How to Always Keep an Eye on the User with a Mobile Robot? *

Alexander Vorndran, Thanh Q. Trinh, Steffen Müller, Andrea Scheidig, and Horst-Michael Groß[a]
[a] *Neuroinformatics and Cognitive Robotics Lab*, *Ilmenau University of Technology*, POB 100565, 98684 Ilmenau, Germany, e-mail: alexander.vorndran@tu-ilmenau.de

Abstract

Using mobile robots in rehabilitation is an upcoming trend in the field of robotic healthcare. A possible application is robot assisted self-training after orthopedic knee or hip endoprosthesis surgery. In this scenario, it is particularly important to be able to continuously observe the patients movements in order to give timely feedback on the ongoing exercise. This paper presents a complementary approach separately taking care of two fundamental prerequisites of an application scenario where a user performs gait retraining in rehabilitation under supervision of a mobile robot: driving at a certain constant distance in front of the user and keeping the user at a suitable position in the sensor's field of view. We show the applicability of our approach under laboratory and real-world conditions.

1 Introduction

Immediate feedback plays a vital role in gait retraining after hip or knee joint endoprosthesis surgery to avoid fallbacks into prior, unnatural and possibly damaging movement patterns. Since this type of surgery becomes more common nowadays, due to an aging population and improvements in medical techniques, fully supervised gait retraining is often not feasible with the available resources. One possible solution is to provide assistance using mobile robots, that may offer such a feedback during normally unsupervised self-training sessions.

Such a training session is usually initiated by the patient who seeks out the robot at a known location in the rehabilitation center. The patient then can watch an introduction video and selects a training duration according to its daily form. Now the patient can start to walk at whatever pace he feels comfortable at while the robot drives in front of him as shown in Fig. 1. The robot observes his gait and posture and should provide immediate feedback on how to improve the movement pattern whenever there is a significant deviation. For this approach to work, it is crucial to continuously keep the patient visible by the sensor needed to evaluate the ongoing exercise.

Therefore, this paper focuses on the aspect of controlling the robot's drive and a pan-tilt mounted camera in order to continuously keep a good view on the user during the training session, laying the foundation for autonomous robot-assisted self-training in this specific rehabilitation scenario. In the following, we describe related state-of-the-art approaches (Sec. 2) before introducing our target platform (Sec. 3) and deriving our system design based on the platform's capabilities and the requirements of our scenario (Sec. 4). After that, we present results obtained while validating our design in controlled circumstances as well as under real-world conditions in an orthopedic clinic (Sec. 5).

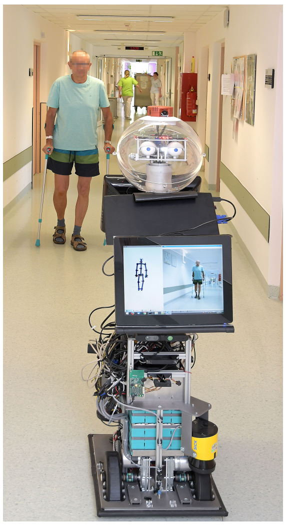


Figure 1 Patient during robot-assisted self-training where the robot drives in front of the user while observing his gait pattern and posture.

2 Related Works

To the best of our knowledge, there are no publications which focus on employing a mobile robot with a pan-tilt unit in orthopedic rehabilitation scenarios. However, using pan-tilt units for person tracking in addition to the robot's driving capabilities is a known concept in the robotics field. There are early publications like [1, 2, 3], which predate most of the modern techniques in computer vision and sensor development, already describing a control design for mobile robots with pan-tilt units to track humans in constrained laboratory environments.

Later approaches like [4] and [5] employ similar techniques but include depth sensor data in order to determine the tracking target position with higher accuracy while being computationally more efficient. Nevertheless, these approaches are still limited to laboratory environments, and in case of [4] to a static target.

In [6], the authors build upon the general trend of using

*This work has received funding from the Thuringian Ministry of Economy, Science and Digital Society and Thüringer Aufbaubank as part of project ROGER (grant agreement no. 2015FE9088) using financial resources of the European Regional Development Fund.

low-cost depth sensors such as Kinect or Kinect2 in fall and gait assessment scenarios and place a Kinect sensor on a small mobile robot. Their robot is able to follow the user autonomously using a gradient-based control-design [7]. The Kinect can then be used to evaluate the user's gait pattern, building the base to make predictions whether the user is about to fall. However, they explicitly state that an obstacle-free environment is assumed.

Following this line, the authors in [8] describe a custom-built six-wheeled mobile robot with a Kinect2 mounted on-top to perform gait analysis on subjects with Parkinson disease while driving in front of them. However, their focus is mostly on general requirements of such a scenario with a strong emphasis on describing design details of the platform and basic preprocessing steps, such as transformation into a fixed reference frame. As a consequence, the system lacks the necessary autonomy required in a complex scenario like ours. For example, the platform has to be controlled remotely to drive in front of its user at the desired distance and does not seem to be equipped with sensors which would allow autonomous navigation and obstacle avoidance.

3 Robot Platform ROGER

Our target platform is a customized SCITOS G3 robot (see Fig. 2) that is based on the robot platform presented in [9] and [10]. The robot's compact footprint of approximately 45 cm × 55 cm at a total height of 1.5 m makes it well suited for operating in crowded public environments. It is equipped with a camera module with four separate RGB cameras for 360° all around view, as well as two Sick S300 laser scanners and two ASUS Xtion RGB-D cameras. In addition to the basic setup, a rear-facing Kinect2 RGB-D sensor is mounted on a Directed Perception pan-tilt unit (PTU).

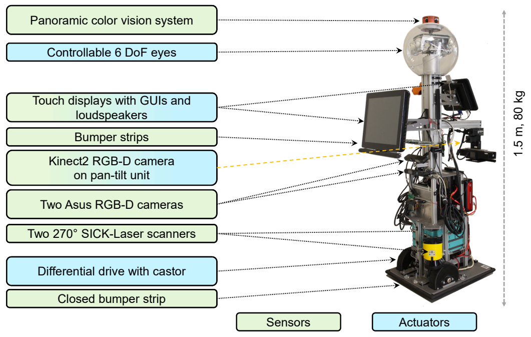

Panoramic color vision system
Controllable 6 DoF eyes
Touch displays with GUIs and loudspeakers
Bumper strips
Kinect2 RGB-D camera on pan-tilt unit
Two Asus RGB-D cameras
Two 270° SICK-Laser scanners
Differential drive with castor
Closed bumper strip
Sensors
Actuators
1.5 m, 80 kg

Figure 2 Sensors and actuators of our robot platform.

Laser and depth data are primarily used in 2D/3D mapping and localization [11] allowing navigation in unstructured real-world environments. Sensor information from color cameras and laser sensors are fed into independent person detection modules (e.g. [12, 13, 14]) whose results are combined in our person tracker [15] allowing us to detect and track people with and without walking aids up to a distance of 10 m with high confidence [13]. To be robust against tracking errors caused by occlusions, the platform uses an appearance-based person re-identification closely linked to the tracking module as described in [16].

The robot middleware MIRA [17] is used to control the platform hardware and to exchange data between sensors and all software modules.

4 System Design

Because our robots operate in a wide range of environments ranging from supermarkets, over public spaces in rehabilitation clinics to senior homes, we need a flexible local path planning algorithm able to navigate even in narrow or human populated environments. Therefore, we use a versatile motion planner based on evolutionary generated trajectories [18] (EMP) developed at our lab. The evolutionary optimization of hypothetical movement trajectories generated by EMP can be controlled through the given objective function. To ease the design of motion behaviors, the objective function is decomposed in multiple less complex objectives representing aspects of the desired behavior. This decomposition allows reoccurring aspects, e.g. obstacle avoidance, to be reused and enables a rapid deployment of our robots in new scenarios.

To accomplish the task of observing users while guiding them to a goal, the motion planning of robot and pan-tilt camera is decoupled (see Fig. 3):

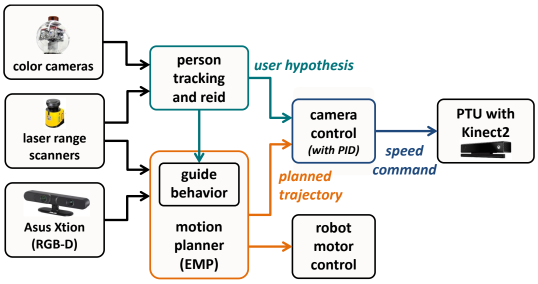


Figure 3 Guide behavior and camera control in system context. Raw sensor observations are processed in person tracking and navigation components, including the guide behavior. High-level data from both instances are then used in the camera control.

The EMP-based motion planning is used to control the robot's drive, implementing a guide behavior and keeping the user at an optimal distance to the camera (lower branch in Fig. 3). Details on the guide behavior implementation follow in section 4.1.

Hence, the camera control problem is reduced to keeping the user in camera view horizontally. Therefore, we designed a camera control algorithm based on a PID controller. We take the user hypothesis delivered by the person tracker and used by the drive behavior and compute the direct line of sight from the pan-tilt unit's mount frame to the user's position. Now the angular difference between current and desired pan angle is computed and used as tracking error in the PID implementation which is detailed further in Sec. 4.2.

4.1 Evolutionary Motion Planning to Keep Optimal Distance

When considering motion planning as an optimal control problem, EMP is best described as a model predictive control approach optimizing over the space of motion command sequences given the objective function and the physical constraints of the robot's drive. To keep EMP adaptable, aside from the robot's dynamics, no restrictions are imposed on these sequences nor the objectives' form. This leads to a high-dimensional optimization problem generally not solvable in closed form. Thus, we rely on evolutionary algorithms to find near optimal solutions. However, we experimentally showed that for our scenarios the quality of the found solutions results in good motions [18].

To implement the desired guide behavior, we build upon our base objective functions, which already realize a goal-oriented movement while avoiding obstacles and keeping a personal space to people, and extend them with a constraint for keeping the distance to a user. Overall, this results in a mutual movement of the user and robot to the goal, i.e. the robot is only moving when the user moves too. The base objectives comprise:

1. a *path objective* responsible for the movement to the goal by approaching the minimum in a globally planned navigation function (using E* planner [19]),
2. a *heading objective* turning the robot towards a given goal orientation when in proximity to the goal position,
3. a *direction objective* preferring forward motion accounting for the motor's limitation with a slower speed when driving backward,
4. a *distance objective* for avoiding collisions with static and dynamic obstacles,
5. a *personal space objective* to keep distance to people in the close proximity of the robot by predicting their movements with a linear model.

Each objective returns a cost value for a given motion command sequence, or can even deny a sequence preventing it from being executed by the robot's motor controller. For optimization, a global cost value is calculated by means of a weighted sum.

Each sequence can be expressed as a vector $(c^{(1)},\dots,c^{(T)})$ with T being the planning horizon and $c^{(t)}$ the command specific to the robot's drive. Further, a sequence can be transformed into the predicted robot's motion trajectory $\tau_R = (\mathbf{x}_R^{(1)},\dots,\mathbf{x}_R^{(T)})$ consisting of poses $\mathbf{x}_R^{(t)} = (x,y,\phi)$ in a planar world. By using a linear motion model, a similar trajectory $\tau_U = (\mathbf{x}_U^{(1)},\dots,\mathbf{x}_U^{(T)})$ can be predicted for the user. The keep distance constraint is realized as an additional objective function

$$f_{KD}(\tau_R,\tau_U) = \frac{1}{T}\sum_{i=1}^{T}\begin{cases}\max(|\delta^{(i)}-\hat{d}|-\hat{t},0) & ,|\alpha^{(i)}|<\frac{\beta}{2}\\ \frac{M}{\pi-\beta}|\alpha^{(i)}| & ,else\end{cases} \quad (1)$$

where $\delta^{(i)}$ is the Euclidean distance from the robot to the user, and $\alpha^{(i)}$ is the angle of the user in the polar coordinate system relative to the robot at time i. The parameters $\hat{d}$ and $\hat{t}$ control the distance kept to the user with $\hat{d}$ being the desired distance and $\hat{t}$ the acceptable error tolerance. Additionally, f_{KD} introduces a circular sector with opening angle β behind the robot where the user is preferably kept. This ensures that the user is kept in the controllable pan limits of the PTU. The parameter M controls the cost of user positions outside of the defined sensor cone. To prefer user positions inside the sensor cone, M should be set to a high value. For our scenario, we set $M = 10$. A visualization of f_{KD} and the used polar coordinate system is depicted in Fig. 4.

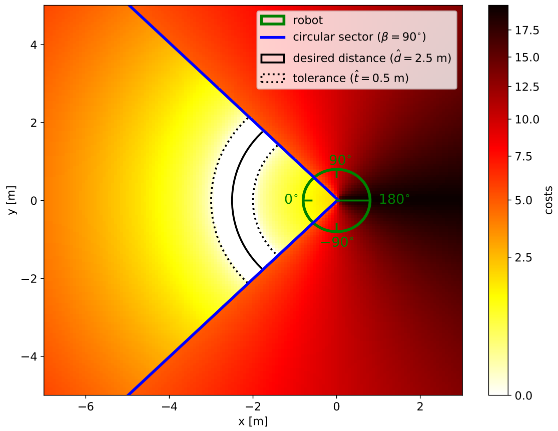


Figure 4 Visualization of the cost terms of f_{KD} by keeping the robot pose $\mathbf{x}_R$ constant and varying the user pose $\mathbf{x}_U$. The robot is centered at $(0,0)$ and 0° is describing the backward direction of the robot.

4.2 Camera Control

Maintaining a defined distance as aspect of keeping the user in good view of the sensor is already covered by the guide behavior of the robot. However, due to the considered application scenario and limitations of the non-holonomic platform, there are situations where the user cannot be kept directly behind the robot, e.g. because the robot has to drive around obstacles or keep the personal space of oncoming persons. Because of the required minimal distance between robot and user, the robot has to start its evasion maneuvers quite a bit earlier than the user. Even though this situation would still allow for an ongoing gait assessment, a fixed sensor might not cover the user sufficiently. Fortunately, a sensor mounted on a pan-tilt unit can compensate the relative orientation as long as it is within reasonable limits, i.e. the user is not occluded by the robot itself or an obstacle.

Since the user's position in respect to the robot in the x-y-plane changes dynamically based on the current scene configuration, the camera control needs to recompute the suitable pan angle as well as the speed at which these changes occur at runtime.

4.2.1 Angle Computation

Baseline approach The most straightforward approach to determine the pan angle is to use the direct „line of sight“ from the PTU's mount frame to the user's position as estimated by the person tracker. In our system, this allows updates at 10 Hz, which is the update frequency of the tracker.

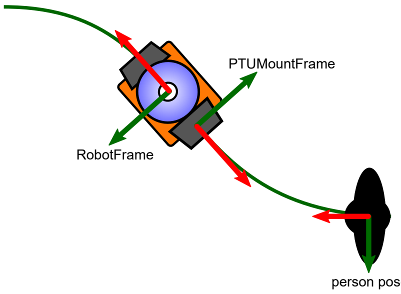


Figure 5 Schematic view of the robot platform and its reference frames. Red indicates the x-axis, green the y-axis, the z-axis is not shown for reasons of clarity.

New estimations of the user's position are transformed in the coordinate frame defined relative to the PTU's mount frame (*PTUMountFrame* in Fig. 5) using the capabilities of our middleware.
The pan angle can then be computed by simple geometric means:

$$\varphi = \operatorname{arctan2}(y_{user}, x_{user}) \tag{2}$$

with x_{user} and y_{user} describing the user's position relative to the PTU's mount frame. Due to the way the PTU has been placed on the robot, we additionally limit the pan angle to be between 90° and $-90°$.

Including prediction Though the baseline approach already provides compelling results in our application scenario, it has some limitations often encountered in simple control problems. Most apparent is the systematic delay in the tracking response caused by physical limitations, like inertia and actuator speed and power limits. To improve the tracking response, it is beneficial to include additional knowledge of the system's dynamics in the computation. As noted in Sec. 4.1 and further detailed in [18], our evolutionary motion planner features characteristics commonly associated with model predictive control. Using the planned trajectory as selected in the cost optimization process, we can obtain a reliable prediction of the robots movement for a limited planning horizon.
While looking at recordings from previous tests in similar applications [9], we found that translational movement only plays a minor role when it comes to changes of the relative pose between robot and user. Of course this only holds true if the time span under consideration is relatively short, like 250 ms in our example.
To include those trajectory data in the pan angle computation, the user position has to be transformed to robot coordinates first. Then, the rotational movement associated with the trajectory is applied. Since the transformation between the PTU mount point and the robots reference frame is fixed, it can easily be applied before the hypothesis is passed to the baseline algorithm, computing the pan angle as described before in Eq. (2).

4.2.2 Angle Trajectory Computation

After the pan angle φ is determined according to the previous section, the camera control has to define the speed trajectory used to reach the desired angle. The approach presented in this paper utilizes a PID controller for this purpose. Therefore, the current pan orientation is retrieved from the hardware controller and used to compute the control deviation φ_{error}.

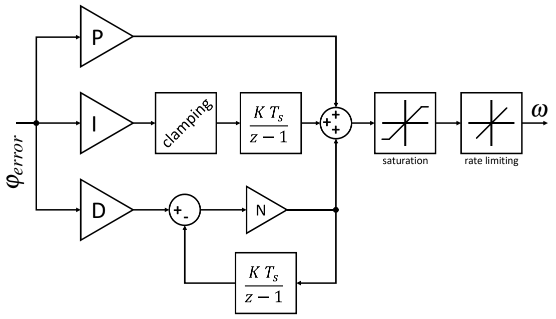


Figure 6 The PID structure used in our implementation. The clamping component [20] to counter integration windup is only shown as black box.

$$\varphi_{error} = \varphi - \varphi_{current} \tag{3}$$

Since only the range ±90° is considered, problems due to angle periodicity do not occur.
The control deviation is passed to a standard parallel PID structure (for implementation details see Fig. 6), which outputs a speed value ω that is then passed to the PTU interface. As the hardware can only realize a limited range of speeds and accelerations, additional output saturation and rate limiting are employed. Because this can lead to a phenomenon known as *integration windup* [21, 22] when the actuators are in saturation, an additional anti-windup component is included. In our implementation, we decided in favor of *integrator clamping* [20] which is reported to have good performance in a wide range of applications while being relatively easy to implement.

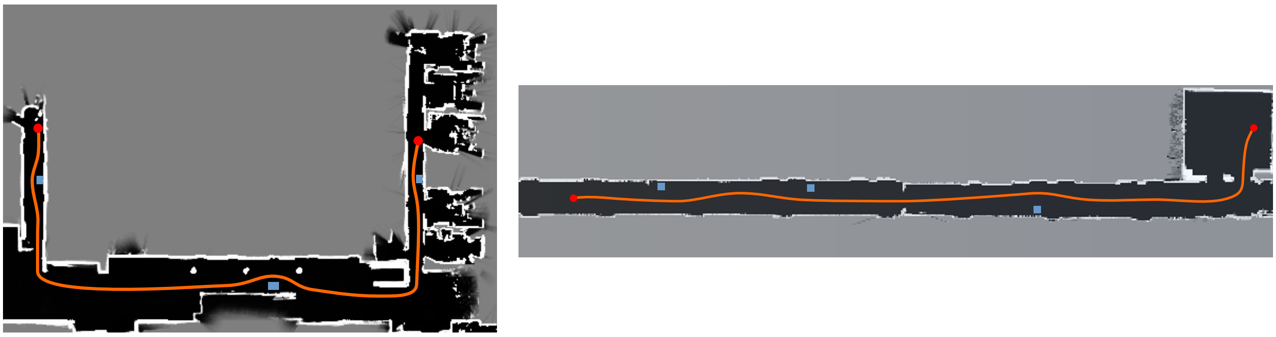
Controller parameters To determine the controller parameters P, I, D and N, we used the *PID Tuner* in MATLAB Simulink. The Simulink PID block has the same structure as shown in Fig. 6 without the rate limiting block. So the rate limiting block was supplemented to the model as well as an integrator block to have a rudimentary simulation of the PTU hardware. Using this model, selecting 250 ms response time, and a robustness factor of 0.65, the PID Tuner tool determined the following set of parameters:

$$P = 8.1778, I = 0.6035, D = -0.1284, N = 9.4601$$

However, some preliminary testing showed that our pan-tilt unit becomes relatively noisy using this parameter set and sometimes looses track of its pan orientation. After iterative testing (stability validation in Simulink and testing with the real hardware), the final configuration used in the experiments was selected as:

$$P = 6.5131, I = 0.0000, D = 0.4480, N = 12.0345$$

As can be seen, the integral gain was eliminated completely, but since the PTU hardware itself has an integrating character, permanent control deviation is not expected to occur.



(a) Test environment in our faculty building

(b) Test environment in the orthopedic hospital

Figure 7 Excerpts from the grid maps of both test environments. Waypoints for robot and user are marked in red, a possible path in orange, and static obstacles, e.g. chairs for patients to rest on, in light blue.

5 Experiments

This section describes how both parts of the proposed system were evaluated. Similar to the previous section, we address them separately: first the guide behavior in section 5.1, followed by the camera control in section 5.2.
First performed extensive tests in our lab's building (Fig. 7a) which shows great similarity to the later application area in rehabilitation facilities. For the evaluation, we recorded several trials for each of the eight test persons while the robot guided them between two fixed navigation points on a U-shaped track, leading to a total of 30 recorded tracks. Some artificial obstacles were added to mimic typical situations encountered during gait training in a public hospital hallway. In addition to these tests, we also performed a smaller study with six patients in a orthopedic clinic (Fig. 7b) under real world conditions. Each participant chose the duration of the test trial as it was appropriate for its state of health. This resulted in trials with a duration of about 10 minutes each (including short pauses if needed) where the patients walked 200 m on average while being guided by the robot.

5.1 Guide Behavior

The desired distance between robot and user for the guiding behavior was set to 2.5 m with a tolerance of 0.5 m, providing a good trade-off between sensor coverage and depth measurement quality.

5.1.1 Faculty building

As Fig. 8 shows, the empirical distribution of the distance between robot and users over all trials shows a clear peak at the desired distance of 2.5 m. In summary, the guide approach is able to keep the desired distance to the user most of the time (82.6%) while driving in our faculty building.
A detailed excerpt of the time profile of the distance and speed is shown in Fig. 9 for a single test trial. As can be seen, the driving speed is adapted when the user slows down (phases II) and increases slightly while the user accelerates again (phase III).

5.1.2 Clinic environment

Applying the same evaluation procedure on data recorded during tests in the clinic environment delivers results as presented in Fig. 10. While still keeping the user within the selected tolerance most of the time (81.0%), there is a slight tendency to drive at the maximum distance allowed (3 m), whereas the peak at the desired distance is much less pronounced. This might be caused by the fact that patients often tend to walk a lot slower than healthy persons.

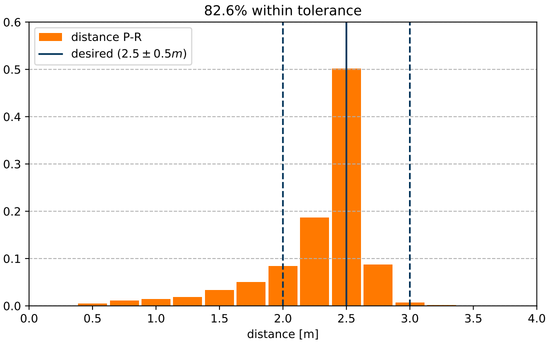


Figure 8 Distance between user and robot over all trials in our faculty building.

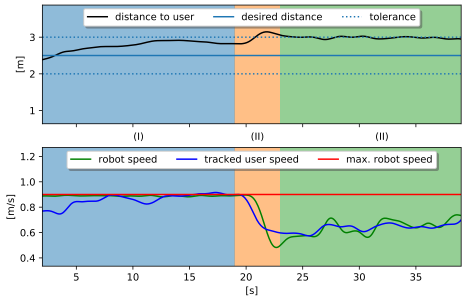


Figure 9 Distance and speed of user and robot over time (example trial). To keep the desired distance, the robot adapts its speed to the user's speed. (I) The robot and user move at an equal pace. (II) The user slows down. A short reaction time is needed where the distance slightly deviates from the tolerance. (III) The robot has adapted to the new speed.

5.2 Camera Control

For evaluating the camera control in the horizontal plane, the field of view of the Kinect2 depth camera is assumed to be narrower than specified (50° instead of 70°). This def-

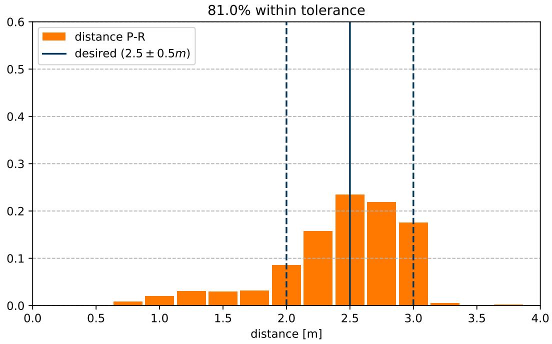


Figure 10 Distance between user and robot over all trials. In general, the distance is within the desired tolerance, though there is a tendency to max out the selected tolerance.

inition mirrors our practical observation that depth values near the image border are distorted by increasing noise and are thus less suited to be used for training evaluation purposes. Based on these definitions, the user is assumed to be visible by the sensor when being within a ±25° cone with its center line following the x-axis of the oriented PTU.

5.2.1 Faculty building

As can be seen in Fig. 11, the camera control algorithm (*Keep In View*, *KIV* for short) is able to increase the probability that the user is within the sensor field of view significantly from around 82% to over 99%.

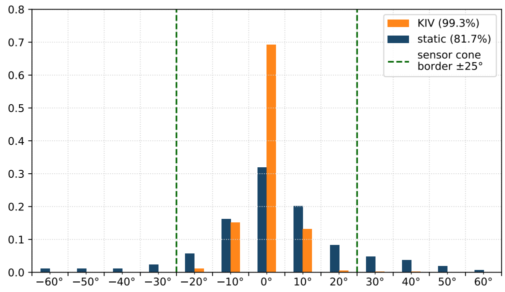


Figure 11 Angular position of the user in the horizontal sensor field of view, with (*KIV*) and without (*static*) our camera control algorithm in our lab's building. The number in parentheses describes the proportion of positions within the ±25° boundaries.

We also evaluated the variance of the values and were able to show that our camera control approach indeed leads to variances that are significantly smaller compared to the static setting. Significance was shown by Levene's test for equal variances [23], using $p < 0.01$.

5.2.2 Clinic environment

As can be seen from Fig. 7b the clinic environment is less challenging in terms of navigation. This claim is supported by the fact that even with no active camera control the user is kept at a suitable position relative to the sensor significantly more often (89.5% instead of 81.7%). Nevertheless, the proposed control algorithm is able to keep up the good performance achieved in our faculty building in the clinic environment as well (see Fig. 12).

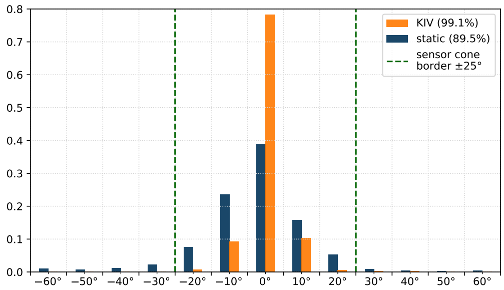


Figure 12 Angular position of the user in the horizontal sensor field of view, with (*KIV*) and without (*static*) our camera control algorithm in the clinic environment. The number in parentheses describes the proportion of positions within the ±25° boundaries.

Similar to the experiments in our faculty building, the variance of the angular position of the user relative to the robot was evaluated. As expected, the test under the same conditions as in section 5.2.1 found the difference in variance to be significant even in the less challenging environment.

6 Conclusion

This contribution presented an efficient approach to control the robot's driving behavior as well as an on-board pan-tilt mounted camera in order to achieve a reliable and continuous user observation in a gait retraining use case in orthopedic rehabilitation.

The experiments have shown that our active camera control approach together with the designed guide behavior is able to satisfy the requirements expected to occur during typical training sessions in our application scenario.

With this system in place, future works are supposed to focus more on autonomous gait and posture analysis tasks based on the sensor data provided by the Kinect2 RGB-D camera, which is the actual core of the robot-based gait training in rehabilitation.

Literature

[1] H. Sidenbladh, D. Kragic, and H. I. Christensen, "A person following behaviour for a mobile robot," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 1, 1999, pp. 670–675.

[2] S. Feyrer and A. Zell, "Detection, tracking, and pursuit of humans with an autonomous mobile robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 2, 1999, pp. 864–869.

[3] D. Burschka and G. Hager, "Vision-based control of mobile robots," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 2, 2001, pp. 1707–1713.

[4] W. Bahn, J. Park, C.-h. Lee, T.-i. Kim, T. Lee, M. M. Shaikh, K.-s. Kim, and D.-i. Cho, "A Motion-Information-Based Vision-Tracking System with a Stereo Camera on Mobile Robots," in *International Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, sep 2011, pp. 252–257.

[5] N. V. Tinh, P. T. Cat, P. M. Tuan, and T. T. Q. Bui, "Visual Control of Integrated Mobile Robot-Pan Tilt-Camera System for Tracking a Moving Target," in *International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, dec 2014, pp. 1566–1571.

[6] A. Staranowicz, G. R. Brown, and G.-L. Mariottini, "Evaluating the accuracy of a mobile kinect-based gait-monitoring system for fall prediction," in *Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments*, ser. PETRA '13. New York, NY, USA: ACM, 2013, pp. 57:1–57:4.

[7] F. Morbidi, C. Ray, and G. L. Mariottini, "Cooperative active target tracking for heterogeneous robots with application to gait monitoring," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Sept 2011, pp. 3608–3613.

[8] O. Ťupa, J. Crha, A. Procházka, and J. Mares, "Gait analysis using ms kinect placed on the mobile robotic platform," in *International Conference on Technical Computing*, 2015, Conference Paper.

[9] H.-M. Groß, S. Meyer, R. Stricker, A. Scheidig, M. Eisenbach, St. Mueller, Th. Q. Trinh, T. Wengefeld, A. Bley, Ch. Martin, and Ch. Fricke, "Mobile robot companion for walking training of stroke patients in clinical post-stroke rehabilitation," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 1028–1035.

[10] H.-M. Groß, A. Scheidig, K. Debes, E. Einhorn, M. Eisenbach, St. Mueller, T. Schmiedel, Th. Q. Trinh, Ch. Weinrich, T. Wengefeld, A. Bley, and Ch. Martin, "ROREAS - robot coach for walking and orientation training in clinical post-stroke rehabilitation: Prototype implementation and evaluation in fields trials," *Autonomous Robots (AR)*, vol. 41, no. 3, pp. 679–698, 2017.

[11] E. Einhorn and H.-M. Groß, "Generic 2D/3D slam with NDT maps for lifelong application," in *European Conference on Mobile Robots (ECMR)*, 2013, pp. 240–247.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 886–893.

[13] Ch. Weinrich, T. Wengefeld, Ch. Schröter, and H.-M. Groß, "People detection and distinction of their walking aids in 2D laser range data based on generic distance-invariant features," in *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*, 2014, pp. 767–773.

[14] M. Eisenbach, D. Seichter, T. Wengefeld, and H.-M. Gross, "Cooperative multi-scale convolutional neural networks for person detection," in *World Congress on Computational Intelligence (WCCI)*. IEEE, 2016, pp. 267–276.

[15] M. Volkhardt, Ch. Weinrich, and H.-M. Groß, "Multi-modal people tracking on a mobile companion robot," in *European Conference on Mobile Robots (ECMR)*, 2013, pp. 288–293.

[16] T. Wengefeld, M. Eisenbach, Th. Q. Trinh, and H.-M. Groß, "May I be your personal coach? Bringing together person tracking and visual re-identification on a mobile robot," in *Int. Symposium on Robotics (ISR)*. VDE, 2016, pp. 141–148.

[17] E. Einhorn, T. Langner, R. Stricker, Ch. Martin, and H.-M. Groß, "MIRA - middleware for robotic applications," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 2591–2598.

[18] St. Müller, Th. Q. Trinh, and H.-M. Groß, "Local real-time motion planning using evolutionary optimization," in *Towards Autonomous Robotic Systems (TAROS)*, ser. LNCS, vol. 10454. Springer, 2017, pp. 211–221.

[19] R. Philippsen and R. Siegwart, "An interpolated dynamic navigation function," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005, pp. 3782–3789.

[20] A. Visioli, "Modified anti-windup scheme for PID controllers," *IEE Proceedings - Control Theory and Applications*, vol. 150, no. 1, pp. 49–54, Jan 2003.

[21] K. J. Åström and T. Hägglund, "Integrator Windup and How to Avoid It," in *American Control Conference*, 1989, pp. 1693–1698.

[22] K. J. Åström and L. Rundqwist, *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society, 2006.

[23] H. Levene, "Robust tests for equality of variances," *Contributions to Probability and Statistics*, vol. 1, pp. 278–292, 1960.