Utilization of the Oculus Rift HMD in Mobile Robot Teleoperation

Tomáš Kot[1,a], Petr Novák[1,b]

[1]VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Robotics, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic

[a]tomas.kot@vsb.cz, [b]petr.novak@vsb.cz

Keywords: teleoperation, mobile robot, head-mounted display, HMD, virtual reality, Oculus Rift

Abstract. This paper mentions some problems related to utilization of a head-mounted display (HMD) for remote control of mobile robots by a human operator and also presents a possible solution. Considered is specifically the new HMD device called Oculus Rift, which is a very interesting device because of its great parameters and low price. The device is described in the beginning, together with some of the specific principles of the Oculus 3D display. Then follows the design of a new graphical user interface for teleoperation, with main focus on visualization of stereoscopic images from robot cameras. Demonstrated is also a way how to display additional data and information to the operator. The overall aim is to create a comfortable and highly effective interface suitable both for exploration and manipulation tasks in mobile robotics.

Introduction

Despite the significant progress in autonomous mobile robotics, there are still many important fields where teleoperated mobile robots (controlled remotely by a human operator) are preferred or even required. This includes for example the very crucial applications related to safety of people during chemical accidents, fires, terrorist attacks or natural disasters. Especially manipulation with some objects (explosives, barrels or containers with dangerous chemical substances etc.) requires the extraordinary decision-making abilities of trained human professionals and cannot be accomplished by generalized algorithms.

The mobile robot in such scenarios is often controlled from a long distance and may be completely out of direct sight of the operator (behind a wall, obstacles or smoke) so he must fully rely on the information provided him by the control system. The primary source of feedback typically is the visual subsystem of the mobile robot containing one or more cameras. Pictures from the camera(s) are typically displayed on a flat screen, together with additional data [1,2,3].

Improved feedback can be achieved using stereovision cameras. Stereovision can provide additional information about depths of individual objects in the camera picture, which can be a great help. There however emerges a question how to display the stereoscopic image to the operator. A common way is to use a *head-mounted display (HMD)* [4,5] that displays different images for each eye. Various HMD devices have been available for few decades already and used not only in military or governmental applications, but also commercially, especially to create *virtual reality (VR)*. However, big prices of the HMD devices (reaching even tens of thousands USD) and mediocre parameters, especially of the cheaper ones [6], were the reasons why VR was a marginal, albeit popular and often mentioned concern.

The upcoming HMD device *Oculus Rift* offers great parameters together with extremely low price and already caused a big wave of new interest in VR, primarily in the gaming industry. We decided to try to utilize it also for teleoperation of mobile robots, together with stereovision. As a testing mobile robot we chose Hercules [7], a four-wheeled mobile robot with a 3-degrees-of-freedom arm fully designed and developed on the Department of Robotics at VŠB-TU Ostrava.

The existing version of control system on the operator’s side uses advanced graphical user interface rendered using Direct3D, which assists the operator in multiple ways [8], but is not able to provide a stereoscopic display of the stereovision cameras mounted on the robot arm.

Fig. 1. Mobile robot Hercules and a screenshot of the existing 2D graphical user interface.

Description of the Oculus Rift HMD device

Oculus Rift is still in development, but a special preliminary Software Development Kit (SDK) version is available for purchase and this version was used in this project. The device offers some major advantages over other similar devices [9]:

- low price (300 USD for the SDK version),
- large field of view (FOV),
- low weight, good comfort while wearing the device,
- ultra-low latency 360° head tracking in 3 axes.

Especially FOV is very important for a good sense of virtual reality. Typical commercial HMD devices use two small LCD displays placed in front of each eye and have a very limited FOV between 30 and 45 degrees. There are few professional devices with larger FOV, but they are very expensive, for example the Sensics xSight with 123° FOV and price around 40,000 USD [10].

Oculus Rift has both the vertical and horizontal FOV larger than 110 degrees while keeping the device hardware very simple and thus cheap. The device consists of a single 7-inch LCD screen situated approximately 4 centimeters in front of the user's eyes and two plastic lenses projecting each half of the screen to the corresponding eye. A notable disadvantage of Oculus Rift is low resolution of the screen of the SDK version – only 1280 x 800 pixels, or 640 x 800 per eye, so individual pixels are clearly visible. The final version is however going to have a better resolution.

Fig. 2. Oculus Rift head-mounted display (SDK version).

Principle of the 3D display

Oculus Rift requires a specific method of rendering, different than most other HMD devices, especially because it contains only one LCD screen. The device can be used to display pictures from real-world cameras or from virtual cameras drawing the content of a virtual 3D world (VR). Two cameras are needed – one for each eye – with the distance between them corresponding to the inter-pupillary distance (IPD) of the user. The cameras must be parallel, with the point of convergence in infinity. Image for the left eye is displayed on the left half of the screen and image for the right eye on the right half.

Besides having the correct resolution (640 x 800) and aspect ratio (4 : 5), each camera also must have the correct vertical FOV, which is given as:

$$\phi_y = 2 \cdot \arctan \frac{h_{lcd}}{2 \cdot d_{lcd}}, \tag{1}$$

where h_{lcd} is physical height of the LCD screen and d_{lcd} is distance of the Oculus LCD screen from user's eyes (Fig. 4).

In VR it is quite simple to meet all the requirements, because optical parameters of a virtual camera (aspect ratio and FOV) are given by *perspective projection transformation* done by a projection matrix:

$$\mathbf{P} = \begin{bmatrix} \frac{1}{s \cdot \tan\left(0.5 \cdot \phi_y\right)} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\left(0.5 \cdot \phi_y\right)} & 0 & 0 \\ 0 & 0 & \frac{z_f}{z_n - z_f} & \frac{z_f z_n}{z_n - z_f} \\ 0 & 0 & -1 & 0 \end{bmatrix}, \tag{2}$$

where s is the aspect ratio and z_n, z_f are depth distances of the near and far clipping planes.

The lenses are placed in a fixed distance L_{IPD} = 0.0635 m (according to the average IPD of human), but this does not correspond to the size of the LCD screen. Physical width of the screen is w_{lcd} = 0.14976 m and the centers of both half-screens are $0.5w_{lcd}$ = 0.07488 m apart. This value differs from L_{IPD}, so each image must be shifted towards the center of the display by h in meters:

$$h_m = \frac{w_{lcd}}{4} - \frac{L_{lcd}}{2}. \tag{3}$$

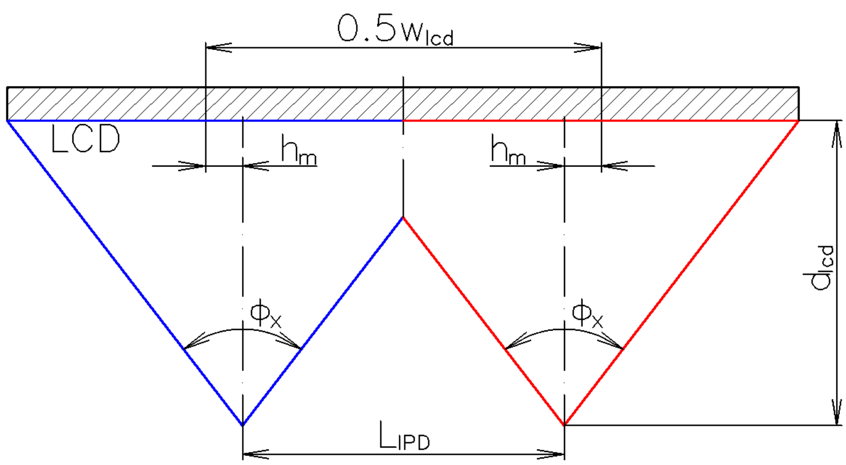

Fig. 3. Physical relation between the eye view cones and the LCD display.

With real cameras, this shifting can be done in pixels by:

$$h_p = W \cdot \frac{h_m}{w_{lcd}}, \tag{4}$$

where W is the horizontal resolution of the Oculus LCD screen in pixels. For VR cameras, it can be simply applied to the projection matrix (plus sign for the left eye, minus sign for the right eye):

$$\mathbf{P}' = \begin{bmatrix} 1 & 0 & 0 & \pm\frac{4h_m}{w_{lcd}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{P} \tag{5}$$

Distortion and chromatic aberration. The lenses in Oculus Rift create a significant pincushion distortion of the image, which can however be canceled out by creating appropriate opposite (barrel) distortion in software before sending the pictures to the HMD device.

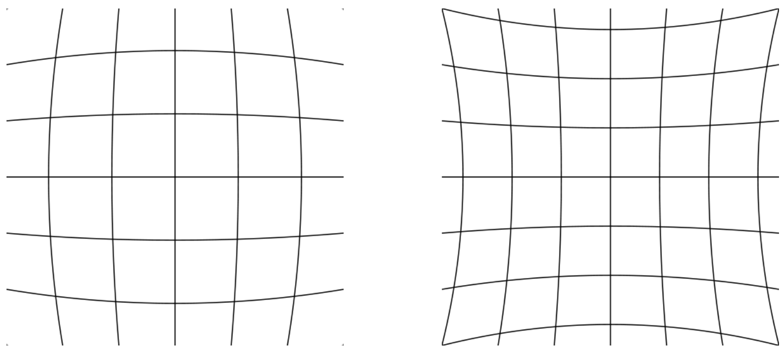

Fig. 4. Barrel and pincushion distortion.

The lenses also introduce the visual defect *chromatic aberration*, which is caused by varying refractive index of different wavelengths of light. Fortunately, also this unwanted effect can be significantly improved by an appropriate pre-transformation. Together with the barrel distortion, it can easily be done in a post-processing *pixel shader*, at the cost of some additional GPU processing time. The code for the pixel shader is available in the Oculus SDK documentation and samples [11].

Convergence. As was already mentioned, the cameras are required to have parallel axes, with the convergence point in infinity. They should not be angled towards each other (so-called “toe-in”) and also *Horizontal Image Translation* (HIT) should not be performed. Because of how the HMD optics works, the user does not have the sense of watching the LCD screen of the device, his eyes are naturally focusing on individual objects in a virtual world in front of him. If an object is very far away (almost infinitely) in reality or in the rendered 3D virtual world, it appears at the same position on both half-screens and thus both eyes look parallel to focus on it. Similarly natural it feels to watch objects in closer distances.

Design of a mobile robot teleoperator user interface

The main motivation of this project is to use Oculus Rift to display stereovision images to the operator of a mobile robot, so this is the first problem that must be solved.

The first obvious solution is to directly display the images from real-world cameras onto the screen of the HMD device, left camera on the left half and right camera on the right half. This creates a virtual reality for the HMD wearer so that he feels like if he was standing at the position of the mobile robot. There are however multiple complications in this case.

First, according to the previous chapter, the physical cameras need to have very specific lenses to exactly or at least very closely match the quite big FOV of the device, which is 98 °. This value can be achieved by using ultra-wide angle or fish-eye lenses, because standard cameras typically have FOV only between 40 and 60 degrees [12]. This was a serious problem with the testing mobile robot Hercules, because its stereovision cameras use a standard 4mm focal-length lenses with 49.1 ° vertical FOV. Another complication is the uncommon aspect ratio of the Oculus half-screens (4 : 5), where height is larger than width. Pictures from the cameras would need to be cropped and the operator would be able to see less of the world around the robot than with the current flat screen without stereovision.

It is still possible to use the existing cameras with normal lenses, without any cropping, if the pictures are scaled down so that they occupy only the middle portion of each half-screen, which exactly matches the corresponding fraction of the total FOV. Practical testing proved that this solution works quite well and a good 3D illusion is achieved, but only a part of the whole FOV is used, which denies one of the main advantages of Oculus Rift.

The biggest problem with direct display of camera images is that it caused nausea to most testing subjects (this applies to both situations – with and without the accurate FOV). The reason behind this is that Oculus Rift makes the user feel really immersed in the VR and so his brain expects all input information from eyes and the vestibular system in his inner ear to match. A possible way how to reduce this source of nausea is to use the head-tracking sensors of Oculus and rotate the physical cameras exactly the same way, with as low latency as possible. The Oculus Rift documentation recommends the maximal latency to be around 40 ms [11], but this is almost impossible to achieve, because during this time we need to process the head tracking data, send a corresponding camera rotation command to the robot, the cameras must mechanically rotate, new images must be read from the camera and transmitted to the operator and finally displayed in the HMD device. Even if the data transfer is very fast, dynamics (acceleration) can possibly be the biggest source of lag. This is the case of Hercules where cameras are mounted on the last link of the manipulator arm and the only way how to look around is to move the whole arm. The acceleration and deceleration times are too high and also the maximal velocity is not large enough to follow head movements in real time.

There is even a bigger problem if the robot moves, because in this case the cameras are moving in a way completely unrelated to the head movement of the operator. And especially on uneven terrain when the images shake, it is extremely inconvenient to look through the robot cameras using a HMD.

Virtual operator station

Because of the described problems, a new system was developed – a *virtual operator station*. The operator wearing HMD is put to a virtual space (“room”) created completely in a computer and can freely look around by head movements. This virtual room contains several 3D objects, primarily a big rectangular plane with images from the stereovision cameras displayed on it. The room itself is fully black, to provide as little distraction as possible.

Watching the video from the cameras this way feels for the operator as naturally as watching a television or cinema screen with a 3D technology in real life. The biggest source of nausea is removed for most testing subjects, because the brain feels to be part of the virtual room and thus expects the virtual screen to be fixed in space, or to move naturally with head movements, which can easily be accomplished.

The virtual room is watched from a pair of imaginary stereovision cameras configured according to the Oculus requirements. This pair of cameras is fixed in one point in the room (see the camera symbols on Fig. 5) and can rotate (look around) in all three axes (yaw, pitch, roll) based on user’s head movements.

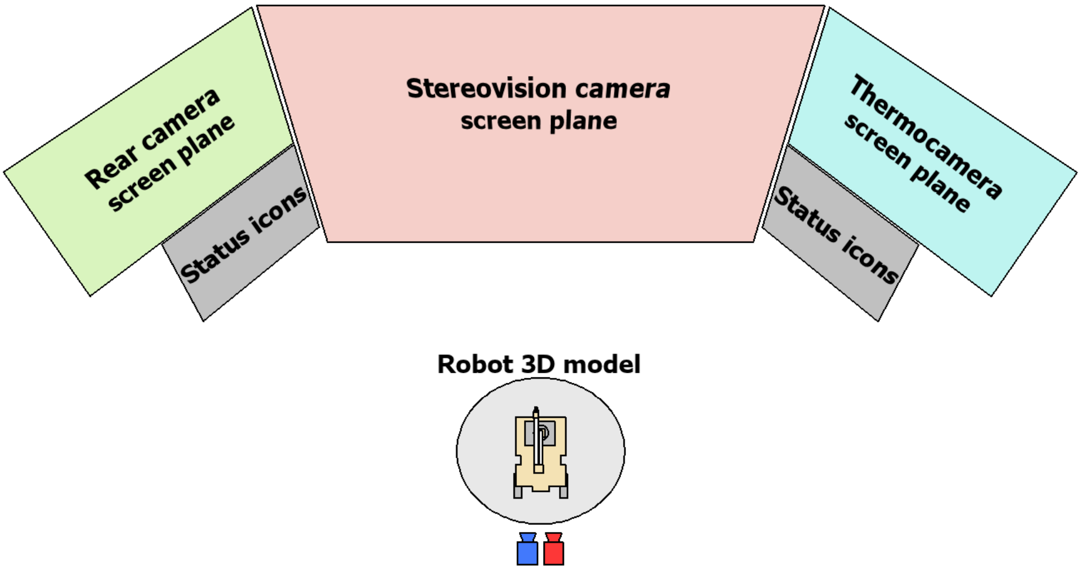

Fig. 5. Schema of the virtual operator station.

The biggest rectangle is a virtual screen plane with images of the stereovision cameras; this rectangle is rendered differently for the left and right eye – each time with the appropriate camera image. The screens on left and right sides display pictures from a secondary (rear) camera and from a thermovision camera. Right in front and slightly down is rendered a 3D model of the mobile robot with the arm at the real actual position, which helps the operator a lot to during manipulation tasks. Additional necessary information about the robot, as operating modes, sensor data etc., are displayed below the smaller camera images. Because of the low resolution of the Oculus SDK version, visual representations of information (icons, symbols, images) are preferred, or a large font must be used for text. After some testing of usability, we decided to display the control elements twice – once on each side, so that they are in the operator's sight most of the time. Important icons can be also displayed over the camera image, as for example the gripper icon on Fig. 6.

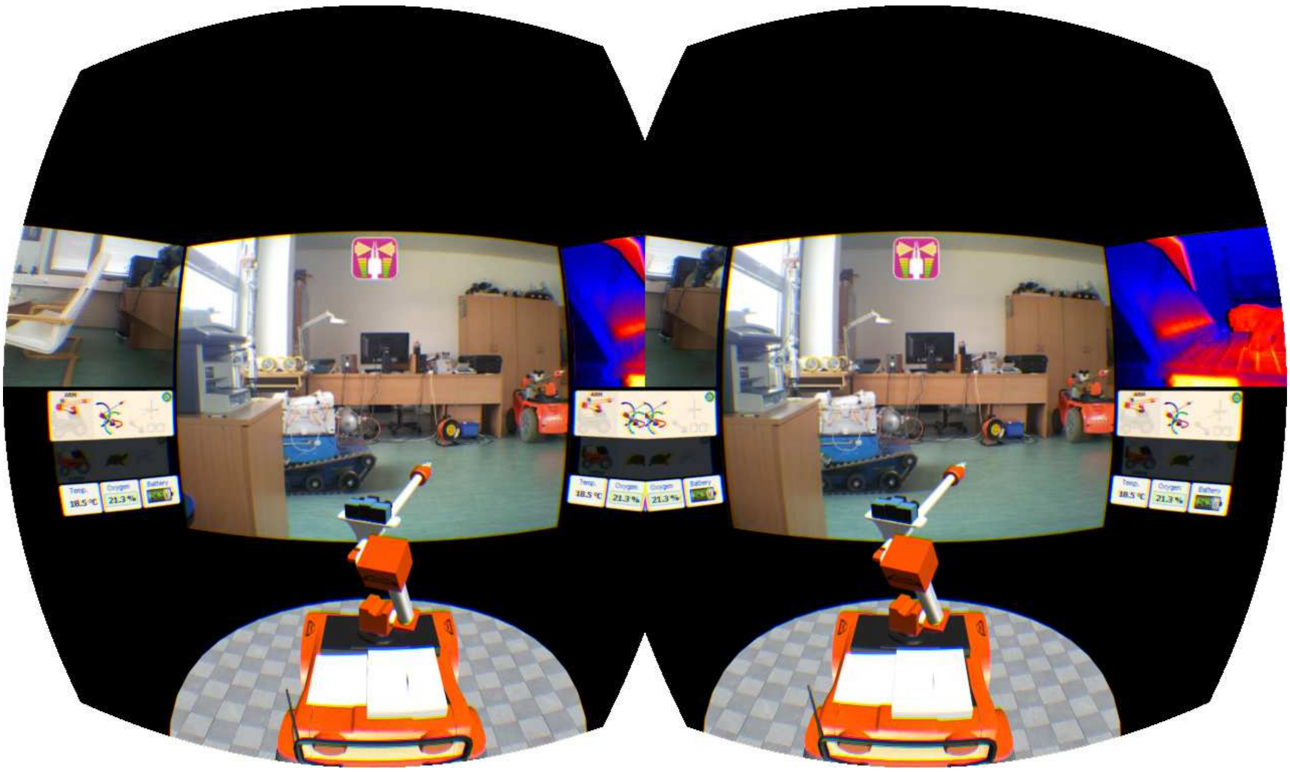

Fig. 6. Image displayed on the Oculus Rift screen: operator is looking ahead.

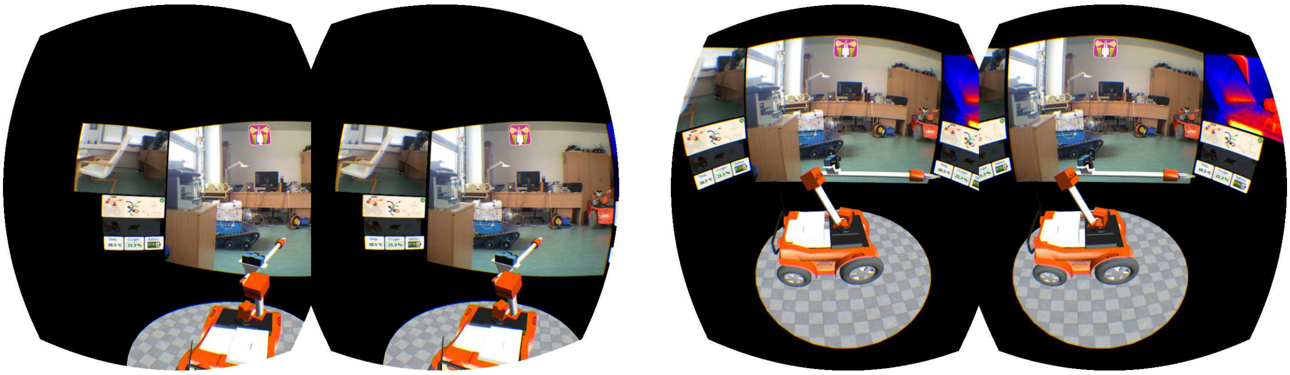

Fig. 7. Images displayed on the Oculus Rift screen: operator is looking to the left and down.

All rectangular planes with textures (camera images or control elements) are oriented vertically in the virtual world, so that their normal vector is perpendicular to the vertical axis of global coordinate system. And at the same time the planes are rotated towards the virtual head position (Fig. 9b).

The previous images show the actual view of the virtual world with applied distortion and chromatic aberration, as it is sent to the HMD device LCD. On Fig. 6 the user is looking straight ahead towards the big screen, on Fig. 7 the user is looking to the left to focus more on the rear camera and then to the bottom to better watch the robot model. The robot model on the last image is rotated sideways, which is a feature that the operator can control manually by a gamepad.

Visibility. Only a part of the Oculus LCD can be easily seen through the lenses. The visible part of Fig. 6 is highlighted on Fig. 8. Content of the virtual room was carefully configured based on testing, to show comfortably the whole plane with the main camera image and the robot arm in front of it, in a very convenient location. The operator can look just slightly to the left or right and will be able to see also most of the secondary camera image and all additional information about the robot.

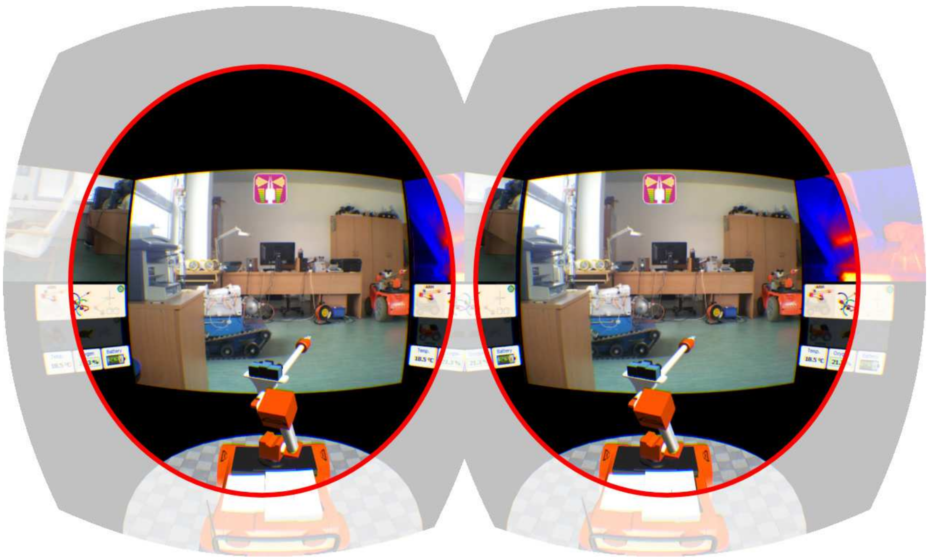

Fig. 8. Physically visible section of the rendered image.

The virtual operator station also provides a “zoom-out” mode, in which all screen planes move away from the user so that they all are visible at the same time without having to look around, at the cost of being smaller and thus less detailed. Frequent transition between the two modes is however not recommended, as it is a potential source of nausea or eye-strain. All screens are still oriented towards the virtual cameras, as shown on Fig. 9b.

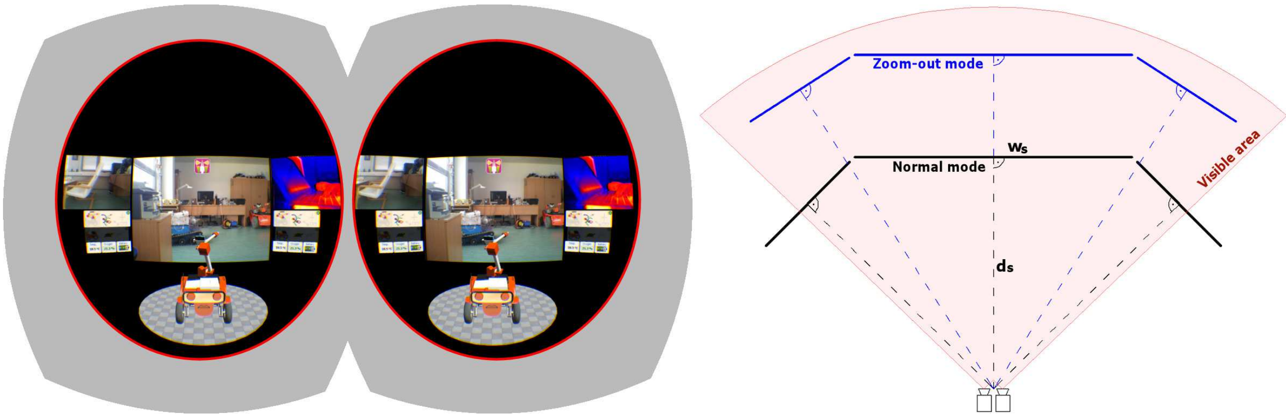

Fig. 9. The zoom-out mode with schematic view of the screens rotating towards the user.

It may be necessary to display some crucial information, for example error messages or symbols, on a floating plane which is in front of any other object and also is moving in the virtual world to be always clearly visible to the operator, regardless how his head is actually rotated. This also must be done with caution and only when really needed, because any similar violations of the expected behavior can be inconvenient for the user.

Stereovision cameras

As already mentioned, the images from stereovision cameras are displayed on the biggest virtual screen. This however is not so straightforward, as attention must be paid to convergence.

In the virtual scene, the screen plane is w_s = 3 m wide and is rendered in a distance d_s = 2.4 m from the user (in the normal mode, Fig. 9b). To focus on this object, the user's eyes naturally converge on it just like if it was a real object 3 meters far away.

There is a stereovision image displayed on this screen, which creates the illusion of additional depth – the objects on the picture appear at different distances than the screen itself, exactly as in a 3D cinema. An object on the image is placed exactly at the distance of the virtual screen if it is at the same pixel position on both images. For physical cameras with parallel axes and the point of convergence in infinity, this is valid only for objects in infinity, or very far away. This in fact means that the whole scene "thrusts out" of the screen into the space in front of it.

This is not very good for immersion, because the 3D model of the robot appears to collide with the image, although it is located in front of the screen, and there is also unmatching depth information near the side edges of the virtual screen, where the edge seems to be at two different depths at the same time.

A possible partial solution is to apply HIT before placing the images on the virtual screen plane, to change the convergence point. This does not violate the rule mentioned earlier, because now the camera images are not displayed directly in the HMD.

The images must be shifted outwards (the left one to the left and vice-versa), to put the convergence point further away. There is one very important limitation to the translation, because if the images are shifted too much, the convergence point of a particular pixel on the image could be "behind infinity" in the HMD virtual world and the eyes would have to rotate outwards to focus on such a point, which is physically not possible. This creates a lot of eye strain, because the brain is not used to this situation and ineffectually keeps trying to focus.

The maximum allowed translation is equal to the parallax p_{max} of the screen plane in VR, so that objects infinitely far on the camera image appear at infinite distance in the HMD:

$$d' = \frac{W}{4 \cdot tg\frac{\phi_x}{2}}, \qquad (6)$$

$$p_{max} = \frac{L_{IPD} \cdot d'}{2 \cdot d_s}, \quad (7)$$

where ϕ_x is the horizontal FOV and *d'* is distance of the projection plane. The p_{max} value is in LCD pixels, but because the camera image pixels do not map 1:1 to LCD pixels, the images must be shifted by p'_{max}:

$$w'_s = \frac{w_s \cdot d'}{d_s}, \quad (8)$$

$$p'_{max} = \frac{p_{max} \cdot W_c}{w'_s}, \quad (9)$$

where w_s represents the width of the virtual screen plane, w'_s is the width of the plane in pixels as it is displayed on the LCD and W_c is horizontal resolution of the camera images.

Shifting the images by this fixed amount does not fully solve the problem with closer objects thrusting out of the screen plane, but at least improves it by placing some objects behind the plane. A better solution would be to analyze the camera images, detect the furthest objects and shift according to them. If the furthest detected objects are not very (infinitely) far way, the HIT may be larger than p_{max}. The value can also be increased without image analysis if the robot is for example designed only for indoor environment, where the maximal possible depth is limited to few meters.

Conclusion

The new graphical user interface of the Hercules operator's control system was practically implemented and tested. The application was programmed in MS Visual C++ and uses Direct3D for graphics rendering. The first reaction of most users was that the system is very impressive, but besides this it is also practical together with stereovision cameras, because the operator can perceive relative depths of obstacles or important objects around the mobile robot.

The idea of a virtual operator station rendered in the HMD device instead of a direct display of the camera images proved to be convenient – while keeping a way how to visualize 3D view of the stereovision cameras, it also reduces possible sources of nausea. The approach also gives additional very interesting possibilities as far as display of other information is concerned. There can be multiple smaller floating screens around the user with images from secondary cameras, an interactive 3D model of the robot showing illustratively its current physical state and also all necessary control elements, information and warning or error icons. Good locations for these additional objects turned out to be on the sides of the main screen or below it, rather than above, because it is less comfortable to look up with the head. Depth in the VR can also be used as a way how to signalize importance of same information, by putting some icons closer to the user than others.

One of the possible problems is that with Oculus HMD on his head, the operator completely loses perception of his surroundings and sees only the VR. This could be improved by mounting two additional cameras directly on the HMD device and rendering their images directly in Oculus instead of the black background of the virtual room to create augmented reality. The cameras however need to have precisely chosen optical lenses to match the Oculus requirements, but this already has been done [13]. The other big problem of the low Oculus Rift resolution will probably be fixed in the final version of the product.

Another important discovery was that the stereovision cameras on the mobile robot Hercules (Fig. 1a) are not placed in a very good location, because both cameras see the last link of the arm from a very close distance (Fig. 1b) and produce extremely different views of it that are almost impossible for the brain to fuse together into a single stereo view. The arm occupies a significant portion of the pictures and thus it breaks the overall depth impression.

Acknowledgement

This article has been elaborated in the framework of the project Opportunity for young researchers, reg. no. CZ.1.07/2.3.00/30.0016, supported by Operational Programme Education for Competitiveness and cofinanced by the European Social Fund and the state budget of the Czech Republic. This article has been also supported by specific research project SP2014/176 and financed by the state budget of the Czech Republic.

References

[1] Cybernet. *Operator Control Unit* [online]. Available from WWW: <http://www.cybernet.com/products/robotics.html>.

[2] *Orpheus Robotic System Project* [online]. Available from WWW: <http://www.orpheus-project.cz/>.

[3] Fong, T., Thorpe, C. Vehicle Teleoperation Interfaces. *Autonomous Robots.* 2001, vol. 11, pp. 9-18. ISSN 0929-5593.

[4] Amanatiadis, A., Gasteratos, A. Stereo Vision System for Remotely Operated Robots. *Remote and Telerobotics*. 2010. ISBN 978-953-307-081-0.

[5] Wikipedia. *Head-mounted display* [online]. Available from WWW: <http://en.wikipedia.org/wiki/Head-mounted_display>.

[6] Virtual Realities, Ltd. *Stereoscopic 3D* [online]. Available from WWW: <http://www.vrealities.com/products/stereoscopic-3d>.

[7] Department of Robotics. *Hercules* [online]. Available from WWW: <http://robot.vsb.cz/mobile-robots/hercules/>.

[8] Kot, T., Krys, V., Mostýn, V., Novák, P. Control System of a Mobile Robot Manipulator. In *Proceedings of International Carpathian Control Conference (ICCC)*. 2014 (in print). ISBN: 978-1-4577-1867-0.

[9] Oculus VR. *Oculus Rift* [online]. Available from WWW: <http://www.oculusvr.com/rift/>.

[10] Sensics. *xSight HMD* [online]. Available from WWW: <http://sensics.com/head-mounted-displays/technology/xsight-panoramic-hmds/>.

[11] Oculus VR. *Oculus Rift SDK Overview, SDK version 0.2.5* [online]. Available from WWW: <https://developer.oculusvr.com/>.

[12] Wikipedia. *Angle of view* [online]. Available from WWW: <http://en.wikipedia.org/wiki/Angle_of_view>.

[13] Steptoe, W. *AR-Rift (Part 1)* [online]. Available from WWW: <http://willsteptoe.com/post/66968953089/ar-rift-part-1>.