

SAM: a robotic butler for handicapped people

Anthony Remazeilles, Christophe Leroux, Gérard Chalubert

Abstract—This article presents a service robotic system for disabled people. SAM is a mobile platform that grasps any textured object in an apartment-like environment. The object grasping is realized with a robotic arm, mounted onto the mobile unit. Thanks to the stereo rig on the arm's gripper, the user defines the object to grasp on a remote screen, in a very simple way, and the grasping is autonomously performed by visual servoing. The scheduling of the operations is defined and controlled with automatons that are dynamically interpreted with a scenario manager. At last, the communication between the different components is realized with web services through a DPWS layer, providing an efficient solution for sharing data between any device of an intelligent home. Experiments confirm the validity of our approach.

I. INTRODUCTION

People presenting a severe motor handicap (like quadriplegia) need a personal assistant to help them in their everyday-life. Among all the operations realized by this person, a majority is relatively simple and frequent, like eating, drinking or bringing an object. Since these operations are well defined, it is interesting to investigate the possibility of having a robotic system performing them. The assistant can then take care of other medical matters, and the disabled person gets also more independence. In this context, this article details a mobile robotic system providing to handicapped people a very easy-to-use solution for the grasping of unknown objects in their environment.

A. State of the art

According to the surveys on assistive robotics [14], four main system types can be distinguished. They are briefly described in the following.

1) *Workstation systems*: Workstation system can be considered as the first category of assistant robot that has been designed (see projects RAID [8], DEVAR [27] and Master [5]). A workstation fits very well the rehabilitation of disabled people in their office environment. It is usually composed of a robotic arm, fixed to a desk, that has access to several elements or objects of its known environment (computer, printer, CD, books). This *a priori* knowledge is a reasonable constraint within the industrial context, since tasks to perform are generic and well-known.

Concerning the grasping task, the control system is usually position-based, requiring thus a perfect knowledge of the environment. A set of preprogrammed tasks enables to access to the different elements of the workstation. In order to

handle small position errors on the objects to grasp, the project PROVAR [28] adds force sensors to the gripper.

More recently, smaller and less expensive systems have been designed, extending the use of workstations from office to daily-life tasks, like feeding with MySpoon [24], or cleaning, painting and even playing with HANDY 1 and RAIL [26]. Nevertheless, these systems are only able to manipulate a restricted set of objects.

2) *Stand alone manipulators*: Lighter than the workstations, since they are reduced to a robotic arm, stand alone manipulators are usually fixed to an element of the environment, like a table (Tou [6] and ISAC [17]).

Stand alone manipulators correspond to a first attempt to avoid the knowledge of the environment, by introducing sensor data within the control loop. Nevertheless, the user remains highly involved, even if ISAC uses a camera to localize by segmentation simple objects, and a face tracking to feed the user. Furthermore, the operation space of the arm remains restricted since it is still mounted onto a fixed structure, while we would like to manipulate objects everywhere.

3) *Wheelchair mounted systems*: By mounting the manipulator arm onto the wheelchair, the user increases its operation space. The arm becomes then a standard device that can be plugged to any wheelchair. Raptor and MANUS are the most used arms in such application [1].

Since the environment can drastically change from one task execution to another, the interest of preprogrammed operations becomes limited, and joystick-based controllers are usually provided instead. Nevertheless, depending on the user's disability, even controlling the joystick can be difficult, and some researchers have thus proposed a speech-based controller [13]. If it provides an easier access to the control commands, it does not reduce the number of actions the user has to schedule and perform in order to manipulate the arm. As studies show [20], in average, the user spends as many times thinking of the action to perform as realizing it.

In this context, the addition of sensors to the robotic arm is a nice solution to reduce the user's actions by introducing sensor-based automatic controls. Cameras appear to be the most used sensor, like in the MUSIIC project where a high intensity and structured light source is used to recognize an object vocally described by the user and previously defined within a database [18]. In the project FRIEND 1, a visual servoing enables to grasp objects with marks [19]. The project FRIEND 2 adds a tray to the wheelchair, a sort of large tactile sensors that facilitates the visual object detection [29]. Objects are not marked within the VICTORIA project [4]. Their silhouette are estimated by active vision, and the grasp-

This work was supported by the fondation des Caisses d'Epargnes pour la solidarité (project AVISO) and the european project ITEA ANSO.

The authors are with the CEA, LIST, 18 route du panorama, BP6, Fontenay aux Roses F-92265, France, `firstname.lastname@cea.fr`

ing procedure is realized by moving the MANUS gripper toward the object center of gravity.

4) *Mobile platforms*: When the arm is fixed onto an independent platform, the obtained system acts as a robot servant providing services to the end-user. Since the mobile platform is required to move autonomously, the problem of Automated Guided Vehicles has to be addressed. Scanners and lasers are used in the projects WALKY [21] and CARE-O-BOT 2 [11], whereas the projects ARPH [15] and HERMES [3] use cameras.

Concerning the object manipulation, the CARE-O-BOT 2 project couples a camera with a laser scanner in order to facilitate the 3D information extraction. Once the object is well-recognized and localized (known objects are defined in a database), a collision-free trajectory is generated and realized in open-loop. In the HERMES project, the camera is principally used to position the mobile platform and the object grasping procedure uses preprogrammed motions. The KARES-2 project proposes several predefined actions to the user, from serving a beverage, to opening and closing doors [2]. The object grasping uses a stereo rig, and the tracking is performed through a modified log-polar mapping algorithm. At last, the MOVAID project [25] chooses the most suited grasping position, by an approach mixing neural network and fuzzy logic (which requires a learning step).

B. Discussion

Workstation are very well adapted for the rehabilitation in the office, but are not able to deal with dynamic environments where elements are not static, since most of the operations are predefined and realized without any sensor control. If the stand-alone manipulators are lighter and less expensive, their operating space remains restricted since the arm is mounted onto a static element.

Mounting the arm onto a mobile system is relevant in the sense that the manipulation can then be performed anywhere. Nevertheless, mounting it directly onto the wheelchair can be considered as a drawback since it increases the occupancy space of the chair [10] and thus limits the possible motions of the user. That is the reason why we have decided to mount the arm onto an independent mobile platform. The wheelchair is not overloaded, and the user can manipulate objects without having to follow the arm.

We propose also an original grasping procedure that relies on a stereo rig mounted onto the gripper. The advantage of this approach is that no *a priori* knowledge on the object is required. Its use is also very simple: two image clicks only are needed to define the object to grasp, and then the grasping operation is totally automatic.

This article gives an overall view of the architecture and capabilities of our system. The next section presents the physical architecture of our mobile robot SAM (*Synthetic Autonomous Majordomo*). Section III deals with the vision-based object tracking and arm guidance procedure. The scenario interpreter ISEN, that monitors the actions of the mobile robot, is then detailed in Section IV, and experimental results in Section V confirm the validity of our approach.

II. OVERALL ARCHITECTURE

The system is mainly composed of two elements, the client and the mobile robot. The client (classical PC) is the system with which the user interacts. It provides different visualizations, depending on the current action of SAM. The client does not deal with low-level sensor outputs but with higher level semantic information. Depending on these data, the scenario interpreter decides the next order to send to SAM.

The second element is the mobile robot, composed of a mobile platform and a robotic arm (see Fig. 1). These two components provide to the client autonomous actions. The communication between the client and the mobile robot uses the wi-fi network, and relies on a web service layer.

A. Mobile platform

We use the mobile platform MP-M470 designed by Neobotix¹. This system provides some autonomous mapping, localization and navigation procedures that we are currently using (in this paper, we do not address the mobile unit localization and control).

The mobile guidance is realized by an embedded PC that uses input information given by its two lasers and its four ultrasound sensors. Laser scanners are used to create a 2D raster image of the robot's neighborhood. The mobile pose is estimated by matching this map with a global one describing the whole environment (learned during an initial exploratory procedure). An extended Kalman filter is used to fuse this matching with the motor encoders. The obtained pose is robust to static or moving obstacles. The path planning method is based on the well-known A^* algorithm [22].

During the environment mapping, the user can define some roadmaps onto which the robot will move in priority (in order to keep it as far as possible from the obstacles). The definition of stations, as a set of preferred destinations, provides a semantic control of the robot (go to the kitchen, or to the bedroom, etc).

B. Manipulator arm

The Assistive Robot service Manipulator, also called MANUS, is developed by Exact Dynamics². It has 8 degrees of freedom, and is usually mounted on electric wheelchairs or mobile platforms. It is used for grasping and manipulating objects, and the control of the arm, either in articular or in cartesian mode, uses a device adapted to the user disability. The gripper can load objects up to 1.2kg, and the maximum spread between the two fingers is 9cm. The MANUS can be controlled with a PC through a CAN bus.

In order to increase the autonomy of the grasping procedure, several sensors have been added to the gripper (see fig. 2): a stereo rig, an optical barrier and two load cells. The cameras enable the user to outline the object to grasp. During the grasping procedure they are also used to define the arm motions depending on the observed object position

¹more technical details on <http://www.neobotix.de>

²<http://www.exactdynamics.nl/>

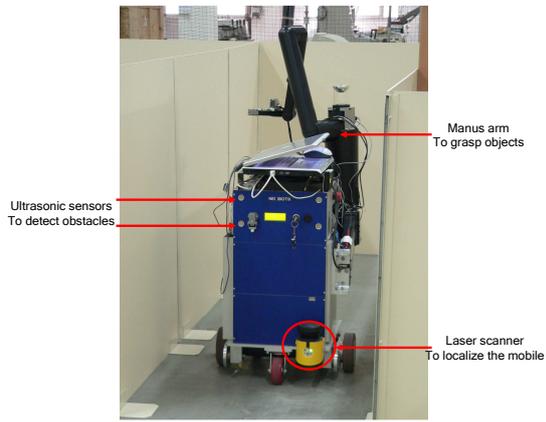


Fig. 1. The MANUS arm is mounted onto the mobile platform MP-M470.

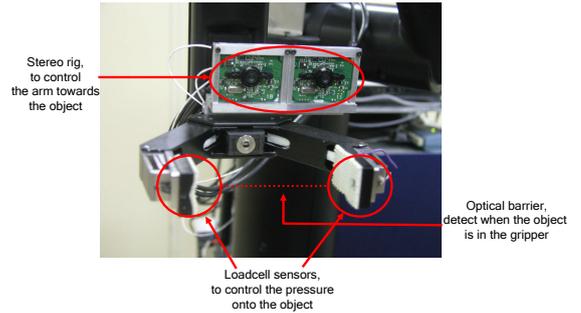


Fig. 2. Sensors added to the gripper for the grasping procedure.

(this is described in the next section). The optical barrier detects the presence of an object inside the gripper's fingers, and the load cells inform on the pressure applied onto the object when the gripper is closed (the use of these sensors will be illustrated in Section IV-B).

C. Communication layer

The communication between the devices relies on web services. Both the arm and the mobile platform are seen as servers to which clients ask for functions or services, through an IP communication. This protocol provides a very interesting interoperability: the arm and the mobile platform become devices similar to any equipment of an intelligent home. They propose well-defined services (motion of the arm, mobile navigation, position information), and totally hide the internal complexity of these actions, not only on the way the information is transmitted, but also on the low-level functions that enable to perform the asked operations.

The protocol used to implement the web service is DPWS [16] (*Device Profile for Web Services*) that relies on SOAP-XML³. It provides also a notion of *Plug and Play*, enabling the devices connected onto the network to detect automatically the appearance of a new one.

III. STEREO VISION-BASED OBJECT GRASPING

Let us suppose that the mobile platform has moved toward a position specified by the user. The arm is automatically positioned at a *grasping configuration*. We suppose here that the object to grasp is already within the two camera fields of view. If it is not the case, buttons of the interface enable to move the arm in its cartesian space, in order to observe the object.

The module described here enables to move the arm just in front of the object. Neither the initial position of the object nor its model is needed. The required information to track the object is automatically extracted from the images during an initialization step described in section III-A.

During the motion of the arm, the object tracking is realized by virtual visual servoing (see section III-B). This

³the design of the protocol has been realized by Odonata and Schneider Electric during the ANSO project.

tracking gives the object pose with respect to the frame of one camera of the stereo rig. This information enables then to deduce the motions of the robotic arm (section III-C) in order to get closer to the object. Note that in this first release, the rotations of the gripper are not controlled. We suppose furthermore that at the initial grasping configuration the gripper is parallel to the floor.

A. Tracker initialization

This step consists in estimating from the two views a set of 3D points describing the object pointed by the user. The operator uses a remote display corresponding to the image acquired by the right camera of the stereo rig. Only two image clicks are needed to define a surrounding box around the object (upper left and bottom right corner). Thus, the information asked to the user is very simple.

Harris points [12] are extracted from this window. Their relatives in the left image are searched by using the differential tracker KLT [23]: the corresponding feature of a salient point is found in the other view by searching the pixel translation of all the points in a neighboring window, through the minimization of a correlation measure. This operation gives a set of matched points $({}^l\mathbf{x}_i, {}^r\mathbf{x}_i)$.

By triangulation⁴, we estimate then for each of these matches the corresponding 3D point, ${}^{c_r}\mathbf{X}_i$, expressed within a frame attached to the right camera. The depth of the four corners ${}^{c_r}\mathbf{B}_i$ of the box defined by the user is also estimated. Since we can not ensure that these four points are associated to real Harris features in the image, we suppose that the depth of these virtual points correspond to the median of all the features depths that have been estimated.

In this initialization step, all the 3D points are expressed in the right camera frame. We suppose that this frame position corresponds to the object frame, which means that at this instant 0, the pose of the object with respect to the right camera is:

$${}^{c_r}\mathbf{M}_o(0) = \begin{bmatrix} {}^{c_r}\mathbf{R}_o & {}^{c_r}\mathbf{t}_o \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0} \\ 0 & 1 \end{bmatrix}, \quad (1)$$

⁴The stereo rig is calibrated with the Matlab ToolBox available at http://www.vision.caltech.edu/bouguetj/calib_doc

and the coordinates of the 3D points in the object frame ${}^o\mathbf{X}_i$ are equals to ${}^{c_r}\mathbf{X}_i$. The estimation of the object pose, as described in the next section, relies on this sparse 3D model, and on the associated image points that are tracked during the arm navigation.

B. 3D object tracking

The objective of this module is to estimate at each instant the position of the object wrt to the right camera. This operation is realized by virtual visual servoing. Visual servoing estimates the motion of a robotic system from an error measured between the current value of visual features $\mathbf{s}(t)$ and their desired value \mathbf{s}^* at the desired robot position. The velocity of the camera τ_c is deduced from the relation [7]:

$$\tau_c = -\lambda \mathbf{L}_s^+ (\mathbf{s} - \mathbf{s}^*), \quad (2)$$

where λ is a positive gain tuning the convergence speed, and \mathbf{L}_s^+ the pseudo-inverse of the interaction matrix linking the variation of the feature value to the motion of the camera.

Virtual visual servoing reuses this formalism to perform pose estimation. Let us suppose that at the instant t , we have an estimation of the object pose wrt to the camera ${}^c\mathbf{M}_o(t)$, and also the current 3D position of the features tracked, ${}^{c_r}\mathbf{X}_i(t)$ (since ${}^{c_r}\mathbf{X}_i = {}^{c_r}\mathbf{M}_o \circ {}^o\mathbf{X}_i$). At instant $t + 1$, a new couple of images is acquired. The desired position of the features \mathbf{s}^* from eq. (2) correspond to the coordinates of the features tracked in this new image. Current values $\mathbf{s}(t)$ are the projection onto the image space of the 3D points ${}^{c_r}\mathbf{X}_i(t)$. The error observed ($\mathbf{s}(t) - \mathbf{s}^*$) is due to the motion of the camera between instants t and $t + 1$, which induces a change of the current pose ${}^{c_r}\mathbf{M}_o$. By using (2), we get a camera velocity that reduces this model pose error. The current pose estimation is then updated, by using the exponential map of τ_c [9]:

$${}^{c_r}\mathbf{M}_o^{k+1} = {}^{c_r}\mathbf{M}_o^k e^{[\tau_c]} \quad (3)$$

This process is applied in close loop until convergence. We get then a novel estimation of the pose of the object wrt to the camera ${}^{c_r}\mathbf{M}_o(t + 1)$.

The virtual visual servoing can be extended to our stereo configuration [9]. At instant t , the current image feature coordinates are $\{{}^l\mathbf{x}_i(t)\}$ and $\{{}^r\mathbf{x}_i(t)\}$, respectively in the left and right view. A new stereo couple of images ($\psi_l(t + 1), \psi_r(t + 1)$) is then acquired. The two feature sets are separately updated by tracking the image points with the KLT tracker, which gives $\mathbf{s}_l^* = \{{}^l\mathbf{x}_i(t + 1)\}$ and $\mathbf{s}_r^* = \{{}^r\mathbf{x}_i(t + 1)\}$. From the current object pose estimation expressed in the right camera, ${}^{c_r}\mathbf{M}_o(t)$, the projection of the model into the two image planes is updated, corresponding to \mathbf{s}_l and \mathbf{s}_r . Then, the pose associated to the new couple of images ${}^{c_r}\mathbf{M}_o(t + 1)$ is updated with eq.(3), where the right camera velocity is deduced from the relation:

$${}^r\tau_c = -\lambda \begin{bmatrix} \mathbf{L}_{\mathbf{s}_r} \\ \mathbf{L}_{\mathbf{s}_l} \end{bmatrix} {}^{c_l}\mathbf{V}_{c_r} \begin{bmatrix} \mathbf{s}_r - \mathbf{s}_r^* \\ \mathbf{s}_l - \mathbf{s}_l^* \end{bmatrix} \quad (4)$$

The twist transformation matrix ${}^{c_l}\mathbf{V}_{c_r}$ enables to map a velocity expressed in the right camera frame to the left

camera frame:

$${}^{c_l}\mathbf{V}_{c_r} = \begin{bmatrix} {}^{c_l}\mathbf{R}_{c_r} & [{}^{c_l}\mathbf{t}_{c_r}]_{\times} \\ \mathbf{0} & {}^{c_l}\mathbf{R}_{c_r} \end{bmatrix}, \quad (5)$$

with $[\mathbf{a}]_{\times}$ the antisymmetric matrix associated to the vector \mathbf{a} .

As proposed in [9], this minimization is embedded into a robust scheme. Weights are associated to each feature, depending on the confidence we have in it. It enables to reject points that have been badly localized by the tracker and could add noise to the pose estimation. Notice that in [9], the visual features used are lines, whereas we use here points.

C. Arm motion control

At each instant, the pose estimated enables to update the current 3D position of the four corners of the box. Its centroid is also estimated: ${}^{c_r}\mathbf{G}(t) = \frac{1}{4} \sum_i {}^{c_r}\mathbf{B}_i(t)$. The robot is asked to bring this centroid toward a desired position just in front of the gripper, ${}^{c_r}\mathbf{G}^*$. We suppose that the object is almost vertical, and we only control the translations of the arm. The motion of the right camera is thus obtained by a 3D servoing:

$${}^r\tau_c = -\lambda \mathbf{e} = -\lambda ({}^{c_r}\mathbf{G}^* - {}^{c_r}\mathbf{G}_d(t)) \quad (6)$$

This velocity is then mapped to the control frame of the MANUS arm. At convergence, the object is just in front of the gripper. Webcams can thus no more be used to control the arm, since the object is too close. In order to finish the grasping procedure, the other sensors are then used, as the next section will show.

IV. THE SCENARIO INTERPRETOR ISEN

The overall application we are considering, a mobile robot operating in a real environment with dynamic objects, can not easily be described with a classical sequential program. The variety of situations that can occur, at every instant, would make such a program increase exponentially if one would try to consider all the possible cases. Thus we have chosen to develop the general controller with a suited system called ISEN (*Interactive Scenarization Engine*). This scenario manager has been developed at the CEA-LIST mainly for augmented reality and virtual applications.

A. Generalities

All the actions than can execute the process are organized as a set of scenarios, scheduling the succession of modular actions, depending on the current system situation and on external information provided either by external devices like sensors or by the operator. To formalize this, scenarios are described by automaton, respecting the following notions.

A *state* is a situation where the behavior of all the system does not evolve. It is characterized by its duration and its stability. An *event* is an instantaneous information, a shutter release, enabling to pass from one state to another. A *transition*, connexion between two states, is dedicated to each event. It depends both on the event that induces it and on the current state of the system. An *action* is a succession of low-level actions that modify the general condition of the system.

A *task* is the description of a work to realized, defined as a set of actions, events and transitions. At last, a *scenario* corresponds to all possible works, *ie* a task or a set of task.

The monitoring of a task execution is realized by an entity called *agent*. By associating an agent to each task, it is possible to realize in parallel several tasks.

The scenarios are defined with a xml language. To control our application, we have designed three main tasks. The *GoToTask*, that controls the navigation of the mobile platform from one position to another, the *GraspObjectTask* and *DropObjectTask* to grasp and drop objects. As an example, the second one is detailed in the next section.

B. Object grasping scenario

Figure 3 illustrates the scenario controlling the grasping procedure. On this figure, ellipses correspond to states (label on the upper part, and action associated on the lower part). Arrows are the events the agent is waiting for, in order to switch from one state to another.

The different states are:

- *InitStateArmGrasp*: initial state of the agent, its action *InitGrasping* initializes the arm as well as the client interface. The event *ManusStartGrasp* is raised by the user, when he switches on the grasping mode.
- *PositArmForGrasp*: the arm reaches a reference grasping position, through an articular motion (action *JointMove*). The event *ArmCommandFulfilled* informs that this position is reached.
- *OpenGripper*: the gripper opening makes the robot ready to grasp an object outlined by the user.
- *WaitDesignation*: the action *waitObject* allows the user to define an object. The box clicked is sent to the tracking module which verifies its validity. If the system validates the box, the tracking starts, but the robot does no move until the user asks it through the event *ObjectSelected* (the user can indeed change its selection).
- *ObjectSelected*: the visual servoing starts, in order to position the gripper just in front of the object. The user can stop at any time the autonomous motion. When the convergence is reached, the event *EndVSM* is raised.
- *BlindGrasp*: the object is just in front of the gripper. A forward translation is performed in order to make it enter the gripper, which will be detected by the optical barrier (event *OpticalBarrierSignal*).
- *CloseGripper*: the gripper is closed, applying onto the object a force controlled by the pressure sensors.
- *ReachSafePos*: the arm is raised a little bit in order to bring the object up.
- *ReachTransportPose*: a transport position, adapted to the mobile navigation, is reached by the arm. The agent returns then to its initial state.

On the top of Figure 3 are presented two events that are managed by all the states (an arm motion fails, or an emergency stop is asked by the user). Whatever is the current state of the agent, it will follow the defined transition. For the moment, simple solutions are proposed to react to these failure situations. If an emergency stop is pressed, the robot is

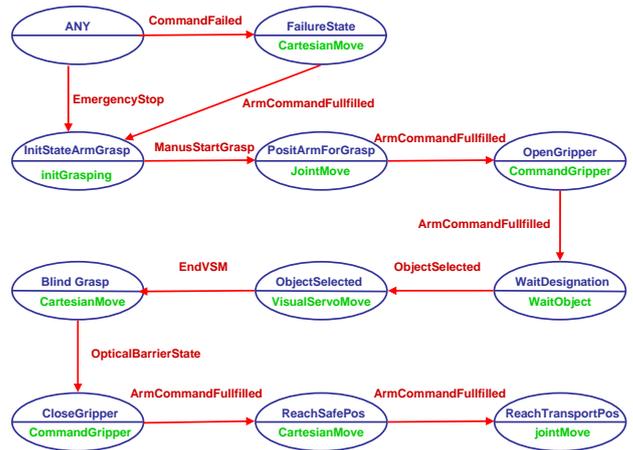


Fig. 3. Scenario associated to the grasping procedure

stopped, and the agent is reinitialized. If a motion command fails, a cartesian displacement is performed to move back the arm, and the agent is also reinitialized.

If an event not scheduled by the current state occurs, it is not taken into account (since it can be waited by another scenario agent). If several agents are able to process the same event at an instant, it will be considered by all these ones.

One of the interests of the ISEN library is that scenarios are dynamically interpreted, which means that the scenario can be changed without recompiling the overall application.

V. EXPERIMENTAL RESULTS

The robotic system SAM has been validated during the evaluation of the ANSO project⁵. The navigation space, simulating an apartment where rooms were realized with vertical boards, had a dimension of 60 m^2 . The robot was asked to go to another room, and the user selected there an object the mobile had to grasp and bring back.

A. Navigation

Figure 4 illustrates the autonomous localization and navigation procedures provided by the Neobotix society for the mobile unit. The environment is represented by orange lines. The laser output are the green points. When it is possible, these points are merged into lines, represented in blue. The white rectangles show the features that are used to localize the mobile. The blue cones correspond to the scanner field of view. On this figure, the mobile platform detects an obstacle (person, trough its legs) onto its nominal path (the roadmap, corresponding to the bright blue lines). A local planning generates the red trajectory that avoids this obstacle.

B. Object grasping

Figure 5 illustrates the diversity of objects the robotic arm is able to grasp. Since no model is needed, any object presenting a sufficiently textured appearance can be considered (*ie* Harris features can be extracted from the object).

⁵a video is available on Anthony.Remazeilles.free.fr

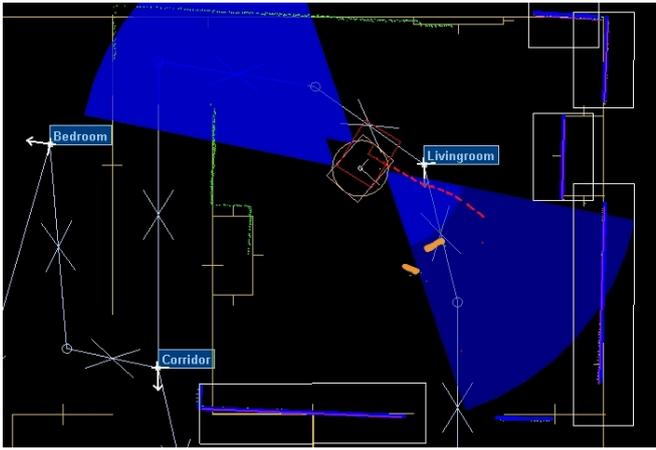


Fig. 4. In order to avoid a person (legs seen as two orange lines in front of the mobile), the robot generates a new trajectory (red dashes), veering away from the robot from its nominal path (snapshot of the Neobotix interface).

Figure 6 presents the tracker output on an image sequence where a planar object is moved in the direction of the stereo rig. In this demonstrative experiment, the box was defined with four points, in order to match with the position of the tag on the object. The points displayed are tracked in the optical flow and used to estimate the pose of the object. The result of the pose estimation is illustrated by the projection of the four box corners. Even if the forward displacement is significant, the tag position is well estimated. Notice that in the final application, the operator only observes on his remote screen the current box position, which is a sufficient information for a non-expert user.

The figure 7 illustrates a vision-based positioning of the arm. The graph presents the evolution of the position error of the centroid of the object during the task. Only the translation part is considered. The artifacts observed at the beginning of the motion are due to the clearances that are quite important on this low-cost arm. But it does not disturb the overall task. The tuning parameter (λ in eq. (6)) is here defined so that the initial velocity is set to 22 mm/s . The usual exponential decrease of the error is observed. At the end of the motion, the error in positioning is less than 5 mm on each direction. This error is sufficiently small to start the blind motion and grasp the object. The gripper frame was required to be at 200 mm of the box centroid.

During experiments with patients, the behavior of the visual servoing was changed in order to get a quicker displacement of the arm. The norm velocity was saturated to 22 mm/s when it was higher. This gives an affine decrease of the error, as it can be seen on the two diagrams of figure 8. The point tracking in both view and the pose estimation is realized in less than 150 ms . The global grasping procedure lasts around 15 seconds, which is an acceptable duration in our context. We are currently investigating an automatic tuning of the convergence parameter, in order to insure that no oscillation happens around the desired position.



Fig. 5. Different kind of objects correctly grasped (pepperpot, can, book, rule, bottle, cup).



Fig. 6. Tag tracking. Points used are displayed (images from right camera).

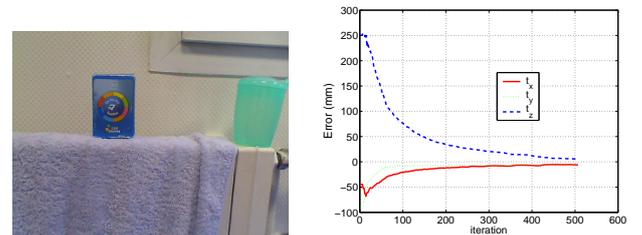


Fig. 7. Evolution of the object position error (in mm) during the visual servoing. The frame $(\bar{x}, \bar{y}, \bar{z})$ is following the classical pinhole model.

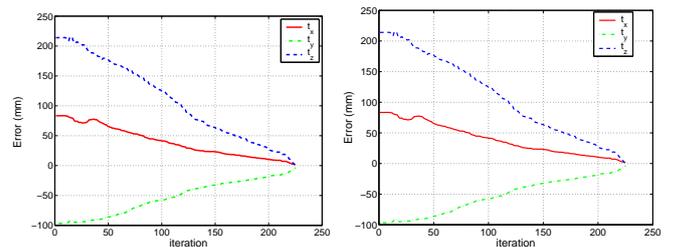


Fig. 8. Error of positioning of the object (in mm) during the navigation, for the pepperpot and the can of Figure 5.

VI. CONCLUSION & FUTURE WORKS

This article gave an overview of the different layers we have compiled to get an autonomous mobile system able to grasp any object outlined by the user. The grasping procedure is very easy to launch, and no *a priori* information

about the object is needed. We use a scenario interpreter to schedule the different actions the robot has to perform. The communication uses a recent protocol, web services, that enables to consider the robotic system as a common device inside an intelligent home, while hiding all the complexity of this communication.

Clinical evaluations have validated our grasping procedure (without the mobile platform). Within four french clinical centers (Garches, Coubert, Reims and Berck), handicapped people and valid persons used our system during one month, at the end of 2007 (the results are currently studied by statisticians). Technical evaluations of SAM realized at the hospital of Brest at the beginning of the year enable us to prepare SAM for clinical evaluations scheduled in two French centers, Berck and Kerpage.

This system represents a research platform that will enable us, according to the end-user comments and advices, to continue and improve the integration of new capabilities. We are currently integrating a novel and alternative grasping procedure that takes into account the shape of the object, in order to select the best grasping position, skill that does not provide our current technique. We will also investigate the use of the object that has been grasp. It is indeed necessary to implement automatic services like drink, or lay down a book. Future work will also consider the implementation of a color-based tracking for the management of non textured objects. We are also planning to add an object retrieval module, which could be used to make the object detection easier, and also give some information on the appropriate grasping position. Furthermore, the association of semantic information to the stored object could improve significantly the interaction between the robot and the user. At last, we would like to take benefit of the dynamical interpretation of the scenarios to give the possibility to the patient assistant to define themselves the different tasks the mobile has to perform, depending on the user needs.

REFERENCES

- [1] R. Alqasemi, E. Mccaffrey, K. Edwards, and R. Dubey. Wheelchair-mounted robotic arms: analysis, evaluation and development. In *IEEE ICAIM*, pages 1164–1169, Monterey, USA, July 2005.
- [2] Z. Bien, D.J. Kim, M.J. Chung, D.S. Kwon, and P.H. Chang. Development of a wheelchair-based rehabilitation robotic system (KARES II) with various human-robot interaction interfaces for the disabled. In *IEEE ICAIM*, volume 2, pages 902–907, 2003.
- [3] R. Bischoff and V. Graefe. Machine vision for intelligent robots. In *IAPR Works. on Machine Vision Applications*, pages 167–176, Tokyo, Japan, Nov. 1998.
- [4] F. Bley, M. Rous, U. Canzler, and K. Kraiss. Supervised navigation and manipulation for impaired wheelchair users. In *IEEE SMC*, volume 3, pages 2790–2796, The Hague, Netherlands, 2004.
- [5] M. Busnel, R. Cammoun, F. Coulon-Lauture, J. Détriché, G. Le Claire, and B. Lesigne. The robotized workstation "MASTER" for users with tetraplegia: Description and evaluation. *JRRD*, 36(3), July 1999.
- [6] A. Casals, R. Villa, and D. Casals. A soft assistance arm for tetraplegics. In *TIDE Congress*, pages 103–107, Bruxelles, Belgium, Apr. 1993.
- [7] F. Chaumette and S. Hutchinson. Visual servo control, part I: Basic approaches. *IEEE RAM*, 13(4):82–90, Dec. 2006.
- [8] J. Dallaway and S. Robin. Raid - a vocational robotic workstation. In *IEEE ICORR*, Keel, UK, Sep. 1992.
- [9] F. Dionnet and E. Marchand. Robust stereo tracking for space robotic applications. In *IEEE IROS*, pages 3373–3378, San Diego, USA, Oct. 2007.
- [10] H. Efrting and K. Boschian. Technical results from MANUS user trials. In *IEEE ICORR*, pages 136–141, Standford, USA, July 1999.
- [11] B. Graf, M. Hans, and R. Schraft. Care-o-bot II-development of a next generation robotic home assistant. *Autonomous Robots*, 16(2):193–205, Mar. 2004.
- [12] C. Harris and M. Stephens. A combined corner and edge detector. In *Avey Vision Conf.*, pages 147–151, Manchester, UK, Aug. 1988.
- [13] H. Heck, M. Larsson, C. Bühler, and G. Bolmsjö. Development of a speech control system for the MANUS manipulator. In *Assistive Technology on the Threshold of the New Millennium*, pages 297–301, Düsseldorf, Germany, Nov. 1999.
- [14] M. Hillman. Rehabilitation robotics from past to present, a historical perspective. In *IEEE ICORR*, Daejeon, South Korea, Apr. 2003.
- [15] P. Hoppenot and E. Colle. Localization and control of a rehabilitation mobile robot by close human-machine cooperation. *IEEE Trans. on Rehabilitation Engineering*, 9(2):181–190, June 2001.
- [16] F. Jammes, A. Mensch, and H. Smit. Service-oriented device communications using the devices profile for web services. In *AINA Work.*, pages 947–955, Washington, USA, May 2007.
- [17] K. Kawamura, S. Bagchi, M. Iskarous, and M. Bishay. Intelligent robotic systems in service of the disabled. *IEEE Trans. on Rehabilitation Engineering*, 3(1):14–21, Mar. 1995.
- [18] Z. Kazi, M. Beitler, M. Salganicoff, S. Chen, D. Chester, and R. Foulds. Intelligent telerobotic assistant for people with disabilities. In *SPIE Telemanipulator and Telepresence Technologies II*, volume 2590, pages 120–130, Orlando, USA, Apr. 1995.
- [19] C. Martens, N. Ruchel, O. Lang, O. Ivlev, and A. Graser. A friend for assisting handicapped people. *IEEE RAM*, 8(1):57–65, Mar. 2001.
- [20] M. Mokhtari, B. Abdulrazak, R. Rodriguez, and B. Grandjean. Implementation of a path planner to improve the usability of a robot dedicated to severely disabled people. In *IEEE ICRA*, volume 3, pages 4023–4028, Taipei, Taiwan, Sep. 2003.
- [21] Neveryd and G. Bolmsj. Walky, an ultrasonic navigating mobile robot for the disabled. In *TIDE Congress*, pages 366–370, Paris, France, 1995.
- [22] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.
- [23] J. Shi and C. Tomasi. Good features to track. In *IEEE CVPR*, pages 593–600, Seattle, USA, June 1994.
- [24] R. Soyama, S. Ishii, and A. Fukase. The development of mealassistance robot 'myspoon'. In *IEEE ICORR*, pages 88–91, Daejeon, South Korea, Sep. 2003.
- [25] D. Taddeucci and P. Dario. Experiments in synthetic psychology for tactile perception in robots: steps towards implementing humanoid robots. In *IEEE ICRA*, volume 3, pages 2262–2267, Leuven, Belgium, May 1998.
- [26] M. Topping. An overview of the development of handy 1, a rehabilitation robot to assist the severely disabled. *Artificial Life and Robotics*, 4(4):188–192, Dec. 2000.
- [27] H. Van der Loos. VA/stanford rehabilitation robotics research and development program: Lessons learned in the application of robotics technology to the field of rehabilitation. *IEEE Trans. on Rehabilitation Engineering*, 3:46–55, Mar. 1995.
- [28] H. Van der Loos, J. Wagner, N. Smaby, K. Chang, O. Madrigal, L. Leifer, and O. Khatib. Provar assistive robot system architecture. In *IEEE ICRA*, volume 1, pages 741–746, Detroit, USA, May 1999.
- [29] I. Volosyak, O. Ivlev, and A. Graser. Rehabilitation robot FRIEND II - the general concept and current implementation. In *IEEE ICORR*, pages 540–544, Chicago, USA, June 2005.