© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Low-Cost 3D-Supported Interactive Control

**Enrico Masala, Antonio Servetti, and Angelo Raffaele Meo**

**Politecnico di Torino, Italy**

***Abstract - Devices for stereoscopic vision are gaining increasing diffusion, but their usage is mostly oriented toward entertainment. A prototype based on consumer devices and open software to achieve low-cost 3D-video-supported interactive control is presented here. This research could stimulate the study and implementation of low-cost general-purpose systems that could be used on a wide spectrum of applications, including remote operation, education, training, and surveillance. There are two related Web extras that provide supplemental material.***

For decades, researchers have been experimenting with real-time multimedia services with 3D technology support. Such services have been used in specialized contexts—such as in the entertainment industry, for teleoperation, and in critical applications—but have mostly relied on special-purpose hardware. Given recent advances in 3D technology and the increasing diffusion of low-cost 3D devices in the consumer market—including 3D TVs, mobile phones, and gaming devices—wouldn't it be great if we could use these ordinary devices to build low-cost 3D communication systems?

We started this line of research in 2012, motivated by the need to add a low-cost stereoscopic remote control module in a research project developing an open source framework for real-time teleoperation. In fact, researchers have demonstrated that teleoperation is an application area that stands to benefit greatly from stereoscopic vision (see the "Related Research in 3D Remote Control" sidebar). 3D images can enable better perception of depth characteristics in the environment—especially in terms of the relative object distance—thus enhancing precision and reducing the time needed to complete tasks involving remote robot piloting and manipulation. Of course, for each type of application, it's necessary to verify that the achievable quality-of-experience—which might be bounded by the limited capabilities of consumer-grade hardware—meets the application constraints and user expectations.

Our goal was to study and design a 3D teleoperation prototype offering good performance at a relatively low cost (less than US$1,500). Here, we highlight the issues involved in designing such a system and present a prototype that demonstrates the feasibility of the low-cost approach, showing its cost-performance tradeoff. The results pave the way for the development of many new interactive real-time systems in different application domains.

## Related Research in 3D Remote Control

The use of 3D vision for teleoperation is a very active research field,[1] because it has been shown that 3D viewing technologies might provide users with higher depth perception. As expected, some tasks benefit more than others from a better comprehension of distance,[2] such as teleguide and obstacle localization. However, even in demanding situations such as in emergency applications, teleoperation with 3D vision typically improves remote-control performance[3]

John McIntire, Paul Havig, and Eric Geiselman have reviewed the literature investigating human factors that have implications on task performance when stereoscopic 3D displays are used[4] The tasks include judging absolute and relative distances, finding and identifying objects, navigating and manipulating objects in terms of position, and performing orientation and tracking.

Researchers have also investigated acquisition and visualization technologies. For example, work has shown that, using efficient algorithms, correctly calibrated stereoscopic images provide a guaranteed depth perception[5] Other work has assessed the performance achieved using different viewing technologies during remote-control operations, providing results in terms of usability while comparing 3D and 2D viewing, 3D and virtual-reality displays, and robot sensors.[6]

Studies have also addressed the problem using simulation. For example, researchers performed a pick-and-place task in a simulated virtual environment using three different systems: multiple 2D displays, a 3D perspective display, and a 3D stereoscopic display.[7] Although the gain in performing simple tasks was limited, introducing a stereoscopic display resulted in better performances than the perspective display for highly difficult tasks.

Finally, Salvatore Livatino and Giovanni Muscato were interested in the effect of video feedback latency, so they performed usability studies that substituted real video images with synthetic images from robot laser data.[8] The authors observed that by using a technique to minimize the amount of bandwidth (and delay) required for the transmission, they could significantly reduce the amount of time spent performing a given task, thus improving system usability.

### References

1. S. Livatino et al., “Mobile Robot Teleguide Based on Video Images,” *IEEE Robotics & Automation*, Dec. 2008, pp. 58–67.
2. D.R. Tyczka et al., “Study of High-Definition and Stereoscopic Head-Aimed Vision for Improved Teleoperation of an Unmanned Ground Vehicle,” *Proc. of Society of Photo-Optical Instrumentation Engineers (SPIE) 8387*, May 2012, article no. 83870L.
3. J.Y. Chen et al., *Effectiveness of Stereoscopic Displays for Indirect-Vision Driving and Robot Teleoperation*, tech. report, Army Research Lab in Aberdeen, Aug. 2010.
4. J.P. McIntire, P.R. Havig, and E.E. Geiselman, “What Is 3D Good For? A Review of Human Performance on Stereoscopic 3D Displays,” *Proc. of Society of Photo-Optical Instrumentation Engineers (SPIE) 8383*, May 2012, article no. 83830X.
5. M. Ferre et al., “3D-image Visualization and Its Performance in Teleoperation,” *Proc. 2nd Int’l Conf. Virtual Reality* (ICVR 07), 2007, pp. 22–31.
6. S. Livatino, G. Muscato, and F. Privitera, “Stereo Viewing and Virtual Reality Technologies in Mobile Robot Teleguide,” *IEEE Trans. Robotics*, vol. 25, no. 6, 2009, pp. 1343–1355.
7. S.H. Park, *The Effects of Display Format and Visual Enhancement Cues on Performance of Three-Dimensional Teleoperational Tasks*, PhD dissertation, Dept. of Industrial Eng., Texas Tech University, 1998.
8. S. Livatino and G. Muscato, “Robot 3D Vision in Teleoperation,” World Automation Congress (WAC), June 2012, pp. 1–6.

## COTS and Open Source Solutions

To reach our goals, we focused our attention on consumer off-the-shelf (COTS) hardware and open source software. COTS components cost much less than dedicated hardware, but they can have (slightly) lower performance.

The scientific literature and open source domain offer many algorithms and reference implementations for problems in the fields of communication, visualization, stereoscopy, and so on. The downside of open source versus commercial software again relates to performance. However, with suitable modifications and component selection, open source software offers good-performance, low-cost solutions that the final installer can customize to address the specific implementation and its goals.

Furthermore, while designing our system, we focused on simplicity of use and customization. We also wanted to create an open-platform-based design and use widespread standards for coding and communication. Although our current solution is only a prototype, we hope it stimulates further research and investments in this research area.

## Architecture Design

A generic structure of a teleoperation system based on real-time 3D video feedback has two main parts: a mobile streaming node equipped with a stereoscopic camera, and a client application for 3D visualization (see Figure 1).

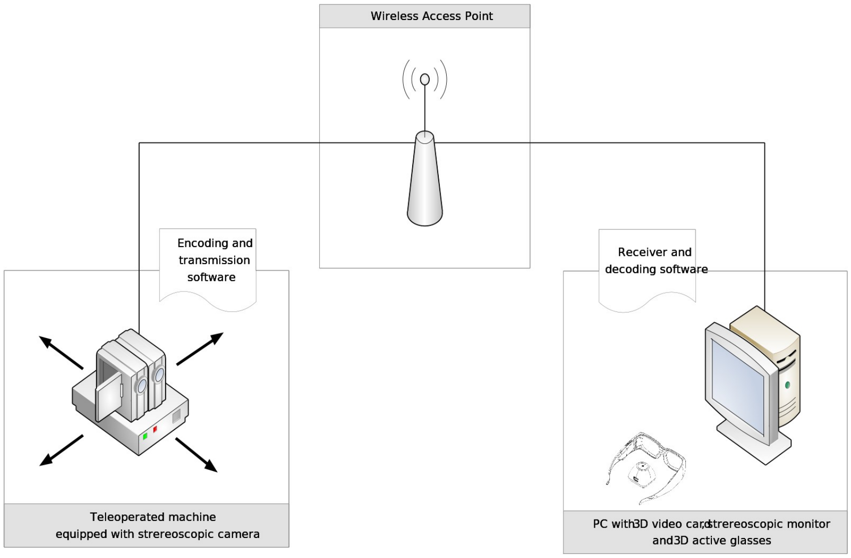

Wireless Access Point
Encoding and transmission software
Receiver and decoding software
Teleoperated machine equipped with strereoscopic camera
PC with3D video card, strereoscopic monitor and3D active glasses

*Figure 1. A block diagram of a 3D video transmission chain suitable for remote teleoperation systems. A stereoscopic camera mounted on the teleoperated machine wirelessly transmits to a receiver suitable for 3D vision.*

The transmitter node is located on the teleoperated machine so that the capture device can record the vision of the area from the device's point of view. The system's core must process the captured images so that a timely robust transmission can occur through a wireless channel to the remote operation side. On the receiver side, the software reconstructs and renders the 3D image from the received data on a graphic display device in a format suitable for stereoscopic vision. The key point for successful system operation is to guarantee low end-to-end latency, which is a fundamental requisite to achieve a good real-time interaction experience.[1]

### Issues with a Consumer-Level Solution

The deployment of consumer-level solutions for remote control 3D video systems is made difficult by constraints that relate to the high cost of industrial-grade components and to the use of proprietary, closed systems. As noted earlier, to enable consumer-level solutions at a reduced cost, we investigated the feasibility of using COTS hardware components, integrated using open source software frameworks. In doing so, we prioritized the following issues.

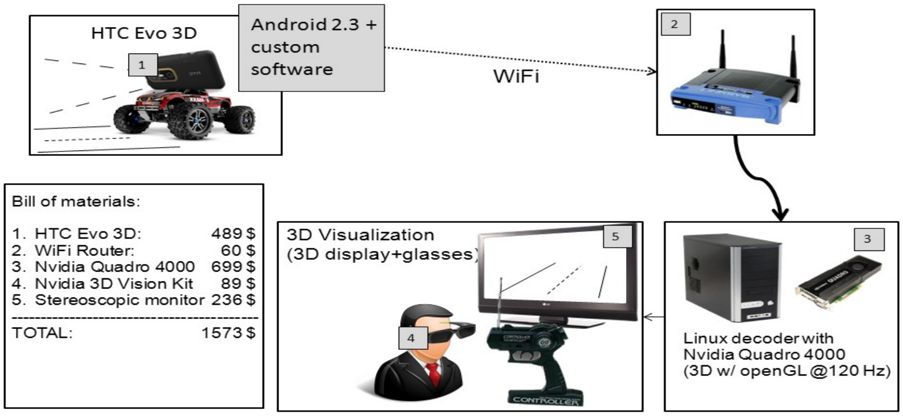
First, we wanted to find a mobile device with suitable resolution and highly responsive output of the captured images. For this reason, we excluded still-picture cameras. We also wanted to reduce the video encoding latency using appropriate encoding algorithms.

Next, we wanted to make the system sufficiently robust to some packet losses—an event possible in any communication scenario that can severely affect the user's perceived video quality. As we will see later, suitable choices of encoding algorithms and related configurations can help address this issue.

Finally, we wanted to render the stereoscopic video using a technology widely supported in an open source environment.

### Component Selection and Evaluation

The previous considerations led us to develop the building blocks of our proposed teleoperation system based on real-time 3D video feedback (see Figure 2 and the related video at http://youtu.be/5zApDO3wSq4, which also shows the components).


HTC Evo 3D
1
Android 2.3 + custom software
WiFi
2
3D Visualization (3D display+glasses)
5
4
CONTROLLER
3
Linux decoder with Nvidia Quadro 4000 (3D w/ openGL @120 Hz)

Bill of materials:

| | |
|---|---|
| 1. HTC Evo 3D: | 489 $ |
| 2. WiFi Router: | 60 $ |
| 3. Nvidia Quadro 4000 | 699 $ |
| 4. Nvidia 3D Vision Kit | 89 $ |
| 5. Stereoscopic monitor | 236 $ |
| TOTAL: | 1573 $ |

*Figure 2. System architecture and components. A mobile phone with stereoscopic capabilities wirelessly transmits to a computer that can render 3D video using active glasses. The total cost of the system is approximately US$1,500.*

Our prototype includes the following five key components.

**A stereoscopic camera.** We used a commercially available 3D mobile phone (HTC Evo 3D) that let us achieve, at a moderate cost, stereoscopic vision, portability, a battery-operated power supply, and wireless communication. In field experiments, we experienced a latency between 150 and 200 ms in the best conditions. This is slightly higher than the ideal requirement (less than 150 ms), but it's probably the most efficient solution that the COTS market can currently provide. In fact, this loss of performance is the price to pay for keeping the overall system cost low.

We also evaluated other solutions, such as 3D cameras, but the majority of them either didn't let us transmit video in real-time, or the format (such as the Digital Video format) required further processing and thus increased system cost, complexity, and latency.

**Encoding and transmission software.** We developed a custom-built open source encoding and transmission software specifically for the HTC Evo 3D phone to minimize encoding and transmission latency.

**Networking hardware.** We used standard networking hardware—a commercially available 802.11 access point for transmission over a standard Wi-Fi channel.

**Receiver and decoder software.** We custom built open source receiver and decoder software on top of the Linux OS and openGL libraries.

**Stereoscopic hardware support.** We used a standard desktop computer with stereoscopic hardware support, including a graphic board, a monitor, and an infrared emitter device for the active glasses. The identified solution for supporting hardware openGL is based on an Nvidia Quadro 4000 graphics board, which provides a direct connection from the video card to the infrared emitter device without the need for specific driver support in Linux (see the specifications at www.nvidia.com/object/product-quadro-4000-us.html). This greatly simplified the management of stereoscopic issues in the receiver but also increased the final system's cost.

### Alternative Solution

Because Nvidia doesn't officially support 3D visualization for the Linux environment, we could have implemented a cheaper solution (for approximately $900) using an experimental driver that's publicly available on the Internet (http://sourceforge.net/projects/libnvstusb. The driver lets you work with much cheaper video cards by synchronizing the infrared emitter using the USB connector. However, we didn't pursue this solution, because it doesn't currently provide stable performance.

## System Implementation

Much of our time was spent developing a video transmission application that could achieve performance comparable to more expensive professional solutions while relying only on low-cost COTS devices. The application (available at http://media.polito.it/itpro) consists of both a transmitter and a receiver part. We implemented the video transmission on top of the Android platform (www.android.com) using both Java code and our optimized routines for video compression, compiled in native code to improve performance. To achieve maximum robustness against data loss and minimum latency, we implemented the video codec, which segments images in independently decodable subareas to allow pipeline processing and better error resilience.

Due to the strict latency constraints, it's not possible to retransmit data; however, encoding and packetization is designed so that the content of each single packet that has been successfully received can be decoded and presented to the user, because no packet depends on other ones for decoding. Missing data is estimated using a frame-copy concealment technique. Data is sent using the standard RTP protocol[2] with a header extension that simplifies recovery in case of packet losses.

On the receiver side, the software features a multithreaded architecture that decouples the packet reception and decoding from data visualization so that the interaction with the device-rendering engine doesn't affect the time required to process the input data. For the visualization part, we used the openGL library (www.opengl.org/sdk), including its extension for stereoscopic visualization. Therefore, the developer needs to draw data only on the correct frame buffer (left or right) and the library automatically handles all visualization issues, including alternating left and right views on the screen at the correct frame rate and synchronizing with the shutter glasses.

Our choice was to rely on open source software at the underlying level, but the whole system can be easily ported and extended to any platform. The transmitter is written partly in Java and partly in C (compression routines), while the receiver is standard C code using sockets, threads, and openGL—a set of libraries available on the vast majority of platforms.

## Results

We tested the system in a practical scenario designed to investigate a set of factors that could affect the operator's performance in remote teleoperation—in particular, the latency of the video feedback for effective control and the quality of the stereoscopic images for effective 3D perception.

The chosen components and system design already take latency and image size into account. Nevertheless, because these two factors conflict with each other, care should be taken to correctly balance the two components. For example, the more we increase image quality, the more we burden system latency and thus reduce usability.

Figure 3a shows the total amount of delay introduced by the different components of the communication chain: acquisition, processing, transmission, and rendering. First of all, an intrinsic lower bound in the overall delay is given by the sum of the camera acquisition latency, the rendering screen refresh rate, and the network transmission time. In our setup, this delay accounted for approximately 130 ms and was the same for any system configuration we tested. Second, we tested whether the time required for image compression—which depends on the device's computational capabilities—heavily influences the maximum frame rate that can be achieved.

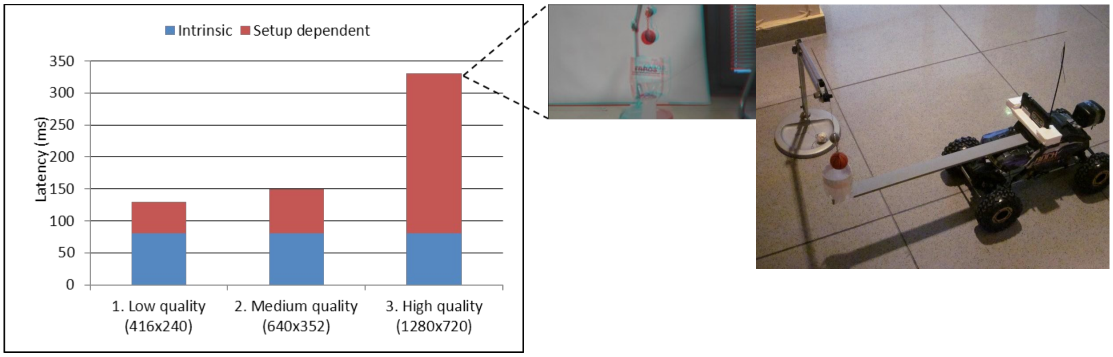


*Figure 3. System prototype: (left) the quality versus latency tradeoff in the test scenario and (right) the 3D vision prototype performing the test.*

The latency measurements with the different image-quality configurations showed that the request for high quality strongly affects the time required for the image-compression process. This behavior limits the achievable frame rate and, in turn, increases the total end-to-end delay. For example, with a frame resolution of 416 × 240 pixels, we were able to encode video at 20 to 22 frames per second, while with a frame resolution of 640 × 352 pixels, we achieved 14 to 18 fps; with 1,280 × 720 pixels, only 3 to 5 fps was possible.

We designed an experiment with the smartphone camera mounted on a radio-controlled toy car and asked users to perform an alignment task using the 3D vision system and the car remote controller (see Figure 3b and the related video at http://youtu.be/1Olg1oV2fIs).

User feedback confirmed that the frame rate allowed by the maximum resolution configuration was too low to effectively drive the car, because its overall latency (more than 300 ms) highly impaired user reactivity to the car movements. On the contrary, users preferred the first two setups—even if the image quality was lower—because their latency was within 200 ms—a value still considered satisfactory for interactive communications.[3] In particular, because the difference in latency between the first two setups wasn't really perceivable, almost all users preferred the second setup (640 × 352). However, recent findings suggest that further quality reductions might not significantly impair the ability to perform given tasks.[4] This could pave the way for adaptively choosing codec and coding parameters, thus varying quality if necessary, depending on the network bandwidth availability.

Moreover, we also considered the case of transmission over the wireless Internet. Experiments show that the system can work with the video qualities shown in Figure 3a using between 800 and 1,700 kbit/s. These rates have been experienced using a 3G connection with a packet loss rate lower than 1 percent. However, latency is approximately 80 ms higher compared to Wi-Fi, and packet-delay variation (jitter) was up to 30 ms, which delays the visualization of some frames. Latency has an impact mainly on the system's perceived responsiveness, while jitter makes the video feedback subject to very short freezes that can be annoying.

In general, the results of the evaluation process indicate that the system can be applied in a context of remote teleoperation where real-time video feedback requirements can be somewhat relaxed by trading off costs with latency. With this setup, we expect that low-cost 3D visualization systems will become popular in myriad new contexts, from domotics and disability assistance to gaming and entertainment. Further studies are in development, with the aim of increasing system timeliness, even for the high-quality video setup.

Future work might include additional optimizations to enhance the user experience. For example, dropping the open source requirement in favor of proprietary software libraries will let us reduce processing time by exploiting specific hardware support for video acceleration. At the same time, the system's total cost can be significantly reduced using much cheaper 3D video cards not yet supported by open source software.

## References

1. J.C. Lane et al., “Effects of Time Delay on Telerobotic Control of Neutral Buoyancy Vehicles,” *Proc. IEEE Int’l Conf. Robotics and Automation* (ICRA 02), vol. 3, 2002, pp. 2874–2879.
2. H. Schulzrinne et al., *RTP: A Transport Protocol for Real-Time Applications*, RFC 3550, July 2003; http://tools.ietf.org/html/rfc3550.
3. *One-Way Transmission Time*, ITU-T recommendation G.114, May 2003.
4. E. Masala and A. Servetti, “Performance vs. Quality of Experience in a Remote Control Application Based on Real-Time 3D Video Feedback,” *Proc. 5th Int’l Workshop on Quality of Multimedia Experience* (QoMEX 13), 2013, pp. 28–29.

***Enrico Masala*** *is an assistant professor at Politecnico di Torino, Italy. His main research interests include the development of algorithms for quality optimization of communications over packet networks, wireline, and wireless, with a particular emphasis on multimedia and specific contexts such as 3D video and dynamic adaptive HTTP streaming. Contact him at enrico.masala@polito.it.*

***Antonio Servetti*** *is an assistant professor at Politecnico di Torino, Italy. His research focuses on speech/audio processing, and real-time multimedia communications. With the advent of video and audio support in HTML5, his interests include also multimedia Web applications and HTTP adaptive streaming. Contact him at antonio.servetti@polito.it.*

***Angelo Raffaele Meo*** *is an emeritus professor at Politecnico di Torino, Italy. He has been involved in numerous national projects in computer engineering, investigating a broad range of topics, including switching theory, hardware design, signal processing, speech analysis and synthesis, and pattern recognition. He also served as the president of the Academy of Sciences of Torino and as the president of the Italian government commission entrusted with the task of promoting open source in the Italian Public Administration. Contact him at meo@polito.it.*