2013 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
AUTONOMOUS GROUND SYSTEMS (AGS) MINI-SYMPOSIUM
AUGUST 21-22, 2013 – TROY, MICHIGAN

TORIS: A SYSTEM FOR SMOOTH GROUND VEHICLE TELEOPERATION IN HIGH LATENCY CONDITIONS

Karl C. Kluge, Ph. D
Alberto Lacaze, Zach La Celle, Steve Legowik, Karl Murphy, Rob Thomson
Robotic Research, LLC
555 Quince Orchard Road, #300
Gaithersburg, MD 20878

ABSTRACT

Latencies as small as 170 msec significantly degrade ground vehicle teleoperation performance and latencies greater than a second usually lead to a "move and wait" style of control. TORIS (Teleoperation Of Robots Improvement System) mitigates the effects of latency by providing the operator with a predictive display showing a synthetic latency-corrected view of the robot's relationship to the local environment and control primitives that remove the operator from the high-frequency parts of the robot control loops. TORIS uses operator joystick inputs to specify relative robot orientations and forward travel distances rather than rotational and translational velocities, with control loops on the robot making the robot achieve the commanded sequence of poses. Because teleoperated ground vehicles vary in sensor suite and on-board computation, TORIS supports multiple predictive display methods. Future work includes providing obstacle detection and avoidance capabilities to support guarded teleoperation and supervised autonomy in high-latency environments.

INTRODUCTION

This paper describes TORIS (Teleoperation Of Robots Improvement System), a tool for teleoperation in high-latency environments. TORIS mitigates the impact of system latency on teleoperation performance through two techniques:

- a predictive display that presents the operator with an estimated latency-corrected view of the robot's position in the environment at the time the current operator command input is received by the robot, and
- joystick command primitives representing desired robot orientation and forward travel distance rather than the usual differential speed and steering rate inputs, removing the operator from the high update rate parts of the robot's control loops.

The first section of the paper reviews related work, broken into two subsections. The first of these discusses human factors research examining the impact of system latency on operator performance in remote driving tasks. The second subsection describes typical examples of related methods applicable to mitigating latency by reducing operator-intensive control of the robot (guarded teleoperation using reactive obstacle avoidance, control via operator-specified waypoints, visual servoing, etc.).

We then present an overview of the current TORIS system design, describing the system architecture and highlighting the predictive display techniques that have been explored and the control strategy. This is followed by a description of planned extensions to TORIS.

RELATED WORK

Multiple system parameters affect ground vehicle operator performance during teleoperation, including display magnification factor, camera field of view, image resolution, frame rate, and latency [8], [2]. This work focuses on mitigating the impact of system latency on driving performance. This section provides a brief overview of human factors studies relating to the impact of latency, followed by a discussion of representative examples of techniques that have been proposed for mitigating the impact of latency by reducing the need for intensive, high update rate operator interaction while teleoperating.

Studies of the impact of latency on teleoperation control

The survey article by Chen, et al [2], lists the results of a number of studies examining the effects of latency on performance:

- System latencies above one second lead operators to switch to a "move and wait" control strategy

- Latencies as small as 300–320 msec significantly degrade operator's compensatory pursuit tracking performance
- "In a simulated driving task, the driver's vehicle control was found to be significantly degraded with a latency of only 170 ms"

This survey also discusses the interaction between latency and frame rate on performance (the studies listed show latency as the more important factor), and the negative impact of latency as a factor responsible for operator motion sickness. Studies are cited demonstrating the value of predictive displays ("Predictive displays, using the teleoperator's control inputs, 'simulate the kinematics without delay and immediately display graphically the (simulated) system output, usually superposed on the display of delayed video feedback from the actual system output'") in reducing the negative effects of latency.

Pausch and Crea's literature survey of military flight simulator visual systems and simulator sickness [7] references several studies that discuss the impact of latency on pilot performance in flight simulation tasks, including one that concludes that a latency of 183 msec is "marginally acceptable for performance."

Luck, et al. report [6] on an experimental test of the impact of latency on ground robot teleoperation performance that examines the impact of two factors:

- Level of Automation: experiments compared teleoperation with full operator control, guarded teleoperation, waypoint navigation with autonomous obstacle avoidance (OA), and full autonomy including path planning
- Latency parameters: independently varied parameters included duration, variability, and direction (user-to-robot vs. robot-to-user)

The task used in the experiments was a building reconnaissance exercise. Both drive errors (collisions, over/undershooting on turns, corrective backup maneuvers, etc.) and time to complete the test course were smaller for the two higher autonomy conditions (waypoint navigation with OA and full autonomy) than in the full and guarded teleoperation conditions. Latency duration had a significant effect on time to complete the test course in all cases; low levels of autonomy (full and guarded teleoperation) showed increases in driving errors with higher latencies.

Techniques for reducing operator intensive interaction during teloperation

A number of techniques are described in the literature to mitigate the effects of system latency and bandwidth limitations by reducing the need for high frequency operator interaction to remotely drive a ground vehicle:

Guarded teleoperation with reactive obstacle avoidance

In this method, perception and control algorithms running on the robotic vehicle detect obstacles in the local environment and prevent the operator from driving the vehicle into those obstacles. A typical example of a reactive obstacle avoidance method is the Vector Field Histogram (VFH) method described by Borenstein and Koren [1]. The VFH algorithm builds a grid-based map of the local environment. The value contained in each cell is an estimate of the probability that the cell contains an obstacle derived from combining multiple sonar readings collected over time as the robot moves. The active region, a moving robot-centered window in the grid map, is processed to create a polar histogram whose elements represent the estimated obstacle density associated with the cells along a given slice of azimuth relative to the robot's heading. The polar histogram is processed to find the best low-density valley in the histogram values to select the current driving direction.

Automated driving between operator-selected waypoints

Interaction intensity can be further reduced by having the operator manually select a sequence of waypoints in imagery from the robot, with the robot autonomously driving along the path connecting those waypoints. If the robot includes a range sensor calibrated with the camera, the 3-D locations of the waypoints can be recovered by intersecting the lines of sight through the selected waypoint pixel locations with the 3-D model constructed from the range data. If no 3-D sensor is available, the 3-D waypoint locations can be approximated using a flat earth model. Kay and Thorpe [3] present STRIPE (Supervised TeleRobotics using Incremental Polyhedral Earth Reprojection), a more sophisticated version of the monocular flat earth technique that uses robot pitch information to incrementally update the estimated camera tilt as the robot drives along the path, updating the estimated 3-D waypoint locations appropriately.

Visual servoing to operator-specified goal locations

The performance of the waypoint based technique described above depends on the accuracy with which the waypoint locations can be estimated from image pixel coordinates. The technique of visual servo control removes this limitation by using the visual appearance of the area around a target waypoint location to track the target location in the robot's camera imagery over time. The results of the target location tracking are used to generate the robot steering commands needed to guide the robot to the desired target location.

A variety of techniques are described in the literature for adapting the tracker's model of the target location appearance to adapt to changes in scale, differences due to motion parallax, etc. Kim, et al. [4] describe visual servo based robot control implemented for use on NASA's Mars Exploration Rover, an especially challenging high-latency teleoperation application given the round-trip signal transmission time between Earth and Mars, and the computing hardware limitations of the planetary rover.

OVERVIEW OF THE TORIS SYSTEM

TORIS uses a different strategy for eliminating the operator-intensive component of teleoperation than the image-based waypoint following and visual servoing approaches described above. Instead, TORIS replaces the differential speed and steering joystick inputs used in traditional teleoperation with absolute forward distance travelled and vehicle orientation inputs. Figure 1 below shows a functional block diagram of the current TORIS system, which we refer to as TORIS Basic.

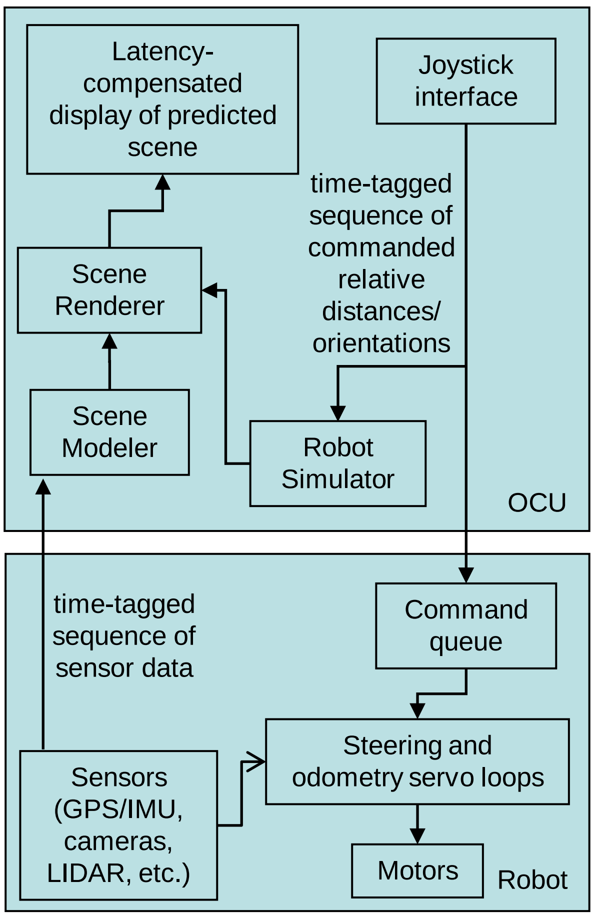

Figure 1: Block diagram of the TORIS system. Clock synchronization allows accurate time tagging of all sensor and command information passing between the OCU and the robot. Control loops on the robot act to make the robot's behavior match the sequence of (time, forward distance, orientation) commands input by the operator.

Clocks on the robot and the Operator Control Unit (OCU) are synchronized at system initialization to permit accurate time tagging of all information transmitted between the robot and the OCU. The system has two main components, a predictive display component that provides the operator with a predicted view of the position of the robot relative to the environment at the time the robot will receive the current command input from the OCU, and a control component that closes the low-level motor servo loops on the robot in order to make the robot motion match the operator-input sequence of desired (time, travel distance, robot orientation) commands.

Figure 2: TORIS OCU interface. The OCU display shows the operator a latency-corrected predicted view of the robot's relationship to the environment, and the user inputs commands representing desired travel distance and robot orientation using a standard game controller.

Predictive display to compensate for system latency

TORIS is designed to support multiple techniques for predictive display in order to allow use of the system on robots and OCUs with differing sensor suites and available computing resources. TORIS development to date has focused on predictive display methods using monocular imagery with no 3-D scene structure estimation; texture mapped range data using either calibrated LIDAR and camera data or imagery from a Kinect or Kinect-like structured light sensor; and monocular imagery with sparse scene geometry estimation using structure from motion. The first two of these modalities are the most mature with regard to development for and integration with the TORIS system, and are described in more detail below.

In the simplest case, the robot has only cameras providing monocular imagery of the environment and proprioceptive sensors providing estimated robot pose (location and orientation). The most recent image received from the robot's camera can be transformed to present a rotated view (without any reconstruction of the 3-D structure of the scene), and the robot can be drawn into the image based on a flat earth assumption and the predicted location and orientation for a given time based on the sequence of operator commands. Figure 3 below shows an example of this predictive display mode. The accuracy of the rendered robot location in the image depends on how well the flat earth assumption applies matches the local environment around the robot.

If the robot is equipped with sensors that provide calibrated range and electro-optical (EO) imagery, then the image data can be texture mapped onto a model of the scene geometry derived from the range data, and the operator command sequence can be used to determine the sensor position and orientation at some future time for the predictive display. Depending on the nature of the sensor suite, the range and EO data may be

- automatically registered (as in the case of a stereo sensor, where the depth image is inherently registered with one of the stereo pair images),
- registered as part of the sensor itself (as in the Kinect and similar structured light sensors, where internal calibration is used to register the color images with the depth data), or
- registered via calibration of the intrinsic and extrinsic parameters of the individual sensors (as in the case of texture mapping color imagery from a camera onto a triangular mesh model of the scene generated from range data coming from a scanning LADAR).

Figure 3: Example of predictive display using monocular imagery. The camera point of view is rotated to track the robot, transforming the image appropriately, and the predicted robot location is drawn in the image using a flat earth model and the trajectory implied by the sequence of commands input by the operator.

Figure 4 below shows an example of a simple predictive display generated using data from an ASUS XtionPro LIVE structured light sensor. Similar in its principle of operation to the Kinect, the XtionPro projects a pattern of infrared (IR) light that is detected by an IR-sensitive camera offset from the projector by a known baseline. Knowledge of the projected pattern allows the sensor to directly measure pixel disparity without solving the stereo matching problem. Color images from a second camera are registered with the depth information using factory-set calibration between the color and IR cameras. While the Kinect and similar sensors suffer from limitations (in particular, poor performance in outdoor environments and limited fields of view), they provide surrogates for use in software development in anticipation of improved sensors.

We are also developing a predictive display mode using color imagery texture mapped onto 3-D data from a scanning LIDAR using the camera/LIDAR sensor head developed by Robotic Research for the MAGIC 2010 competition [5]. This sensor head provides 360 degree coverage of the environment around the robot, providing clear field of view advantages relative to the structured light sensors, but the need to correct the range data for robot motion during the LIDAR scan process requires more complicated processing,

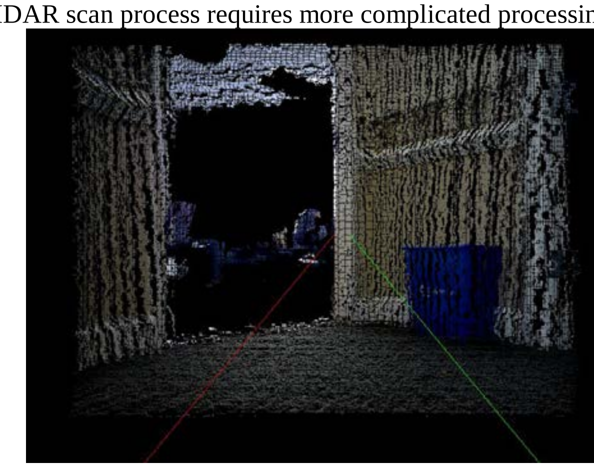

Figure 4: Predictive display using colored point cloud data from an ASUS XtionPro LIVE structured light sensor. This mode is being enhanced to use texture mapping of the color image data onto a mesh created from the point cloud data.

Robot control in TORIS

In conventional teleoperation, joystick inputs are normally used to command rates – larger deflections are used to indicate higher speeds and turn rates, with the direction of deflection indicating the sign or direction of the motions. Including the remote operator in the high-frequency parts of the robot control system through rate inputs of this type is a major source of performance problems in high latency scenarios.

TORIS uses the magnitude of the joystick deflection to indicate relative distance to travel and relative change in orientation rather than rates. The high-frequency control loops to make the vehicle perform those actions are closed on the robot instead of involving the operator directly.

The clocks on the robot and OCU computers are synchronized to permit accurate time tagging of all sensor and command information passing between them. A control module on the robot is responsible for making the vehicle adhere to (as far as physically possible) the operator-specified sequence of (time, distance, orientation) commands in order to keep the predictive display on the

OCU matching with the robot location and pose in the environment.

NETWORK SETUP FOR TESTING TORIS

In order to support controlled evaluation of TORIS performance, we have implemented a network setup that allows independent control of network latency in both directions (robot → OCU and OCU → robot). System changes are normally tested with end-to-end latencies of ~1.5 seconds. We can also control sensor data frame rate, with tests showing reasonable performance at 5 frames/second and 2 seconds round trip latency. Tests with round trip latencies of 4 seconds show some degradation in TORIS performance, but control is still qualitatively better than standard teleoperation under those conditions. We plan to perform more quantitative evaluations during the coming year.

PLANNED EXTENSIONS TO TORIS

TORIS Intermediate: TORIS with reactive obstacle avoidance for guarded teleoperation

The ability of TORIS' predictive display capability to present an accurate latency-corrected prediction of the robot's future relationship to the environment is limited by deviation of the robot from the commanded sequence of actions due to slip or other causes, and by the presence of moving obstacles in the environment. In order to mitigate the possibility of collisions with objects near the robot as a consequence of these limitations, the TORIS Intermediate system will enhance the TORIS Basic system by adding reactive obstacle avoidance to provide guarded teleoperation.

We plan to adapt the Vector Field Histogram method for reactive obstacle avoidance described above in the related work section, and illustrated below in Figure 5. The operator will be allowed to specify the maximum allowable deviation from the commanded (time/forward distance/orientation) sequence, and the system will stop and signal the operator when an excessive deviation forces resynchronization of the OCU and the robot.

TORIS Advanced: TORIS with autonomous path planning and obstacle avoidance to reach operator-specified goal locations

The TORIS Advanced system will further reduce operator interaction and loading and improve reliability by adding higher level autonomous capabilities to the system. The operator will be able to specify a goal location in imagery in a manner similar to the waypoint or visual servo methods described in the related work section, and sensing and processing on board the robot will autonomously detect and avoid obstacles while driving to the goal. TORIS Advanced will utilize the mapping and planning capabilities Robotic Research has developed for applications like the MAGIC 2010 competition [5].

Robotic Research's High Maneuverability Planner (HMP) will be a key component of TORIS Advanced's autonomy package. This kino-dynamic planner has been utilized by U.S. Army unmanned platforms for a variety of programs including the ARL Robotic Collaborative Technology Alliance, TARDEC's Safe Ops program, and the Army Future Combat System Autonomous Navigation System.

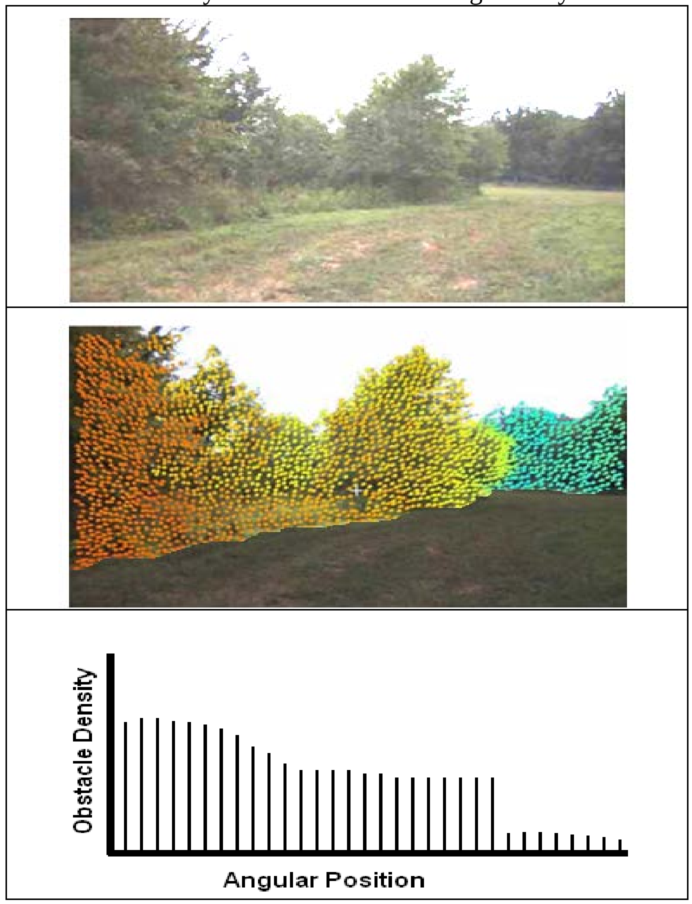

Figure 5: Illustration of Vector Field Histogram (VFH) reactive obstacle avoidance. The top picture shows a color image of an outdoor scene. The middle picture shows the same scene with the range to obstacles above the ground surface color coded (warmer red shades = closer, cooler blue shades = further). The bottom picture shows the obstacle density VFH – the horizontal axis is azimuth relative to the current vehicle heading; the vertical axis is obstacle density along that azimuth, with bearings having closer obstacles showing a higher obstacle density.

The HMP generates trajectories for the robot that avoid obstacles while meeting the constraints imposed by the operator (no-go areas, speed limits, etc.). The input to the HMP is a moving robot-centered 3-D representation of the local environment. It outputs a trajectory to be followed by a path tracking module on the robot that controls wheel speeds and steering.

The HMP combines all sensor information into a single representation of the environment that is then utilized to evaluate the cost of performing different actions. As such, the environmental representation includes morphological information, as well as slippage characterization. The resulting trajectories are sequences of vehicle state/time pairs that the vehicle control processes follow. Among other things, the state information includes the desired vehicle position and attitude. By including kino-dynamic constraints, it can protect the robot from accidents involving operator over-steering as well as collisions with obstacles. Figure 6 below shows an example in simulation of the HMP planning a robot trajectory to climb a staircase, turn, and go down a hallway while avoiding rubble obstacles and meeting kinematic vehicle constraints.

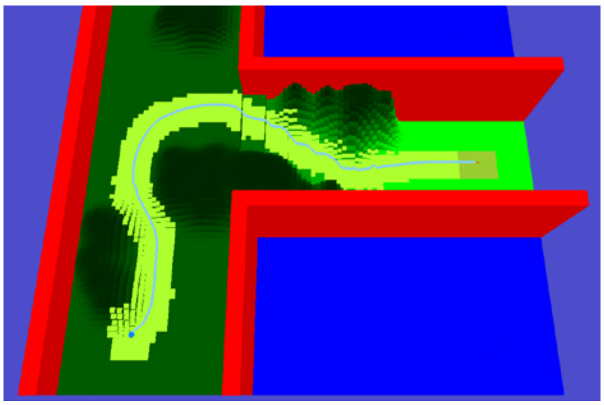

Figure 6: Simulation of the HMP generating a trajectory through a staircase with rubble. The HMP can handle both holonomic and non-holonomic platforms.

CONCLUSION

System latency from a variety of sources – signal transmission, data loss due to noisy communications channels, data processing time, etc. – adversely impacts ground vehicle teleoperation for delays as low as 170 ms. The TORIS Basic system addresses this problem through a combination of

- predictive display techniques supporting multiple sensing modalities to support robotic systems with varying price points and sensor package complexity, and
- control methods that remove the operator from the high-frequency wheel and steering control loops, which are instead closed on the robot.

Planned enhancements to TORIS Basic include adding reactive obstacle avoidance to support guarded teleoperation (TORIS Intermediate), and adding fully autonomous driving to operator-selected goals locations (TORIS Advanced).

We have implemented a network environment for TORIS testing that supports controlling communications latency in both directions (robot → OCU and OCU → robot) independently, and have done developmental testing of the system showing reasonable performance with round trip latencies as high as 4 seconds. More systematic testing to quantitatively measure the performance benefits of TORIS relative to standard teleoperation is planned for the current (second) year of the Phase II SBIR contract supporting TORIS development.

ACKNOWLEDGMENTS

The authors would like to acknowledge Dhruv Bhatt, who served as PI for TORIS Phase I. TORIS is funded under a Phase II SBIR award from TACOM, contract number W56HZV-12-C-0014. The views expressed in this paper are those of the authors, and do not necessarily represent those of the sponsor.

REFERENCES

[1] J. Borenstein and Y. Koren, “The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pages 278-288, 1991.

[2] J. Y. C. Chen, E. C. Haas, and M. J. Barnes, “Human Performance Issues and User Interface Design for Teleoperated Robots,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 37, no. 6, pages 1231-1245, 2007.

[3] J. Kay and C. Thorpe, “An Examination of the STRIPE Vehicle Teleoperation System,” in *Proceedings of IROS '97*, pages 1152-1157, 1997.

[4] W. S. Kim, I. A. Nesnas, M. Bajracharya, R. Madison, A. I. Ansar, R. D. Steele, J. J. Biesiadecki, and K. S. Ali, "Targeted Driving Using Visual Tracking on Mars: from Research to Flight," *Journal of Field Robotics*, vol. 26, no. 3, pages 243-263, 2009.

[5] A. Lacaze, K. Murphy, M. Del Giorno, and K. Corley, “Reconnaissance and Autonomy for Small Robots (RASR) team: MAGIC 2010 Challenge,” *Journal of Field Robotics*, vol. 29, no. 5, pages 729-744, 2012.

[6] J. P. Luck, P. L. McDermott, L. Allender, and D. C. Russell, “An Investigation of Real World Control of Robotic Assets under Communication Latency,” in *Proceedings of HRI ’06*, pages 202-209, 2006.

[7] R. Pausch and T. Crea, “A Literature Survey for Virtual Environments: Military Flight Simulator Visual Systems and Simulator Sickness,” Computer Science Report No. TR-92-25, University of Virginia, Aug. 1992.

[8] J. B. F. van Erp and P. Padmos, “Image Parameters for Driving with Indirect Viewing Systems,” *Ergonomics*, vol. 46, no. 15, pages 1471-1499, 2003.