Vision-Based Assistive Navigation for Robotic Wheelchair Platforms

P.E. Trahanias, M.I.A. Lourakis, A.A. Argyros and S.C. Orphanoudakis

Institute of Computer Science
Foundation for Research and Technology – Hellas
P.O.Box 1385, Heraklion, 711 10 Crete, Greece
and
Department of Computer Science, University of Crete
P.O.Box 1470, Heraklion, 714 09 Crete, Greece
e-mail:{trahania,lourakis,argyros,orphanou}@ics.forth.gr

Abstract: *In this paper we present an approach towards providing advanced navigational capabilities to robotic wheelchair platforms. Contemporary methods that are employed in robotic wheelchairs are based on the information provided by range sensors and its appropriate exploitation by means of* obstacle avoidance *techniques. However, since range sensors cannot support a detailed environment representation, these methods fail to provide advanced navigational assistance, unless the environment is appropriately regulated (e.g. with the introduction of beacons). In order to avoid any modifications to the environment, we propose an alternative approach that employs computer vision techniques which facilitate space perception and navigation. Computer vision has not been introduced todate in rehabilitation robotics, since the former is not mature and reliable enough to meet the needs of this sensitive application. However, in the proposed approach, stable techniques are exploited that facilitate reliable, automatic navigation to any point in the visible environment. This greatly enhances the mobility of the elderly and disabled, without requiring them to exercise fine motor control. Preliminary results obtained from the implementation of this approach on a laboratory robotic platform indicate its usefulness and flexibility.*

1 Introduction

Current advances in robotics have facilitated the introduction of related technologies in many application areas, such as surveillance systems, autonomous vehicles, delivery robots and cleaning machines [1, 2]. A distinctive and very important application sector is that of rehabilitation robotics [3]. The latter has been greatly advanced through the introduction of flexible manipulators (actuators), mobile platforms, fixed robotic workstations and sensor devices. Such hardware components have supplied rehabilitation robots with capabilities such as (manually-operated) manipulation, handling and low level navigation. The navigational capabilities are typically based on range (sonar, laser) and/or proximity (infrared) sensor measurements, with the former indicating the distance of environment objects from the robotic platform and the latter signaling the presence of objects close to it [4].

*This work was partially supported by the General Secretariat for Research and Technology, Greece, under Grant No. 6060. The authors wish to acknowledge the staff of the Computer Vision and Robotics Laboratory of ICS-FORTH for their help in developing the proposed application on TALOS –the laboratory's mobile robotic platform– as well as in the experiments conducted.

In this work we are interested in developing navigational capabilities for robotic wheelchairs, that will provide mobility and navigational assistance to elderly and disabled users. More specifically, we aim at enhancing current robotic wheelchairs by supplying them with the capability of automatic "targeted navigation" (move to *that* point). Therefore, a robust and effective navigation approach is required, that will work synergistically with motor impaired users, ultimately providing assistive navigation in uncontrolled environments. In the remainder of this section, the navigational capabilities and related shortcomings of current robotic wheelchairs are shortly presented, followed by a brief overview of the proposed enhancements.

Sonars, in the form of a ring [5], are the most commonly used type of range sensor in current generation robotic wheelchairs. In addition to being cheap, they provide measurements that can be directly interpreted with minimal processing overhead. Thus, they are very well suited to low level navigation tasks such as obstacle avoidance [6, 5, 7, 8]. There are, however, certain limits to what can be achieved by using only local range measurements. Although they provide information that can easily support reactivity to local environment features, they are inadequate for the more demanding needs of autonomous navigation. This stems from the fact that local range information constitutes a very restricted environment representation.

To overcome the above problem, various methods that make use of environment maps have appeared in the literature [9, 10, 11, 12, 13]. Such methods rely either on *a priori* knowledge or on an initial *learning* phase for obtaining a map of the environment, which is constantly updated by the local sonar measurements. Whenever movement to a particular location in the environment is desired, motion planning methods such as those described in [14, 15, 16] are employed to yield an optimal[1] path from the starting configuration (position and orientation) to the target one. Tracking of the established path is then accomplished by some sort of odometric (dead-reckoning) technique. A non-trivial problem encountered in these approaches is that path following is not error-free [17]. Due to imperfect environment representations, sonar inaccuracies (specular reflections, crosstalks, e.t.c.) and drifts of the robot motors, errors are usually introduced during this phase. Unfortunately, these errors accumulate in the whole process and may result in completely erroneous motion. A major flaw, inherent in these navigation approaches, which also shares responsibility in the above, is the lack of any intermediate information concerning localization of the robot with respect to some known features of the environment. Although there have been some efforts towards fully automatic localization (e.g. [18]), they are expected to perform poorly in environments cluttered with obstacles.

To deal with the localization problem, radio beacons [19, 20, 21] or easily recognized patterns [22, 23] are usually introduced. Beacons, combined with planning techniques that take uncertainty into account [17, 24, 25], facilitate accurate localization and, therefore, path following. Such environment modifications are, however, very restricting since they confine robot roving in a predefined space. Especially in the case of robotic wheelchair platforms, where the goal is to support user mobility in various environments, beacon-based approaches do not seem very appropriate. Even in the case that beacons are employed, the assignment of semantic information to individual environment segments and/or objects can not be supported, due to lack of any relevant information. In other words, "targeted navigation" cannot be supported, although highly desired in order to increase the autonomy of the robotic wheelchair.

Vision, being a more powerful sense, can be employed towards this end. Indeed, rich information regarding the environment can be extracted from images. Moreover, it is fairly straightforward to assign semantic content to this information. On the other hand, one cannot neglect the fact that most computer vision techniques are not yet reliable enough for coping with the uncertainty and unpredictability of the real world. It should also be noted that, relying on computer vision alone for navigation, may incur prohibitively high computational costs. Thus, the proposed approach is semi-automatic, relying on the user for some decisions that are hard to make automatically, and trying to combine some of the advantages of range sensing and vision by fusing information

[1] Optimality is meant here according to a set of criteria that are imposed on the path, i.e. total length of the path, minimum distance of the robot to the environment objects, etc.

acquired by a sonar ring and a camera. Computer vision techniques are involved for target tracking; sonar-based reactivity is employed for local, fine control of motion. More specifically, the camera "locks" on a user-selected target by fixating it throughout a navigation session, while at the same time the sonars are checking for obstacles that may be in the wheelchair's course. Whenever the wheelchair completes a detour for avoiding an obstacle, the camera instructs it to move in the direction of the target and approach the desired destination. For the sake of reliability, higher level, cognitive procedures (e.g. environment-feature recognition, scene interpretation) are not employed, since current methods that address them are not very accurate. As will be shown later, the exclusion of higher level capabilities contributes towards a robust approach, whereas the visual capabilities introduced facilitate an environment representation that is quite appropriate for the task addressed. This is the crux of this work, which integrates existing, robust methods for achieving accurate (targeted) navigation in uncontrolled environments.

Computer vision has not been introduced todate in rehabilitation robotics, since the former is not mature enough to meet the needs of this sensitive application. In the proposed approach, however, proven techniques are exploited that facilitate reliable, automatic navigation to any point in the visible environment. In the rest of this paper, the proposed approach is elaborated and preliminary results from its implementation are presented. Section 2 describes in detail the proposed navigation approach and Section 3 focuses particularly on its adaptation to robotic wheelchair platforms. The implementation of this approach on a laboratory experimental platform is presented in Section 4 and results obtained from our experiments are also given. Section 5 discusses issues regarding platform training in the user's environment and presents possible extensions of this approach, giving also directions for their implementation. Finally, Section 6 concludes the paper with a brief discussion.

2 Platform Navigation

As already mentioned, current approaches to automatic navigation employ beacons and environment maps [15, 20]. This limits their usability, since they are confined to specific environments and, even then, detailed maps are difficult to construct and maintain. To compensate for that, in our approach we completely avoid space charting. Instead, we introduce a deictic, visual representation of the target pattern [26] which the user wants to reach. More specifically, we do not employ any kind of workspace maps but use visual images to represent the viewed scene. Since this representation is at the lowest possible level (image level), it does not introduce any errors as would be the case with higher level (cognitive) representations. On the other side, however, it can not support *fully* automatic navigation since it is lacking a detailed model of the viewed scene. To overcome this handicap, the target selection process is entrusted to the operator; in other words, our approach simply ignores the issue of platform localization (including also determination of the initial robot configuration) and lets the user pick-up a desired environment pattern from an image of the viewed scene; the selected pattern constitutes the target position of the system.

In order to reach the selected target, we employ a *visual fixation capability* and a *hierarchical robot motion-planner*. Visual fixation is triggered by the target selection and is maintained throughout the whole navigation process. The motion planner operates in two levels. At the higher level, the *global planner* consults the fixation module and commands the motion of the robotic platform towards the direction pointed to by the vision system, i.e. the platform moves in a straight path towards the target. At a lower level, the *tactical planner* controls local, fine platform motion. To achieve this, it constantly checks for environment obstacles in the direction of motion using the platform sensors. In case that an obstacle is encountered, the tactical planner takes over control temporarily from the global planner and performs a motion to avoid the obstacle. During this motion, the global planner is idle; visual fixation is, however, active by tracking the icon of the target. When the obstacle is no longer detectable from the platform's sensors, the global planner resumes control of the platform and redirects its motion towards the target. The above procedure is performed in a closed loop until the target is reached, yielding assistive navigational capabilities to the platform. The two level scheme just described can also be considered as a realization of the subsumption architecture proposed by Brooks [27]. According to the latter, a hierarchy of concurrent processes are pursuing a common goal, with processes higher in the hierarchy having more general subgoals and lower priority processes inhibiting higher level ones whenever dynamic changes occur in the environment. Figure 1 presents an algorithm in pseudocode that implements the steps described above; technical issues involved in that are elaborated in the following sections.

---

0. *Select* Visual_Target /* Target selection in the viewed scene */
1. *Continuously Fixate* on Visual_Target /* Active throughout the whole navigation session */
2. *Invoke* Global_Planner
    2a. *Initiate* Robot_Motion Towards Visual_Target /* Start moving towards target */
    2b. **While** VisualTargetNotReached **do**
        Acquire Sensor_Measurements
        Check For_Obstacles
        If ObstacleFound
            Invoke Tactical_Planner /* Avoid obstacles cutting the platform's path */
            *Avoid*_Obstacle
            Resume Robot_Motion Towards Visual_Target /* After obstacle avoidance
                is completed, platform motion is again redirected towards target */
        EndIf
    EndWhile
3. **End.**

Figure 1: Algorithmic implementation of the proposed approach for assistive navigation.

2.1 Environment Representation

An accurate environment representation requires advanced perceptual capabilities that are very difficult to be implemented in practice. Moreover, any information about the external world is inherently characterized with "uncertainty" and "inaccuracy", due to the measurement process involved. Therefore, many navigation approaches reject any environment representations [28, 12], regarding the real world as a *representation of its own* [29]. These methods are very well suited for reactive behaviors (e.g. collision avoidance); however, they suffer from the fact that they lack a general (global) model of the workspace, which makes them inappropriate for more complex navigation scenarios.

Motivated by recent works on *deictic representations* [30, 31, 26], the proposed navigation approach employs them as an effective model for the external world. A deictic representation is initiated by a higher level process; for the sake of reliability and robustness of the approach, this is performed by the user, who selects a target in the viewed scene using appropriate man-machine interfaces.

Upon selection of a visual target, a *marker* M is *bound* to it [26]. This binding is kept permanent for the period of the navigation session, i.e. until the selected target is reached. Markers, a central concept in deictic representations around which perception and action revolve, can be thought of as *pointers* to environment objects. When a marker is set to point to an object, it registers the object's features and initiates specific action(s). Formally, we define a marker M as

$$M(F, A)\ ,\ F \in S_F\ ,\ A \in S_A$$

where, S_F denotes the set of object feature values and S_A denotes the set of associated actions. The issue of feature values is deferred until the next subsection where target tracking is discussed. In our model, S_A has only *one* member; in other words, regardless of the target selection (and the registered feature values), the same action is always invoked. The latter consists of a two-step procedure, corresponding exactly to the two steps 1 and 2 of the algorithm presented in Fig 1. More specifically, S_A is defined as

$$S_A = \{Continuously\ Fixate \text{ on Visual\_Target ; } Invoke \text{ Global\_Planner}\}$$

Intuitively, the above definition associates with each user selection one, twofold system action: (a) A fixation on the target which is active at all times during a navigation session, and (b) invocation of the global planner which commands platform motion towards the target, until the latter is reached.

The above described representation, although rudimentary, is quite adequate for the needs of the application addressed. Moreover, since it remains at the low, image level for all computations involved, it is quite robust and not susceptible to ambiguities introduced by object- or scene-based representations [32, 33].

2.2 Visual Target Fixation and Tracking

The deictic representation of the target described above facilitates visual fixation and tracking of the target. Since the camera is moving due to the motion of the robotic platform, this is actually the case of a moving observer tracking a target that can be either stationary or moving. This problem has been studied by various researchers, using mainly predictive techniques [34]. However, these techniques have considerable computational requirements and are, therefore, inappropriate for our case where real time performance is required. Consequently, we have adopted a "template-based" approach, where a template is used to represent the target selected and tracking is achieved by a "template-pursuit" technique.

More specifically, user pick-up of a target triggers the construction of a template, which covers an *area of interest* around the target. A region growing [35] technique is employed at this point to delineate the area of interest. This starts with a small window centered at the point of user selection, which is used as a "seed" for subsequent growing. A criterion is used to determine the bounds where window expansion should terminate. This criterion is based on the distribution of the color histogram. The contents of set S_F, introduced previously in section 2.1, can now be defined. For a particular object (template), they consist of a representation of the distribution of the object's color histogram. In other words, the object features used for pattern matching refer to the latter distribution. As shown in [36, 37], color distributions can be used to index objects with a good accuracy. On the other side, the color distribution does not change significantly when computed in sub-parts of an object. Based on this observation, growing of the seed template is terminated when, at a certain point, the color histogram change, with respect to the initial one, is above a predefined threshold.

Following template construction, fixation of the vision system to the selected target is performed. For the case of an intrinsically calibrated camera[2], this is easily achieved by *pan* and *tilt* motions with angles ϕ_p and ϕ_t, respectively, given as

$$\tan\phi_p = \frac{x}{f} \quad , \quad \tan\phi_t = \frac{y}{f}$$

where, f is the focal length of the camera and (x, y) are the image coordinates of the center of the constructed template. The above is illustrated schematically in Fig 2. When f is not known, iterative fixation procedures (feedback control loop) can be employed for this task [38, 39].

Target fixation is maintained during the course of platform motion by tracking the template across subsequent frames. Template tracking is accomplished by minimizing a sum of squared differences criterion. Since the color

[2] Actually, for the task of target fixation, only the camera focal length f is needed.

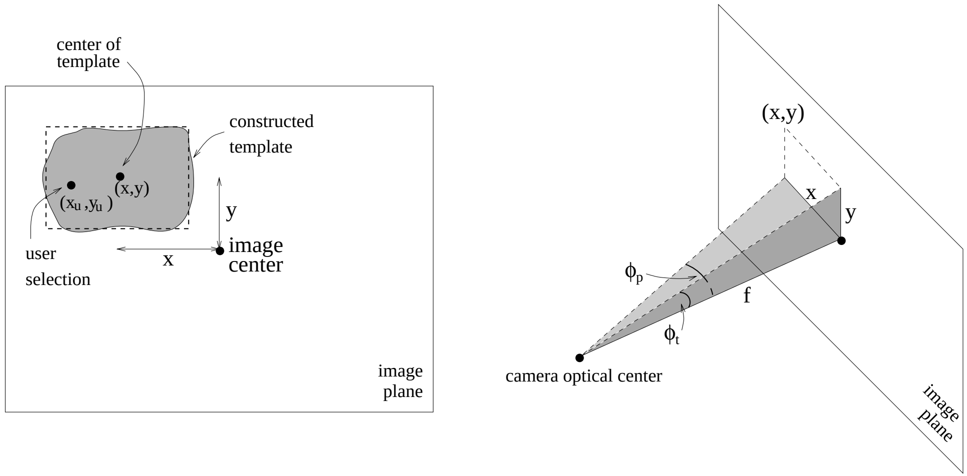


Figure 2: Angles ϕ_p and ϕ_t used to perform target fixation.

histogram exhibits invariance under camera transformations [37], template tracking can be effectively achieved in neighboring frames. In order to avoid template comparisons between distant (in time) image frames, the current (stored) template is continuously updated. This involves replacement of the template with the one that constitutes the best match in the next frame.

Visual fixation on the target results in a direction D_v, pointed to by the vision system. D_v is completely determined by ϕ_p and ϕ_t. This is subsequently utilized for guiding the motion of the robotic platform which is performed on the horizontal, 2D plane. Therefore, D_v is simply computed on the horizontal plane, i.e. it is set equal to ϕ_p.

2.3 Navigation Approach

Using the fixation capability described above, platform navigation is implemented in two levels. At the higher level, the global planner is responsible for achieving the navigation goal, i.e. reaching the selected target. Towards this end, it instructs a motion of the robotic platform in a direction D_p, which coincides with D_v, the direction pointed to by the vision system. Due to motor drifts and inaccuracies of the target tracking module, D_p and D_v may change slightly during the course of this motion. This is easily adjusted, however, by using a feedback loop between the output-input of the global planner, as shown in Fig 3(a). The global planner simply corrects its motion, so that the two inputs match.

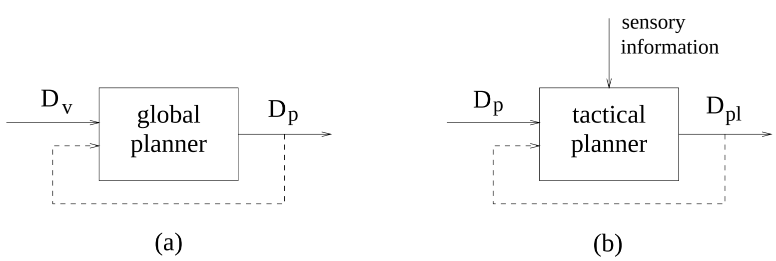


Figure 3: (a) Global planner, (b) tactical planner.

The tactical planner operates at a lower level, being invoked only when fine, local motion control is required. Towards this, it constantly checks for environment obstacles using the platform range sensors. In case that an obstacle is detected at a nearby distance, the tactical planner takes over control from the global planner. It uses

D_p (provided by the global planner) and the information returned by the sensors to render the *local platform direction* D_{pl}, as shown in Fig 3(b).

The latter is determined as illustrated in Fig 4(a). Let us denote with s_0 the sensor pointing in the direction of the platform motion, D_p. Then, $s_1, s_2, \ldots,$ denote the sensors on the one side of s_0, whereas $s_{-1}, s_{-2}, \ldots,$ denote the sensors on the other side of s_0. Let us also assume that all sensors between s_{-i} and s_j (included) indicate the presence of an obstacle. Then, the platform local direction of motion D_{pl} is set in the direction pointed to by s_{j+1}, if the angle between the latter direction and D_p is smaller than the angle between D_p and the direction pointed to by $s_{-(i+1)}$ (see Fig 4(a)). Otherwise, it is set in the direction pointed to by $s_{-(i+1)}$. In other words, the platform is set to avoid the obstacle in a way that its initial direction (D_p) is minimally modified.

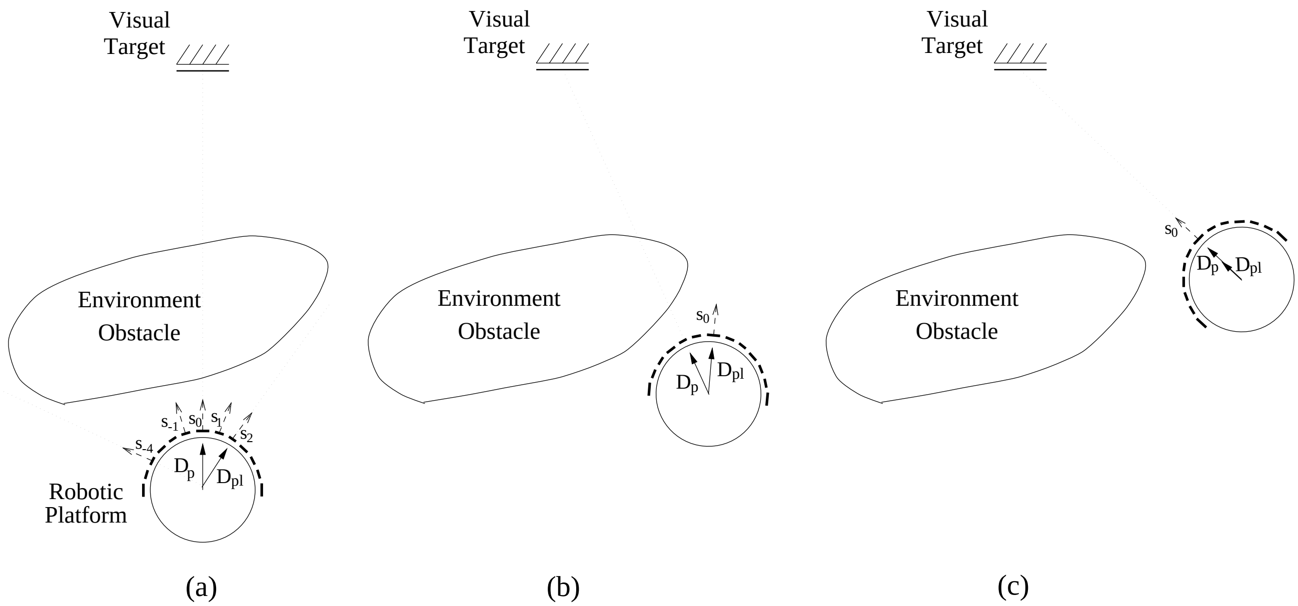


Figure 4: Obstacle avoidance with simultaneous target tracking; see text for explanation.

During this motion in the direction D_{pl}, the vision system maintains D_v, by virtue of the target fixation capability. Therefore, the global planner determines a direction of motion, $D_p = D_v$. The tactical planner tries to minimize the difference between D_p and D_{pl} using, however, the sensory information. Figure 4(b) shows an intermediate snapshot of the motion commanded by the tactical planner, where this concept is illustrated. When matching between D_p and D_{pl} is made possible (Fig 4(c)), then the selected target is again visible. The tactical planner is deactivated at this point and the global planner resumes control again.

3 Application to Robotic Wheelchairs

In this section we focus particularly on robotic wheelchair platforms and consider the adaptation of the proposed navigation approach in this case. Moreover, cost considerations and performance trade-offs are briefly examined.

3.1 Platform Configuration and Workspace Environment

The proposed approach for assistive navigation of robotic wheelchairs presupposes a certain platform configuration. This is shown schematically in Fig 5 in the form of a geometrical configuration.

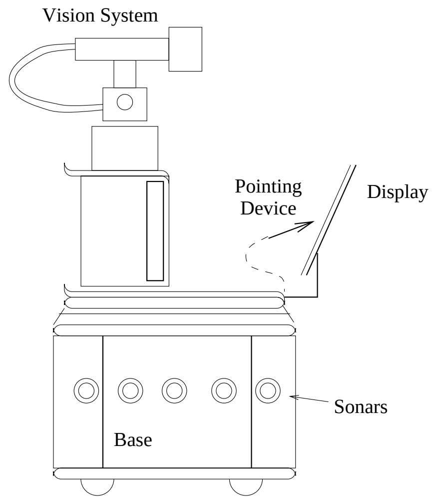


Figure 5: Robotic platform layout.

The platform is assumed to be equipped with a vision system, consisting of an image acquisition device (camera) mounted on an active head. The head supports at least two degrees of freedom: pan and tilt. The whole system is placed at an appropriate height on top of the platform, so that it overlooks the workspace, without any parts of the scene being occluded by other parts of the platform or by the user. It is also assumed that the head provides a 360^o pan capability. Finally, an image display and a pointing device are assumed to be available on the platform. The display is simply connected to the output of the camera, whereas the pointing device facilitates target selection in the viewed scene. The latter device may be user specific for this particular application. However, a detailed analysis of available technical solutions for this subject is beyond the scope of this paper. In the rest we will simply regard this as a *black-box device* that returns a selected point from the viewed image.

At a relatively low height, a ring of sonars is attached on the platform. Typically [4], sonars are placed in 15^o intervals, which requires a total of 24 sonars if we consider a cylindrical base. It should be noted, however, that for most practical cases a semicircle of sonars would suffice, since obstacles are only expected to be encountered during forward motion. In that case, the sonars should be placed in the front half of the platform. The whole configuration is completed with on-board processing power and mechanical motors that execute the platform-motion commands.

Typical navigation sessions are assumed to take place indoors. Although this is by no means an inherent limitation of the method, current technology and the strict requirements of this sensitive application area suggest indoor environments as an ideal workspace for robotic wheelchair platforms employing the proposed approach. Other than this, no further assumptions are made concerning the environment.

Since the only environment representation that is employed is the visual image of the viewed scene, the approach is inherently limited to automatic navigation only to points in the visible environment. Although this may not be a serious drawback for small workspaces, it could eventually annoy the user when navigation to a distant target is desired; in this case, the user has to select intermediate targets, based on his/her own knowledge about the environment. Directions, however, towards overcoming this limitation will be given in a later section.

Another handicap of this approach is the lack of memory of navigation sessions. That is, if at a later time the user wants to repeat a navigation session, he/she has to go through all the steps involved in it. However, after a sufficiently long period of use, especially in the user's residential or vocational environment that undergo little or no changes at all, it would be desirable to accumulate a workspace memory that would ease the navigation process. This is also further discussed later in the paper, where it is shown how this issue can be tackled.

3.2 Performance Trade-offs and Cost Considerations

Various approaches towards robotic platform navigation range considerably in the degree of autonomy they support. In the one end, one may regard "manually-controlled" platform motion, whereas, in the other end, fully autonomous navigation in unstructured environments is considered. For the case of robotic wheelchair platforms, the former approach has already led to market products. Operator controlled electric wheelchairs are routinely used by elderly and disabled. Regarding autonomous navigation approaches, they are currently far from being reliable in order to be introduced in this sector. However, there is a clear demand for technology that would increase the independence of people with special needs and the number of tasks that can be performed by them [40, 41].

Towards this end, the proposed approach for assistive navigation presents a very good compromise regarding the reliability/autonomy trade-off. It relieves the user from the continuous operation of the wheelchair, involving him/her only in the target selection process. On the other side, it does not support navigation in non-visible areas, neither pattern recognition capabilities regarding the target objects. It is, however, of utmost importance that the enhanced navigation capabilities, compared to contemporary ones, are offered without compromising robustness and reliability in platform operation. This is due to the fact that no higher-level, cognitive procedures are involved in any of the steps employed. Moreover, performance reliability has been verified experimentally, as will be presented in detail in the next section.

The proposed approach seems also attractive from the cost-effectiveness point of view. Current electric wheelchairs are in the order of 10 K ECU. Implementation of our approach on top of such a wheelchair would incur the extra cost for the parts needed (camera, sonars, pan-tilt head, image display and pointing device) plus the cost for development and integration efforts. Since the approach is based mostly on available technology from the computer vision and sonar fields, development costs can be kept low. Moreover, technological advances have contributed to considerable reductions in the prices of the hardware items mentioned above. Consequently, the proposed approach may lead to implementations that can constitute affordable, add-on modules to contemporary electric wheelchairs.

4 Experimental Results

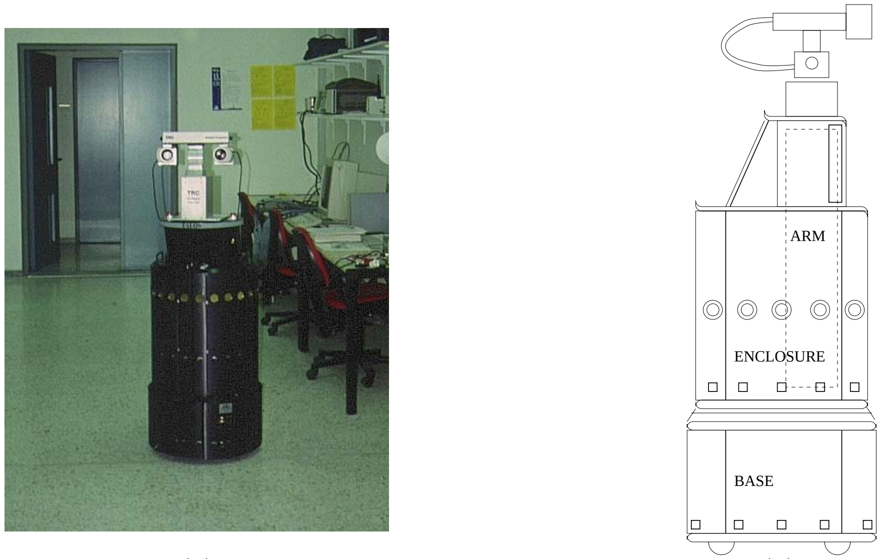
A prototype of the approach described above for assistive navigation has been implemented and tested in a laboratory environment. Preliminary results are reported here, which demonstrate the feasibility of this approach in real scenarios.

4.1 Implementation

The mobile robotic platform available at the Computer Vision and Robotics Laboratory (CVRL) of ICS-FORTH, namely TALOS, has been used as a testbed in all our experiments. TALOS includes:

- A mobile robotic platform (equipped with a 486 and a PENTIUM processors running Linux, wireless communications, sonar, infrared, and tactile sensors).
- A binocular, active vision head (independent control of pan, tilt, left and right vergence).

Figure 6(a) shows a picture of TALOS and Fig 6(b) shows the platform geometry. The system is configured so that the PENTIUM processor is responsible for vision processing and control of the head, while the 486 processor controls the motion of the robot as well as the sonar, infrared and tactile sensors. Communication and coordination between the two processors is facilitated by the TCX package [42].



(a) (b)

Figure 6: (a) The mobile robotic platform TALOS and, (b) platform's geometry.

The prototype developed on TALOS does not include any advanced user interface for scene viewing and, especially, target selection. In its current version this is simply done interactively by the system operator.

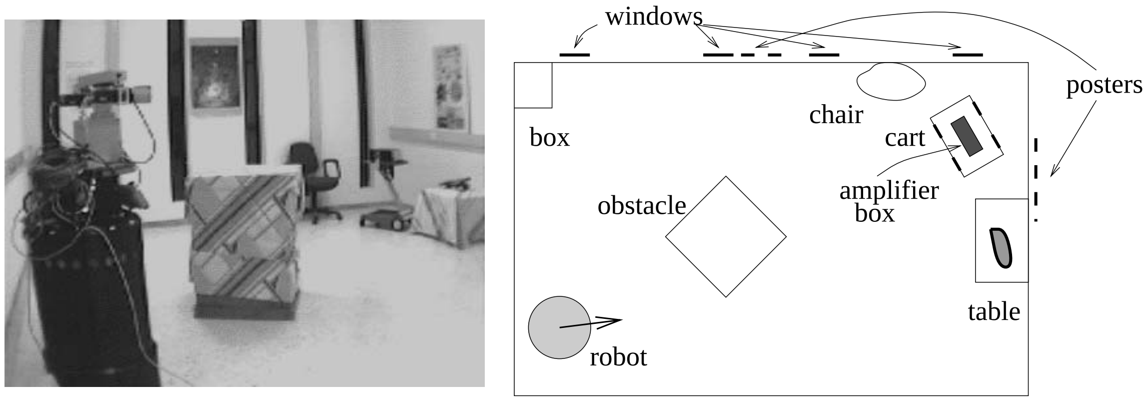
4.2 Laboratory Experimentation

Several experiments have been conducted to test the effectiveness of the proposed approach as well as its "behavior" in indoor environments. In these experiments we have considered various workspace environments and target objects. Moreover, obstacles have been artificially placed to obstruct the initial, straight path to the target. These experiments have verified the appropriateness of this approach for assistive navigation. Results from a sample experiment are presented here for demonstration purposes.

The workspace for this experiment is shown in Fig 7(a). It consists of a room with various "objects" placed in it. The platform's initial position was at the one end of the room. More specifically, Fig 7(b) shows a top view of the workspace; the robotic platform is denoted with the filled circle, whereas its initial orientation is indicated with the corresponding arrow. Some objects (chair, table, box, cart with an amplifier on top of it) are placed in various spots in the room. An obstacle has been intentionally placed as indicated in Fig 7(b).

In this experiment we have simulated a navigation session to reach the amplifier box, lying on the cart at the far end corner of the room. The scene, as viewed by the robot's camera is shown in Fig 8. The target selection, performed by the user, is depicted in Fig 8(a) with an arrow. As can be observed, this selection corresponds to the amplifier box. The template constructed for this user selection is shown in Fig 8(b), as a window superimposed on the actual image.

Following that, target fixation has been performed and platform navigation has been initiated in a straight line towards the selected target. This is shown in the first few images of Fig 9. Actually, Fig 9 shows a sequence of snapshots of the whole navigation session, until the target selection was reached. After a short motion in a straight path, the obstacle, cutting the robot's way, has been detected. The images in the second and third row of Fig 9 illustrate the detour performed by TALOS in order to avoid it. During this motion, fixation of the vision system to the target has been maintained. After completion of the obstacle avoidance motion, TALOS has again initiated a motion in a straight line towards the target. This is shown in the images in the last two



(a) (b)

Figure 7: (a) Workspace for navigation experiments, (b) top view of the workspace.

(a) (b)

Figure 8: Scene viewed by the camera mounted on TALOS; (a) user selection, (b) constructed template.

rows of Fig 9.

The result presented throughout this section, although it refers to an artificial (laboratory) workspace, demonstrates clearly our approach for assistive navigation and serves as an indication of its performance in indoor environments.

5 Future Work and Platform Enhancements

As already mentioned in section 3.1, the proposed navigation approach is lacking any memory of past navigation sessions. However, such a feature would greatly enhance the navigational assistance offered to the user, especially in his/her *everyday* workspaces (e.g. residential or vocational environment). In this section we will provide some hints towards incorporating such features in the basic approach described above. Moreover, other enhancements in the platform functionalities are also discussed.

In the normal mode of operation, functioning of the robotic wheelchair as described in the previous sections is performed. More specifically, target selection is performed by the user in the visual image of the viewed scene. At a higher level of operation, however, the targets and/or objects selected most often and their spatial relationships can be memorized. Technically speaking, the extracted templates that correspond to the user selections can, actually, be memorized as well as spatial relations between them. The latter can be extracted

Figure 9: Snapshots from a navigation session.

using the actual path that was tracked by the platform in order to accomplish a navigation goal. By allowing the user to assign semantics to these targets or objects (environment landmarks), a spatial memory can be constructed, enabling the definition of more complex navigational tasks. More specifically, the user may assign a semantic label (e.g. *"telephone"*) to an object that has been selected as a navigational goal in a previous session. By appropriately storing this information and relating it to the actual object (template), certain tasks can be effectively automated.

This leads to a semantically-driven mode of operation, where the user may request navigation to certain, memorized environment targets. Therefore, requests of the type *"take me to the telephone"* can be granted by the system, relaxing thus the demand on target selection from the user. Since execution of such tasks does not involve any user intervention, the definition of advanced "service tasks" that require automatic navigation is made possible. This, of course, presupposes effective visual search methods, at least for the objects of interest. The reason for this stems from the fact that the system's knowledge about the environment is vague, in the sense that during training it does not acquire a detailed model (map) of the environment, but only a partial representation of objects and their spatial relationships. However, this information can effectively limit the search space to very few objects, which can subsequently be handled by more elaborate visual search methods.

The proposed approach concerns only user-selected targets. Towards assisting the user in the target selection process, vision techniques that will suggest regions of interest can be employed. Such techniques will extract precategorical visual information that corresponds to potentially interesting features (e.g. color, symmetry) or dynamic events of the environment. The case of dynamic events is of particular importance, since they signal changes in the environment. For example, recent work on motion perception [43, 44, 45] and tracking of moving objects [46, 47, 39] can be used for enabling a robotic wheelchair to follow a person that moves in the static environment.

It should be noted at this point that motion analysis techniques can also be used in order to compensate for the motion of the visual sensor and *actively* fixate on the target. Another way for achieving the same result is by tracking tokens (lines and corners) belonging to the target, using methods such as those described in [48]. However, the applicability of such techniques is currently limited by the computational costs they involve.

Since the proposed approach does not make any limiting assumptions about the "mobile platform", it can effectively be exploited by other robotic actuators, e.g. flexible manipulators, in tasks such as manipulator motion, object grasping, etc. User selection of an object (via advanced man-machine interfaces) triggers fixation on that object, which can then be easily manipulated (e.g. picked up). Such functionalities can be effectively integrated with the navigational functionalities presented above, resulting in robotic platforms with advanced navigational and manipulation capabilities.

6 Discussion

In this paper we have presented an approach towards introducing assistive navigational capabilities in robotic wheelchair platforms. Contemporary approaches usually fail to support autonomous navigation. In most cases, some low level functionalities are provided (obstacle avoidance, collision warning, etc.), whereas the strategic motion of the platform is user controlled. This can be attributed to the fact that these approaches lack some kind of *environment representation* that would facilitate execution of navigation goals.

In our approach, we overcome this limitation by introducing visual representations of the selected navigational target. More specifically, a user target selection initiates a deictic representation of the target, which then facilitates target tracking throughout the navigation session. This is coupled with sonar-based obstacle avoidance techniques. The resulting approach exhibits a navigational behavior that may be useful in workspace environments of elderly and disabled. This has been demonstrated by preliminary experimental results obtained from an implementation of this approach on a mobile robotic platform.

To the best of our knowledge, this is the first time that computer vision techniques have been introduced in this sensitive application area. This is mainly due to the fact that computer vision is still not reliable enough to be employed in cases where safety and robustness are at a premium. By excluding, however, high level cognitive tasks (target recognition and selection) from the vision system and relying on the user for performing them, we have been able to achieve reliable system performance in indoor environments.

The work presented here constitutes a first step towards the development of *autonomous* robotic platforms offering advanced functionalities to the user. The approach presented is amenable to various enhancements that may contribute to *added-value* platform functionalities. Platform training in user environments has been presented as a means to fully automate routine tasks. The user is responsible in the training procedure to assign semantics to the selected targets. Moreover, we have briefly discussed other extensions concerning tracking of moving objects, motion compensation to facilitate more accurate target fixation and manipulation functionalities. Although further research and experimental work is definitely needed in this application area of robotics and computer vision, we believe that the guidelines presented in this paper constitute a good starting point towards the development of autonomous wheelchair platforms.

References

[1] R.C. Dorf. *Concise International Encyclopedia of Robotics: Applications and Automation.* Wiley, New York, NY, 1990.

[2] W. Stadler. *Analytical Robotics and Mechatronics.* McGraw-Hill, Inc., Singapore, 1995.

[3] E. Donna, C. Bacon, T. Rahman, and W.S. Harwin. *Fourth Intl. Conf. on Rehabilitation Robotics: Conference Proceedings.* Applied Sci. & Eng. Lab., Univ. of Delaware/A.I. duPont Inst., Wilmington, Delaware, USA, 1994.

[4] H.R. Everett. *Sensors for Mobile Robots: Theory and Application.* A K Peters, Ltd., Wellesley, MA, 1995.

[5] S.A. Walter. The sonar ring: Obstacle detection for a mobile robot. In *Proceedings of the IEEE Int. Conf. Robotics Automat.*, pages 1574–1578, Raleigh, NC, Mar. 31 - Apr.3 1987.

[6] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robotics Res.*, 5(1):90–98, 1986.

[7] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. on Sys. Man and Cybern.*, SMC-19(5):1179–1187, Sept. - Oct. 1989.

[8] J. Borenstein and Y. Koren. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Trans. on Robotics and Autom.*, 7(3):278–288, Jun. 1991.

[9] A. Elfes. *Occupancy Grids: A Probabilistic Framework for Mobile Robot Perception and Navigation.* PhD dissertation, Electrical and Computer Engineering Dept./ Robotics Institute, Carnegie Mellon University, 1989.

[10] J.Borenstein and Y.Koren. Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance. *IEEE Trans. on Robot. and Automation*, 7(4):535–539, Aug. 1991.

[11] A. Zelinsky. *Environment Exploration and Path Planning Algorithms for Mobile Robot Navigation using Sonar.* PhD dissertation, Computer Science Dept., University of Wollongong, Australia, 1991.

[12] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Trans. on Robotics and Autom.*, 8(3):304–312, Jun. 1992.

[13] I. J. Cox. Blanche: An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Trans. on Robotics and Autom.*, 7(2):193–204, Apr. 1991.

[14] J. F. Canny. *The Complexity of Robot Motion Planning.* MIT Press, Cambridge, MA, 1988.

[15] J.-C. Latombe. *Robot Motion Planning.* Kluver Academic Publishers, Boston, MA, 1991.

[16] Y. K. Hwang and N. Ahuja. Gross motion planning – a survey. *ACM Computing Surveys*, 24(3):221–291, Sept. 1992.

[17] H. Takeda, C. Facchinetti, and J.-C. Latombe. Planning the motions of a mobile robot in a sensory uncertainty field. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:1002–1015, 1994.

[18] M. Drumheller. Mobile robot localization using sonar. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9(2):325–332, Mar. 1987.

[19] G. Giralt, R. Chatila, and M. Vaisset. An integrated navigation and motion control system for autonomous multisensory mobile robots. In M. Brady and R. Paul, editors, *First International Symposium on Robotics Research*, pages 191–214, Cambridge MA, 1984. MIT Press.

[20] R.R. Rathbone, R.A. Valley, and P.J. Kindlmann. Beacon-referenced dead reckoning: A versatile guidance system. *Robotics Engineering*, Dec. 1986.

[21] S.H. Dodington. Electronic navigation systems. In D. Christiansen and D. Fink, editors, *Electronic Engineer's Handbook, 3rd edition*, pages pp.76–95. McGraw Hill, 1989.

[22] M. Magee and J.K. Aggarwal. Robot self-location using visual reasoning relative to a single target object. *Pattern Recognition*, 28(2):125–134, 1995.

[23] A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. In *10th National Conference on Artificial Intelligence*, pages 697–702, San Jose, CA, July 1992.

[24] H. Takeda and J.-C. Latombe. Sensory uncertainty field for mobile robot navigation. In *IEEE International Conference on Robotics and Automation*, pages 2465–2472, Nice, France, 1992.

[25] P.E. Trahanias and Y. Komninos. Robot motion planning: Multi-sensory uncertainty fields enhanced with obstacle avoidance. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS'96*, Osaka, Japan, Nov. 4-8 1996.

[26] S.D. Whitehead and D.H. Ballard. Learning to perceive and act by trial and error. *Machine Learning*, 7:45–83, 1991.

[27] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE J. Robotics Auromat.*, RA-2(7):14–23, Apr. 1986.

[28] V. Lumelski and A. Stepanov. Dynamic path planning for a mobile automation with limited information on the environment. *IEEE Trans. on Automat. Control*, CA-31(11), Nov. 1986.

[29] R.A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–160, 1991.

[30] P.E. Agre. *The Dynamic Structure of Everyday Life.* PhD dissertation, MIT Artificial Intelligence Laboratory, Cambridge, MA, 1988.

[31] P.E. Agre and D. Chapman. Pengi: An implementation of a theory of activity. In *Sixth Nat. Conf. on Artificial Intelligence*, pages 268–272, Los Altos, CA, 1987. Morgan Kaufmann.

[32] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, Aug. 1988.

[33] K. Pahlavan, T. Uhlin, and J.O. Eklundh. Active vision as a methodology. In Y. Aloimonos, editor, *Active Perception, chapter 1.* Lawrence Erlbaum Associates, 1993.

[34] A. Blake and A.L. Yuille, editors. *Active Vision – Chapter 1: Tracking.* The MIT Press, Cambridge, MA, 1992.

[35] R.C. Gonzalez and R.E. Woods. *Digital Image Processing.* Addison-Wesley, Reading, MA, 1992.

[36] M. Swain and D. Ballard. Color indexing. *Intl. Journal of Computer Vision*, 7(1):11–32, 1991.

[37] D. Slater and G. Healey. The illumination-invariant recognition of 3d objects using local color invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(2):206–210, 1996.

[38] T.J. Olson and D.J. Coombs. Real-Time Vergence Control for Binoculare Robots. *International Journal of Computer Vision*, 7(1):67–89, 1991.

[39] D. Murray and A. Basu. Motion Tracking with an Active Camera. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(5):449–459, May 1994.

[40] R. Foulds. Interactive robotic aids - one option for independent living: An international perspective. *Monograph 37, World Rehabilitation Fund, New York*, pages 7–17, 1986.

[41] J.L. Dallaway, R.D. Jackson, and P.H.A. Timmers. Rehabilitation robotics in Europe. *IEEE Trans. Rehabilitation Eng.*, 3(1), 1995.

[42] C. Fedor. *TCX: An Interprocess Communication System for Building Robotic Architectures, Programmer's Guide to version 10.XX*, January 1994.

[43] A.A. Argyros, M.I.A. Lourakis, P.E. Trahanias, and S.C. Orphanoudakis. Independent 3d motion detection through robust regression in depth layers. In *British Machine Vision Conference (BMVC '96)*, Edinburgh, UK, Sep. 1996.

[44] A.A. Argyros, M.I.A. Lourakis, P.E. Trahanias, and S.C. Orphanoudakis. Qualitative detection of 3d motion discontinuities. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS'96*, Osaka, Japan, Nov. 4-8 1996.

[45] R. Sharma and Y. Aloimonos. Early detection of independent motion from active control of normal image flow patterns. *IEEE Trans. Syst. Man, Cybern.*, 26(1):42–53, Feb. 1996.

[46] N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade. Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision. *IEEE Trans. on Robot. and Automation*, 9(1):14–25, Fe. 1993.

[47] P.K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(2):152–165, April 1993.

[48] O. Faugeras. *Three-Dimensional Computer Vision*, chapter 8. MIT Press, Cambridge, MA, 1993.