been since his previous position is still in sight.

The use of landmarks is not an issue since 360 degrees field of view can be utilized. No longer will the robot have to reverse its direction to locate the landmark since the panospheric camera allows the operator to see behind the robot.

Nomad Robot

The Nomad robot avoids many of these difficulties with a panospheric camera developed by John Murphy of Carnegie Mellon's Field Robotics Center. The color camera in this system images the underside of a reflective sphere above it. The reflections from the spherical mirror include information about the scene all the way around the robot, with elevation angles ranging from straight down to 0.436 radians above the horizon.

An example image from the panospheric camera is shown in Figure 2. The large circle in the center of this image is the reflection of the camera lens. Surrounding that is the body of the robot, which is about the size of a compact car. To the left and right are the robot's navigation stereo camera pairs. Below is the directional pointing antenna, which provides a high-bandwidth communication channel for sending the images off the robot. Around the robot is the rocky terrain of the Atacama desert and the blue sky.

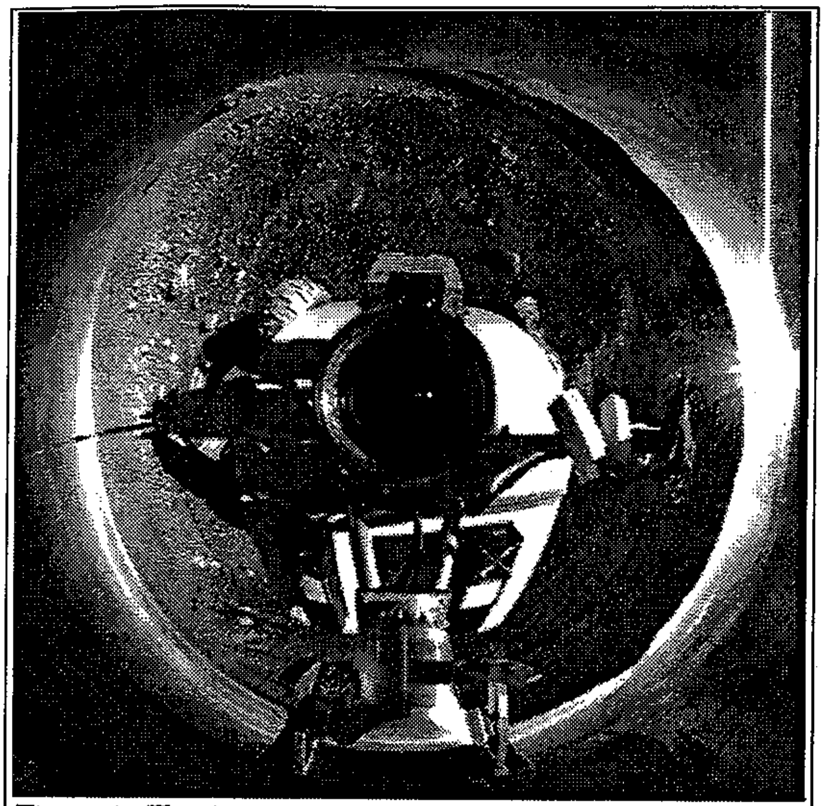
Figure 2: The Atacama Desert in Chile viewed with Nomad's panospheric camera.

During operations in Chile's Atacama Desert in June and July 1997, a two-processor PC compressed each image and transmitted them via a pointed antenna system base station, a satellite link, and T1 lines. The images were received at the Carnegie Science Center in Pittsburgh and the NASA Ames Research Center in Moffett Field, California. At both locations, a Silicon Graphics Workstation decompressed, dewarped, and rendered the images. A Magellan device, a 6-degree-of-freedom input device similar to a spaceball, enabled the operator to pan and zoom in the image and to tilt viewing direction from the horizon to the straight down. At the Carnegie Science Center, the thirty-person audience in the panoramic Electric Horizon Theater controlled the pan direction by voting with buttons in front of each seat. The 512 X 512, 24-bit images were received, decompressed, dewarped, and swapped in as new textures at a maximum rate of 6 frames per second on both a two-processor Octane and a four-processor Onyx. Interaction with the 3D model, such as turning the viewpoint from straight ahead to the left, flowed smoothly at 30 frames per second, independent of the image acquisition rate.

Apple's QuickTime VR™ provides a similar experience for interacting with an immersive image, except that the image does not change [3]. Only the operator's viewpoint within the image changes. Dr. Shree Nayar at Columbia University has also pursued real-time panospheric cameras and displays [14]. His work differs from the work reported here in two ways: his Omnicamera displays 1) are lower resolution for use on PCs, and 2) require the lens to have a single center of projection, which is not necessary with our texture mapping approach. A number of other researchers have also investigated panospheric imaging [13, 16, 17], including the use of spherical reflectors [11].

This next section describes the imaging geometry of the panospheric camera and the software techniques used to achieve the high update and interaction frame rates.

The Imaging Geometry

The images from the panospheric camera were mapped onto a curved surface that was tailored to the imaging geometry so the image would be correctly dewarped. This approach was not immediately obvious. Other techniques seemed to be less complex, but ultimately did not provide the flexibility and speed afforded by our texture-mapping approach.

For example, if the interface restricted the viewpoint to directions along the horizon, dewarping the image to a single Mercator projection would have been sufficient. Paging through vertical strips of the image could efficiently render the cylindrical projection. This approach might lead to large interactive frame updates, but would distort the image of the ground just in front of the robot, an important image region. As Figure 3 illustrates, Mercator projections present the least directional distortion about some great circle in the viewing sphere. However, for regions far from this circle, the image is highly distorted. Mercator projections enlarge features far from the horizon, which would cause several undesirable and distracting artifacts for the operator. For example, a Mercator projection would visually distort the robot's body and cause rocks to appear to swell as they moved closer to the robot's wheels. An accurate projection of the ground in front of the robot would require an oblique projection, which would unevenly distort points along the horizon. Furthermore, the projection would need to be recalculated if the operator wished to look at the ground to the left or right of the robot, at the cost of rendering efficiency. Our texture-mapping approach enables the correct projection to be efficiently calculated, regardless of the viewing direction.

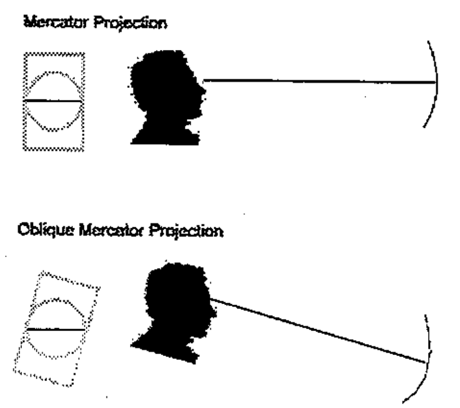

Figure 3: A Mercator projection provides an undistorted view anywhere on the horizon, but has to be recalculated as an oblique projection if the operator wishes to view the ground in front of the robot.

Our texture-mapping approach makes use of the software and hardware already available on graphics workstations. For example, a simple and effective technique to implement the Mercator projection would be to create a three-dimensional model of a cylinder and map the precalculated Mercator projection onto its surface. The projection origin could be placed in the cylinder's center and rotated as the operator panned through the image. The texture-mapping hardware and software would attend to the details, shifting appropriate portions of the image from texture memory to the display, and scaling, warping, and wrapping around as necessary. Many three-dimensional modeling languages, such as the Virtual Reality Modeling Language (VRML), define a default image mapping for textures that would adequately map the Mercator projection onto the cylinder. Using one of these languages would make the Mercator approach easy to apply.

But the Mercator projection approach distorts some image regions. Rather than risk introducing confusing misinformation to the robot interface, we chose to develop a different projection strategy which creates a unique, efficient dewarping process that is optically correct in every viewing direction.

The Nomad interface texture-maps the image onto a hemispheric surface that is perpendicular to the imaging rays at every point. The relationship between a position in the image and a ray in three-dimensional space is described by the following equation:

$$F(x, y) = P_0 + \alpha P_1 \qquad [1]$$

where x, y are image coordinates with respect to the lower, left-hand corner, P_0 is the origin of the ray, P_1 is direction of the ray, and $\alpha >= 0$.

This relationship between the imaging ray and the image coordinates will be developed in a series of three steps. Step 1 finds the ray's heading, assuming the origin is on the camera's axis of symmetry. Step 2 finds the ray's elevation, assuming that all the reflections off the spherical surface happen negligibly close to the hemisphere's center. Finally, Step 3 adds a term that corrects for the assumption of Step 2.

Step 1: The Azimuth orientation vector

For any image location (x, y), the ray's azimuth, γ, depends on (Cx, Cy), the point where the axis of symmetry passes through the image plane according to the relationship

$$\gamma = \arctan[(y - Cy)/(x - Cx)] \quad [2]$$

The values of Cx and Cy may be determined directly from a sample panospheric image by finding the center of the camera lens's circular reflection in the image (Figure 4).

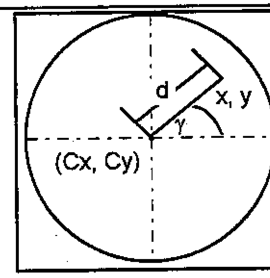

Figure 4: Each point in the image (x, y) is characterized in cylindrical coordinates by γ and d.

Step 2: The elevation angle

Each ray's elevation angle, ϕ, depends on θ, the angle at the focal point between the axis of symmetry and a point of reflection on the mirror's surface. Let f be the distance along the axis of symmetry between the focal point and the image plane (Figure 5). Simple trigonometry defines θ as:

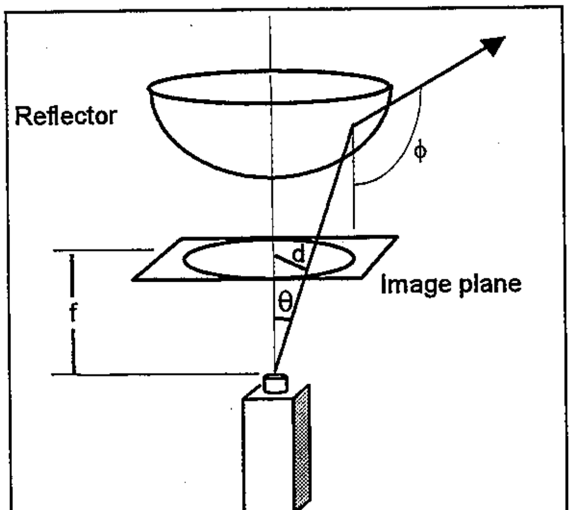

Figure 5: The measured distances of D and d define θ.

$$\theta = \arctan(d/f) \quad [3]$$

Now consider Figure 6. Let ρ be the angle at the center of the sphere between the axis of symmetry and the ray's point of reflection on the sphere's surface. Let r be the radius of the sphere. It is necessary to find ϕ, the angle between the vertical axis (assumed to be parallel to the axis of symmetry) and the incoming ray of light that would be imaged in the camera by following the ray along θ.

Summing to π the angles of the triangle formed by θ, D and ρ and applying the law of sines gives:

$\phi = 2 \arcsin(D \sin\theta / r) - \theta$ [4]

Step 3: Correction for the Offset

Step 2 assumed that the ray's focal point was at the center of the reflector, which is not precisely correct. All rays intersect with the reflector's axis of symmetry, but rays imaged near the reflector's edge intersect this axis lower than rays near the center of the reflector's image. The distance from the intersection point to the reflector's center is h. Referring again to Figure 6, by the law of sines,

$h = r \sin(\rho + \phi)/\sin(\pi - \phi)$ [5]

Combining the information obtained in the above steps, the vector equation for imaged rays is given by:

$f(x, y) = [0, 0, D - h] + \alpha\, [(\sin\theta)(\cos\gamma), \sin\phi \sin\gamma, -\cos\phi]$, [6]
for any $\alpha >= 0$

Construction of the Three-Dimensional Viewing Screen

The curved surface is simply a connected mesh of points some distance along selected image rays. The pattern for connecting the points on the rays is the same as the pattern in the original image. The normalized image coordinates were also associated with each vertex to form the (u, v) texture map coordinates between the image and the surface.

For the Nomad project a tesselated surface was generated with points 1000 units along imaging rays. The selected rays were associated with a square grid subsample of the image pixels. Large grid steps (greater than 32 pixels) created surfaces with easily visible facets. Small steps (less than 16 pixels) created surfaces with complex geometry; but these surfaces were indistinguishable from the simpler surfaces created with 16-pixel steps.

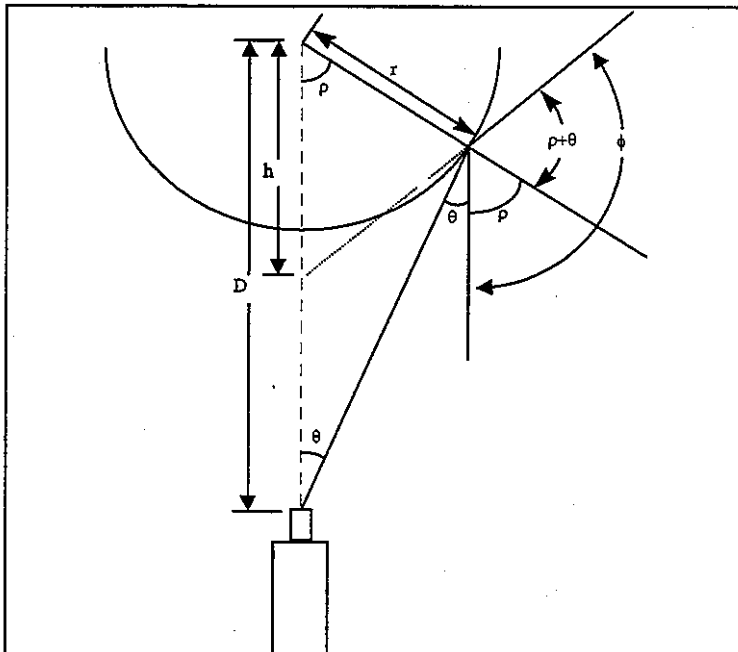

Figure 6: Incoming light rays strike the spherical mirror at the top and are reflected into the camera at the bottom. The reflected rays make the angle $\rho + \theta$ with the normal of the reflecting surface.

With the viewpoint at the sphere's center, the texture-mapped surface presents a correctly dewarped version of the remote scene. A zoom feature enables the operator to move toward or away from the spherical surface along the line of sight. When the viewpoint moves from the center of the hemisphere, the image becomes gradually distorted, but in an intuitive manner, as shown in Figure 7.

The-End-To-End process

The final process consisted of the following steps. An image collected by the CCD was sent to a frame grabber on the robot's two-processor Windows NT machine. This PC reduced the image from 786,432 bytes to approximately 75,000 bytes by a lossy wavelet compression algorithm. This image was sent from the robot to a relay station through a pointing antennae pair, and then to Pittsburgh via satellite, where a T1 line carried the image to the Carnegie Science Center, and to the Silicon Graphics four-processor Onyx. On one of these processors a program (developed by Kurt Schwehr, of the NASA Ames research center) collected and assembled the compressed image packages and sent them via shared memory to the dewarping program. The dewarper converted the wavelet-compressed image back to a standard image, then sent it through another shared memory segment to the rendering process. Finally, the rendering process loaded the image into texture memory using SGI's fastload extension to the OpenGL standard. Since the image was collected, decompressed, and formatted in a separate, asynchronous process, loading new images did not affect the rendering frame rate during interaction with the sphere and the viewpoint. Once in texture memory, the new image was texture-mapped onto the curved survace. Three images of this sphere were taken with each refresh. The images differed by 30 degrees, to match the angular separation of the three projectors in the Electric Horizon Theater. These projectors displayed the images onto the curved projection screen. The delay from the robot to the theater was approximately 500 milliseconds, and the images were refreshed up to 6 times a second. When the operators moved their viewpoint within the sphere, the changing viewpoint refreshed at 30 frames per second to create a smooth, continuous animation of movement around the curved display in the virtual environment.

Figure 7: The hemispherical surface with a panospheric image seen from outside the surface. The wide field of view image is distorted, but is still easy to interpret.

CONCLUSIONS

This paper presented three items: 1) A discussion about mobile robotics and their limitations, 2) the mathematics involved in panospheric imaging, a technique to dewarp and display panospheric images conveniently to an operator, and 3) the design of an imaging system that allows a robot on one continent to be viewed and driven by operators on another continent. An increasingly important issue in mobile robotics is the realistic sense of remote telepresence. It is essential that the operator has an intuitive sense of the robot's environment. Time delay demands a heavy cognitive burden since the operator must remember where he or she has been. The panospheric camera relieves the operator of this mental effort by displaying the full 360 degrees of the remote environment.

Wide field of view imaging is also useful when landmarks are infrequent or long delays between images confuse the operator. A panospheric camera provides a reliable and powerful mechanism to acquire wide field of view images. These distorted images can not only be dewarped and presented to an operator efficiently and simply in real time, they can also enable the operator to interactively adjust the center of attention and the level of magnification. The key to this approach is to take advantage of the texture mapping hardware that is becoming widely available on a number of graphics workstations and PCs. This application of texture mapping provides the operator with a complete sense of the surrounding terrain and lets him or her explore without being hamstrung by time lags to control remote pan and tilt gimbals. This technology demonstrates how computer graphics can fundamentally improve the effectiveness of remote operations.

ACKNOWLEDGEMENTS

The author would like to thank the Iowa Space Grant Consortium and the Intelligent Mechanisms Group (IMG) at NASA Ames, who funded this work. Particular thanks are due to Michael Sims and David Wettergreen, who managed the IMG's participation in the project; Kurt Schwehr, who wrote the image communication code; Maria Bualat, who wrote the operator interface; and Hans Thomas, who helped debug the final code and interface device. I would also like to thank the members of the Field Robotics Center at Carnegie Mellon University, including William "Red" Whittaker, who conceived of and led the Nomad project, and John Murphy, who designed and built the panospheric camera. Thanks also to Mark Maimone and Eric Rollins who managed and led the team's operations, Deepak Bapna who built the pointing antenna and arranged the satellite communication system, and Martin Martin who coordinated the stateside efforts. Many grateful thanks to Lowry Burgess, Carl Loeffler, Summus and the Carnegie Science Center.

BIBLIOGRAPHIC REFERENCES

1. Backes, P.G., and Tso, K.S. (1990). UMI: an interactive supervisory and shared control system for telerobotics. *1990 IEEE International Conference on Robotics and Automation,* **2**: 1096-1101.
2. Cannon, D.J., Thomas, G., Wang, C., and Kesavadeas, T. (1994). A virtual reality based point-and-direct robotic system with instrumented glove. *International Journal of Industrial Engineering,* **1**(2): 139-148.

3. Chen, S.E. (1995). QuickTime VR – an image based approach to virtual environment navigation. *Computer Graphics: proc. of SIGGRAPH '95*, 29-38.
4. Conway, L., Volz, R.A., and Walker, M.W. (1987). Tele-autonomous systems: methods and architectures for intermingling autonomous and telerobotic technology. *1987 IEEE International Conference on Robotics and Automation*, **2**: 1121-1130.
5. Choi, S.K., and Yuh, J. (1993). Design of advanced underwater robotic vehicle and graphic workstation. *1993 International Conference on Robotics and Automation*, **2**: 99-105.
6. Christensen, B., Drotning, W., and Thunborg, S. (1991). Graphical model based control of intelligent robot systems. *1991 IEEE International Conference on Systems, Man, and Cybernetics*, **2**: 1069-1075.
7. Christensen, B.K., and Desjarlais, L.M. (1990). A graphical interface for robotic remediation of underground storage tanks. *Proceedings of the First IEEE Conference on Visualization*: 449-456.
8. Diner, D.B., and Venema, S.C. (1989). Graphic overlays in high-precision teleoperation: current and future work at JPL. *Proceedings of NASA Conference on Space Telerobotics*, **3**: 511-520.
9. Fong, T., Pangels, H., et al. (1995). Operator Interfaces and Network-Based Participation for Dante II, SAE 25th International Conference on Environmental Systems, San Diego, CA, July 1995.
10. Hine, B.P. III, Stoker, C., et al. (1994). The application of telepresence and virtual reality to subsea exploration, *AIAA*, Reno, NV 1996.
11. Hong, J. (1991). Image based homing. *Proc. of IEEE Int. Conf. On Robotics and Automation.*
12. Kim, W.S. (1993). Graphical operator interface for space telerobotics. *1993 IEEE International Conference on Robotics and Automation*, **3**: 761-768.
13. Nalwa, V. (1996). A true omnidirection viewer. Technical report, Bell Laboratories, Holmdel, NJ.
14. Nayar, S. (1997). Catadioptric omnidirectional camera. *Proc. of IEEE Computer Vision and Pattern Recognition Conference*, 482-488.
15. Thomas, G., Blackmon, T., Sims, M. and Rasmussen D. (1997). Video engraving for virtual enviornments; *Electronic Imaging: Science and Technology '97*, San Jose, California.
16. Yagi, Y. and Kawato, S. (1990). Panoramic scene analysis with conic projection. Proc. of Int. Conf. On Robots and Systems (IROS).
17. Yamazawa, K., Yagi, Y., and Yachida, M. (1995). Obstacle avoidance with Omnidirectional image sensor HyperOmni Vision. *Proc. of IEEE Int. Conf. On Robotics and Automation*, 1062-1067.