A Virtual Presence System Design with Indoor Navigation Capabilities for Patients with Locked-In Syndrome

Jens Garstka*, Simone Eidam†, and Gabriele Peters*

Human-Computer Interaction
Faculty of Mathematics and Computer Science
FernUniversität in Hagen – University of Hagen
D-58084 Hagen, Germany

Email: {jens.garstka, gabriele.peters}@fernuni-hagen.de*, eidam.simone@gmail.com†

***Abstract*—In this article, we present a prototype of a virtual presence system combined with an eye-tracking based communication interface and an indoor navigation component to support patients with locked-in syndrome. The common locked-in syndrome is a state of paralysis of all four limbs while a patient retains full consciousness. Furthermore, also the vocal tract and the respiration system are paralyzed. Thus, the virtually only possibility to communicate consists in the utilization of eye movements for system control. Our prototype allows the patient to control movements of the virtual presence system by eye gestures while observing a live view of the scene that is displayed on a screen via an on-board camera. The system comprises an object classification module to provide the patient with different interaction and communication options depending on the object he or she has chosen via an eye gesture. In addition, our system has an indoor navigation component, which can be used to prevent the patient from navigating the virtual presence systems to critical areas and to allow for an autonomous return to the base station using the shortest path. The proposed prototype may open up new possibilities for locked-in syndrome patients to regain a little more mobility and interaction capabilities within their familiar environment.**

***Index Terms*—biomedical communication; human computer interaction; eye tracking; indoor navigation; virtual presence.**

I. INTRODUCTION

This article describes an extension of the previous work of Eidam et al. [1]. Undoubtedly, it is a major challenge for locked-in syndrome (LIS) patients to communicate with their environment and to express their needs. Patients with LIS have, for example, to face severe limitations in their daily life. LIS is mostly the result of a stroke of the ventral pons in the brain stem [2]. The incurred impairments of the pons cause paralysis, but the person keeps his or her clear consciousness. The grade of paralysis determines the type of LIS and has been classified in classic, total and incomplete LIS. Incomplete LIS means that some parts of the body are motile. Total LIS patients are like classic LIS patients completely paralyzed. However, the latter ones still can perform eyelid movements and vertical eye movements that can be used for communication. Therefore, several communication systems for classic LIS patients have been designed in the past.

This article introduces an eye-gesture based communication interface for controlling movements of a virtual presence system (VPS) and for selecting objects of the environment with the aim to interact with them.

In the presented prototype, the patients will see exemplary scenes of the local environment instead of the typically used on-screen keyboard. These scenes contain everyday objects, e.g., a book the impaired person wants to get read, which can be selected using a special eye gesture. After selection, the patient can choose one of various actions, e.g., “I want to get read a book” or “please, turn the page over”. A selection can either lead to a direct action (light on/off) or to a notification of a caregiver via text-to-speech. Moreover, the prototype allows the LIS patient to control a VPS. For this purpose, different eye gestures controlling the VPS are presented and discussed in this article. We also show an effective but cheap implementation of an indoor navigation component to enable the VPS to maneuver itself back to the base station taking the shortest way possible.

In a long-term perspective, the aim is to build a system where the object selection screen mentioned above shows the live view of the environment captured by the on-board camera of the VPS. This requires the implementation of an object classification approach for the most common objects. Each of the recognizable object classes provides an adjustable set of particular interactions/instructions. By this means, a VPS enables the LIS patient to interact with an environment in a very direct way.

The article is organized as follows: Section II gives a short introduction to eye tracking and describes different existing communication systems for LIS patients using eye tracking approaches. Furthermore, the section provides a brief overview of indoor navigation approaches. In Section III the concept and implementation details of our object-based interaction are presented. In the subsequent Section IV we introduce the models of the eye tracking interface controlling the VPS and the indoor navigation used to enable the VPS to autonomously move to the base station on the shortest path. Finally, the evaluation results will be presented in Section V and discussed in Section VI. The article concludes with a description of future work in Section VII.

II. RELATED WORK

This section starts with a brief overview on eye tracking techniques and already existing systems that support LIS patients with their communication. Finally, a short sub-section gives an overview on methods for indoor navigation with focus on impaired persons.

A. Eye Tracking

Many existing eye tracking systems use the one or other kind of light reflection on eyes to determine the direction of view. The human eye reflects incident light at several layers. The eye tracking device used for controlling the prototype employs the so-called method of dark-pupil tracking. Dark-pupil-tracking belongs to the video-based eye tracking methods. Further examples are bright-pupil- and dual-Purkinje-tracking [3].

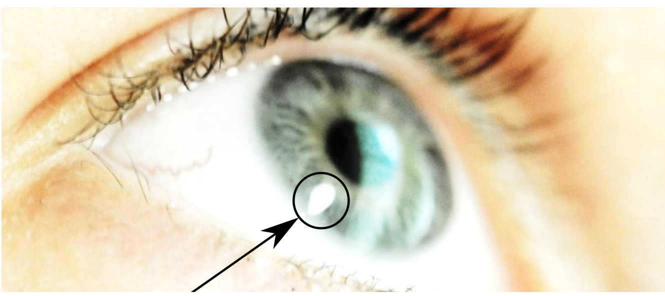

Fig. 1. The light reflection used by many eye trackers is called the glint.

For video-based systems, a light source (typically infrared light) is set up in a given angle to the eye. The pupils are tracked with a camera and the recorded positions of pupil and reflections are analyzed. Based on the pupil and reflection information, the point of regard (POR) can be calculated [3]. In Figure 1, the white spot just below the pupil shows a reflection of an infrared light on the cornea. This reflection is called the glint. In case of dark-pupil tracking, it is important to detect both, the pupil center and the glint. The position of the pupil center provides the main information about the eye gaze direction while the glint position is used as reference. Since every person has individually shaped pupils, a onetime calibration is needed. In case of a stationary eye tracker, also the distance between the eyes is determined to calculate the position of the head relative to the eye tracker.

B. Communication Systems for LIS Patients

There are many prototypes that have been developed in order to support LIS patients with their communication. Many of them are video-based eye tracking systems. One of the first systems was the communication project ERICA developed in 1989 [4]. With the help of the system users were enabled to control menus with eyes. They were able to play computer games, to hear digitized music, to use educational programs and to use a small library of books and other texts. Additionally, ERICA offered the possibility to synthesize speech and control nearby devices. Currently available and commercial communication systems for LIS patients are basically based on ERICA. These systems include the Eyegaze Edge Talker from LC Technologies and the Tobii Dynavox PCEye Go Series. The Tobii solution provides another interaction possibility called “Gaze Selection” in addition to an eye controlled mouse emulation. It allows a two stage selection, whereas starring at the task bar on the right side of the screen enables a selection of mouse options like right/left button click or the icon to display a keyboard. Subsequently, starring on a regular GUI-element triggers the final event (such as “open document”). Two-stage means that the gaze on the target task triggers a zoom-in event. It is said, that this interaction solution is more accurate, faster and reduces unwanted clicks in comparison to a single stage interaction.

Furthermore, current studies present alternative eye based communication systems for LIS patients. For example, the prototype developed by Arai and Mardiyanto, which controls the application surface using an eye gaze controlled mouse cursor with the eyelids to trigger the respective events [5]. This prototype offers the possibility to phone, to visit websites, to read e-books, or to watch TV. An infrared sensor/emitter-based eye tracking prototype was developed from Liu et al., which represents a low-cost alternative to the usual expensive video-based systems [6]. With this eye tracking principle, only up/down/right/left eye gaze moves can be detected as well as staying in the center using the eyelids to trigger an event. By using the eye movement, the user can move a cursor in a 3×3 grid from field to field. And by using the eyelids, the user can finally select the target field. Barea, Boquete, Mazo, and Lpez developed another prototype that is based on electrooculography [7]. This prototype allows by means of eye movements to control a wheelchair allowing an LIS patient to freely move through the room.

All prototypes that have been discussed so far are based on an interaction with static contents on screen, for example a virtual keyboard. However, the prototype presented in this contribution shows a way to select objects in images of typical household scenes by a simulated object classification. This allows an evaluation of the system without the need of a full classification engine. The latter will lead to a selection of real objects in the patient’s proximity.

C. Indoor Navigation

The use of GPS for indoor navigation is often not possible as ceilings and walls almost completely absorb the weak GPS signal. However, there are numerous alternatives including ultrasonic, infrared, magnetic, and radio sensors. Unfortunately, in many cases the position is not determined by the mobile device. Instead, it is determined from the outside. This requires a permanent electronic infrastructure, which often can not be retrofitted without major effort.

The following publications provide a brief overview of indoor navigation solutions. The survey by Mautz and Tilch [8] contains a good overview of optical indoor positioning systems. Nuaimi and Kamel [9] explore various indoor positioning systems and evaluate some of the proposed solutions. Moreover, Karimi [10] provides a wide overview of general approaches to indoor navigation in his book.

Considering that QR codes are used for positioning in our approach, an overview of recent publications focusing on QR codes follows.

The indoor navigation described by Mulloni et al. is an inexpensive, building-wide orientation guide that relies solely on mobile phones with cameras [11]. The approach uses bar codes, such as QR codes, to determine the current position with a mobile phone. This method was primarily used at conferences. Information boards containing appropriate QR codes were used to determine the current location of visitors.

The work of Li et al. is focused on robot navigation and the question of how QR codes can be identified and read even under bad lightning conditions [12]. For this purpose, they combine and optimize various image filters for the mentioned use case.

Gionata et al. use a combination of an IMU (rotational and translational sensors) and QR codes for an automated indoor navigation of wheelchairs [13]. The QR codes are used as initial landmarks and to correct the estimated position of a wheelchair after driving a certain period. The movement of the wheelchair between two QR codes is approximated with an IMU.

A somewhat different intended use of the QR codes is shown in the paper of Lee et al. [14]. They use QR codes to transfer navigational instructions to a mobile robot along a predefined route. These instructions hint the robot where it needs to turn, for example, left or right.

Zhang et al. use QR codes as landmarks to provide global pose references [15]. The QR codes are placed on the ceilings and contain navigational information. The pose of the robot is estimated according to the positional relationship between QR codes and the robot.

In brief, it has been found that that QR codes or similar markers represent an effective and proven means for indoor navigation. In context of our work presented in this article, we will combine the work of Zhang et al. with a simple floor plan [15]. This will be discussed more in detail in Section IV.

III. INTERACTION

This section describes the concepts and the implementation of our interface for object-based interaction and communication using an eye tracker.

A. Concept

The following section provides an overview of the basic concept of this work. As already mentioned, the impaired person will see an image of a scene with typical everyday objects. This image is representative for a real scene, which is to be captured by a camera and analyzed by an object classification framework in future work. Figure 2 shows an image of one possible scene. The plant can be used by a LIS patient to let a caregiver know, that one would like to be in the garden or park, the TV can be used to express the desire to watch TV, while the remote control directly relates to the function of the room light. The red circle shown at the center of the TV illustrates the POR calculated by the eye tracker. The visual feedback by the circle can be activated or deactivated, depending on individual preferences.

Fig. 2. An example scene used with this prototype.

An object is selected by starring a predetermined time on the object, what we call a “fixation”. With a successful fixation a set of options will be displayed on the screen. A closing of the eyelids is used to choose one of these options. Depending on the selected object, a direct action (e.g., light on/off) or an audio synthesis of a corresponding text is triggered (e.g., “I want you to read me a book.”).

Furthermore, other eye gestures have been implemented to control the prototype. By means of a horizontal eye movement, the object image is changed. However, the latter is only an aid during the test phase without an implementation of a real object classification to avoid the use of a keyboard or mouse. By means of a vertical eye movement, the object-based interaction and communication mode is switched to the robot controlling mode and vice versa.

B. Implementation

The eye tracking hardware used is a stationary unit with the name RED manufactured by SensoMotoric Instruments (SMI). RED comes with an eye tracking workstation (a notebook) running a software, which is named iView X. The latter provides a network component to allow an easy communication between the hardware and any software through a well-defined network protocol.

Figure 3 gives a brief overview of all components of our prototype. Area 1 shows the patient’s components to display test scenes with different objects. The stationary eye tracking unit is shown in area 2. Area 3 shows the eye tracking

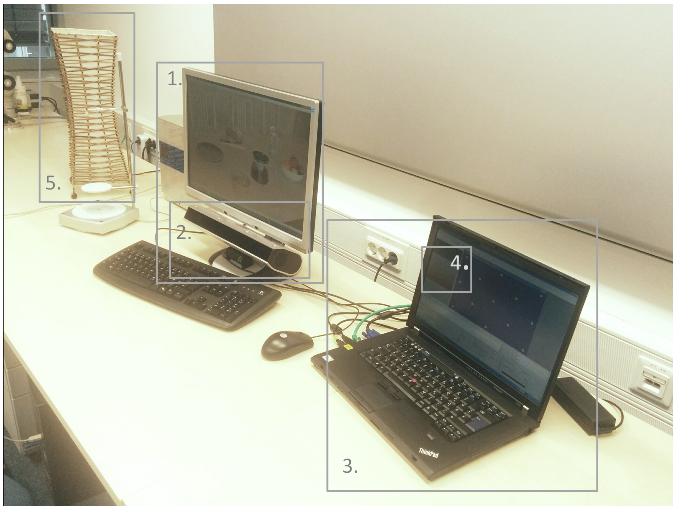

Fig. 3. The eye tracking components of the prototype.

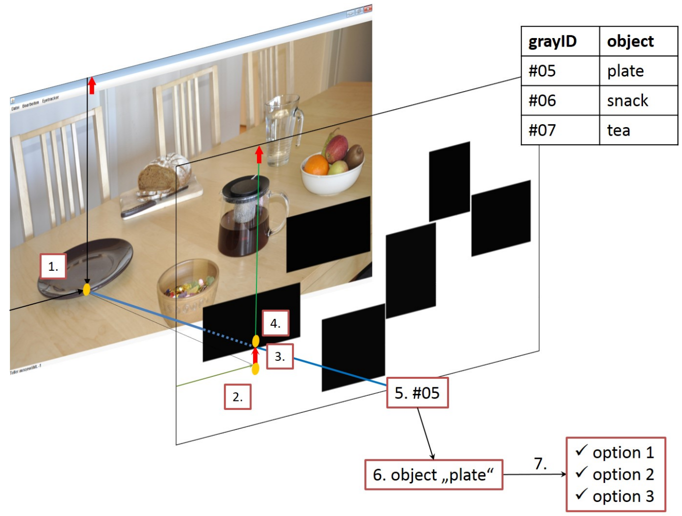

Fig. 4. Elements used to simulate the object classification.

workstation with the eye tracking control software in area 4. Finally, area 5 contains a desk lamp, which can be turned on and off directly with a fixation of the remote control shown in Figure 2.

C. Eye Gesture Recognition

Eye gesture recognition is based on the following principle: the received POR-coordinates from the eye tracker are stored in circular buffer. At each coordinate insertion the buffer is analyzed for eye gestures. These eye gestures are a fixation, a closing of the eyelids, and a horizontal/vertical eye movement. The following values can be used to detect these eye gestures:

- the maximum x- and y-value: $x_{\max}$, $y_{\max}$
- the minimum x- and y-value: $x_{\min}$, $y_{\min}$
- the number of subsequent zero values: c

The detection of the fixation is performed as follows:

$$|x_{\max} - x_{\min}| + |y_{\max} - y_{\min}| \leq d_{\max}, \quad (1)$$

where $d_{\max}$ is the maximum dispersion while the eye movements are still recognized as fixation. The value of $d_{\max}$ is individually adjustable.

The detection of a closing of the eyelids is realized by counting the amount c of subsequent coordinate pairs with zero values for x and y. Zeros are transmitted by the eye tracker, when the eyes couldn't be recognized. This occurs on the one hand when the eyelids are closed, but on the other hand when the user turns the head or disappears from the field of view of the eye tracker. Therefore, this event should only be detected if the number of zeros corresponds to a given time interval:

$$(c > c_{\min}) \wedge (c < c_{\max}) \quad (2)$$

All variables $c_{\min}$ and $c_{\max}$ can be customized by the impaired person or the caregiver, respectively.

The combination of these two different approaches is a benefit, because object selection is realized through the fixation while option selection is done by closing the eyelids. The latter allows the LIS patient to rest the eyes while the option panel is open. Hence, the patient can calmly look over the offered options in order to get an overview.

For the horizontal eye gesture detection, a given range of x-values must be exceeded while the y-values remain in a small range, and vice versa for the vertical eye gesture. As already mentioned, the horizontal eye movement is used to switch between different images. But this functionality is not a part of a later system and is merely a simple additional operation to present a variety of objects while using this prototype. The vertical eye movement (vertical eye gesture) is used to switch between the object-based interaction and communication mode and the robot controlling mode.

D. Simulated Object Classification

Figure 4 shows schematically the principle of the simulated object classification. It is based on a gray-scale image that serves as a mask for the scene image. On this mask the available objects from the scene image are filled with a certain gray value. Thus, each object can be identified by a unique gray value ($grayID$). The rear plane illustrates the screen. The coordinates that correspond to a fixation of an object (1.) refer to the screen and not to a potentially smaller image. Thus, these raw coordinates require a correction by an offset (2. & 3.). The corrected values correspond to a pixel (4.) of the gray-scale image whose value (5.) may belong to one of the objects shown. In case of the example illustrated in Figure 4 this pixel has a gray value of 5 and corresponds to the object "plate" (6.). Finally, either all available options will be displayed (7.) or nothing will happen in the case the coordinates do not refer to a known object.

IV. NAVIGATION

Control and navigation of a VPS should primarily take place through eye gestures of an impaired person. But the system should autonomously return to the base station in times when the VPS is not in use. If the latter shall be achieved without boring random movements, as it can be observed frequently on robotic vacuum cleaners, the system must have knowledge of the local environment. For the tasks outlined in this article, QR codes are an effective means, mainly because they are very inexpensive and they are easy to install. However, the location by itself as described by Alessandro Mulloni et al. is not enough [11]. Even the approach of Zhang et al. putting some navigational information in the QR codes is not sufficient for some application scenarios [15].

Ideally, the robot knows a complete map of the local indoor environment.

A. Maps

One possibility to achieve a map of the local environment can be a commonly used method with the acronym SLAM ("Simultaneous Localization and Mapping"). Jeong and Lee describe a SLAM approach where they only use ceilings captured with a camera pointing upwards to create a map of the indoor environment [16]. Using this method, it is possible to identify QR codes that are placed on the ceiling (see Zhang et al. [15]) and put them into the map.

Alternatively, one can use a manually created floor plan. The latter would have the advantage that the floor plan is complete and can contain various extra information. These additional information may include:

- The exact position of the base station.
- The exact position and orientation of each QR code placed on the ceilings.
- The ceiling height.
  This is mainly of interest for a precise positioning or position correction of the robot based on the QR codes, which are placed on the ceiling. With the knowledge of the ceiling height, the opening angle of the camera, and the viewing direction upwards, the relative displacement of the robot with respect to the QR codes can easily be triangulated.
- Regions that should not be entered.
  Considering the fact that the target group of the approach presented in this article will have difficulties to control the VPS even with simplest eye gestures, it is useful to be able to mark certain regions that should be avoided. This could be, for example, a table with chairs where the robot can get stuck, or an area with sensitive objects like plants.

An exemplary floor plan is shown in Figure 5. It contains the positions and orientations of the QR codes. The QR codes themselves initially contain only an ID for the identification of each code. However, there is also the possibility to encode extra information in each QR code.

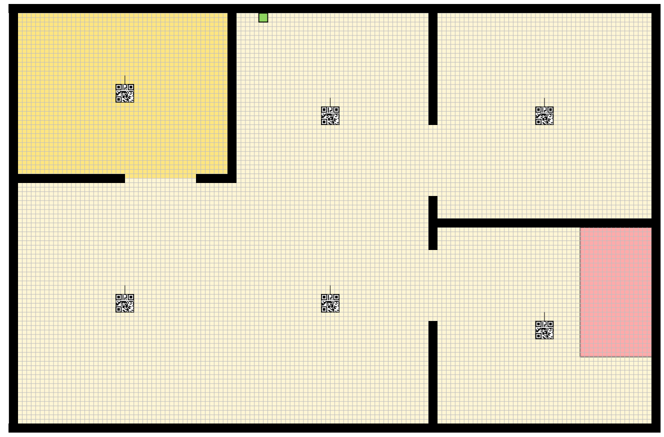

Fig. 5. An exemplary floor plan used for indoor navigation.

The floor plan can be implemented as a pixel image. In our case (see Figure 5) each pixel has an edge length of $5 \times 5\,cm$. The different yellow shades shown in Figure 5 indicate different ceiling heights. The area marked in red indicates a region that should be avoided.

B. Control

When controlling a robot with eye gestures several questions have to be answered:

- What eyes gestures can be used to activate or deactivate the control?
- What should happen if the eye tracker fails to detect the eyes?
- What eye gestures should be used to control the VPS?
- When should the robot return to the base station?
- Are there ways to define regions on the screen where the eyes can rest without triggering an eye gaze event?

To enable and disable the VPS control, we use an eye lid closure similar to Subsection III-C, i.e., the eye lid closure is within a given time interval $(c > c_{\min}) \wedge (c < c_{\max})$, where $c_{\min}$ and $c_{\max}$ can be customized. When switched off, an impaired person can switch between the object-based interaction and communication mode and the robot controlling mode by a vertical eye movement.

If the eye tracker fails to detect the eye gaze position for a period of $> c_{\max}$, it gets into a fail state. This results in an immediate stop of the VPS. To continue, a patient needs to reactivate the eye gaze control with a lid closure.

In general, a live view of the area in front of the VPS is always visible on the screen. This ensures that a patient can examine how and where the VPS is moving. Three different models of eye gestures to control the VPS are currently tested. The first model, shown in Figure 6, corresponds to the model of a joystick.

This means that an eye gaze pointing to the upper half of the screen accelerates the VPS in a forward motion. Pointing to the left and to the right causes a corresponding rotation. Since

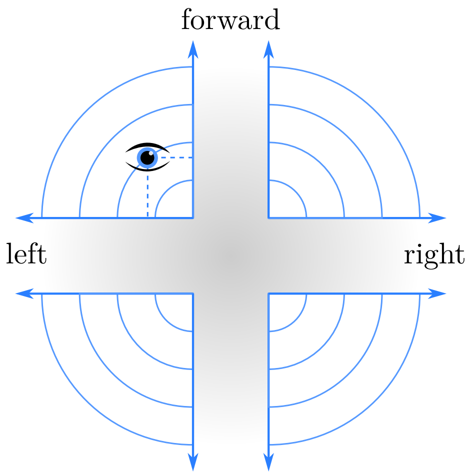

Fig. 6. Eye gaze control model I: the joystick mode.

an exact positioning of eye gazes can be very stressful, the area of a neutral position has been widened. This is visualized through the gray gradient shown in Figure 6.

There is also the possibility to drive backwards. However, according to the current prototype, the VPS has no rear camera. Thus, a reverse drive would be a blind drive. For this reason this ability has been removed in a second control model (see Figure 7).

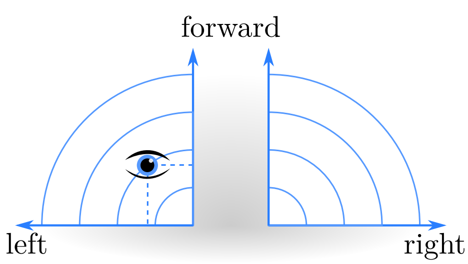

Fig. 7. Eye gaze control model II: the half joystick mode.

The latter model has another advantage: If the control sensitive area is located only on the upper half of the screen, the entire lower half of the screen can be used to rest the eyes.

The third model corresponds to a vertical slider. It can be used to do a turn-on-the-spot or to move straight forward by pointing to the upper or lower half of the screen. To switch between the two control states, we will use a fixation in a small area in the center of the screen (see Figure 8).

The horizontal region left and right of this central area (gray-shaded region in Figure 8) can be used to rest the eyes. Moreover, it will make no difference where the eye gaze position is located horizontally. Therefore, this model is suitable especially for the aforementioned LIS patients whose movements have been degraded to the extent that they are limited to vertical eye movements.

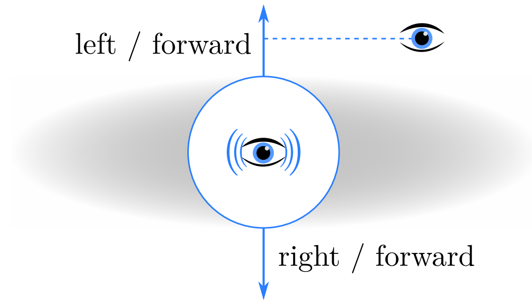

Fig. 8. Eye gaze control model III: slider mode.

C. *Shortest Path to Base Station*

An autonomous movement of a robot from a point A to a point B is a common and well solved problem in robotics. Path planning algorithms are measured by their computational complexity. The results depend on the accuracy of the map (floor plan), on the robot localization and on the number of obstacles.

If the underlying map is a raster map (e.g., a pixel image), one of the many variants of the A^* algorithm introduced by Hart et al. [17] is often used. Modern modifications and improvements like the work by Duchoň et al. [18] optimize the A^* algorithm for fast computation and optimal path planning in indoor environments.

In order to avoid contact with walls and doors and to pass the restricted areas in sufficient distance, the thickness of wall and blocked regions is enlarged by dilatation. Our robot has a radius of about $15\,cm$. In addition to the radius $10\,cm$ safety distance are used, to take account of inaccuracies in localization and movement of the robot. Accordingly, a dilation by $25\,cm$ or 5 pixels in the case of the presented map in Section IV-A is applied to the base map.

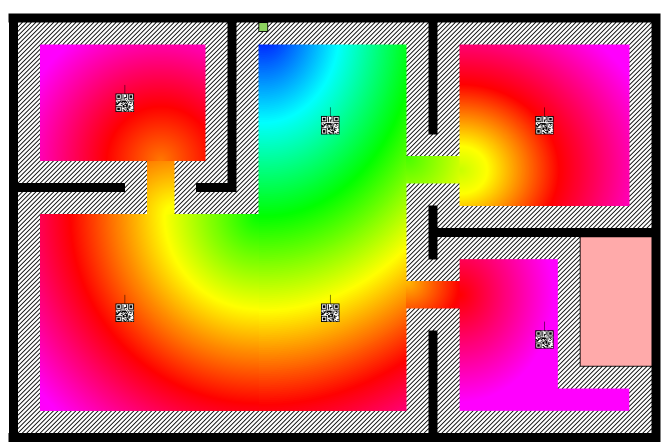

Fig. 9. Color gradient of shortest path to base station.

Figure 9 illustrates with a color gradient how a robot can find a direct path to the base station through gradient descent. The base station is depicted by the small green rectangle on the upper wall of the middle room. The colors indicate from purple

($> 8.5\,m$), over red ($\approx 6.8\,m$), yellow ($\approx 5.1\,m$), green ($\approx 3.4\,m$), and cyan ($\approx 1.7\,m$) to blue, the shortest path distance from an arbitrary point on the floor plan to the base station. The shaded areas show the $25\,cm$ wide safety distance along the walls.

D. Prototype

To build a prototype, an iRobot Roomba 620 vacuum cleaning robot is used as platform. It was extended by an access point and a USB to UART converter to send serial control command via network. In addition, two wireless cameras were mounted on top of the Roomba. One camera points forward, while the other camera points towards the ceiling. All devices get their electricity from the batteries of the Roomba. The prototype is shown in Figure 10.

Fig. 10. Prototype configuration based on a vacuum cleaning robot.

The Roomba has two separately controllable drive wheels. This enables the system to do a turn-on-the-spot and easily enables the implementation of the above-mentioned joystick mode.

Let x and y be the coordinates of the eye gaze position on the screen and c_x and c_y be the center coordinates of the screen. Further let s be a configurable speed factor. Then the speed values of the left and right wheel are:

$$v_l = ((x - x_c) + (y - y_c)) \cdot s \text{ and} \quad (3)$$

$$v_r = ((x - x_c) - (y - y_c)) \cdot s, \quad (4)$$

where v_l and v_r are the velocities of the left and right driving wheel. Figure 11 shows an exemplary view of the front camera.

V. RESULTS

The results can be divided into two parts. The first part deals with the object-based interaction, while the second part deals with the control of the robot.

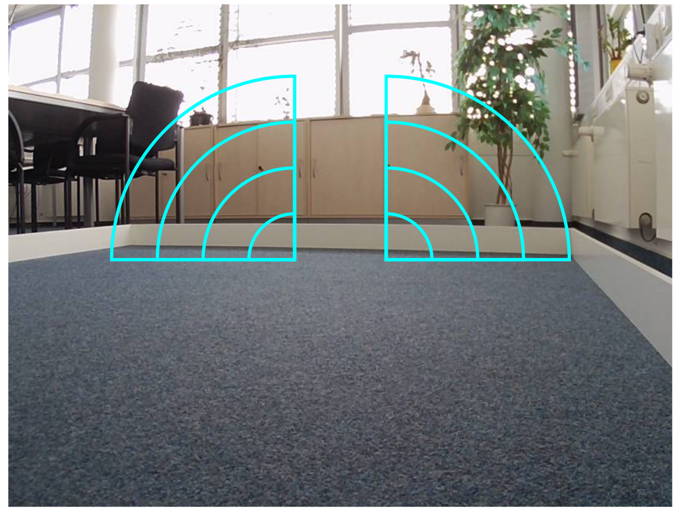

Fig. 11. An exemplary view of the front camera.

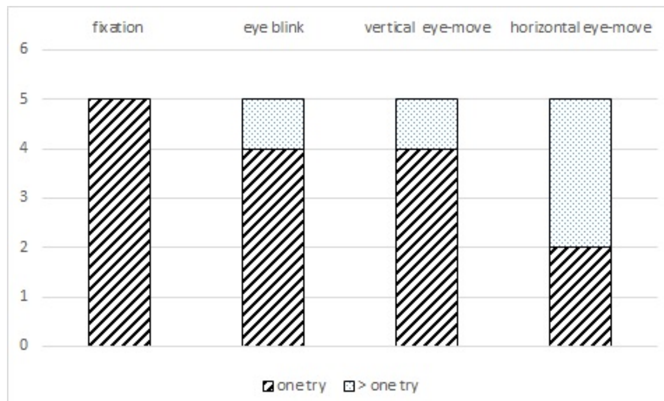

Fig. 12. Bar diagram of the eye gesture recognition.

A. Object-Based Interaction

The interface for object-based interaction has been tested by five persons to analyze its basic usability. Figure 12 briefly illustrates the results of the usability test. It shows whether a test person (subject) required one or more attempts to use a specific function successfully. During these tests, the subjects were able to validate the detected position of the eye tracker by means of the POR visualization. The diagram shows that none of the test persons had problems with the fixation. While the options were selected due to closing the eyelids, only one subject required several attempts. The same applies to the vertical eye movement. In a second pass, it turned out that precisely this subject requires other settings for a successful eye gesture recognition. Thus, more time for training and personal settings will help to achieve better results. However, it should be stated that this combination of object selection via fixation and option selection by closing the eyelids turned out to be a workable solution.

Figure 12 further shows that three of five test persons had difficulties to deal with the horizontal eye movement. Interviews with the subjects showed that it appears to be very difficult to control the horizontal eye movement to get a straight motion. Apart from that, it must be considered that in general LIS patients are not able to do horizontal eye movements.

In summary, it can be noted that the usability can be assessed as stable and accurate. With a well-calibrated eye tracker, the basic handling consisting of the combination of fixation and closing the eyelids is perceived as comfortable. Additionally, it is possible to adjust the eye gesture settings individually at any time. This enables an impaired person to achieve optimal eye gesture-recognition results and a reliable handling.

B. Controlling the Robot

The development of the controlling interface of the robot has nearly been completed. It stands to reason that the second model seems to be the the interface with the easiest control and the least symptoms of fatigue for the eyes. However, a detailed test is still pending.

VI. RESULTS AND DISCUSSION

Since this work is in progress, there are different parts of this work that need to be discussed, implemented and evaluated in the near future. We list the main points – even in parts – below:

- Currently, the LIS patient can only deactivate the eye tracking during the object-based interaction mode by switching to robot control mode. Thus, there should be a way to disable the fixation detection. Since eye gestures based eye movements have proved to be difficult, our idea is a combination of two consecutive fixations, e.g., in the upper left and lower right corners.
- Instead of the currently used static pictures displayed in object-based interaction mode, a live view of the VPS should be shown. But this requires a well functioning object classification.
- Thus, a major part of this work will be the classification of a useful set of everyday objects. Recently, deep convolutional neural networks trained from large datasets have considerably improved the performance of object classification (e.g., [19], [20]). At the moment, they represent our first choice.

In addition, there are many other minor issues to deal with. However, at this point these issues are not listed individually.

VII. CONCLUSION AND FUTURE WORK

The presented prototype demonstrates an interface to drive a VPS through a local environment and offers a novel communication and interaction model for LIS patients, where visible objects selected by eye gestures can be used to express the needs of the patients in a user-friendly way.

In contrast to the discussed state-of-art methods, which are based on an interaction with static content on screen, the direct interaction with the environment is a benefit in two ways. On the one hand, compared to the methods that use a virtual keyboard, our method is faster and less complex. And on the other hand, compared to the methods where pictograms are used, our method eliminates the search for the matching icon. Thus, the advantage of such a system is a larger flexibility and a greater interaction area, i.e., a direct connection to controllable things like the light, a TV, or a radio.

Our current work examines different models to control the movements of the prototype with eye gestures in a live view from the on-board camera of a VPS. Moreover, an autonomous navigation of a VPS using QR codes and a floor plan is currently tested to fit the particular situation of LIS patients.

Future work will include the ability to select objects individually from the local environment. This will enable the patients to use real objects for communication tasks with the help of an eye tracker. The interaction with the real environment via a live view will ensure a more intuitive interaction than the communication via static screen content and thus will provide LIS patients with even more freedom. In addition, in this scenario dynamic changes within the room (displacement or exchange of objects) will not affect the interaction range of a patient.

Independently of this, a LIS patient should alway have the ability to select a virtual keyboard to send individual messages as fall-back option.

REFERENCES

[1] S. Eidam, J. Garstka, and G. Peters, “Towards regaining mobility through virtual presence for patients with locked-in syndrome,” in *Proceedings of the 8th International Conference on Advanced Cognitive Technologies and Applications*, Rome, Italy, 2016, pp. 120–123.

[2] E. Smith and M. Delargy, “Locked-in syndrome,” *BMJ: British Medical Journal*, vol. 330, no. 7488, pp. 406–409, 2005.

[3] A. Duchowski, *Eye Tracking Methodology, Theory and Practice*. Springer-Verlag, 2007, ch. Eye Tracking Techniques, pp. 51–59.

[4] T. E. Hutchinson, K. P. White, W. N. Martin, K. C. Reichert, and L. A. Frey, “Human-computer interaction using eye-gaze input,” *IEEE Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1527–1534, November 1989.

[5] K. Arai and R. Mardiyanto, “Eye-based human computer interaction allowing phoning, reading e-book/e-comic/e-learning, internet browsing, and tv information extraction,” *IJACSA: International Journal of Advanced Computer Science and Applications*, vol. 2, no. 12, pp. 26–32, 2011.

[6] S. S. Liu, A. Rawicz, S. Rezaei, T. Ma, C. Zhang, K. Lin, and E. Wu, “An eye-gaze tracking and human computer interface system for people with als and other locked-in diseases,” *JMBE: Journal of Medical and Biological Engineering*, vol. 32, no. 2, pp. 111–116, 2011.

[7] R. Barea, L. Boquete, M. Mazo, and E. Lpez, “System for assisted mobility using eye movements based on electrooculography,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 4, pp. 209–218, 2002.

[8] R. Mautz and S. Tilch, “Survey of optical indoor positioning systems.” in *IPIN*, 2011, pp. 1–7.

[9] K. A. Nuaimi and H. Kamel, “A survey of indoor positioning systems and algorithms,” in *Innovations in Information Technology (IIT), 2011 International Conference on*. IEEE, 2011, pp. 185–190.

[10] H. A. Karimi, *Indoor wayfinding and navigation*. CRC Press, 2015.

[11] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg, “Indoor positioning and navigation with camera phones,” *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 22–31, 2009.

[12] W. Li, F. Duan, B. Chen, J. Yuan, J. T. C. Tan, and B. Xu, “Mobile robot action based on qr code identification,” in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*. IEEE, 2012, pp. 860–865.

[13] C. Gionata, F. Francesco, F. Alessandro, I. Sabrina, and M. Andrea, “An inertial and qr code landmarks-based navigation system for impaired wheelchair users,” in *Ambient Assisted Living*. Springer, 2014, pp. 205–214.

[14] S. J. Lee, J. Lim, G. Tewolde, and J. Kwon, “Autonomous tour guide robot by using ultrasonic range sensors and qr code recognition in indoor environment,” in *Electro/Information Technology (EIT), 2014 IEEE International Conference on*. IEEE, 2014, pp. 410–415.

[15] H. Zhang, C. Zhang, W. Yang, and C. Chen, “Localization and navigation using QR code for mobile robot in indoor environment,” in *2015 IEEE International Conference on Robotics and Biomimetics, ROBIO 2015, Zhuhai, China, December 6-9, 2015*, 2015, pp. 2501–2506.

[16] W. Y. Jeong and K. M. Lee, “CV-SLAM: a new ceiling vision-based SLAM technique,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada, August 2-6, 2005*, 2005, pp. 3195–3200.

[17] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[18] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, “Modelling of Mechanical and Mechatronic Systems Path Planning with Modified a Star Algorithm for a Mobile Robot,” *Procedia Engineering*, vol. 96, pp. 59–69, 2014.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, 2012, pp. 1106–1114.

[20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.