

Combining Semi-autonomous Navigation with Manned Behaviour in a Cooperative Driving System for Mobile Robotic Telepresence

Andrey Kiselev^(✉), Annica Kristoffersson, and Amy Loutfi

Örebro University, örebro, Sweden

{andrey.kiselev,annica.kristoffersson,amy.loutfi}@oru.se

Abstract. This paper presents an image-based cooperative driving system for telepresence robot, which allows safe operation in indoor environments and is meant to minimize the burden on novice users operating the robot. The paper focuses on one emerging telepresence robot, namely, mobile remote presence systems for social interaction. Such systems brings new opportunities for applications in healthcare and elderly care by allowing caregivers to communicate with patients and elderly from remote locations. However, using such systems can be a difficult task particularly for caregivers without proper training. The paper presents a first implementation of a vision-based cooperative driving enhancement to a telepresence robot. A preliminary evaluation in the laboratory environment is presented.

Keywords: Human-robot interaction · Mobile robotic telepresence · Teleoperation · User interfaces

1 Introduction

Mobile Remote Presence (MRP) [1] is an emerging field of assistive robotics which aims to bring mobile robots equipped with certain telepresence and teleoperation capabilities into domestic environments. The main goal of those systems is to allow rich natural human-human interactions from distant locations. Many systems are now available for research and some are already commercial products. The application domains of such systems may vary and define the appearance and functionality of a particular robot to certain degree. [2] provides an overview of modern MRP systems for different applications.

One of the important application domains for MRP systems is healthcare and elderly care, where telepresence robots can be profitable [3] and contribute to prevention of problems related to loneliness and social isolation [4]. At the same time, with the spread of the technology into consumer fields, scientists and developers are facing problems related to the nature of end-users who may have no proper skills, knowledge and experience on using such systems and who cannot be exposed to extensive training which is normally acceptable in other application domain for telepresence and teleoperation. Particularly, previous research

has shown that some novice users of MRP system can experience high workload even in a relatively simple driving task [5]. This actually might leave no room for actual interaction between local and remote people diminishing the utility of such solutions [6].

The problem of high mental workload during driving/piloting tasks has been previously studied in automotive industry [7], aviation [8] and related fields. However, in a vast majority of cases, it is possible or assumed to be possible to overcome the problem with training and experience [9], [10]. Unfortunately, it is not always possible to use extensive training for those who operate telepresence robots in domestic environments as the type of end-users are diverse and the care structures may not be suitable to enable this type of training. Another possible solution to coping with inexperienced users is to provide certain semi-autonomous functions that can predict and avoid possible hazardous situations. Such systems are growing rapidly for example in automotive industry [11] [12].

One important function is to avoid collisions and utilizing various sensors and processing techniques to detect possible collisions. For example, [13] presents a system for real-time obstacle and lane detection in a structured environment based on stereoscopic cameras, [14] proposes the fusion algorithm for combining stereo vision and laser scanner data, [15] introduces the system for safe car navigation using multiple sensors and software algorithms, [16] describes the system for collision-free navigation for humanoid robot using laser range finder and camera, [17] investigates the effectiveness and drivers' acceptance of collision avoidance system. For MRP systems, a good balance between autonomous functions and manual driving is needed. Pilots must be able to not only use video-conferencing, but also to express behaviours through the mobile robot. For example, during the conversation the remote user might want to nod using the robot camera, or screen tilt function or to show impatience by slightly rotating left-right. This basic problem can be found in many different domains. Particularly, [18] and [19] discuss the problem of balanced control for urban search and rescue (USAR) robots, a constrained-based method for vehicle semi-autonomous operation is presented in [20], shared control for robots is discussed in [21]. Different operation modes for cooperative control are shown in [22].

In this paper we present an image-based system for drivable area discovery and collision avoidance for telepresence robot which implements the cooperative control paradigm in a parallel mode in combined case [22]. The system only uses the video stream from the robot to find drivable areas based on floor samples. The current paper is based on the previous research reported in [23].

The rest of the paper is organized as follows: Section 2 depicts our cooperative driving system along with the drivable area discovery algorithm. First tests are described in Section 3. The paper is concluded by Section 4.

2 Semi-autonomous Cooperative Driving

When mobile robots are used in indoor environments and especially in homes of older adults, there are certain considerations that must be taken into account.

First of all, the safety aspect of a fully autonomous domestic robot might be questionable [24]. And this need special attention concerning that MRP systems naturally have human in the loop in most of cases, which makes the entire system even more complicated.

Another important aspect of a telepresence robot is its ability to keep the balance between autonomy or semi-autonomy with a possibility for human users to express behaviours through the robot as briefly mentioned in Section 1. It is important to understand that a MRP system is not only a video-conferencing system on wheels. Its ability to mediate user's statements through body language is also important. In this light, our proposed system attempts to combine robot's semi-autonomy with a possibility for manned behaviours. We use parallel shared cooperation control to combine manned driving performed by an operator with an autonomous collision avoidance algorithm.

2.1 System Architecture

The system which we use for this study is the Giraff MRP system [25]. It is a split system in which there are two essential components: the Giraff robot and pilot interface called Giraff Pilot. Since both parts of the system are actually PCs running Windows®OSs, and several levels of API are currently available for developers, there are several possibilities to inject drivable area discovery algorithm into the basic system. Currently, all semi-autonomous driving functions are implemented on the side of pilot. There are advantages and drawbacks of such solution in comparison to the robot-side implementation of collision avoidance. There are two factors which influence the decision on pilot side implementation:

- When all processing is done on the pilot side, there is no dependency on the robot itself. Thus, the tests can be carried out with different robots and in case of lacking computational resources, a pilot side computer's performance can be increased in an easy way.
- The problem of synchronization under conditions of variable control latency is also eliminated. The control latency of the system depends on many factors and cannot be easily controlled except the case when there is a direct network connection between the robot and the pilot. Performing all processing on the same side eliminates the need of synchronization.

The downsides of this decisions are the reduced quality of the video stream (frames may contain compression artefacts) and a varying framerate.

2.2 Image-Based Drivable Area Discovery

The current implementation of the drivable area discovery algorithm is based on a combination of image thresholding in HSV-space [26] and flood filling. Thus, the method only uses the robot's camera for drivable area discovery. In this paper, the algorithm is called "image-based" because the whole processing is done from scratch on each frame and there is not between-frame persistence of the discovered drivable area.

While the work presented here could utilize similar or better such as depth sensors, various laser range finders or a combination of techniques, our motivation to only use images provided by the onboard camera is driven by the intent to use a standard platform with no hardware modification, to reduce overall system cost and to ease portability to other robotic platforms.

Method. The frames received from the robot have the resolution of 640×480 *px* at the rate of $15\text{-}25$ *FPS*. Therefore, our goal was to develop an algorithm simple enough to be able to process frames in real time using a standard office PC for pilot. The full processing chain for the algorithm is shown in Fig. 1.

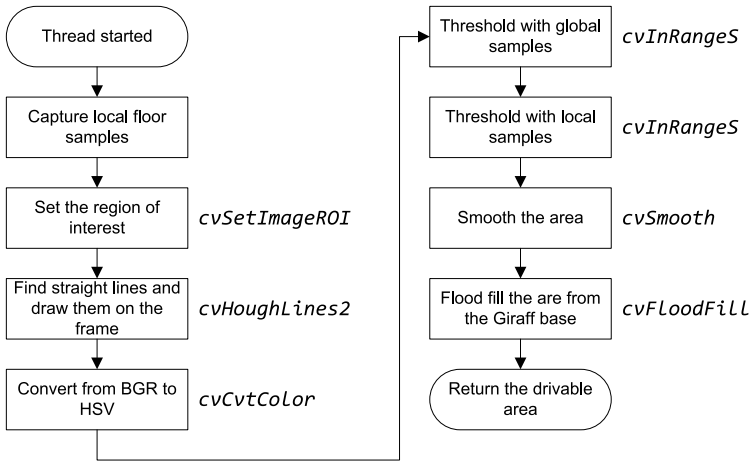


Fig. 1. Algorithm of the frame processing. Note that global floor samples are being loaded outside the frame processing.

The algorithm uses samples of known surfaces called “*floor samples*” to discover the drivable area from frames. There are two kinds of floor samples used by algorithms. First, global floor samples are those which are previously known to belong to floor surfaces allowed to drive. Those floor samples come from initial teaching and experience. Additionally, local floor samples are taken from each frame to allow traversal of unknown surfaces further algorithm learning. Pilot users can choose to store recently taken local floor patterns into the permanent database. Local floor samples are taken in front and on both sides of the robot’s base. The actual capturing area is shown on Fig. 2. In the preparation phase, additionally to selecting the region of interest, the Canny edge detection and Hough transform [27] is applied to find any long straight lines and draw them back to the frame. This is done to emphasize the margins between segments of the frame (e.g. floor and wall) which may have not significantly different color to be found by thresholding, but the margin can be still extracted by edge detection and Hough transform.

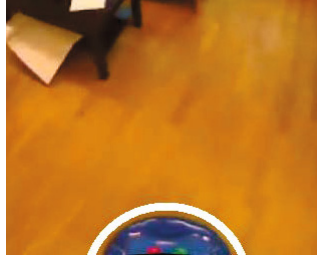


Fig. 2. Sampling area (white) in front of the robot's base used to take local floor samples

After that, the frame is converted into HSV color space and thresholding is performed using the global floor samples loaded from the library and local floor samples taken in front of the robot.

After the main plugin loop received any discovered drivable area, it attempts to avoid possible collisions or to minimize impact. The algorithm for collision avoidance is shown of Fig. 3.

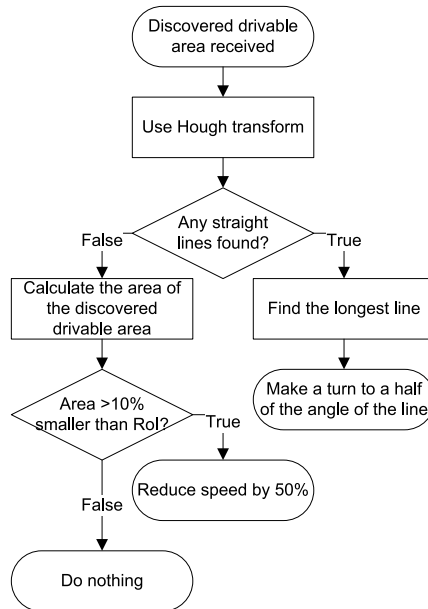


Fig. 3. Algorithm for collision avoidance

The proposed system implements the cooperative control paradigm in a parallel mode in combined case. Therefore, the user input is modified by the system

to avoid obstacles if there are any in the drivable area. As it can be seen from Fig. 3, the system attempts to find two types of obstacle. First, if the drivable area is fully cut by some straight line (like from the wall or furnitures), the collision avoidance algorithm attempts to make a turn to avoid collision. Then, if the drivable area contains some obstacle, but no straight line is found, the system reduces the current speed.

Implementation. All image processing is done on the pilot side. Giraff Pilot plugin API [25] was used to embed the code into the basic Giraff Pilot software. Development is accomplished in Java, using JavaCV¹ [29] computer vision library.

Giraff Pilot plugin API offer two interfaces: 1. GiraffViewPlugin interface, which allows to access the video stream received from the robot and manipulate the video stream on the user interface; 2. GiraffDrivePlugin interface, which extends the previous one with hardware access and driving capabilities. The latter is used in this study.

Essentially, the Cooperative Driving plugin replaces the original driving UI in that sense that it implements all basic robot driving functions. but also incorporates cooperative driving and collision avoidance parts.

When the pilot application starts it also loads and instantiates all available plugins. The cooperative driving plugin is designed in a way to also load all dependencies during this stage to avoid any delays if the plugin is actually selected and started by users. Then, during the call when the plugin is actually started by user it identifies available CPU cores and memory and starts processing each new frame in a new thread. The framerate of the video stream received from the robot depends on many factors, such as selected resolutions, network conditions, available PC resources, etc. The plugin attempts to use maximum available CPU cores and starts processing each frame, but may skip some frames if the received framerate is too high in comparison to computational resources available.

3 Preliminary Evaluation

Two variations of the visualization method were developed and incorporated into the Giraff pilot software. They are shown on Fig. 4 and Fig. 5. On the user interface, the main video area shows the video stream received from the robot along with some overlays containing important information (resolution and framerate, rotation and back up buttons, battery status), a driving line (green on both screens) and a drivable area discovery visualization. The driving is done with a mouse. Our previous research [30] shows that using projected image which corresponds to the real robot’s size in user interface can assist novice users to drive in narrow passages. This was used for the second visualization method shown in Fig. 5. At the same time, the interface with round shape visualization

¹ JavaCV is a wrapper library for OpenCV [28] library.

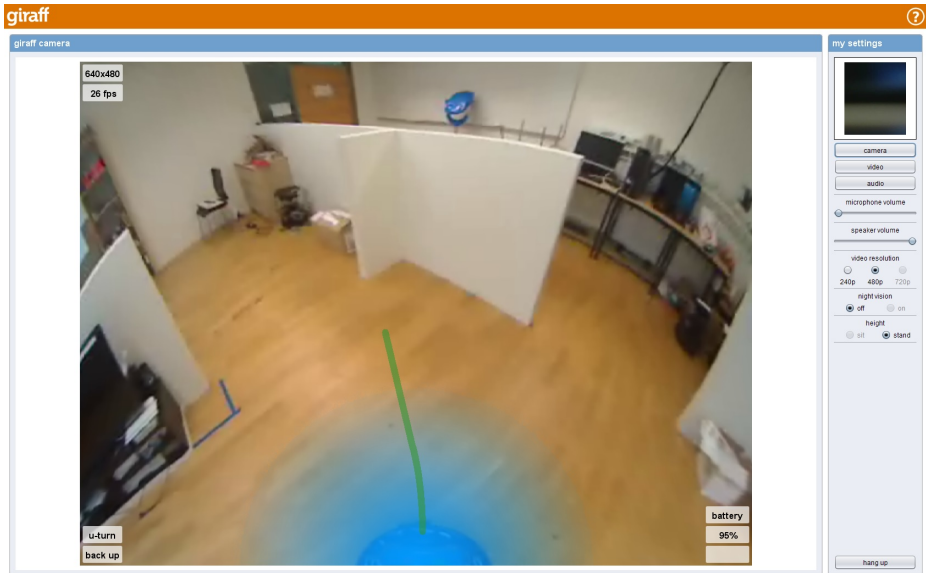


Fig. 4. Pilot interface with round shape drivable area visualization

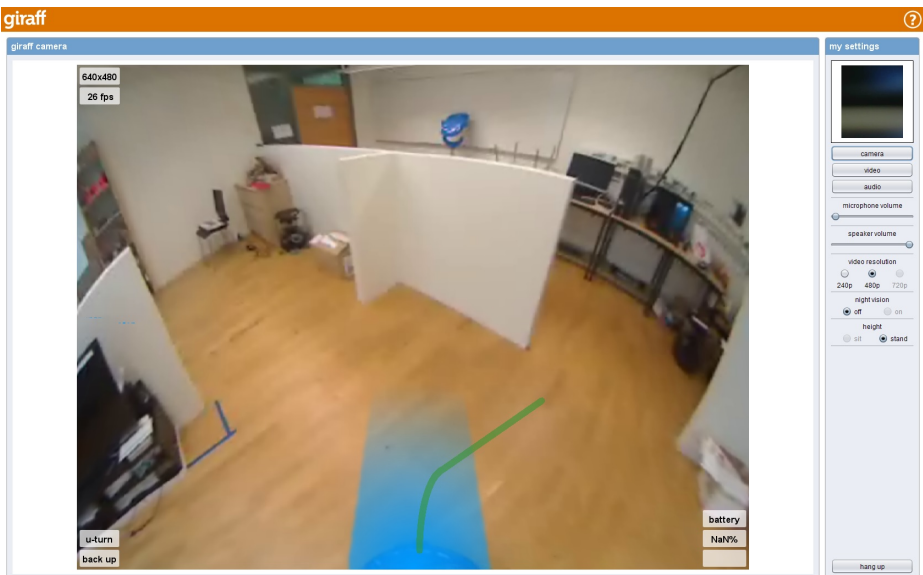


Fig. 5. Pilot interface with a corridor-like drivable area visualization

helps users to identify obstacles in the sides of the robot. Nevertheless, the drivable area discovery and obstacle avoidance algorithms behind both visualization methods are identical and use the same region of interest to cope with obstacles in front and on the sides of the robot.

Some examples of drivable area discovery is shown in Fig. 6. In the case (b) the collision avoidance algorithm extracts the straight line in the drivable area (shadow of the bed) and attempts to turn left to avoid front collision. Of course, this implementation of collision avoidance inevitably falls into a local minima, for example, in the corner of the room, which was observed during the trials. On other cases no long enough straight lines is found and the system reduces the forward speed. Fig. 6d shows the work of the drivable area discovery in a fast motion when the robot is being rotated. It can be clearly seen that there is a gap between the margin of the extracted drivable area and the actual obstacle² caused by the frame processing lag. In this test the system was trained and used both local floor samples along with the previously captured and stored global samples. The video demonstration of this test is available also in [23]. In another

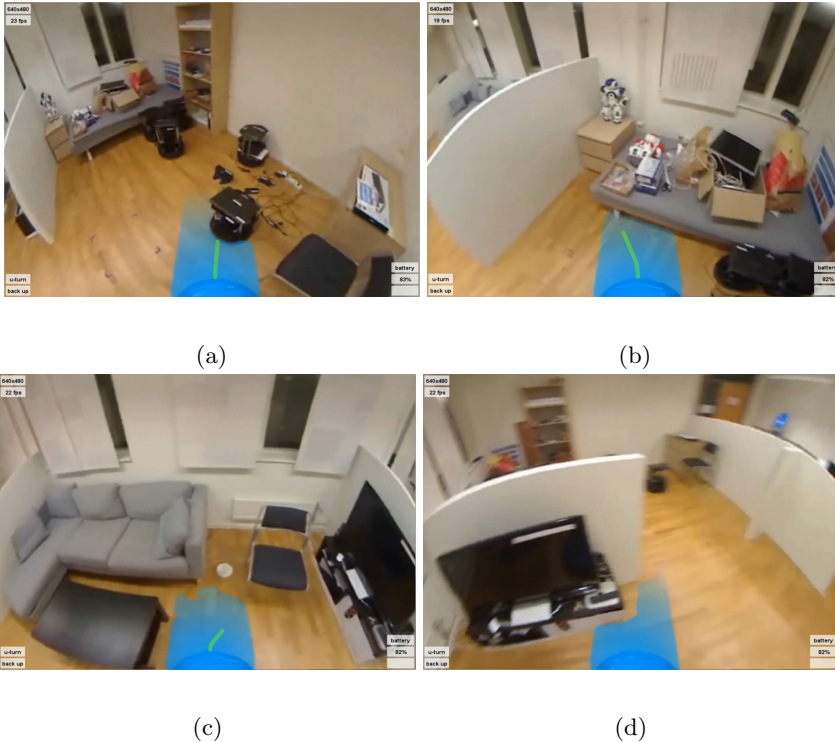


Fig. 6. Examples of drivable area discovery in a lab environment with wooden floor pattern

² The robot is rotated clockwise at a rotation speed of 60 degrees/s

test, the robot was put into an environment with a coloured and reflective floor without any global samples to work with (see Fig. 7). Here, several issues with the drivable area discovery system were discovered. First, the system is not capable to cope with light *blobs* (see Fig. 7b and Fig. 7c). This is due to a restriction in the algorithm which does not permit to classify overexposed segments as drivable. This limitation does not also allow the robot to traverse the overexposed area in Fig. 7d. One possible way to address the issue with light blobs in the future is to track them in motion according to the robot's actual speed.

Another issue seen in Fig. 7c is the margin between two different areas. Although both areas are classified as drivable according to the local floor samples captured in both areas, the margin between them is still considered as an obstacle due to the straight line detection stage. In both tests, the pilot software was run

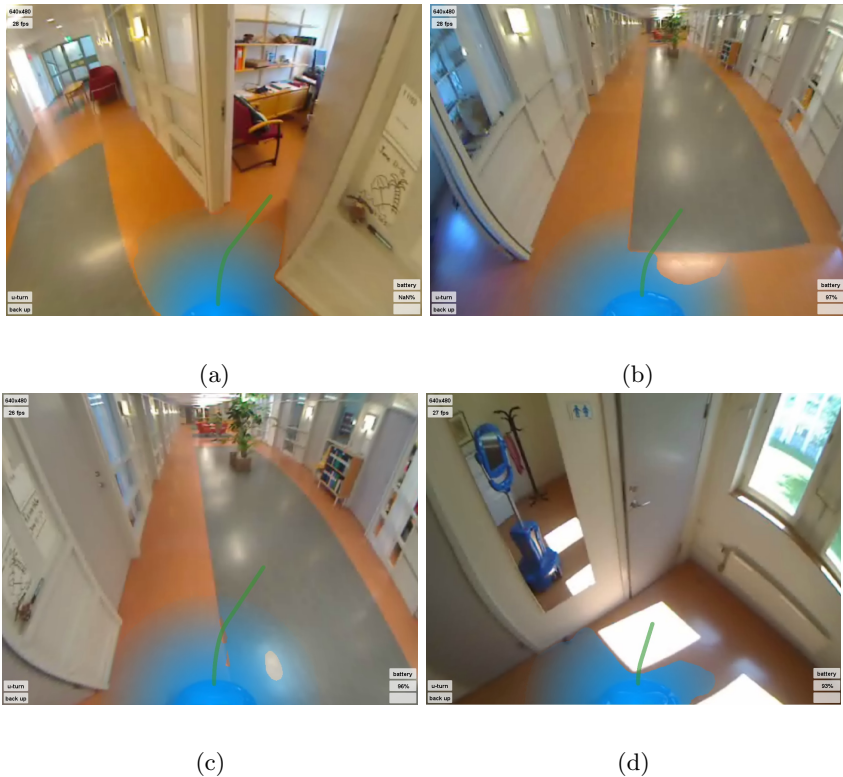


Fig. 7. Examples of drivable area discovery in a lab environment with coloured floor

on an average laptop PC³ using Oracle®Java™SE Runtime Environment 1.7.0. In this configuration, the average measured performance was 18.7 FPS (average frame processing time is $\mu = 53.4ms$, $\sigma = 10.0ms$).

³ Intel®CORE™i5-2520M@2,50GHz, Intel®HD Graphics, 4.0GB RAM, Windows®7 x64.

4 Discussion and Future Work

This paper outlines an attempt to incorporate active collision avoidance into a teleoperated mobile robot. This issue is currently well known and recent research shows rapid development of new technologies and their application in automotive industry. However, it appears quite a new and different problem for teleoperation and mobile robotic telepresence in particular when users have different level of immersion and depth perception.

There are two main directions of future work. First of all, the system reported in this paper was primarily developed for conducting studies in the HRI field. Specifically, the possible difference in technology perception and pilots' self confidence depending on the system autonomous behaviour is of a particular interest. We are also interested in evaluating semi-autonomous systems in conversational situations.

At the same time, reliability of the system must be improved in order to allow more complicated experimental scenarios and thorough evaluations. The currently used drivable area discovery algorithm does not guarantee stable work on surfaces with complex patterns. To overcome this limitation, another or additional methods must be implemented. Such methods can be based for example Local Binary Patterns [31] or Gabor filters [32].

References

1. Beer, J.M., Takayama, L.: Mobile remote presence systems for older adults: acceptance, benefits, and concerns. In: Proceedings of the 6th International Conference on Human-Robot Interaction HRI 11. HRI 2011, pp. 19–26. ACM (2011)
2. Kristoffersson, A., Coradeschi, S., Loutfi, A.: A Review of Mobile Robotic Telepresence. *Advances in Human-Computer Interaction* **2013**, 1–17 (2013)
3. Gandsas, A., Parekh, M., Blech, M.M., Tong, D.A.: Robotic telepresence: profit analysis in reducing length of stay after laparoscopic gastric bypass. *Journal of the American College of Surgeons* **205**(1), 72–77 (2007)
4. Coradeschi, S., Kristoffersson, A., Loutfi, A., Von Rump, S., Cesta, A., Cortellessa, G., Gonzalez, J.: Towards a methodology for longitudinal evaluation of social robotic telepresence for elderly. In: Proceedings of the HRI 2011 Workshop on Social Robotic Telepresence, pp. 1–7 (2011)
5. Kiselev, A., Loutfi, A.: Using a Mental Workload Index as a Measure of Usability of a User Interface for Social Robotic Telepresence. *Workshop in Social Robotics Telepresence* (2012)
6. Lee, M.K., Takayama, L.: Now, i have a body. In: Proceedings of the 2011 annual conference on Human factors in computing systems - CHI 2011, p. 33. ACM Press, New York (2011)
7. Engström, J., Johansson, E., Östlund, J.: Effects of visual and cognitive load in real and simulated motorway driving. *Transportation Research Part F: Traffic Psychology and Behaviour* **8**(2), 97–120 (2005)
8. Hankins, T.C., Wilson, G.F.: A comparison of heart rate, eye activity, EEG and subjective measures of pilot mental workload during flight. *Aviation space and environmental medicine* **69**(4), 360–367 (1998)

9. Parasuraman, R., Riley, V.: Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society* **39**(2), 230–253 (1997)
10. Patten, C.J.D., Kircher, A., Ostlund, J., Nilsson, L., Svenson, O.: Driver experience and cognitive workload in different traffic environments. *Accident Analysis & Prevention* **38**(5), 887–894 (2006)
11. Vlassenroot, S., Broekx, S., Mol, J.D., Panis, L.I., Brijs, T., Wets, G.: Driving with intelligent speed adaptation: Final results of the Belgian ISA-trial. *Transportation Research Part A: Policy and Practice* **41**(3), 267–279 (2007)
12. Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M.N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T.M., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.W., Singh, S., Snider, J., Stentz, A., Whittaker, W.R., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., Ferguson, D.: Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics* **25**(8), 425–466 (2008)
13. Bertozzi, M., Broggi, A.: GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* **7**(1), 62–81 (1998)
14. Labayrade, R., Royere, C., Gruyer, D., Aubert, D.: Cooperative Fusion for Multi-Obstacles Detection With Use of Stereovision and Laser Scanner. *Autonomous Robots* **19**(2), 117–140 (2005)
15. Aufrère, R., Gowdy, J., Mertz, C., Thorpe, C., Wang, C.C., Yata, T.: Perception for collision avoidance and autonomous driving. *Mechatronics* **13**(10), 1149–1161 (2003)
16. Maier, D., Stachniss, C., Bennewitz, M.: Vision-Based Humanoid Navigation Using Self-Supervised Obstacle Detection. *International Journal of Humanoid Robotics* **10**(02), 1350016 (2013)
17. Itoh, M., Horikome, T., Inagaki, T.: Effectiveness and driver acceptance of a semi-autonomous forward obstacle collision avoidance system. *Proceedings of the Human Factors and Ergonomics Society* **3**, 2091–2095 (2010)
18. Wegner, R., Anderson, J.: Balancing robotic teleoperation and autonomy for urban search and rescue environments. In: Tawfik, A.Y., Goodwin, S.D. (eds.) *Canadian AI 2004. LNCS (LNAI)*, vol. 3060, pp. 16–30. Springer, Heidelberg (2004)
19. Doroodgar, B., Ficocelli, M., Mobedi, B., Nejat, G.: The search for survivors: Cooperative human-robot interaction in search and rescue environments using semi-autonomous robots. In: *2010 IEEE International Conference on Robotics and Automation*, pp. 2858–2863. IEEE (May 2010)
20. Anderson, S.J., Karumanchi, S.B., Iagnemma, K., Walker, J.M.: The intelligent copilot: A constraint-based approach to shared-adaptive control of ground vehicles. *IEEE Intelligent Transportation Systems Magazine* **5**(2), 45–54 (2013)
21. Law, C.K.H., Xu, Y.: Shared control for navigation and balance of a dynamically stable robot. *Proceedings - IEEE International Conference on Robotics and Automation* **2**, 1985–1990 (2002)
22. Yokokohji, Y., Ogawa, A., Hasunuma, H., Yoshikawa, T.: Operation modes for cooperating with autonomous functions in intelligent teleoperation systems. In: *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 3, pp. 510–515. IEEE (1993)

23. Kiselev, A., Mosiello, G., Kristoffersson, A., Loutfi, A.: Semi-autonomous Cooperative Driving for Mobile Robotic Telepresence Systems. In: Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction. HRI 2014, 104. ACM New York (2014)
24. International Organization for Standardization: ISO 13482:2014 Robots and robotic devices - Safety requirements for personal care robots (2014)
25. Giraff Technologies AB: Giraff Technologies AB (2013)
26. Joblove, G.H., Greenberg, D.: Color Spaces for Computer Graphics. SIGGRAPH Comput. Graph. **12**(3), 20–25 (1978)
27. Stockman, G., Shapiro, L.G.: Computer Vision, 1st edn. Prentice Hall PTR, Upper Saddle River (2001)
28. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
29. Bytedeco: JavaCV (2014)
30. Mosiello, G., Kiselev, A., Loutfi, A.: Using Augmented Reality to Improve Usability of the User Interface for Driving a Telepresence Robot. Paladyn, Journal of Behavioral Robotics **4**(3), 174–181 (2013)
31. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7), 971–987 (2002)
32. Fogel, I., Sagi, D.: Gabor filters as texture discriminator (1989)