Journal of Intelligent & Robotic Systems (2023) 107:10
https://doi.org/10.1007/s10846-023-01807-4

SHORT PAPER

Egocentric Computer Vision for Hands-Free Robotic Wheelchair Navigation

Mohammed Kutbi[1] · Haoxiang Li[2] · Yizhe Chang[3] · Bo Sun[4] · Xin Li[5] · Changjiang Cai[6] · Nikolaos Agadakos[7] · Gang Hua[2] · Philippos Mordohai[8]

Received: 24 July 2022 / Accepted: 4 January 2023 / Published online: 14 January 2023

Abstract

In this paper, we present an approach for navigating a robotic wheelchair that provides users with multiple levels of autonomy and navigation capabilities to fit their individual needs and preferences. We focus on three main aspects: (i) egocentric computer vision based motion control to provide a natural human-robot interface to wheelchair users with impaired hand usage; (ii) techniques that enable user to initiate autonomous navigation to a location, object or person without use of the hands; and (iii) a framework that learns to navigate the wheelchair according to its user's, often subjective, criteria and preferences. These contributions are evaluated qualitatively and quantitatively in user studies with several subjects demonstrating their effectiveness. These studies have been conducted with healthy subjects, but they still indicate that clinical tests of the proposed technology can be initiated.

Keywords Robotic wheelchair · Egocentric camera · Assistive technology

1 Introduction

Wheelchairs are essential for people who face difficulties while walking, and are required by about 1% of the population. According to World Health Organization's Assistive Technology Fact Sheet (2016) [1], 75 million people need a wheelchair and only 5% to 15% of those in need have access to one. In the United States (US), per the US 2010 census, there are 3.6 million wheelchair users nationwide [2]. While in Canada, 49% of seniors in institutional settings are wheelchair users [2]. Moreover, Europe has around 5 million wheelchair users and, sadly, 2 million of them require alternative interfaces to control their wheelchairs [3] as they suffer from upper-limb mobility difficulty that prevent them from using the traditional joystick. Additionally, approximately 10% of wheelchair users need help controlling their wheelchair [4].

Our work is focused on wheelchair users with limited or no upper-limb mobility, such as those who suffer from quadriplegia, cervical spinal cord injury or other disabilities and the elderly. These people may have difficulty controlling the wheelchair via the joystick [5]. A number of alternative wheelchair control approaches, reviewed in Section 2, have been developed to give access to wheelchairs to some of these people.

At the core of our research lies a novel wheelchair motion control approach based on an egocentric wearable camera. In our prototype, a web camera mounted on a cap is used, but other implementations are possible. To drive the wheelchair using our design, users move their heads within a small range to control a virtual joystick, which tracks the motion of the head and is shown on a frontal display. The movement of the head is slight to keep the amount of efforts required small and apply no pressure on the user's neck by external forces [6]. The frontal display serves as feedback helping the user understand the state of the robot and learn how to control it. As described in Section 3, in addition to the camera and the display, our system includes a consumer depth camera and a laptop. An initial implementation of this approach was presented by Li et al. [7].

There are three advantages due to this setup [7]. The first is robustness compared to computer vision-based systems relying on gesture recognition or face detection, which are harder problems than the one our system faces in uncontrolled environments. Factors such as cluttered background, illumination and face pose variations can be

✉ Mohammed Kutbi
m.kutbi@seu.edu.sa

Extended author information available on the last page of the article.

very challenging for real-time gesture recognition and face detection [8]. The second is that our system requires only small head motions, which feel more natural to the users and reduce their self-consciousness compared to having to perform pre-determined expression or motion sequences to control the wheelchair [6]. The third is that the egocentric camera allows the robot to see what the user sees.

In contrast to our proposed method, most of the existing hands-free control methods require the user's full attention during navigation [9–11]. (See Section 2 for details.) For certain categories of users, such as the elderly or those with disabilities, increasing the level of autonomy is desirable in order to decrease effort and safety risks. Therefore, we provide users with multiple levels of autonomy and navigation control options to fit their individual needs.

We present four hands-free use cases. In the first use case, the user can select a room in a known map via a voice command. In the second use case, the user can trigger navigation to an object that does not have to be known a priori via an attention-based mechanism initiated by the user's gaze. The third use-case is similar, but navigation is towards a known person identified by the robot via face detection. Finally, in the fourth use-case, the wheelchair follows an unknown person in a potentially unknown environment.

In the above scenarios, the wheelchair navigates autonomously towards destinations specified by the user. Following previous research in human-robot interaction [12], we would like our system to consider user preferences and comfort, in addition to safety and efficiency, during navigation. As preferences vary across users, there is a need for a shared autonomy approach that allows path planning to be individually customized. In our approach, the system involves the user in path planning only when the system's estimated uncertainty is high. For example, destination is set by the user, then, several path are planned by the wheelchair to reach that destination, instead of one destination as it is in a typical fully autonomous navigation system. If there is an obvious preferred path among these paths according to learned user's preferences, the wheelchair immediately navigate according to the preferred path. Otherwise, if the provided paths are similarly preferred by the user, the user is asked to make a choice. This research was first presented by Chang et al. [13].

This paper provides a comprehensive description of a long-term effort in the development of the robotic wheelchair. It focuses on the technical aspects of the system, while a companion paper [6] presents multiple studies to evaluate its usability. The contributions of the current paper include:

1. A hands-free navigation approach for wheelchairs based on egocentric computer vision;
2. Four use cases for the autonomous navigation mode of the wheelchair;
3. A user interface to enable multiple levels of autonomy and involve the user only in hard decisions and, whenever user is making decisions, system learns from these decisions.

We are not aware of any other robotic wheelchair that provides a set of capabilities at different levels of autonomy similar to the ones presented in this paper. The rest of paper is organized as follows: in Section 2, we briefly review related publications from the literature; Section 3 is an overview of our system; Section 4 contains a description of our egocentric hands-free wheelchair control approach; in Section 5, we present autonomous navigation use cases implemented by our system; in Section 6.2, we describe an approach for learning user preferences in wheelchair navigation; Section 7 contains descriptions and results from our experiments; we summarize our conclusion in Section 8.

2 Related Work

In this section, we review technologies for controlling wheelchairs without use of the hands and for learning the user's navigation preferences.

2.1 Hands-free Methods for Wheelchair Control

We begin with methods that do not rely on computer vision and then turn our attention to those that do.

In the sip-n-puff system [14], user control wheelchair by "sipping" and "puffing" on a pneumatic tube in variant levels of strength. The approach is suitable for users who suffer from both upper and lower-limb mobility difficulties. However, it disrupt their breathing by requiring them to switch between deep and shallow inhales and exhales affecting their natural breathing. Moreover, the user cannot communicate with others during navigation. Chin-based [15] or head-based control [9, 16] is feasible for users who can move their heads. In a head-control system, switches mounted on the headrest are operated by head movement. In a chin-control system, the user's chin sits in a cup-shaped joystick. Users control the wheelchair by neck flexion, extension, and rotation. Similar to ours, both control schemes require frequent neck movement, but they also require the users to apply forces on the tactile sensors, which can be wearisome.

Tongue-based human machine interfaces [17, 18], brain-controlled [19, 20] and voice-controlled wheelchairs [21–23] are among recent work. Tongue-based control may rely on inductive devices installed in the user's mouth and on the user's tongue [17], or non-invasive techniques

[18] that capture the tongue movement via the induced ear pressure. All tongue-based solutions, unlike our system, interfere with the user's ability to speak.

Another class of control methods is the brain-computer interface (BCI) which receives input by sensing the electrical activity of the user's brain. To drive the wheelchair, users are not required to perform any physical action nor use any mechanical control device, however, they are required to place a set of electrodes on their scalp and imagine a movement such as the kinaesthetic movement of one or more limbs [19, 20, 24]. These motor imageries can then be captured and classified generating different commands to drive the wheelchair.

A semi-autonomous BCI system using electroencephalogram (EEG) signals to interface with the user has been proposed to enhance mobile robot navigation in an uncertain environment [25]. It uses vanishing points and door plate as environmental feature in performing Simultaneous Localization and Mapping (SLAM) to allow the system to build obstacle-free trajectories to the destination. BCI has the potential to help severely disabled users, but requires their full attention. On the other hand, our system require less attention from the users.

Computer vision has been used to develop assistive wheelchair controls. Previous methods either use fixed cameras on the wheelchair to sense the environment [26, 27] or use a camera looking at the user to translate the user's motion into control signals [28, 29]. The user simply selects the target objects or locations on a screen and the robot approaches and grasps the object. Pasteau et al. [27] use computer vision to activate automatic trajectory corrections and avoid collisions. Purwanto et al. [28] detect the user's gaze and blinks with a user-facing camera to control the wheelchair. Similarly, Gray et al. [29] control the wheelchair with head gestures recognized by a camera focused on the user's face. Xu et al. [30] designed a wheelchair that can be driven by recognizing the gaze orientation of the user. Additionally, with the help of beacons and markers places in the scene the wheelchair navigate and avoid obstacle.

The use of outward-facing cameras is more challenging, especially due to motion and illumination changes in the scene. According to Halawani et al. [31], an outward-facing camera is superior to an inward-facing one due to its wider field of view. Their system tracks head motion through a camera on the user's hat that is looking downwards towards the user's clothes and the wheelchair. As a result, the observed motion is caused by head-motion instead of wheelchair-motion and, therefore, allows the activation of five discrete commands. Kim et al. [32] localize a robotic wheelchair based on special visual markers whose appearance changes according to viewing angle. Our design requires a single marker printed on plain paper.

Zolotas et al. [33] propose a user interface based on the Microsoft Hololens, which enables the system to augment the user's view of the scene with virtual annotations and explanations to objects that are shown on the Hololens screen. While the user drives the wheelchair via a joystick, the Hololens localizes the user's head in space and place the virtual content accordingly. More recently, Chacón-Quesada and Demiris [34] enabled the user to control the wheelchair via an eye-gaze tracker. The egocentric camera in our system is not in the user's field of view.

2.2 Learning from Previous Navigation Experience

In this section, we review work on robotic wheelchairs that considers shared autonomy. The relevant approaches vary in terms of their criteria for measuring comfort or safety and the human-robot collaboration mechanism. Gulati et al. [35] assess how comfortable motions are based on features including normal and tangential jerk, angular velocity and acceleration. However, in contrast to our work, obstacles in the scene are not included among the criteria. Shiomi et al. [36] use the behavior of caregivers to customize the behavior of the wheelchair. Our approach learns directly from the users performing relevant actions themselves.

Parikh et al. [37, 38] consider user inputs, reactive behavior and deliberate notion plans to enhance semi-autonomous navigation by helping users avoid collisions. Ceres et al. [3] proposed a robotic platform with five levels of autonomy to enable users, especially children, to navigate safely. It uses an automatic obstacle avoidance technique to increase safety. Zeng et al. [39] presented a user-wheelchair collaborative system that requires user to provide the destination and preferred speed while the system plans and navigates accordingly. During navigation, users have the option to modify the path. There are some wheelchair platforms that allow the user to complement the robot using a human-robot interaction interface e.g. door opening which is a task that is easy for users to do but challenging for a robot [40].

Urdiales et al. [41] proposed an efficiency measure, based on smoothness, directness and safety, which is later used to integrate user and robot-generated commands to accomplish navigation goals. Another shared control platform developed by Li et al. [42] integrates the commands from the robot and the user while considering a measure for the level of assistance adapted based on the user's capabilities. Carlsson and Demiris [43] propose an approach that allows users to control the wheelchair, while the robot can modify the control signals for safety reasons. A different approach for combining human and robot commands relies on machine learning to achieve optimal navigation [44]. Speech recognition is a popular technique which have been used in many wheelchair shared control applications [21], including our approach.

Lastly, we turn our attention to methods operating in dynamic settings where the robot has to interact or avoid people in the scene. For additional information on this topic, we refer readers to a survey [45]. A motion planning approach designed for dynamic environments was developed by Sisbot et al. [12]. It uses safety and comfort in its cost function, as most other approaches, but additionally, its safety measure uses distance to humans while its comfort measure requires the robot to be visible by people in the scene. Kirby et al. [46] presented a framework, dubbed the Companion, which integrates features for static and dynamic elements of the scene, such as travel distance and distances to obstacles and people. Companion takes into consideration social norms such as not getting into personal spaces and passing from the right. Cosgun et al. [47] developed a path planner that uses the social force model [48] to predict how people would react to the robot to enhance navigation. It uses a static and a dynamic planner, with the former computing an optimal path based on path length, distance from people and disturbance of groups of people, and the latter refining sections of the path that are within a certain distance to people. Morales et al. [49] focus on finding the optimal set of parameters to increase comfort for pedestrians and the passenger of the wheelchair. A user study shows that subjects favor the proposed planner to one returning the shortest path. Our robotic wheelchair does not currently model human behavior. It can only learn the behavior of its user around people. This is an area for future research for us.

3 Robotic Wheelchair

In this section, we introduce our robotic wheelchair. It is designed as a modification of a commercially available power wheelchair driven by a joystick, the Drive Medical Titan Transportable Front Wheel Power Wheelchair, as shown in Fig. 1. To control the wheelchair, our navigation system uses an Arduino micro-controller (Arduino Mega 2560) to generate electrical signals mimicking those of the joystick. The Arduino-generated signals are connected to the joystick interface and passed to the motor.

The prototype described in this paper uses the following sensors and software. A Kinect v2 sensor is mounted on the wheelchair to enable RGB-D visual SLAM and to measure the distance to objects. A tablet is mounted and used as a display device in front of the user. A webcam, either a Logitech C270 or a Logitech c930e, is attached to a baseball cap which is worn by the user and shares the user's egocentric point of view with the robot. An additional webcam is mounted on the back of the wheelchair and its video feed is displayed on the tablet when the wheelchair moves in reverse. The resolution of the Kinect v2 RGB camera is 960 × 540 and its focal length is 540.68 pixels, while the resolution of the egocentric camera is 640 × 480 and its focal length is 823.1 pixels. Lastly, a QR visual marker is attached to on top of the tablet mount facing the user to help track his head pose.

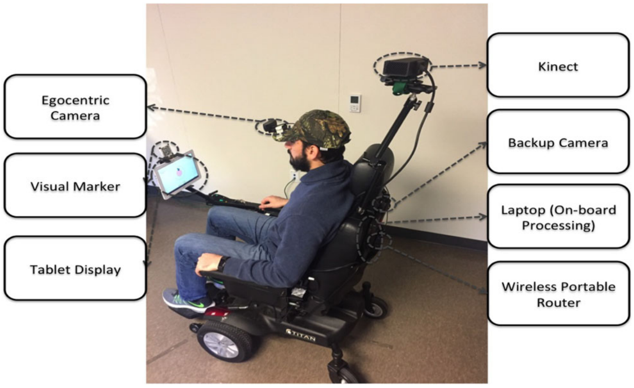


Fig. 1 The robotic wheelchair

The software system is built on the Robot Operating System (ROS) [50]. Figure 2 shows a diagram of our system. In this work, we focus on indoor navigation and rely on the RGB-D camera to sense the environment. We use Real-Time Appearance-Based Mapping (RTAB-Map) [51] to build a map of the scene beforehand. RTAB-Map is a RGB-D graph-based SLAM approach with an appearance-based loop closure detector. The loop closure detector uses a bag-of-words technique to determine whether a new image is from a previously observed location or a new location. A new constraint is added to the map's graph when a loop closure hypothesis is accepted. Then, a graph optimizer minimizes the errors in the map. For path planning in autonomous navigation, we use the ROS

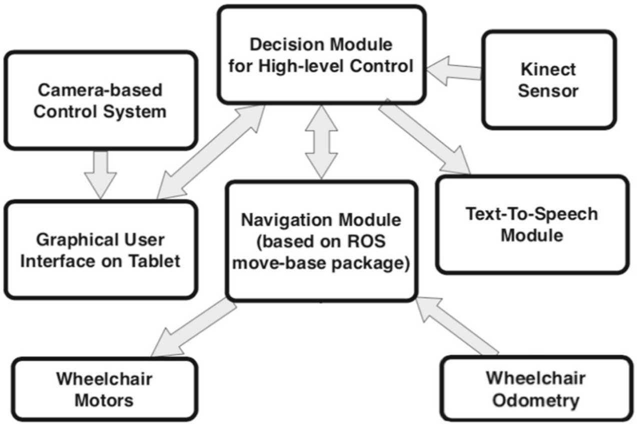

Decision Module for High-level Control
Camera-based Control System
Kinect Sensor
Graphical User Interface on Tablet
Navigation Module (based on ROS move-base package)
Text-To-Speech Module
Wheelchair Motors
Wheelchair Odometry

Fig. 2 System Diagram: the flow of commands and information is represented by the direction of the arrows. Components are activated based on the mode of navigation: manual or autonomous

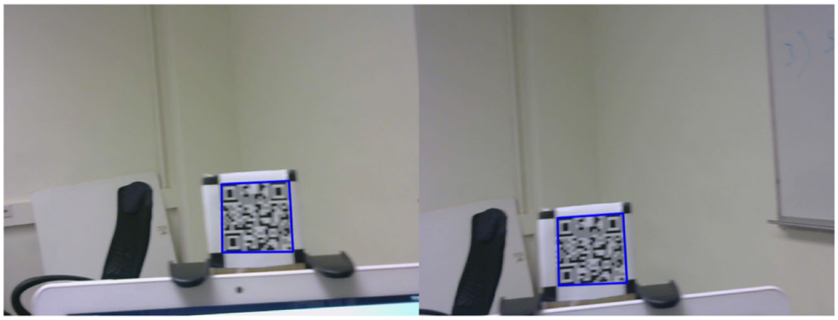


Fig. 3 The visual marker is reliably tracked in the presence of motion blur. See blue outlines

navigation stack [52]. All software runs on a laptop mounted on the wheelchair.

4 Egocentric Hands-free Control

In this section, we describe the hands-free wheelchair control approach in detail. Conceptually, the approach is divided in two parts: head motion tracking and user interface.

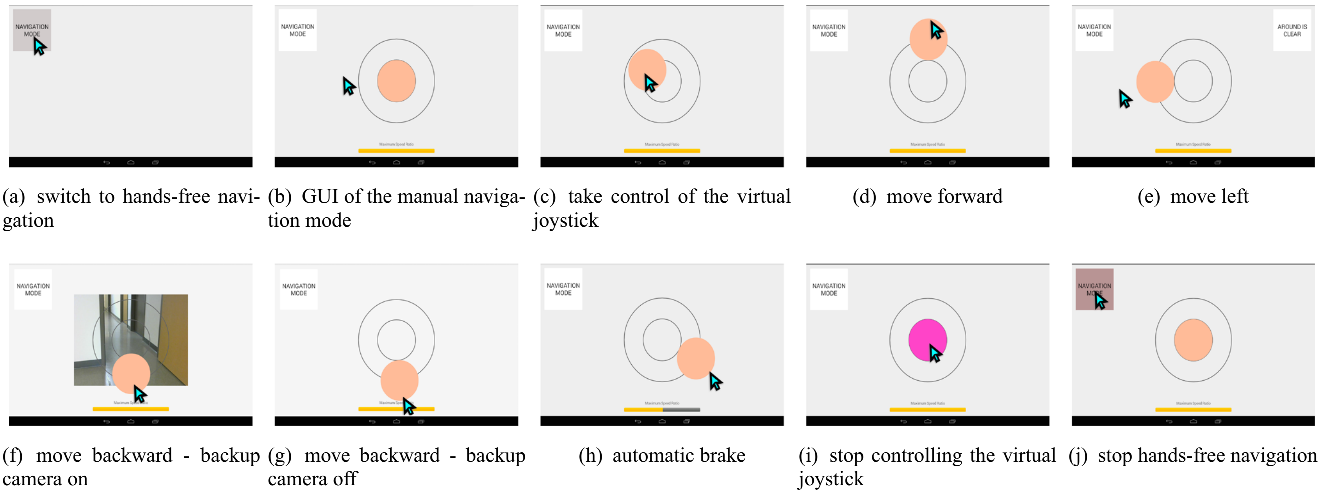
4.1 Head Motion Tracking

As shown in Fig. 1, the head motion tracking relies on two components: a head-mounted camera and a visual marker facing the user mounted on the wheelchair. We use a Quick Response (QR) code marker, which can be reliably detected in the images captured by the head-mounted camera via the ViSP library [53]. After the marker is detected initially, we use the Consensus-based Matching and Tracking of Keypoints (CMT) tracker [54] to track it. The resulting technique is robust to blur due to the use of a distinct marker and tracking, rather than re-detection in each frame (see Fig. 3). This allows reliable control of the wheelchair.

The motion of the tracked visual marker is converted to motion of the cursor on the display. Because one of the objectives of our system design is to keep head motion small, and also due to size mismatch between the display and the range of motion of the user's neck, we scale the tracked motion before updating the location of the cursor.

Because the proposed head motion estimation module can capture even small head movements, the cursor, and as a result the wheelchair, can be controlled with very small head motion and user effort. This is advantageous compared to alternatives, such as chin and head-based motion control devices that require forces to be applied by the user's head and indirectly by the neck. The frequent movement of the neck combined with these forces may cause neck problem due to repetitive stress injury or repetitive motion injury. It should be noted that we have not assessed whether our technology or alternatives strain the user's neck, but users preferred our contact-free control mechanism to a chin-based alternative in a previous study [6].

An additional advantage of our approach is that it enables the user to give continuous commands to the robot. In contrast, most previous hands-free mobility solutions provide discrete motion commands, such as move backward or forward, turn right or left, and stop.



(a) switch to hands-free navigation (b) GUI of the manual navigation mode (c) take control of the virtual joystick (d) move forward (e) move left

(f) move backward - backup camera on (g) move backward - backup camera off (h) automatic brake (i) stop controlling the virtual joystick (j) stop hands-free navigation

Fig. 4 The GUI of the manual control

4.2 User Interface

In the proposed system, the Graphical User Interface (GUI) is presented to the user on the tablet, as shown in Fig. 4(b). The position of the cursor on the display is controlled by the user's head motion, while actions can be deployed by hovering with the cursor over pre-specified locations on the screen for a given amount of time.

A typical work-flow is shown in Fig. 4. The hands-free navigation mode is engaged by holding the cursor on the corresponding button for a pre-specified amount of time. The virtual joystick is picked up by placing the cursor on the center of the display. Once the virtual joystick has been picked up, the direction and speed of the wheelchair are proportional to the placement of the cursor relative to the center of the display. As shown in Fig. 4(h), the bar at the bottom indicates the maximum speed of the wheelchair. To exit hands-free navigation, the virtual joystick must be released at the center and the "navigation mode" button must be pressed again for the same amount of time.

For safety, an automatic brake has been designed to slow down and eventually stop the robot when the marker is not in the view of the wearable camera which could be due to user being distracted or rapid change in lighting conditions affecting the tracker's ability to track the marker.

The similarity of the our control mechanism to conventional joysticks makes its use intuitive for users with experience using joysticks. Subjects in our experiments were able to learn how to operate the virtual joystick quickly.

5 Autonomous Navigation Use Cases

In this section, we present four use cases with our wheelchair in autonomous navigation mode.

5.1 Map-based Navigation

Map-based navigation allows the user to communicate with the system and specify the destination verbally. The system offers multiple routes to the user to pick from, as discussed in Section 6. This use case requires a pre-built map and a set of pre-defined locations on that map, each associated with a word or phrase. Paths are automatically generated and colored, while a voice recognition interface [55] is used to interpret commands and locations.

A typical interaction with the wheelchair in this context involves the following steps. The user engages the voice recognition system by saying "attention" followed by one of the pre-specified locations. The system provides feedback informing the user that it has received the location and, then, starts searching for available paths using a path planner (see Section 6), gathers and evaluates all available paths. If two or more paths are similarly preferred according to the user's learned preferences, then the decision is given to the user. The system shows the available paths on the map each with specific color e.g. "red" or "green" and asks the user to select from them using a voice command by naming the path color. Otherwise, the system picks the most preferred path without involving the user, and begins navigation. The preferences are learned gradually and a completely new preference can be re-learned gradually if the user's choices and decision changed drastically.

5.2 Attention-driven Navigation

Autonomous navigation is only applicable when a representation of the destination is given to the motion planner. In this section, we present an attention-driven method which provides an interface for users to specify the destination they want to navigate to if there is no name attached to it. The egocentric camera naturally follows the user's attention since its field of view largely overlaps with the center of the user's field of view. When an object persistently appears in the video steam of the wearable camera that means the user is staring at it and it is regarded as an object of interest. Then, the system can navigate to it autonomously upon the user's request.

The system switches to attention-driven navigation mode based on a voice command, typically "attention". In this mode, the user finds an object of interest and places it in the center of her/his field of view for a predefined period of time. Since the egocentric camera shares the user's viewpoint, the system can identify the attentional object, and set its location as the destination. The system, then, displays the attentional object on the frontal display and requests user confirmation. After receiving confirmation, the system hands over the image of the object taken from the egocentric camera to the RGB-D camera. The robot, then, navigates autonomously towards the object, first by rotating to the left or right according to the pose of the egocentric camera relative to the RGB-D camera, which is fixed on the frame of the robot, and then by translating once the object becomes visible in the RGB-D camera. During the autonomous phase, the user does not have to keep fixating at the object and can focus on other tasks, including setting the next destination.

5.2.1 Attentional Object Detection

To enable the above functionality, we developed an efficient object detection technique, shown in Fig. 5. We begin by detecting contours in the images of the egocentric camera [56] and obtain the bounding rectangles of all closed contours as hypotheses for the attentional object. To identify the bounding rectangle of the attentional object, we use

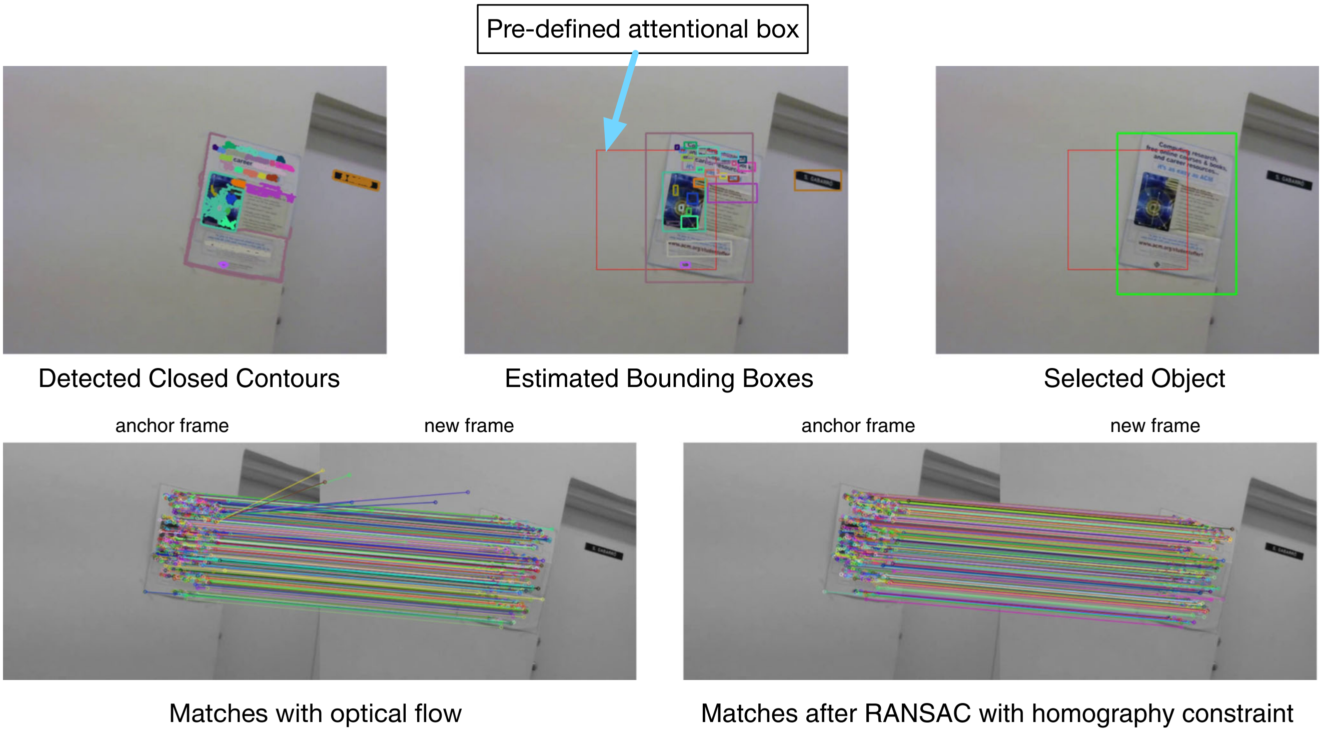


Fig. 5 Work-flow of the proposed attentional object detection. The top row shows how the system detects the attentional object. The bottom row shows how the system matches images from two cameras

the Intersection-Over-Union (IoU) of the hypotheses with a pre-specified attentional area in the center of the image as a score. If the score exceeds a threshold, set to 0.2 here, the current frame is identified as the anchor and the object enclosed by the corresponding contour as a potential attentional object. Then, as shown in the bottom row of Fig. 5, the system tracks the object in the video. Optical flow is estimated between the anchor and current frame and a homography is estimated using Random sample consensus (RANSAC) [57]. To evaluate the quality of the matching, we use the ratio of the inliers of the estimated homography over the total number of matches. If at least 50 frames with an inlier ratio of at least 50% are accumulated, the attentional object is considered initialized and it is handed over to the RGB-D camera.

5.2.2 Object Hand-over

After the object of interest has been initialized, it is presented in the frontal display and the user is asked to confirm that navigation to the object should commence. After user confirmation, detecting and tracking the object is handed over to the RGB-D camera, which, however, may not initially observe the object. If that is the case, the robot rotates to the left or to the right guided by the user's estimated head pose until the object is detected in the RGB-D camera.

We use feature correspondences to make the initial detection in the RGB-D camera. Specifically, we apply the Features from Accelerated Segment Test (FAST) [58] feature detector and the Oriented FAST and rotated BRIEF (ORB) [59] feature descriptor on the anchor frame from the egocentric camera and the frames of the RGB-D camera. Potential correspondences are used as input to RANSAC for estimating a homography between the two views. The relative pose of the object with respect to the robot is then estimated using the depth of the matches features in the RGB-D frame.

In order to evaluate the quality of the matching, we measure how well the projected object bounding box approximates the estimated bounding box. The metric is designed under the mild assumption that the distances from

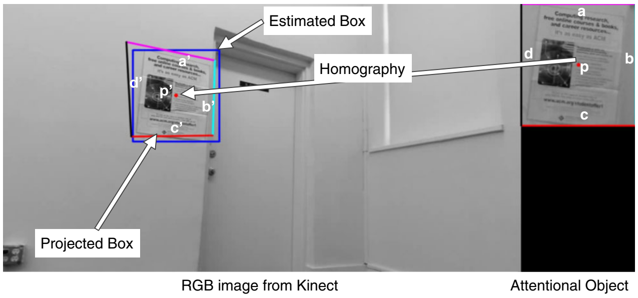


Fig. 6 Attentional object localization

the object to the RGB-D sensor and egocentric camera are very similar, after compensating for differences in the resolution and focal length of the two sensors (see Section 3). As shown in Fig. 6, the projected bounding box is obtained by mapping the object boundaries using the estimated homography.

5.3 Person-guided Navigation

In this navigation scenario, the user wishes to approach a specific person in a scene with one or more people. This can be accomplished as above with the difference that the target people are known a priori. The motivation behind this use case is that it is inefficient and uncomfortable for the robotic wheelchair to locate a person of interest by searching the environment. The search process would require primarily rotating the wheelchair in place until the person of interest is seen by the RGB-D sensor. We propose to leverage the user's ability to quickly locate the person of interest in the egocentric camera, use the orientation of the egocentric camera relative to the frame of the robot to rotate to the appropriate direction, and also leverage the autonomous driving capability of the robot to relieve the user from having to operate the wheelchair.

Prior to the deployment of this mode, an enrollment process is required to register the group of known people into the face recognition system. During enrollment, we ask each person to stand in front of the wheelchair and record a 5 seconds video while the person is looking at the RGB-D camera. These short videos are associated with that person and serve as the representation of a person in our system.

The work-flow starts with selecting a target person in the user's field of view. Since the user shares the field of view with the egocentric camera, as the user searches for the target person, all faces in front of the camera are detected and compared to the database. Until the target person is identified, head motion history is used to provide insight into the location of the person relative to the wheelchair. Thus, the wheelchair turns to the correct direction to approach the target person. Once the RGB-D sensor sees the target person, the wheelchair autonomously navigates to the target person. While navigating, the user is free to look around. We consider this a great example of combining the strengths, and fields of view, of the user, the egocentric camera and the stationary RGB-D camera mounted on the wheelchair to achieve high efficiency in a useful task.

For face detection, recognition and tracking, we use the Cascade Convolutional Neural Network based face detector [60], the Probabilistic Elastic Part based model face recognition system [61], and the Consensus-based Matching (CMT) face tracker [62], respectively.

5.4 People-following

In this navigation scenario, the wheelchair follows a person without requiring pre-registration of people in the system or any information about them. The system detects the nearest person, regardless of their orientation relative to the RGB-D camera or whether they are standing or walking. In this mode, the wheelchair aims to move close to the detected person up to a pre-specified distance.

For input, the system relies exclusively on the RGB-D sensor, utilizing a depth-based upper-body detector and an RGB-D tracker. Given the location of the detected person, appropriate motion commands based on the distance and angle to the person are issued and executed. The Spencer-project person detector [63, 64], specifically the depth-based upper-body detector [65], is used for people detection and the multi-object tracker of Ess et al. [66] for tracking.

6 Learning to Navigate

In this section, we present an approach for learning the user's path selection preferences by observing choices users make when navigating the wheelchair and in simulation. We accomplish this by training a Support Vector Machine (SVM) to rank paths according to the observed choices. This formulation is more effective than a set of rules for assigning appropriate weights to speed, distance, comfort etc. This work was initially presented in [13].

6.1 Planning Multiple Paths

To learn users' preferences through their selections of paths, we need a path planner that generates multiple paths, instead of just one as typical path planners do [67]. In this subsection, we present two planners for finding multiple paths: one generates homotopically distinct paths using the Generalized Voronoi Diagram (GVD) [68], while the other generates multiple paths, that are not necessarily homotopically distinct, by applying a modified A* algorithm iteratively.

The inputs here comprise a map in 2D occupancy grid format, which can be acquired with the RGB-D sensor, the current position, and the destination. Path planners typically generate one path according to their objectives. But in our case, we need to generate several paths for a given scenario. Here, we use two planning methods, separately, to generate a diverse set of paths. The goal is to generate at least one path that matches the user's preferences. A way to determine whether two paths are different is based on the notion homotopy [68–70]. Homotopically distinct paths are paths that have at least one obstacle separating them, preventing a continuous transformation from one path to another. This,

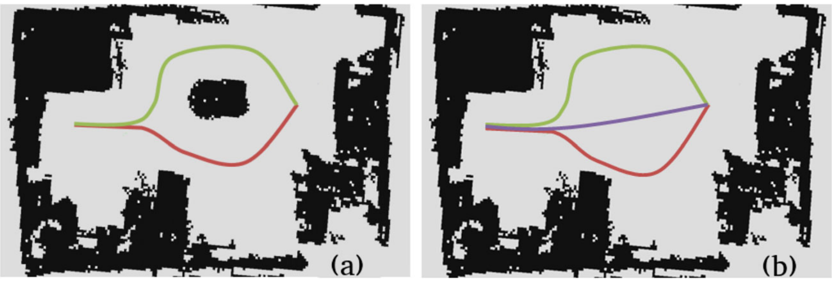


Fig. 7 2D occupancy grids of a room containing: (a) homotopically distinct paths, and (b) distinct paths in the same homotopy class

however, is not the only criterion. For instance, in example (a) in Fig. 7, the table placed in the middle of the room separates the space and creates two homotopy-classes of paths, one on each side of the table. When the table is removed in Fig. 7(b), all paths are homotopically the same and cannot be distinguished based on homotopy. However, wheelchair users have different preferences: some may prefer driving close to a wall and leave space for people to walk for example, while others may prefer to stay as far from obstacles as possible.

6.1.1 Generating Homotopically Distinct Paths Using the GVD Planner

We use the Generalized Voronoi Diagram (GVD) to generate homotopically distinct paths, as in the approach of Kuderer et al. [68]. This yields a graph in which all paths are homotopically distinct. As a result, the k-shortest paths connecting two vertices on the graph correspond to the k-shortest homotopically distinct paths between the vertices on the map. Finding shortest paths on a graph is a well-studied problem in computer science [71].

Figure 8 illustrates the four-step process of finding homotopically distinct paths.

(a) Construction of the GVD starts with a map of the environment, provided in the form of an occupancy grid. We require a *generalized* Voronoi diagram because the obstacles in the map are not restricted to points. The boundaries of the polygonal cells of the GVD are free locations in the map that are equidistant to the two (or more) nearest obstacles [72]. To estimate distances from all locations in free space to the obstacles efficiently we compute the Euclidean distance transform [73] and use it to identify the edges of the GVD. The output is a binary map identifying whether cells belong to the edges of the GVD or not.

(b) Then, the start and end point are connected to their closest point in the GVD with straight lines.

(c) The GVD graph representation $G = (V, E)$ is then extracted from the GVD binary occupancy grid. Vertices V are detected as cells of the occupancy grid with more than two incoming edges. The start and end points are considered special vertices (the only ones with exactly one edge). The weight of each edge is set equal to the distance between the two vertices it connects.

(d) The K-shortest paths in this graph are found using Dijkstra's algorithm. By construction, each path is homotopically distinct.

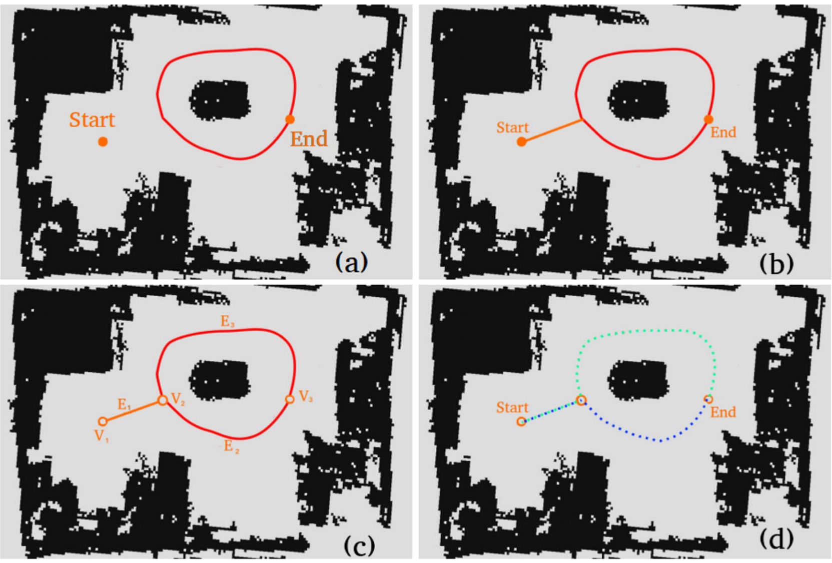


Fig. 8 Overview of the use of GVD for finding homotopically distinct paths in a map.
(a) Construction of the GVD.
(b) The start and end point are connected to their closest points in the GVD. (c) Graph representation of the GVD.
(d) K-shortest paths in the graph ($K = 2$)

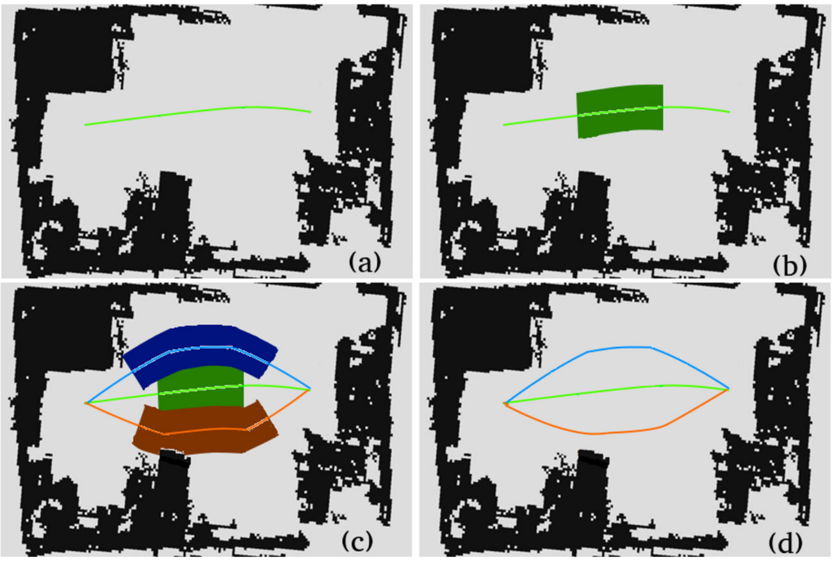


Fig. 9 Overview of the Iterative A* Algorithm. (a) Apply A* algorithm to find the shortest path. (b) Block the detected shortest path with a virtual obstacle. (c) Repeat steps (a) and (b) until there are no more paths. (d) Collect all paths

6.1.2 Path Planning using the Iterative A* Algorithm

The above approach is inapplicable for generating multiple paths in wide spaces without obstacles that can be circumnavigated. We would still like to capture user preferences, such as driving down the middle of a wide hallway, or keeping to the right, or staying away from people. To handle situations where we would like to differentiate between paths in the same homotopy class, we developed a variant of the A* algorithm.

The Iterative A* Algorithm, shown in Fig. 9, works as follows:

(a) Use the A* algorithm to find the shortest path in the occupancy grid. Exit to step (d) if no path can be found.
(b) Block the most recently found shortest path with a virtual obstacle. (This is implemented by relabeling the occupancy grid cells as occupied in a rectangle of fixed size around the midpoint of the path. The area around the start and end of the path are left unchanged to avoid blocking additional acceptable paths.
(c) Go back to step (a) using the modified map from step (b).
(d) Return all paths that have been detected.

6.2 Learning User Preferences

To be able to collect data to train and test our approach, we have designed two user interfaces (UIs): one on a simulated wheelchair and the other on the actual robotic wheelchair. Regardless of the UI, users are offered multiple paths and asked to choose their preferred one. Given these path selections, our approach learns to rank paths according to the preferences of the user. We formulate the problem as pairwise selection and train an SVM to perform ordinal regression [74].

First, we collect a dataset comprising selections made by the same user, from which we obtain ranking constraints. For each scenario, we obtain pairwise constraints between the selected path and those that were not selected, but no constraints among paths that were not selected.

We use the term *scenario* to refer to a tuple of the robot's initial location and goal on a given map. Our system plans multiple paths for every scenario and asks the user to pick one of them. On the simulator, this is implemented as a single click on the path, while on the actual robot it is implemented as a voice command. Data collection and annotation effort in the simulator is less time consuming compared to the same task on the actual robot. However, since path selection on the simulator is not equivalent to collecting actual data, we evaluate the effectiveness of both types of data in Sections 7.1.3 and 7.1.4.

In order to train the path ranking system, we define a set of features to represent each path. Those features convey information that affects user selections in both static and dynamic environments, where people must be considered by the planner.

6.2.1 Features for Paths in Static Environments

Users differ in their path selection criteria. In a static environment, for a path $\boldsymbol{X}$ consisting of a sequence of poses $\boldsymbol{x}$, the feature vector $\boldsymbol{f}(\boldsymbol{X})$ has the following five elements:

(a) Path length: users prefer the shortest path among equivalent options. We calculate the length of the path by summing the straight-line distances between consecutive poses in the path.

$$l_p = \sum_{i=1}^{n} l(x_i, x_{i+1}) \tag{1}$$

(b) Narrow passage length: we define as a "narrow passage" a segment of the path that is near obstacles on both sides. Such segments may be undesirable to certain users due to the increased chance of collision. We define a narrow passage segment as a set of consecutive poses $\boldsymbol{X_n}$ among all poses $\boldsymbol{X}$ such that $d(x_i) < d_n$ and $d(x_{i+1}) < d_n$ where d_n is a constant, set as $d_n = 0.5m$ here.

$$l_n = \sum l(x_i, x_{i+1}), \quad \{x_i, x_{i+1}\} \in \boldsymbol{X_n} \tag{2}$$

(c) Average distance to obstacles: the distance between every pose to the nearest obstacle. This feature is an indication of comfort and safety, similar to corresponding features in the literature [21, 41, 46].

$$\bar{d} = \frac{\sum ||d(x_i)||}{N} \tag{3}$$

(d) Minimum distance to obstacle: the minimum among all distances from any pose to the respective nearest obstacle.

$$d_{\min} = \min(||d(x_i))|| \tag{4}$$

(e) Sum of turning angles: is an indication of user comfort [35, 41, 75], since the angles between consecutive poses are related to angular velocity and acceleration of the wheelchair. Angular velocity and acceleration may cause discomfort if they are large.

$$\delta_{sum} = \sum \|\delta(x_{i+1}) - \delta(x_i)\| \tag{5}$$

6.2.2 Features for Paths in Environments with People

Our approach also takes people in the scene into consideration. People move, and, more importantly, affect the wheelchair in different ways than static obstacles [12, 45–47, 49]. Our path planner only considers instantaneous observations and re-planning is needed as people move in the environment. To facilitate path ranking in the presence of people, we define two additional features.To acquire data for these two features, which are defined below, we use the SPENCER person detector [63, 64].

(f) Average distance to people: this feature is analogous to the average distance to obstacles, but the average is taken over distances to people.

$$\bar{d_p} = \frac{\sum ||d_p(x_i)||}{N} \tag{6}$$

(g) Minimum distance to people: this feature is analogous to the minimum distance to obstacles.

$$d_{p\_min} = min(d_p(x_i)) \tag{7}$$

6.2.3 Support Vector Pairwise Ranking

So far, we have presented ways to generate multiple paths and to encode each of them in a feature vector. The next step is to predict the most preferable path to the user, which is a ranking, not a classification, task. In each training scenario, the user selects one path and rejects all other options. Thus, what we can learn from the training data is that the user preferred a given path every other available path. However, no knowledge is available about the relative ranks of the paths that were not chosen. Taking this into account, the problem is formulated as an ordinal regression task and solved according to the approach of Herbrich et al. [74] which relies on an SVM subject to pairwise constraints.

We use the same formulation for both static and dynamic environments. We begin by normalizing the features to have mean equal to zero and variance equal to one, thus, obtaining normalized feature vectors denoted by $\hat{\boldsymbol{f}}(\boldsymbol{X})$. The training set is generated by forming pairs comprising a preferred and a not-preferred path from the same scenario. Then, an element-wise vector subtraction is preformed on the feature vectors of the paired paths resulting in a vector we refer to as the *preference vector*. Half of these subtractions are performed by subtracting the feature vector of the preferred path from that of the not-preferred path, while the other half are carried out in the opposite direction. The resulting preference vectors are labeled as follows:

$$y(\boldsymbol{p}_{ij}) = \begin{cases} +1 & \text{if user prefers path } X_i \text{ to } X_j \\ -1 & \text{if user prefers path } X_j \text{ to } X_i. \end{cases} \tag{8}$$

A Linear SVM is trained to predict the label of an input preference vector, which is equivalent to predicting the most preferable path. Because the SVM is linear, we can multiply its weight vector $\boldsymbol{w}$ with the feature vectors $\hat{\boldsymbol{f}}(\boldsymbol{X})$ before the subtraction. The resulting product is the score of a path; paths with higher scores are preferred over alternatives with lower scores.

$$S(\boldsymbol{X}) = \boldsymbol{w}^T \hat{f}(X) \tag{9}$$

For fully autonomous navigation, the path with the maximum score is selected. But in cases where the top two ranked paths have scores within a predefined margin, we engage the user to break the tie. Whenever the user is involved in the selection, additional training data are generated.

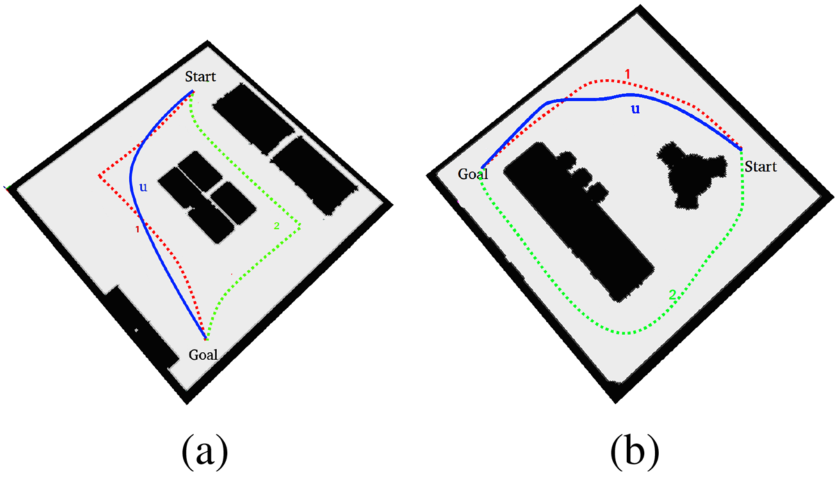


Fig. 10 Mapping user paths to those generated by the planner. Solid curves marked with a *u* correspond to paths traced by the user, while dashed curves correspond to paths generated by the GVD planner. The path followed by the user is mapped to the automatically generated path in the same homotopy class, which must have been generated by the planner. (Not all homotopically distinct paths are shown to reduce clutter)

6.3 Learning from Demonstration

Every time we record a path driven by the user, we assume that the user has selected that path over a number of alternatives. To obtain all possible paths we use the GVD planner described in Section 6.1.1 and shown in Fig. 8. The set of paths must contain a path that is homotopically similar to the recorded path, as shown in Fig. 10. Therefore, we can map any path, complete or partial, the user has followed to its homotopically equivalent path. This is done using the vector of winding angles as the descriptor of a path, according to Kuderer et al. [68]. We then attach features to the paths, as described in Section 6.2.1, and use the classifier to compare two or more paths using the user choice to generate labels, as described in Section 6.2.3.

7 Experiments

In this section, we present experiments on learning user preferences in path selection, attention-driven navigation, person-guided navigation and people following.

7.1 Learning User Preferences in Path Selection

In this section, we present an experiment on our approach for learning personalized user preferences. We attempt to learn navigation models for two subjects based on a set of 144 scenarios for each subject. *A scenario is defined as a navigation between specific start and goal positions in a specific environment.*. Specifically, for each subject there were 36 scenarios for each planner in static scenes and 36 more for each planner in dynamic scenes, that is in the presence of people. Both real and simulated platforms are used for this experiment.

7.1.1 Experimental Setup

An experiment has been designed to evaluate the path ranking approach on the simulated and actual robotic wheelchair. The goal is to learn the preferences and create a customized path planner for each user, rather than the "average" preference of the crowd [76]. Two users with different preferences performed both the simulated and physical experiments.

A ROS-based simulator that uses our path planners has been developed to collect the users' path selections for training. The trained planner was then deployed in both simulated and physical environments for testing.

7.1.2 Data Collection in the Simulator

Data were collected in four virtual maps of similar complexity to the space used for experiments (see Fig. 11 for examples). Each map has a different layout and contains furniture, such as a bed, a desk, a dining table, a sofa, etc. We intentionally designed maps to have one, two or more homotopically distinct paths. We generated scenarios by asking the subjects to pick the start and end points. Then, the system generates a set of possible paths using one of the planners and the subject selects among them.

An additional dataset was collected in the same four maps after adding people to the scene to mimic dynamic environments to measure the effects of the presence of people on navigation choices. In total, each of the static and dynamic datasets contains 36 scenarios per planner per user.

7.1.3 Results in Simulated Maps

The subjects were offered a set of paths to select from, as shown in Fig. 11. For maps that have only one

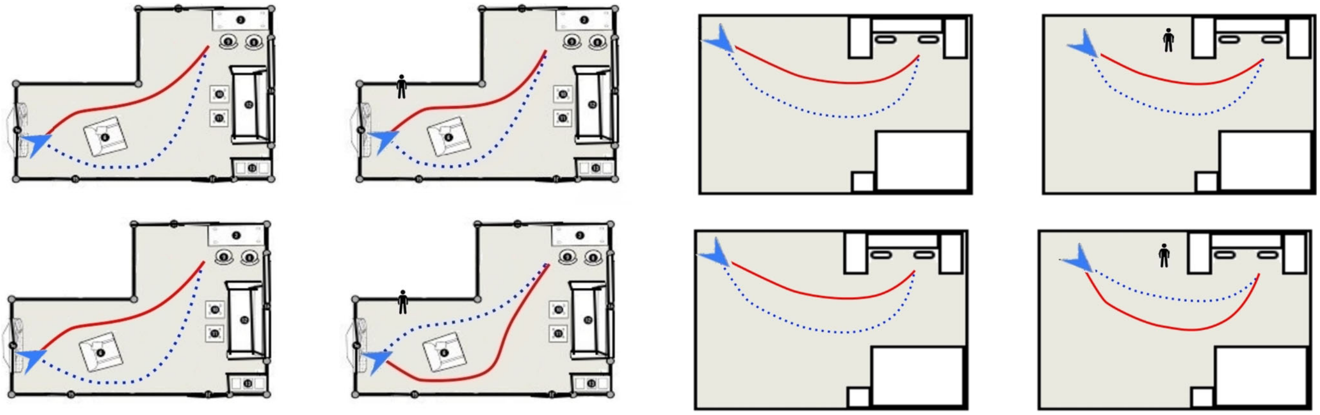


Fig. 11 Path selection examples in the simulator. Solid red curves represent the preferred path and dotted blue ones the paths that were not chosen. The selections of subject 1 are shown at the top and of subject 2 at the bottom. Note: subject preferences are different when people are present in the environment

homotopically distinct path, the GVD planner was not used, since it cannot generate multiple options in them. Figure 11 also shows the selected paths by both subjects in two of the maps.

For each planner and each environment category, we split each subject's training data into two thirds for training and one third for testing. This split results in 8 pairs of training and testing datasets, and 8 corresponding SVMs trained as detailed in Section 6.2.3.

To limit the number of times the system asks for user involvement, we define a confidence calculated by subtracting the scores of the top two ranked paths. We can control how often the system decides autonomously, *the system decision rate*, by thresholding this confidence. The ROC curves of system accuracy over system decision rate as the threshold varies is shown in Fig. 12. Accuracy is 85-90% when the system is in full autonomous mode and never involves the user. As the system starts involving the user when confidence is low, the accuracy increases. In the rest of the section, the confidence threshold is set to 0.2.

Average results after randomly splitting each dataset 20 times are shown in Table 1. System accuracy in selecting the preferred path is always above 76%, at most 14.4% of system choices are incorrect, while 2-12% of the decisions are deferred to the user.

It should be noted that for some subjects who are consistent in path selection such as subject 2, who said "I always pick the path that was away from the person, especially for open spaces," the system accuracy increases using the iterative A* planner.

7.1.4 Results on Robotic Wheelchair

After training our approach in simulation, we tested it in a real scene: a studio shown in Fig. 13. The studio has a bed, a dining table and areas labeled study area and kitchen. The dining table at the center produces two homotopically distinct paths in most scenarios. In some cases, the two available paths have similar feature vectors, such as the paths between the kitchen and bed. In other cases, such as between the bed and the study area, the paths have very different feature vectors making the choice more obvious.

Our physical experiments test the ability to train the system in simulation and deploy the learned ranking model in real environments. Preliminary results are encouraging. The observed differences in users' preferences persist across the simulator and the physical wheelchair, especially in dynamic environments. Pictures and video from the physical experiments are shown in Fig. 14 and in the supplemental material, respectively.

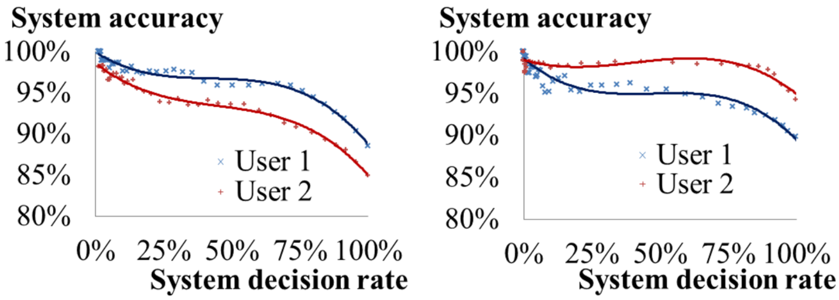


Fig. 12 Plot of accuracy vs. system decision rate for static (left) and dynamic (right) scenes, averaged over 72 scenarios per user (36 scenarios per planner)

Table 1 Results in the simulator for both users in static (top) and dynamic (bottom) scenes averaged over 36 scenarios in each category

	GVD Path Planner			Iterative A* Path Planner		
	Preferred Path was Chosen	Not Preferred Path was Chosen	Asked user to decide	Preferred Path was Chosen	Not Preferred Path was Chosen	Asked user to decide
User 1	91.8%	2.4%	5.6%	76.3%	12.0%	11.5%
User 2	79.6%	10.7%	9.6%	81.8%	11.1%	7.0%
User 1	93.9%	2.1%	3.9%	79.9%	14.4%	5.6%
User 2	84.9%	6.7%	8.3%	97.3%	0.5%	2.0%

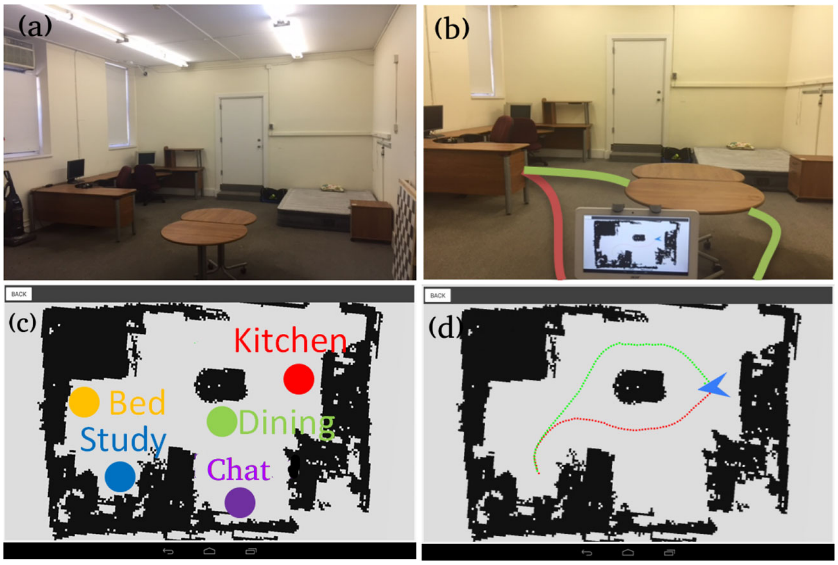


Fig. 13 Learning user preferences in an actual scene. (a) Studio for navigation, (b) paths between kitchen and study, (c) map of studio, (d) paths displayed on tablet

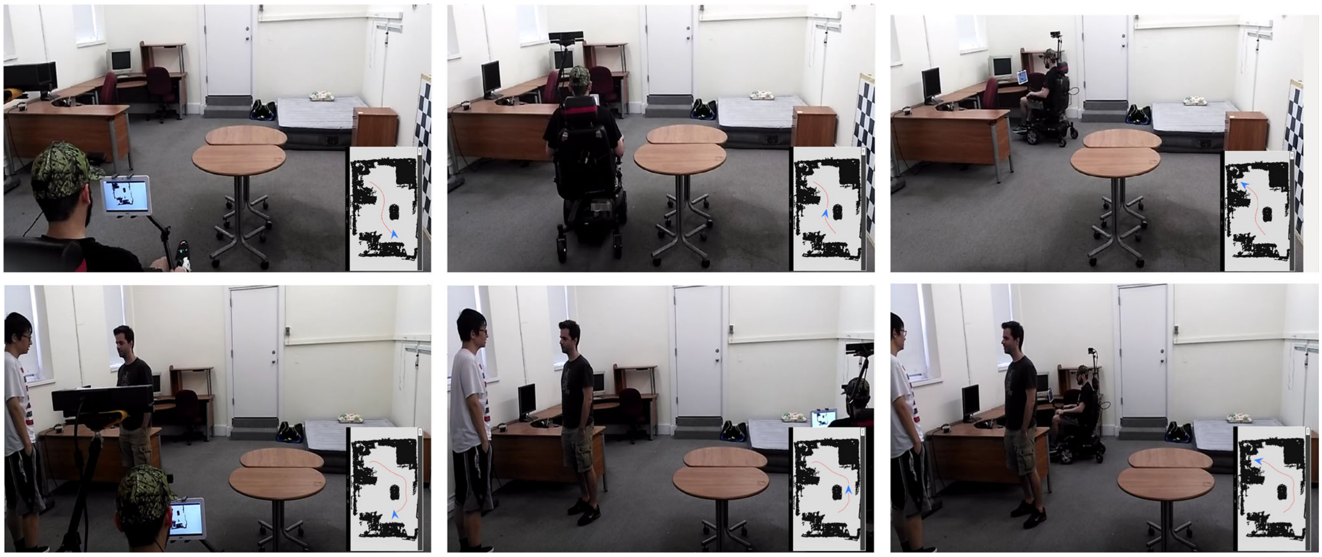
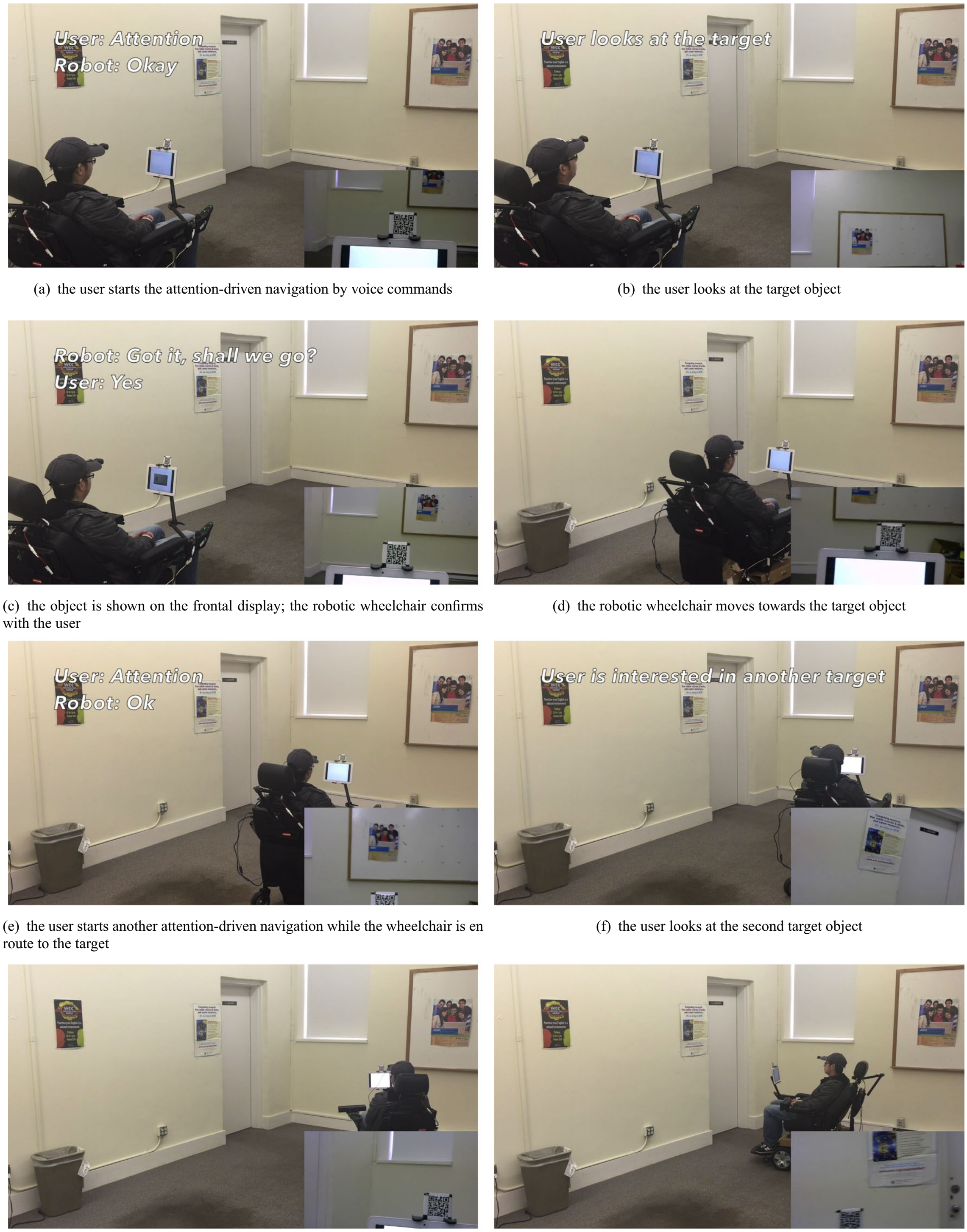


Fig. 14 Pictures from an experiment with the second subject. A static and a dynamic scene are shown at the top and bottom row respectively. The presence of people affect the subject's choice of paths, even though the people were not blocking any path



(a) the user starts the attention-driven navigation by voice commands

(b) the user looks at the target object

(c) the object is shown on the frontal display; the robotic wheelchair confirms with the user

(d) the robotic wheelchair moves towards the target object

(e) the user starts another attention-driven navigation while the wheelchair is en route to the target

(f) the user looks at the second target object

(g) the robotic wheelchair turns towards the direction the user looked at to localize the object

(h) the robotic wheelchair moves towards the new target

Fig. 15 Attention-driven navigation: the egocentric view is shown in the bottom right corner

7.2 Attention-driven Navigation

In this previously unpublished experiment, we demonstrate attention-driven navigation. The user initiates attention-driven navigation to the poster on the whiteboard by staring at it. While the robotic wheelchair moves towards the whiteboard, the user initiates another attention-driven navigation to a different poster. The robotic wheelchair

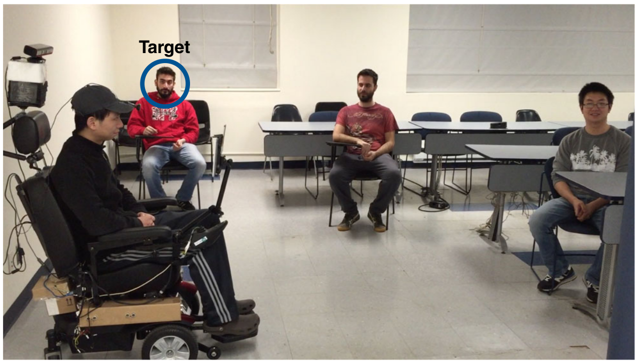


Fig. 16 An example of person-guided navigation. The user detects the person of interest to the left, which informs the wheelchair of the correct direction to turn. The face recognition task is then handed over from the egocentric camera to the RGB-D sensor

then changes the navigation target to the second poster. Prior knowledge about the posters is not required. We show images from one run in Fig. 15. Please see the supplemental material for a video of the full experiment here.

7.3 Person-guided Navigation

Person-guided navigation leverages the user's ability to locate objects or people in the scene to speed up and improve autonomous navigation. For example, the user wants to navigate to a person in a room and this person is not visible to the wheelchair from its current position. The user can indicate to the wheelchair the correct direction to rotate until the person enters the field of view of the RGB-D camera. The workflow for this example is as follow:

(a) the user searches for the target person in the room;

(b) the face detection and recognition system detects and recognizes the person from the video stream of egocentric camera;

(c) the relative pose of the wearable camera with respect to the wheelchair is used to determine the wheelchair's rotation direction;

(d) the wheelchair starts turning towards the person;

(e) when the person is detected and recognized in the video stream coming from the RGB-D camera, the wheelchair navigates to the person.

An example is shown in Fig. 16 and in the supplemental videos. In the experiment, as shown in Fig. 16, three people other than the wheelchair user are present in the scene, two of which are registered in the face detection and recognition system. The target person is not initially visible to the wheelchair RGB-D camera. The wheelchair user locates the target person with the egocentric wearable camera. Then, the wheelchair start turning leftward in-place until it is facing the target person. See Section 5.3 for more details.

7.4 People Following

In this experiment, a person is in front of the wheelchair. Once people following mode is activated, the wheelchair detects, tracks and follows the person in the scene. More details can be found in Section 5.4, while an example is shown in Fig. 17 and in the supplemental video.

8 Conclusion

We have presented a novel approach for wheelchair navigation using a wearable egocentric camera based on

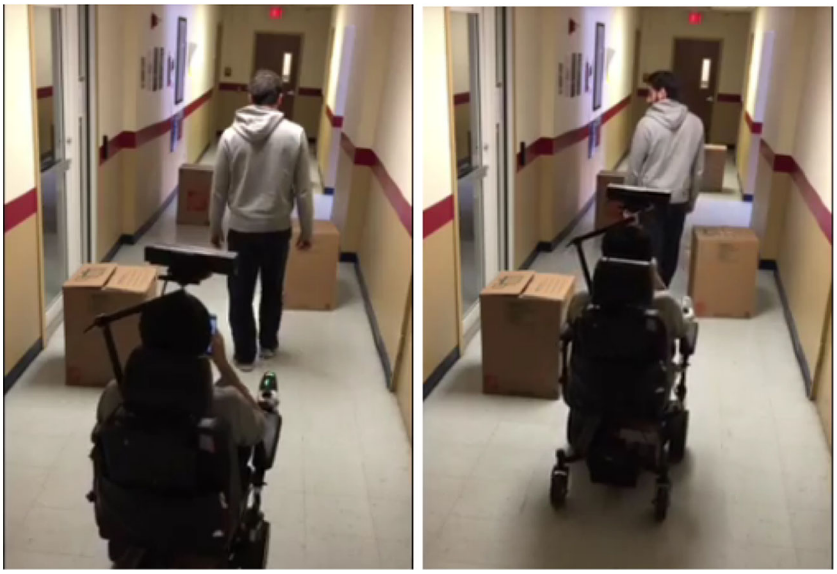


Fig. 17 An example of people following

computer vision technology. It allows hands-free operation of the wheelchair and gives the user the ability to control direction and speed in a continuous way. Multiple levels of autonomy are supported. The user can operate the wheelchair at a low level with direction and velocity commands, or can initiate complex maneuvers with single instructions. The viability and practicality of the system and its capabilities have been shown experimentally.

In addition, we have presented an approach for integrating user preferences in path planning, learning from the user's interaction with the system. These interactions occur either when users select paths in autonomous mode or drive the wheelchair manually. We designed and conducted experiments on both real and simulated platforms. Results in both cases demonstrate the system's ability to learn from relatively small amounts of user inputs. Encouraged by the success of learning in the simulator, we investigated whether we can train using simulated data only. In a separate effort [77], we showed promising results in a setting where we no longer train personalized navigation models for shared autonomy, but a global model for fully autonomous navigation. One promising direction for future work is better modelling of people in the simulator, since there is some discrepancy between their influence and that of real people on the robot.

The next step for this technology is broader testing. The contributions described in this paper were evaluated in user studies with healthy subjects. The results of these trials indicate that clinical tests of the proposed technology should be considered.

Acknowledgements Research reported in this publication was primarily supported by the National Institute of Nursing Research of the National Institutes of Health under Award Number R01NR015371 and also partially supported by the National Science Foundation under Awards IIS-1527294 and IIS-1637761. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health or the National Science Foundation. The main contributions of all authors took place while they were with Stevens Institute of Technology. Mohammed Kutbi prepared this manuscript while at the Saudi Electronic University.

Author Contributions Mohammed Kutbi prepared the manuscript and contributed to conception, design and implementation, Haoxiang Li contributed to conception, design and implementation, Yizhe Chang contributed to conception, design and implementation, Bo sun contributed to design and implementation, Xin Li contributed to implementation, Changjiang Cai contributed to implementation, Nikolaos Agadagos contributed to implementation, Gang Hua contributed to conception, and Philippos Mordohai contributed to conception and manuscript preparation.

Data Availability Data are not publicly available to preserve the subjects' privacy.

Code Availability Our code is not available because it requires a custom modification of a particular wheelchair, and thus is not of broad interest. It can be made available upon request.

Declarations

Ethics approval All studies involving non-researchers as participants received IRB approval at Stevens Institute of Technology.

Consent to participate Informed consent was obtained from all individual participants included in the study.

Consent for Publication No images of subjects are shown in the paper. All individuals depicted are authors of the manuscript.

Conflict of Interests The authors have no relevant financial or non-financial interests to disclose.

References

1. Organization, W.H.: Assistive technology, fact sheet (2016)
2. Shields, M.: Use of wheelchairs and other mobility support devices. Health Rep. **15**, 37—41 (2004)
3. Ceres, R., Pons, J.L., Calderon, L., Jimenez, A.R., Azevedo, L.: A robotic vehicle for disabled children. IEEE Engineering in Medicine and Biology magazine **24**(6), 55–63 (2005)
4. Fehr, L., Langbein, W.E., Skaar, S.B.: Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey. J. Rehabil. Res. Dev. **37**(3), 353 (2000)
5. Simpson, R.C., LoPresti, E.F., Cooper, R.A.: How many people would benefit from a smart wheelchair? J. Rehabil. Res. Dev. **45**(1), 53 (2008)
6. Kutbi, M., Du, X., Chang, Y., Sun, B., Agadakos, N., Li, H., Hua, G., Mordohai, P.: Usability studies of an egocentric vision-based robotic wheelchair. ACM Transactions on Human-Robot Interaction (THRI) **10**(1), 1–23 (2020)
7. Li, H., Kutbi, M., Li, X., Cai, C., Mordohai, P., Hua, G.: An egocentric computer vision based co-robot wheelchair. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2016)
8. Wang, M., Deng, W.: Deep face recognition: a survey. Neurocomputing **429**, 215–244 (2021)
9. Chen, S.-H., Chen, Y.-L., Chiou, Y.-H., Tsai, J.-C., Kuo, T.-S.: Head-controlled device with M3S-based for people with disabilities. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), vol. 2, pp. 1587–1589 (2003)
10. Simpson, R.C.: Smart wheelchairs: A literature review. J. Rehabil. Res. Dev. **42**(4), 423 (2005)
11. Carlson, T., Leeb, R., Chavarriaga, R., Millán, J.D.R.: The Birth of the Brain-Controlled Wheelchair. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5444–5445 (2012)

12. Sisbot, E.A., Marin-Urias, L.F., Alami, R., Simeon, T.: A human aware mobile robot motion planner. IEEE Trans. Robot. **23**(5), 874–883 (2007)
13. Chang, Y., Kutbi, M., Agadakos, N., Sun, B., Mordohai, P.: A shared autonomy approach for wheelchair navigation based on learned user preferences. In: Fifth International Workshop on Assistive Computer Vision and Robotics, pp. 1490–1499 (2017)
14. Cook, A.M., Polgar, J.M.: Assistive technologies: Principles and practice. Elsevier Health Sciences (2014)
15. Lipskin, R.: An evaluation program for powered wheelchair control systems. Bull. Prosth. Res **6**, 121–219 (1970)
16. Nguyen, H., King, L., Knight, G.: Real-time head movement system and embedded linux implementation for the control of power wheelchairs. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), vol. 2 (2004)
17. Lund, M.E., Christiensen, H.V., Caltenco, H., Lontis, E.R., Bentsen, B., Struijk, L.N.A.: Inductive tongue control of powered wheelchairs. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (2010)
18. Mace, M., Abdullah-Al-Mamun, K., Vaidyanathan, R., Wang, S., Gupta, L.: Real-time implementation of a non-invasive tongue-based human-robot interface. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2010)
19. Lopes, A.C., Pires, G., Nunes, U.: Robchair: Experiments evaluating brain-computer interface to steer a semi-autonomous wheelchair. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2012)
20. Carlson, T., Millan, J.D.R.: Brain-controlled wheelchairs: a robotic architecture. IEEE Robot. Autom. Mag. **20**, 65–73 (2013)
21. Boucher, P., Atrash, A., Kelouwani, S., Honoré, W., Nguyen, H., Villemure, J., Routhier, F., Cohen, P., Demers, L., Forget, R., Pineau, J.: Design and validation of an intelligent wheelchair towards a clinically-functional outcome. J. Neuroeng. Rehabil. **10**(1), 1 (2013)
22. Fezari, M., Bousbia-Salah, M.: Speech and sensor in guiding an electric wheelchair. Autom. Control. Comput. Sci. **41**(1), 39–43 (2007)
23. Nishimori, M., Saitoh, T., Konishi, R.: Voice Controlled Intelligent Wheelchair. In: SICE (2007)
24. Gandhi, V., Prasad, G., Coyle, D., Behera, L., McGinnity, T.M.: EEG-Based mobile robot control through an adaptive brain–robot interface. IEEE Transactions on Systems Man, and Cybernetics: Systems, **44**(9), 1278–1285 (2014)
25. Liu, Y., Li, Z., Zhang, T., Zhao, S.: Brain-Robot interface-based navigation control of a mobile robot in corridor environments. IEEE Transactions on Systems, Man, and Cybernetics Systems (2018)
26. Quintero, C.P., Ramirez, O., Jagersand, M.: Vibi: Assistive vision-based interface for robot manipulation. In: Proc. of the IEEE International Conference on Robotics and Automation (2015)
27. Pasteau, F., Krupa, A., Babel, M.: Vision-based assistance for wheelchair navigation along corridors. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2014)
28. Purwanto, D., Mardiyanto, R., Arai, K.: Electric wheelchair control with gaze direction and eye blinking. Artificial Life and Robotics **14**(3), 397–400 (2009)
29. Gray, J., Jia, P., Hu, H.H., Lu, T., Yuan, K.: Head gesture recognition for hands-free control of an intelligent wheelchair. Industrial Robot: An International Journal **34**(1), 60–68 (2007)
30. Xu, R., Hartshorn, R., Huard, R., Irwin, J., Johnson, K., Nelson, G., Campbell, J., Ay, S.A., Taylor, M.E.: Towards a semi-autonomous wheelchair for users with ALS. In: Workshop on Autonomous Mobile Service Robots (At IJCAI) (2016)
31. Halawani, A., Ur Réhman, S., Li, H., Anani, A.: Active vision for controlling an electric wheelchair. Intell. Serv. Robot. **5**(2), 89–98 (2012)
32. Kim, B.K., Tanaka, H., Sumi, Y.: Robotic wheelchair using a high accuracy visual marker lentibar and its application to door crossing navigation. In: Proc. of the IEEE International Conference on Robotics and Automation (2015)
33. Zolotas, M., Elsdon, J., Demiris, Y.: Head-mounted augmented reality for explainable robotic wheelchair assistance. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1823–1829 (2018)
34. Chacón-Quesada, R., Demiris, Y.: Augmented reality control of smart wheelchair using eye-gaze–enabled selection of affordances. In: Robots for Assisted Living Workshop (2018)
35. Gulati, S., Jhurani, C., Kuipers, B., Longoria, R.: A framework for planning comfortable and customizable motion of an assistive mobile robot. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4253–4260 (2009)
36. Shiomi, M., Iio, T., Kamei, K., Sharma, C., Hagita, N.: Effectiveness of social behaviors for autonomous wheelchair robot to support elderly people in Japan. PloS one, vol. **5**, 10 (2015)
37. Parikh, S.P., Grassi, V., Kumar, V., Okamoto, J.: Incorporating user inputs in motion planning for a smart wheelchair. In: Proc. of the IEEE International Conference on Robotics and Automation, vol. 2, pp. 2043–2048 (2004)
38. Parikh, S.P., Grassi, V., Kumar, V., Okamoto, J.J.: Usability study of a control framework for an intelligent wheelchair. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 4745–4750 (2005)
39. Zeng, Q., Teo, C.L., Rebsamen, B., Burdet, E.: A collaborative wheelchair system. IEEE Trans. Neural Syst. Rehabil. Eng. **16**(2), 161–170 (2008)
40. Galindo, C., Gonzalez, J., Fernandez-Madrigal, J.-A.: Control architecture for human–robot integration: application to a robotic wheelchair. IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics) **36**(5), 1053–1067 (2006)
41. Urdiales, C., Peula, J.M., Fdez-Carmona, M., Barrué, C., Pérez, E.J., Sánchez-Tato, I., del Toro, J.C., Galluppi, F., Cortés, U., Annichiaricco, R., Caltagirone, C., Sandoval, F.: A new multi-criteria optimization strategy for shared control in wheelchair assisted navigation. Auton. Robot. **30**(2), 179–197 (2011)
42. Li, Q., Chen, W., Wang, J.: Dynamic shared control for human-wheelchair cooperation. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 4278–4283 (2011)
43. Carlson, T., Demiris, Y.: Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload. IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics) **42**(3), 876–888 (2012)
44. Goil, A., Derry, M., Argall, B.D.: Using machine learning to blend human and robot controls for assisted wheelchair navigation. In: IEEE International Conference on Rehabilitation Robotics (ICORR) (2013)
45. Kruse, T., Pandey, A.K., Alami, R., Kirsch, A.: Human-aware robot navigation: a survey. Robot. Auton. Syst. **61**(12), 1726–1743 (2013)
46. Kirby, R., Simmons, R., Forlizzi, J.: Companion: a Constraint-Optimizing Method for Person-Acceptable Navigation. In: RO-MAN, pp. 607–612 (2009)
47. Cosgun, A., Sisbot, A., Christensen, H.I.: Anticipatory robot path planning in human environments. In: The 25Th IEEE International

Symposium on Robot and Human Interactive Communication, pp. 562–569 (2016)
48. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**(5), 4282 (1995)
49. Morales, Y., Miyashita, T., Hagita, N.: Social robotic wheelchair centered on passenger and pedestrian comfort. Robot. Auton. Syst. **87**, 355–362 (2017)
50. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: An open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
51. Labbe, M., Michaud, F.: Online global loop closure detection for large-scale multi-session graph-based SLAM. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2661–2666 (2014)
52. Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B., Konolige, K.: The office marathon: robust navigation in an indoor office environment. In: International Conference on Robotics and Automation (2010)
53. Marchand, E., Spindler, F., Chaumette, F.: Visp for visual servoing: a generic software platform with a wide class of robot control skills. IEEE Robotics & Automation Magazine **12**(4), 40–52 (2005)
54. Nebehay, G., Pflugfelder, R.: Clustering of static-adaptive correspondences for deformable object tracking. In: CVPR (2015)
55. Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., Wolf, P., Woelfel, J.: Sphinx-4: A flexible open source framework for speech recognition (2004)
56. Suzuki, S., Abe, K.: Topological structural analysis of digitized binary images by border following, Computer Vision. Graphics, and Image Processing **30**(1), 32–46 (1985)
57. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Commun ACM (1981)
58. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: ECCV, pp. 430–443, Springer (2006)
59. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to sift or surf. In: ICCV (2011)
60. Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: CVPR (2015)
61. Li, H., Hua, G., Lin, Z., Brandt, J., Yang, J.: Probabilistic elastic matching for pose variant face verification. In: CVPR (2013)
62. Nebehay, G., Pflugfelder, R.: Consensus-based matching and tracking of keypoints for object tracking. In: IEEE Winter Conference on Applications of Computer Vision (WACV) (2014)
63. Linder, T., Arras, K.O.: People detection, tracking and visualization using ros on a mobile service robot. In: Robot Operating System (ROS), pp. 187–213, Springer (2016)
64. Linder, T., Breuers, S., Leibe, B., Arras, K.O.: On multi-modal people tracking from mobile platforms in very crowded and dynamic environments. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 5512–5519 (2016)
65. Jafari, O.H., Mitzel, D., Leibe, B.: Real-Time Rgb-D based people detection and tracking for mobile robots and head-worn cameras. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5636–5643, IEEE (2014)
66. Ess, A., Schindler, K., Leibe, B., Van Gool, L.: Object detection and tracking for autonomous navigation in dynamic environments. The International Journal of Robotics Research **29**(14), 1707–1725 (2010)
67. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
68. Kuderer, M., Sprunk, C., Kretzschmar, H., Burgard, W.: Online generation of homotopically distinct navigation paths. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 6462–6467 (2014)
69. Bhattacharya, S., Likhachev, M., Kumar, V.: Topological constraints in search-based robot path planning. Auton. Robot. **33**(3), 273–290 (2012)
70. Mandel, C., Frese, U.: Comparison of wheelchair user interfaces for the paralysed: head-joystick vs. verbal path selection from an offered route-set. In: EMCR (2007)
71. DIJKSTRA, E.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
72. Choset, H., Burdick, J.: Sensor-based exploration: The hierarchical generalized voronoi graph. The International Journal of Robotics Research **19**(2), 96–125 (2000)
73. Lau, B., Sprunk, C., Burgard, W.: Efficient grid-based spatial representations for robot navigation in dynamic environments. Robot. Auton. Syst. **61**(10), 1116–1130 (2013)
74. Herbrich, R., Graepel, T., Obermayer, K.: Support vector learning for ordinal regression. International Conference on Neural Networks (1999)
75. Park, J.J., Kuipers, B.: A smooth control law for graceful motion of differential wheeled mobile robots in 2D environment. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 4896–4902 (2011)
76. Jain, A., Das, D., Gupta, J.K., Saxena, A.: Planit: A crowdsourcing approach for learning to plan paths from large scale preference feedback. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 877–884 (2015)
77. Kutbi, M., Chang, Y., Sun, B., Mordohai, P.: Learning to navigate robotic wheelchairs from demonstration: is training in simulation viable? In: Seventh International Workshop on Assistive Computer Vision and Robotics (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mohammed Kutbi is an assistant professor and Chair of Information Technology Department at Saudi Electronic University in Jeddah, Saudi Arabia. He received his M.S. and Ph.D. degrees in Computer Science from Stevens Institute of Technology in 2014 and 2018, respectively. He received his B.S. degree in Information Technology from King Abdulaziz University in Jeddah, Saudi Arabia. His research interests include robotics, egocentric vision and motion segmentation.

Haoxiang Li received the BS degree in computer science from the University of Science and Technology of China, in 2010. He received the PhD degree from the Department of Computer Science at Stevens Institute of Technology, in 2016. He is currently a senior researcher with Wormpex AI Research. Before that, he was a Researcher with AiBee and Adobe Research. His research interests include human face processing and analysis, computer vision, and robotics. He holds 7 US patents.

Yizhe Chang is an assistant professor in the Mechanical Engineering Department, California State Polytechnic University, Pomona. He received his M.S. and Ph.D. degrees in mechanical engineering from Stevens Institute of Technology in 2012 and 2016 respectively. He received his B.S. degree in Instrumentation technology from Tianjin University, Tianjin, China. His research focuses on robotics, especially human-robot interaction, in physical and simulation scenarios.

Bo Sun is a researcher at Adobe Inc. He earned his PhD from the Stevens Institute of Technology and earned his Bachelor and Master degree from Nankai University, China. His research interests span 3D reconstruction from images, robotics and panoptic segmentation.

Xin Li is now the Director of Operation Strategy at Huolala. Before that, he was the Director of Applied AI at HelloBike, a Senior Staff Engineer at ELEME, and a Senior Research Engineer at Didi Research. He received his B.E. in Software Engineering from South China University of Technology and M.S. in Computer Science from Stevens Institute of Technology.

Changjiang Cai is a senior research scientist at InnoPeak Technology (aka OPPO US Research Center). He earned his Ph.D. in Computer Science and M.S. degree in Electrical Engineering from Stevens Institute of Technology, US, and received the B.E. degree in Mechanical Engineering from Northwestern Polytechnical University, China. His research interests focus on 3D computer vision, representation learning and human pose estimation.

Nikolaos Agadakos is a PhD student in the University of Illinois at Chicago. He has worked as an intern Machine Learning research scientist in Stanford Research Institute (SRI). He was the recipient of the Peter and Deborah Wexler Graduate Student Award Scholarship, in 2019. His research interests are in Computer Vision, Adversarial and Robust Learning, and Deep Learning. He received his Engineering Diploma and M.S. in Computer and Electronics Engineering in 2012 and 2015 from the Technical University of Crete, Greece and his M.S. from Stevens Institute of Technology in 2018, USA.

Gang Hua (M'03-SM'11) was enrolled in the Special Class for the Gifted Young of Xi'an Jiaotong University (XJTU), in 1994, and received the BS degree in automatic control engineering from XJTU, in 1999, the MS degree in control science and engineering from XJTU, in 2002, and the PhD degree in electrical engineering and computer science from Northwestern University, in 2006. Gang Hua is the Vice President and Chief Scientist of Wormpex AI Research. His research focuses on computer vision, pattern recognition, machine learning, robotics, towards general Artificial Intelligence, with primary applications in cloud and edge intelligence, and currently with a focus on new retail intelligence. Before that, he served in various roles at Microsoft (2015-18) as the Chief Science/Technical Adviser to the CVP of the Computer Vision Group, Director of Computer Vision Science Team in Redmond and Taipei ATL, and Senior Principal Researcher/Research Manager at Microsoft Research. He was an Associate Professor at Stevens Institute of Technology (2011-15). During 2014-15, he took a leave from Stevens and worked on the Amazon-Go project. He was a Visiting Researcher (2011-14) and a Research Staff Member (2010-11) at IBM Research T. J. Watson Center, a Senior Researcher (2009-10) at Nokia Research Center Hollywood, and a Senior Scientist (2006-09) at Microsoft Live labs Research. He has published more than 160 peer reviewed papers in top conferences and journals. To date, he holds 19 US patents and has 15 more patents pending. He is an IEEE Fellow, an IAPR Fellow, and an ACM Distinguished Scientist. He is the recipient of the 2015 IAPR Young Biometrics Investigator Award.

Philippos Mordohai is a professor of Computer Science at Stevens Institute of Technology. He earned his PhD from the University of Southern California and held postdoctoral appointments at the University of North Carolina and the University of Pennsylvania. His research interests span 3D reconstruction from images and video, range data analysis, perceptual organization and active vision. He has served as area chair for CVPR, ICCV and ECCV and program co-chair of the International Conference on 3D Vision (3DV), 2019.

Affiliations

Mohammed Kutbi[1] · **Haoxiang Li**[2] · **Yizhe Chang**[3] · **Bo Sun**[4] · **Xin Li**[5] · **Changjiang Cai**[6] · **Nikolaos Agadakos**[7] · **Gang Hua**[2] · **Philippos Mordohai**[8]

Haoxiang Li
lhxustcer@gmail.com

Yizhe Chang
chang@cpp.edu

Bo Sun
bosu@adobe.com

Xin Li
li.xin.cv@gmail.com

Changjiang Cai
ccai1@stevens.edu

Nikolaos Agadakos
nagadakos@googlemail.com

Gang Hua
ganghua@gmail.com

Philippos Mordohai
pmordoha@stevens.edu

[1] Saudi Electronic University,
Jeddah, 23442, Makkah, Saudi Arabia

[2] Wormpex AI Research, Bellevue, 98004, WA, USA

[3] California State Polytechnique University,
Pomona, 91768, CA, USA

[4] Adobe Inc, San Jose, 95110, CA, USA

[5] Huolala, Shenzhen, China

[6] Innopeak Technology, Palo Alto, 94303, CA, USA

[7] University of Illinois, Chicago, 60605, IL, USA

[8] Stevens Institute of Technology, Hoboken, 07030, NJ, USA