Sensor-Based Real-Time Adaptation of 3D Video Encoding Quality for Remote Control Applications

Enrico Masala
Control and Computer Engineering Department, Politecnico di Torino
corso Duca degli Abruzzi 24 — 10129 Torino, Italy
masala@polito.it

***Abstract*—The availability of stereoscopic mobile devices, such as mobile phones, on the consumer market allows to attempt the development of low-cost remote control systems that can provide a real-time 3D video feedback. In this work we show how implement such a communication system by considering the stringent latency constraints of the remote control scenario. To reduce the impact of this issue, we observe that part of the latency is due to the limited processing power of the mobile device that cannot sustain video transmission at high quality with low latency. Thus, we propose to dynamically change the latency-quality trade-off at the transmitter to optimize the quality of experience as perceived by the operator of the remote control system, by taking into account, in real-time, the dynamics of the control operations. In more details, low-cost accelerometer and gyroscopic sensors are employed to decide in real-time how much latency has to be privileged over quality and vice versa, by selectively reducing the quality of one of the views in favor of a reduced overall latency. Comparisons with a non-adaptive higher-quality but also higher-latency system show that the operators prefer the adaptive system despite the video quality is slightly reduced in dynamic control conditions.**

I. Introduction

Stereoscopic video communications have been shown to provide distinctive advantages in a number of contexts, ranging from entertainment to telepresence [1], [2]. Among the range of possible applications, teleoperation can also greatly benefit from stereoscopic vision [3], [4]. However, the stringent low-delay constraint, which is an important requirement for teleoperation, often does not allow to use commercial off-the-shelf (COTS) hardware due to the high-performance requirements of many elements of the transmission chain, e.g., video capture and compression.

Recently, the increasing diffusion of mobile devices allowed to find some devices with stereoscopic image acquisition capabilities on the consumer market at a reasonable price. Therefore, it is interesting to explore the limits of such devices, in particular to investigate if they can be used as building blocks for a low-latency video communication system suitable for teleoperation.

Many difficulties have to be addressed, especially how to achieve low latency and efficient real-time video encoding despite the limited processing power of such devices. Although software can be developed and specifically optimized for these type of devices, this is often not sufficient to achieve satisfactory performance, therefore alternative approaches should be pursued.

An approach could be to reduce the amount of information to process, e.g., by reducing image resolution, so that also the computational requirements are reduced. However, in order not to impact negatively on the quality of experience (QoE) of teleoperators, some understanding of the current dynamics of the remote control operations would be useful to optimize the QoE at each time instant. However, rather than continuously processing images as captured by the camera, for instance to detect movements, the same goal can be very effectively achieved by means of using some sensors, such as accelerators and gyroscopes, fixed on the teleoperated object. Such sensors are currently available at no additional costs in many smartphones. Therefore, if the video capturing device is fixed onto the teleoperated object, the sensor integrated in the mobile device can effectively detect movements at no additional computational cost.

Some studies have also proposed to use the values provided by similar sensors to speed up video processing operations, such as video encoding, as in [5]. In that work a sensor is used to suggest a global motion vector later used by the motion estimation algorithm, resulting in a reduced complexity of the motion estimation algorithm.

This work focuses on the specific scenario of telecontrol supported by a low-latency 3D video feedback, proposing both a low-cost integrated architecture to build the video communication system as well as an algorithm to adapt the latency-complexity trade-off of the video transmission algorithm by taking into account, in real-time, the dynamics of the telecontrol operation, with the final aim to improve the QoE for the remote operator. In more details, accelerometer and gyroscopic sensors are employed to decide, at each time instant, how much latency has to be privileged over quality and vice versa, by selectively reducing the quality of one of the views. This has only a limited effect on the stereoscopic image quality, as shown in [6]. Comparisons with a non-adaptive higher-quality but also higher-latency system show that the operators prefer the adaptive system despite the video quality is slightly reduced in dynamic control conditions.

The paper is organized as follows. Section II provides a brief background on the requirements of interactive remote control applications. The architecture of the proposed low-cost

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
978-1-4799-0125-8/13/$31.00 ©2013 IEEE.

3D video communication system is detailed in Sec. III, then Sec. IV presents the proposed adaptive algorithm to optimize the quality-latency trade-off. The simulation setup is described in Sec. V followed by results in in Sec. VI. Finally, conclusions are drawn in Sec. VII.

II. Requirements of Teleoperation Scenarios

The considered teleoperation scenario relies on the availability of a low-latency video feedback to an operator to perform some tasks by means of remote controls. The digital video communication channel is typically separated from the control channel which is strongly dependent on the type of the remote-controlled device.

An important QoE factor in remote control applications is the latency of the communication, since it directly impacts on the ability of the operator to interact effectively with the remote system. Latency must be low enough to allow proper reactions to events happening in the remote scenario. For instance, the danger of hitting obstacles should be perceived sufficiently in advance to be able to adjust the trajectory of the remote-controlled objects. Thus, in general the allowed latency also depends on the speed at which the remote-controlled objects move.

Communication robustness is also important. Specific video encoding strategies should be adopted to make the communication robust, since it is extremely annoying for the operator to experience distortion or even freezes of the video feedback even for short periods of time. In this context, differential encoding with motion compensation typically employed in video coding schemes can be problematic since it may lead to error propagation which can last over time. In addition, an approach based on differential encoding with motion compensation is also computationally complex for resource-limited devices. Although strategies to overcome this issue in interactive scenarios have been proposed [7], independent encoding of each image is often used in practical scenarios [8] since it presents very low computational complexity, high robustness and low latency at the same time. The main drawback is bandwidth demand. However, this requirement is compatible with many teleoperation scenarios where connectivity can be provided by local area networks (LAN) or wireless LAN communication technology. Examples include teleoperation where the operator is at a moderate distance from the controlled device, e.g., for safety or similar reasons.

III. The Proposed Architecture of the Low-Cost 3D Video Feedback System

A. General Structure

Many consumer devices with video capture capabilities (e.g., mobile phones) exhibit low image acquisition latency, which is close to the requirements of a typical real-time remote control system. In addition, many mobile phones now run an operating system, e.g., Android [9], that allows to develop and run custom software on the device themselves. Therefore, a software can be written to acquire images, compress and send them by means of the network interface to a remote receiver. Unfortunately, the limited processing power of the devices often imposes stringent constraint on the image quality since operations have to be carried out in real-time.

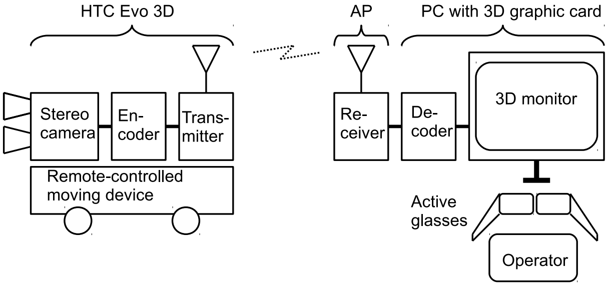

Fig. 1. Block diagram of the 3D video communication system for real-time remote control.

Nevertheless, we managed to build a low-cost, low-delay 3D video communication system by using commercial off-the-shelf (COTS) components and developing our custom transmitter and receiver software. The block diagram of such a system is shown in Fig.1. The main components are:

- Acquisition device: a mobile phone with 3D capabilities (HTC Evo 3D), which features a stereoscopic camera. The device is fixed on a telecontrolled object; in our experiments we used a radiocontrolled (RC) toy car.
- Encoder: our custom-developed software running on the mobile device, written partly in Java for Android and partly in C using the Java native code interface (JNI) in order to maximize the performance of the video compression routines, i.e., minimize encoding latency.
- Transmitter: the integrated 802.11 [10] module in the mobile phone is used to transmit data as packets to the receiver.
- Receiver: a standard 802.11 access point (AP), connected to a personal computer.
- Decoder: our custom-developed software running on a Linux PC, equipped with an Nvidia graphics board and the Nvidia 3D Vision Kit [11] to add 3D visualization capabilities.
- Visualization device: a 120-Hz monitor, and active shutter glasses synchronized with the PC by means of the infrared emitter of the 3D vision kit.

To simplify the development process of the client software, we choose to use the Nvidia Quadro 4000 graphic board [12] which provides hardware support for the stereoscopic extension of the OpenGL libraries [13], so that the hardware automatically synchronizes the glasses with the monitor without the need to explicitly interface the receiving software with the driver controlling the infrared emitter.

B. Latency and Frame Rate Issues

All the previous elements in the transmission chain introduce a certain amount of latency. The stereoscopic camera of the HTC Evo 3D mobile phone is able to capture images at 30 frames per second (fps). Depending on the required image quality and resolution, our custom-made encoding software may not be able to encode and transmit it within 33 ms, i.e., the

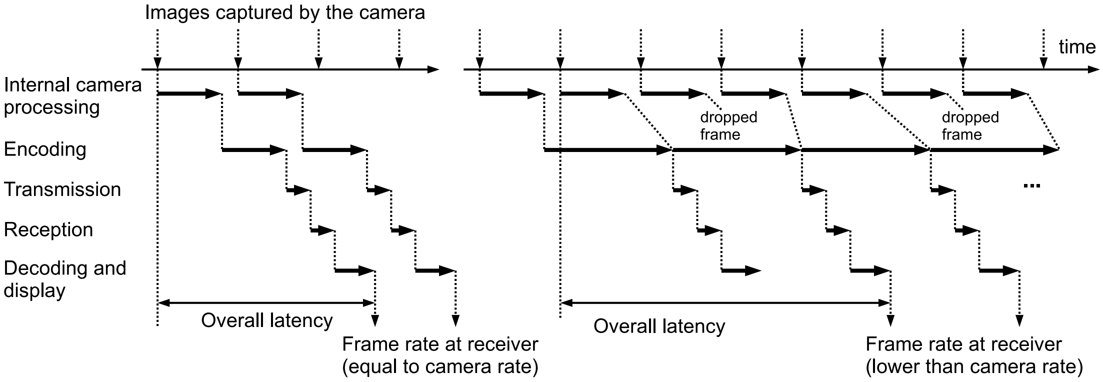
Images captured by the camera
time
Internal camera processing
Encoding
Transmission
Reception
Decoding and display
dropped frame
dropped frame
...
Overall latency
Overall latency
Frame rate at receiver (equal to camera rate)
Frame rate at receiver (lower than camera rate)

Fig. 2. Latency and frame rate at the receiver, when the time needed to encode data captured by the camera is lower than the camera frame rate (left) and vice versa (right). In the latter case some data from the camera is dropped. If encoding time varies over time the frame rate at the receiver is variable.

time budget available for processing the whole frame data at 30 fps. If more time is needed for encoding and transmission, some of the captured frames will be dropped at the transmitter, as shown in Fig. 2. In other words, latency due to encoding and transmission has a direct impact on the frame rate at the receiver. Therefore, by carefully adjusting the amount of time needed for encoding and transmission, e.g., by reducing image resolution, it is possible to change the latency of the whole system as well as the actual frame rate achieved at the receiver.

C. *Video Encoding and Packetization Algorithm*

Due to the processing power limitations of the capture device, for the transmitter we adopted a coding scheme which relies on independent image encoding. The encoding algorithm relies on JPEG image encoding. It operates in the YCrCb color space with 4:2:0 chrominance subsampling, then it performs transform coding, quantization and entropy coding of the coefficients. However, a number of modifications with respect to the standard have been introduced to boost transmission efficiency on an 802.11 WLAN.

When a large number of packets are transmitted in an 802.11 network, the effective bandwidth tends to decrease due to the use of a contention-based medium access control (MAC) layer mechanism for wireless channel access. Moreover, MAC-level packet headers further decrease efficiency if data is split into small packets. Thus, the encoding algorithm has been designed to fill the packet payload as much as possible until reaching the maximum size imposed by the Ethernet standard, i.e., 1500 bytes including IP and UDP headers. However, for robustness purposes, the encoder always inserts an integer number of blocks in the packet, i.e., it never fragments blocks across packets. This allows to fully decode any received packet in case of loss of previous or subsequent packets.

From the implementation point of view, such an algorithm requires a roll-back mechanism when a threshold amount of encoded bits is reached in the encoding software. This feature is typically not provided by standard JPEG libraries whose interface only support encoding of a given number of macroblocks regardless of the resulting bit size. Moreover, header information which does not change over time (e.g., quantization tables) has been dropped to save bandwidth. Tables are hardcoded or transmitted out-of-band on a side channel before the low-latency video communication starts. In our prototype implementation the quantization parameter is decided in advance and fixed for the whole duration of the transmission, therefore quantization tables can be precomputed. To maximize execution efficiency, all image encoding routines have been written in C and they have been compiled for the target device (an ARM-based platform) as native code.

IV. Low-Complexity Optimization of the Quality-Latency Trade-off

Experimentally we measured that transmitting stereoscopic compressed image data with the proposed algorithm at the maximum resolution introduces an image encoding latency equal to about 200 ms, which is near the maximum tolerable latency for real-time remote control operations.

However, in static conditions, e.g., when evaluating the situation without moving the telecontrolled device, this latency may be reasonable, especially considering the high quality of the video feedback presented to the user.

When the operator starts moving the telecontrolled device, lower latency is needed, especially as moving speed increases. Otherwise, the operator may be limited in the operating speed. This would negatively affect the QoE. Therefore, we propose to detect the movement conditions and to adapt the latency of the system, in real-time, to the dynamics of the control operation. This can be done at the expense of image quality, since coding a smaller resolution image is less computationally demanding.

Detecting the remote operation dynamics by means of image analysis would be possible in principle, but this would demand significant computational power to the mobile device to analyze the image content. In addition, these operations would be carried out at the expense of the computational power available for the video coding algorithm, which is already a critical resource in the proposed low-cost setup.

Therefore we employed an alternative approach. Since the mobile phone that captures the video is fixed onto the telecontrolled object, we relied on the presence of two sensors inside the mobile device to determine the current dynamics of the remote operation. In particular, the accelerometer and the gyro-

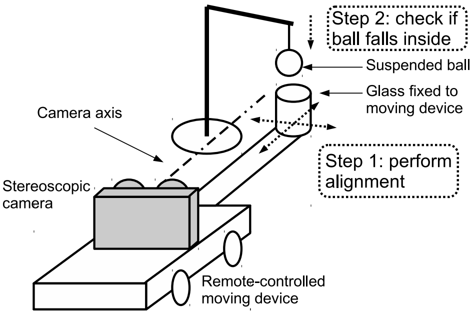
Step 2: check if ball falls inside
Suspended ball
Glass fixed to moving device
Camera axis
Step 1: perform alignment
Stereoscopic camera
Remote-controlled moving device

Fig. 3. The alignment experiment: first the remote operator align the glass under the ball, then the wire that suspends the ball is released to check if the ball falls inside the glass or not.

scope have been used to compute the linear acceleration of the device. More technically, linear acceleration is determined by the fusion of the values provided by the accelerometer and the gyroscope, which is performed directly by routines integrated in the Android operating system. The gyroscopic information is needed to correctly subtract the gravity acceleration vector from the measures coming from the accelerometer.

The linear acceleration value is used to determine if movements are taking place or not, by means of a threshold value. If acceleration is greater, it is assumed that vibrations due to movements are its cause. In this case, the resolution of the right image is reduced so that the latency due to the encoding of the left and right image is reduced to a level comparable with the latency imposed by coding the left image only.

In such a way the responsiveness of the video feedback is improved, and the quality of the stereoscopic image is not reduced excessively. In fact, the work in [6] showed that the spatial resolution of one of the two images in a stereoscopic pair can be reduced to some extent without significantly affecting both image quality and depth perception. However, if the resolution is strongly reduced, some quality decrease will be perceived, therefore a good trade-off point must be found so that the quality reduction, if present, is reasonably compensated by the lower latency of the video feedback which has positive effects on the perceived QoE.

Note that the sensors integrated with the mobile phone often provide noisy values, probably due to the fact they are cheap and they are used in applications, such as video games, where precision has only limited importance. Nevertheless, in our experiments their values have been shown to be reliable enough for our aims.

Finally, note that, due to the use of mechanical sensors, only movements of objects mechanically connected to the sensor can be promptly detected and communication latency correspondingly reduced. Although this approach may not be suitable for all possible telecontrol scenarios, we believe that it is still reasonable if the sensor is fixed on the telecontrolled part, so that the occurrence of not immediately detected movements is limited to the case in which the telecontrolled part is in a static state.

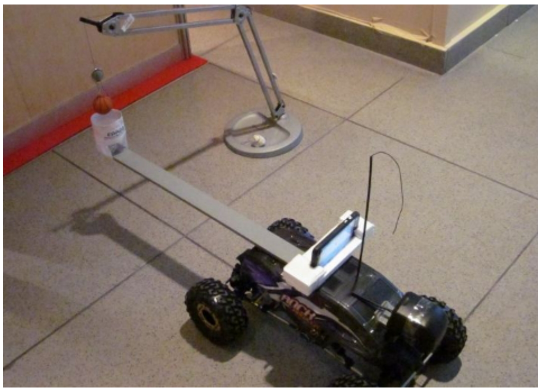

Fig. 4. Picture of the prototype in an experimental setup.

V. Simulation Setup

We setup a remote control experiment in order to test the effectiveness of the proposed adaptive system. More specifically, the experiment consists in a remote operator, looking at the 3D video feedback, which is required to align a glass, fixed on a RC toy car, under a suspended static ball, so that if the ball is dropped it would fall into the glass. Fig. 3 illustrates the experiment. A picture of the prototype used for the experiments is shown in Fig. 4. Due to the position of the stereoscopic camera axis the correct depth alignment can be achieved only when the depth perception is good. Fig. 5 shows a sample image captured from the camera. The stereoscopic image has been converted to a red-cyan anaglyph for printing purposes. Remote operators are also asked to perform the alignment in the least possible time, thus forcing them to move very frequently to achieve the goal as fast as possible.

We considered two cases. In the first one, the system is configured to always transmit both the left and the right side image to the receiver, therefore the latency-quality trade-off has been fixed in this scenario, as well as the frame rate which depends on the encoding and transmission latency, as explained in Sec. III-B.

In the second one, the sensor-based adaptive transmission feature is enabled, therefore when the operator is moving the RC toy car the latency of the system is reduced, as well as the quality of the right-side image. However, it is expected that the presence of movement in the video as well as the quality reduction affecting only one of the two images mitigate the negative effects. Note that in this second case the video frame

Fig. 5. Sample stereoscopic image captured during the remote control experiment. Converted to red-cyan anaglyph for printing purposes.

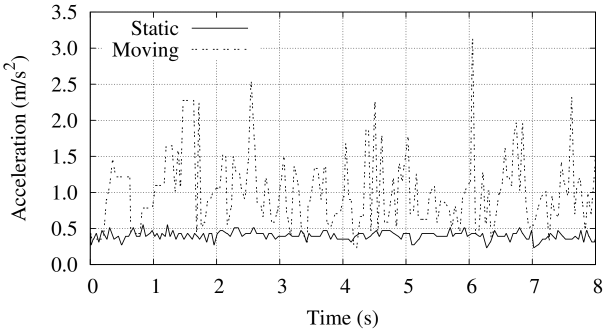

Fig. 6. Sample of acceleration values as a function of time in two different operating conditions: static and moving.

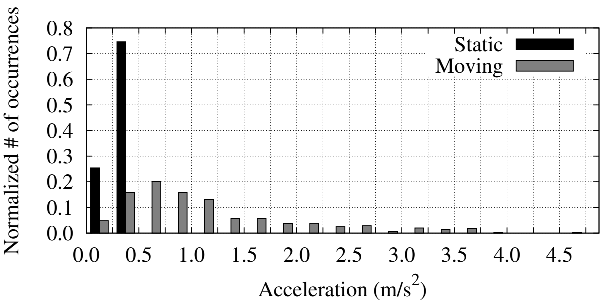

Fig. 7. Histogram of the acceleration values in two different operating conditions: static and moving.

rate is variable, i.e., it is lower with higher quality images and higher with lower quality images, as explained in Sec. III-B. The quantization parameter is always the same in both cases.

The responsiveness of the sensors has also been evaluated. In all our experiments, the acceleration variations have always been detected fast enough so that the acceleration increase is detected before that movement can be perceived from the video. This has been experimentally verified by transmitting the acceleration values within the same packets of the video frames, logging the uncompressed video sequence to a file and checking the sensor values when the movement started to be perceived in the video sequence.

VI. Results

A. Sensor Characteristics

Fig. 6 shows a sample of linear acceleration values as a function of time in different conditions, i.e., static and moving. The sensors have been configured to provide new values at least as every 33 ms, i.e., the time corresponding to the camera frame rate. The graph shows that the two conditions can be easily distinguished by means of a threshold mechanism, as it is also confirmed by a histogram plot shown in Fig. 7. From both figures, in our setup a good threshold value to distinguish between the static and moving conditions appears to be about 0.5 m/s^2.

Note that the non-zero value in static conditions is probably due to the fact that there are a number of small vibrations onboard the RC toy car which are present also in static conditions, e.g., due to the cooling fan for the power electronic components that directly drive the actuators of wheels and steering.

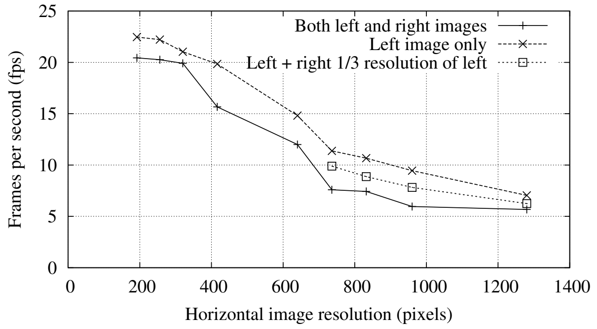

Fig. 8. Frames per seconds as a function of the image resolution depending on the resolution at which left and right frames are transmitted.

B. Video quality and Remote Control Performance

Fig. 8 shows the frame rate that can be achieved for different image resolutions when both left and right images are sent. When the image resolution is high, the latency introduced by the encoding process is high, therefore the average frame rate is low, as explained in Sec. III-B and shown in Fig. 2, despite the camera continuously captures data at 30 fps. The overall latency of the system typically ranges between 150 and 250 ms.

If the resolution of the right image is reduced the latency of the encoding process decreases, thus the overall latency of the 3D video feedback system decreases and at the same time the frame rate experienced at the receiver increases. Fig. 8 shows that, if the right image is not processed at all, for the case of high resolution it is possible to nearly double the frame rate (e.g. from about 5 to about 10 fps). If the resolution is low the frame rate is already high since it is possible to encode the image fast enough to provide approximately real-time performance, therefore the performance gap between the two cases is much lower. The achieved value is slightly lower than the camera frame rate since the operating system of the mobile phone does not provide real-time services, hence occasionally video encoding routines are delayed by the scheduler.

The middle curve in Fig. 8 shows that, when the resolution of the right image (horizontal and vertical) is reduced at 1/3 of the left one, the frame rate is increased and latency is consequently reduced. The value 1/3 has been chosen experimentally as a good compromise between increased performance due to the reduced amount of data to compress, which results in up to 40 ms latency decrease, and the overall quality reduction of the stereoscopic picture. Lower reduction values do not provide significant latency improvements while higher values excessively deteriorates video quality. Therefore, the system switches the between full and reduced (1/3) resolution cases in real-time depending on the value provided by the accelerometer sensor.

The image quality reduction corresponding to various resolutions reductions are shown in Fig. 9 and 10 for a sample image (Fig. 5) of our test scenario. However, it must be noted that these values refers to only one of the two images of the stereoscopic pair, therefore they are shown as a reference value for completeness but they cannot be used to directly infer the

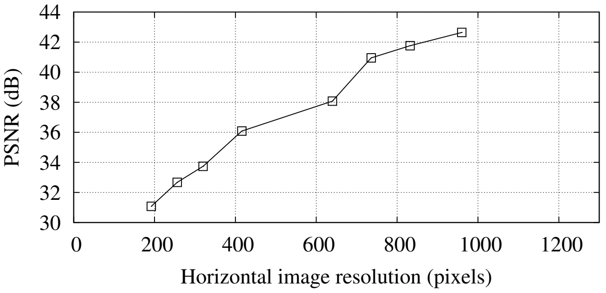

Fig. 9. PSNR as a function of the resolution used to send the right image to the receiver. The image is rescaled by the receiver to the same resolution of the left one.

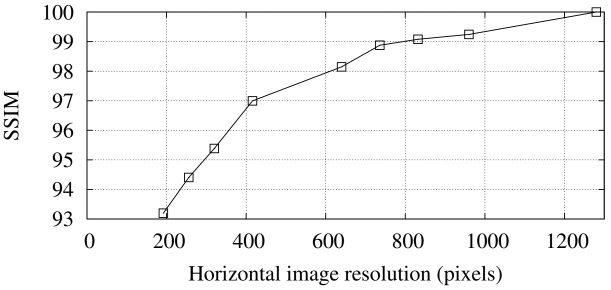

Fig. 10. SSIM as a function of the resolution used to send the right image to the receiver. The image is rescaled by the receiver to the same resolution of the left one.

quality of the stereoscopic pair.

For this purpose, Fig. 11 shows a sample stereoscopic picture where the right image has been processed using the same resolution reduction routines used in our system, i.e., downsampled by a factor of three at the transmitter and upsampled again at the receiver. By observing the picture it is possible to note that depth perception is substantially unaffected by the resolution reduction of the right image. Details in the picture, such as the text on the glass, appear partially blurred, due to the resolution reduction. However, the contours of the objects are still clear, therefore the image, despite its limitations on details, is useful to perform the alignment task. Moreover, note that the image is part of a video sequence, thus when it is observed as a part of a video sequence, the resolution reduction is much less perceivable.

To confirm this impression, a few informal subjective tests have been carried out by asking three subjects to perform the alignment experiment described in Sec. V. They performed the experiment both when the adaptive algorithm was in use and when the configuration was set to provide high quality for both views but also high latency. All the subjects preferred the setup with the adaptive algorithm. Although more formal tests with more subjects are needed, this could already be an indication that the QoE improves. This is probably due to the fact that, in static conditions, the image quality is equivalent in the two setup, but while moving the RC toy car, the system is slightly more reactive in providing the video feedback with positive impact on the perceived QoE. Also, the adaptation of the image resolution is immediate when the operator starts moving the RC toy car since, as explained in Sec. V, the value of the acceleration sensor increases immediately over the threshold, before starting the compression of the frame that shows the first movement.

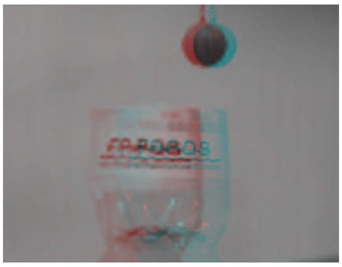

Fig. 11. Sample with right image transmitted at 1/3 resolution of the original. Converted to red-cyan anaglyph for printing purposes.

VII. Conclusion

This work proposed a low-cost remote control systems with real-time 3D video feedback using a stereoscopic mobile device for video capturing. To reduce the negative effects of latency due to the limited processing capabilities of the mobile device, we propose to optimize the latency-quality trade-off by taking into account, in real-time, the dynamics of the telecontrol operation, so that, at each time, the quality of experience delivered to the users of the remote control system is optimized. In order not to impact on the computational complexity of the system, accelerometer and gyroscopic sensors are employed to decide, in real-time, how much latency has to be privileged over quality and vice versa. Comparisons with a fixed high quality system show QoE improvements when subjects are asked to perform simple control operations.

Acknowledgment

This work was partially funded by the Regione Piemonte through FAROS project (POR.FESR07-13.I.1.3, pr. 147-311).

References

[1] A.M. Tekalp, E. Kurutepe, and M.R. Civanlar, “3DTV over IP,” *IEEE Signal Process. Mag.*, vol. 24, no. 6, pp. 77–87, Nov. 2007.

[2] J.G. Apostolopoulos, P.A. Chou, B. Culbertson, B. Kalker, M.D. Trott, and S. Wee, “The road to immersive communication,” *Proc. IEEE*, vol. 100, no. 4, pp. 974–990, Apr. 2012.

[3] S. Livatino et al., “Mobile robot teleguide based on video images,” in *IEEE Robotics & Automation Mag.*, Dec. 2008, pp. 58–67.

[4] M. Ferre et al., “3D-image visualization and its performance in teleoperation,” in *Proc. Intl. Conf. on Virtual Reality (ICVR)*, Beijing, China, Jul. 2007, pp. 22–31.

[5] X. Chen, Z. Zhao, A. Rahmati, Y. Wang, and L. Zhong, “Sensor-assisted video encoding for mobile devices in real-world environments,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 3, pp. 335–349, Mar. 2011.

[6] L. Stelmach, W.J. Tam, D. Meegan, and A. Vincent, “Stereo image quality: effects of mixed spatio-temporal resolution,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 2, pp. 188–193, Mar. 2000.

[7] B. Girod and N. Farber, “Feedback-based error control for mobile video transmission,” *Proc. IEEE*, vol. 87, no. 10, pp. 1707–1723, Oct. 1999.

[8] B. Yamauchi and K. Massey, “Stingray: high-speed teleoperation of UGVs in urban terrain using driver-assist behaviors and immersive telepresence,” in *DTIC Conference*, Dec. 2008.

[9] “Android resources,” 2013. [Online]. Available: http://www.android.com

[10] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 2012.

[11] Nvidia Corporation, “Nvidia 3D vision kit,” 2012.

[12] ——, “Nvidia quadro 4000 graphics board,” 2012. [Online]. Available: http://www.nvidia.com/object/product-quadro-4000-us.html

[13] “OpenGL software development kit,” 2013. [Online]. Available: http://www.opengl.org/sdk