Localisation by camera of a rehabilitation robot

O. Ait Aider, P. Hoppenot, E. Colle
CEMIF - Complex System Group - University of Evry, 40, rue du Pelvoux, 91020 Evry Cedex, France.
e-mail: oaider | ecolle | hoppenot @cemif.univ-evry.fr

Abstract : *Robotics can provide technological solutions for improving the quality of life of motor disabled or elderly people. The main objective is to give person hours of independence by using a mobile base mounted arm. Because of particular constraints of this field of application the machine is semiautonomous and requires a close human machine co-operation for its control. In order to develop a user oriented machine control it is significant to determine the limits of autonomy of an affordable robot.*
After a description of the whole assistance system the paper focuses on the localisation problem of the mobile robot in a partly known environment.

Key-words : *mobile robot, man-machine cooperation, camera pose determination.*

1. Introduction

The day-to-day difficulties of people with disabilities are being more seriously taken into account with respect to accessibility, job market integration, medical assistance, etc. The primary objective of rehabilitation robotics has been to either fully or partially restore the disabled user's manipulative function by placing a robot arm in the user's task environment. Assistance systems currently available on the market require major transformations of the house. On the contrary, semiautonomous mobile robots are relevant configuration due to their potential for minimising the required degree of home adaptation.

The emergence of robotic solutions in performing aid tasks for people with disabilities is only realistic under two conditions. The first one concerns the very principle of the aid function. The system must not substitute, but rather compensate for the activity deficiency of people with disabilities. The second condition is the cost of providing this assistance. Cost effectiveness constraints imply the reduction of complexity and hence the system's autonomy. This loss of autonomy must be compensated by a close human machine co-operation.

The person intervention degree during the task progress is variable. It can begin by taking part in perception or decision functions until a remote control of the system. The partial autonomy of the system completes the field of person abilities either the handicap or to realise tedious actions.

Among the main today’s life functions listed by WHO (World Health Organisation) ([1]), several actions like carrying, grasping, picking up, moving, are “robotisable”. Different kinds of project have been presented in [2]. First ones are workstation-based systems. A table-mounted robot arm works in an environment where the position of different objects is known by the system. HANDY1 [3] and DeVAR [4] are two examples. Second kinds of projects are stand-alone manipulator systems where the object position is not known. This allows more flexibility but needs sensors for the environment perception: Tou system [5] and ISAC [6]. Other solutions are wheelchair-based systems. The most well known system is MANUS [7]. Mobile robot systems are also used: WALKY [8], Health Care Robot [9], URMAD [10], MOVAID [11] and MOVAR [12]. The last kind of systems proposed is collaborative robotic aid systems where multiple robots perform several tasks for the user [13].

The project ARPH (Assistance Robotics to Handicapped Person) is developed in collaboration with AFM (French Association against Myopathies). It belongs to the mobile robot system category. A manipulator arm is mounted on a mobile robot (Figure 1). The mission consists in carrying and manipulating an object in a partially known environment such as a flat.

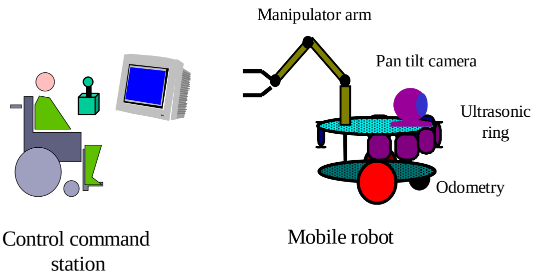

Figure 1: System architecture

The paper focuses on the displacement of the mobile robot.

Before defining precisely a well-adapted co-operation it is necessary to evaluate limits of robot autonomous abilities.

Section 2 presents briefly the assistance system called ARPH. Solutions imple-mented for the robot displacement are developed in section 3. Localisation of the mobile base is a key function but difficult to solve in a partially known environment with a limited perception system. The last section proposes an approach based on a camera.

2. Assistance system

The assistance system is composed of an arm mounted mobile robot and a control station.

Mobile robot

Robot is cylindrical-shaped, 90 cm high and 70 cm large. The manipulator arm is a MANUS developed by Exact Dynamics. Perception system is composed of an odometer , an ultrasonic ring and a camera.

The camera mounted on a pan and tilt base is a commercial device dedicated to general surveillance applications. In auto-tracking mode that camera can automatically follows the movement of an object. Camera plays three roles: i) a perception device which provides video feedback during the robot displacement, ii) a control device which provides robot the direction to follow or the object to reach or follow (if the object is mobile), iii) a localisation device to palliate odometrical errors.

Control station

The Control station is composed of:

i) control devices adapted to the handicap of the disabled person

ii) a screen which displays different types of information via enhanced reality techniques, such as video image of what is seen by the robot, virtual aids superimposed onto the video image, robot position on a 2D flat plan, virtual camera point of view, robot operating indicators (Figure 2)[14].

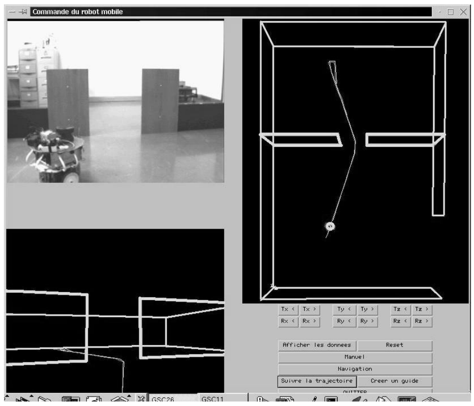

Figure 2: *Example of information displayed on the screen.*

Control modes

There are three main mode types to control of the robot displacement: automatic, manual and shared modes. For all the missions, the operator chooses the goal of the mission. It can be performed by pointing out the destination on a 2D flat plan displayed on a screen, by designing an object at a known position (the fridge for example) or by designing an object at an unknown position (a book for example). In the automatic mode, the robot computes its trajectory and reaches the destination avoiding obstacles. In the manual mode, the person teleoperates the robot manually via a joystick or any control device. The operator chooses the goal on-line. In shared modes the control of the degrees of freedom of the machine is shared between man and machine. Different combinations can be

imagined , for instance, the person defines the goal driving the tilt and pan base of the camera. The auto-tracking function of the camera is used to pilot the robot to the goal. The manual mode can also become a shared mode if the user is assisted by the robot to avoid obstacles automatically.
In order to realise a mission the person builds strategies based on a succession of control modes. Multiple strategies can be developed depending on the operator, the mission or the situation of the robot.

3. Autonomous displacement

A displacement of a mobile robot requires three functions: planning, navigation and localisation. Planning determines the best path to go from one point to another. Navigation ensures the robot follows the planned path avoiding obstacles. Localisation provides the position and the orientation of the robot in the flat at any time. In ARPH these functions imitates human-like behaviours.

Planning

The problem is to reach a goal. A person uses different strategies of planning. For a far destination a plan is used to find a way to go from one point to another. If the destination is within sight the person reaches the interest point following the direction he looks at.
In our application the system has the same human behaviour. In a classical robotic approach the robot computes a path through the flat to reach the goal using the known flat plan [15].
The second way to plan a trajectory is to use the camera in an auto tracking mode. The person points out a goal with the camera. The goal must be within sight of the camera. The camera tracks the goal, for example object, automatically. The robot moves in the direction pointed out by the camera. This is a human like behaviour. The object is considered as a target that can be mobile. The remaining issue is only to avoid obstacles on the path. This is a navigation problem.

Navigation

The problem is to follow the planned trajectory. A person divides navigation into two behaviours: goal-seeking and obstacle avoidance. A fusion of the two behaviours is performed during the displacement. The orientation of the head defines the direction for goal seeking. If an obstacle is on the way, the trajectory is deviated locally to avoid it. Usually people try to walk as far as possible from obstacles, for example in the middle of corridors.
Automatic navigation imitates the human behaviour making the fusion of goal-seeking and obstacle avoidance. For goal-seeking direction is defined by relative positions of the robot and the goal. If a non-modelled obstacle stands on the robot path, it must be avoided. Ultrasonic sensors detect these obstacles and fuzzy logic manages obstacle avoidance. As human like behaviour, the robot goes in the middle of the free space.

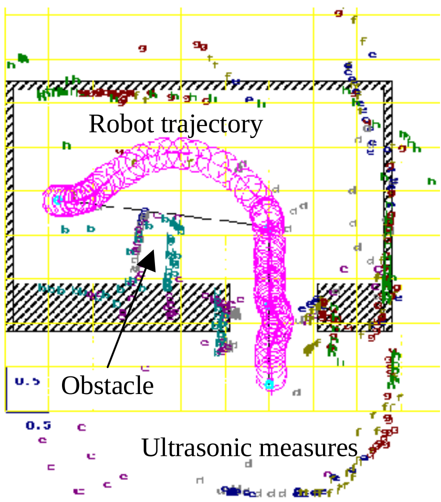

Figure 3: *Fusion of two behaviour, obstacle avoiding and goal-seeking for robot navigation.*

The fusion of two behaviours takes into account only obstacle avoidance when an obstacle is near the robot. When the distance between obstacles and the robot grows up, goal-seeking behaviour takes more importance in the robot command. Figure 3 shows a trajectory followed by the robot with a non modelled obstacle in the room. All these results are detailed in [16].

Localisation without camera

The first approach has been based on ultrasonic and dead reckoning measures. Two levels of localisation are provided. The first level assumes robot knows approximately position and orientation. The location is updated on-line by dead reckoning. A line fitting technique applied to ultrasonic measures and a matching with a 2D-model of the flat allows the correction of dead reckoning errors regularly and on line. In the second level, robot is considered lost and the mission in progress is interrupted . In this case dead reckoning is inefficient and only ultrasonic measures can be used. The robot localisation is found by matching between a great amount of US measures and a 2D-model which is the plan of the Flat. As person lost in a town, the robot looks for landmarks in the room. If some landmarks are recognised, the robot is able to compute its location with the help of the map. Results are detailed in [17][18].

In summary, the limit of the approach is due to objects which are not modelled. Because of the physical characteristics of the ultrasonic wave (specular reflection and multi-bound), objects mask the modelled part of the environment and prevents matching between measures and 2D-model.

The following section proposes another approach based on the camera.

4. Localisation by camera

The location of a robot can be computed from a single image of the environment provided by an on-board camera. It is assumed that the co-ordinates of the camera are known in the robot frame. Some 2D relevant features are extracted from the video image and matched to 3D landmarks of the environment. Several methods based on a set of points or straight lines has been proposed in the literature and can be divided into analytical and numeral methods.

Dhome in [19] proposes an analytical solution using a straight lines matching. Equations of the inverse perspective projection is solved by several intermediate co-ordinate frames in order to reduce the number of unknowns. The method leads to multiple solutions. The ambiguity is removed by applying a set of logical rules. In [20], an algebraic method using 3, 4, 5 or n matchings and a study about the number of solutions are presented.

Numerical methods determine the camera location by computing an approximation of optimal rotation and translation by iterative algorithms which minimise an error function. The expression of the error function depends on the formalism which represents transformation, for example for rotation, quaternion in [21] or Euler angles in [22]. Liu presents in [22] an iterative algorithm to determine rotation after an equation linearisation. Translation is then computed by the least square method. Results remain accurate as long as angles of rotation are less than 30° in the work space frame.

As seen before our approach consists of a two level localisation. In the first level dead reckoning provides an approximate location of the robot. In this case camera replaces ultrasonic ring to limit increasing error and reach a defined accuracy. Dead reckoning data initialise camera location algorithm. The second level corresponds to the case robot is completely lost without even an approximate knowledge of its position.

The following work deals with the first level which considers the following assumptions:
i) Environment is partly known, the model is a Brep representation in which walls, edges and corners of the flat are integrated ;
ii) Dead reckoning provides a robot approximate location about 10° for orientation and 50cm for x and y axes;
iii) Camera is calibrated.

The approach follows five stages: images acquisition, image segmentation and 2D features extraction, matching between 2D-image and 3D-model features, camera co-ordinate computing and finally robot co-ordinate computing. That work concerns the fourth step only.

Camera perspective projection model

This is the most used model [21, 23, 24]. (O_c, X_c, Y_c, Z_c) is the camera frame. Z_c corresponds to the optical axis. Image plane is perpendicular to Z axis at the focal length f (figure 4.1). The projection of a 3D point p (x_c, y_c, z_c) is p' (fx_c/z_c, fy_c/z_c, f). (u,v) is the pixel coordinates of p in the display plane.
The model is represented by a 3x4 matrix $\mathbf{M_{int}}$ which allows the computation of the 2D projection in display plane of a 3D point p(x_c,y_c,z_c) expressed in the camera frame.

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \mathbf{M_{int}} \cdot \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}$$

with $\mathbf{M_{int}} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

α_u, α_v, u_0, v_0 are called intrinsic parameters of the camera and are determined by calibrating [21,24,25].

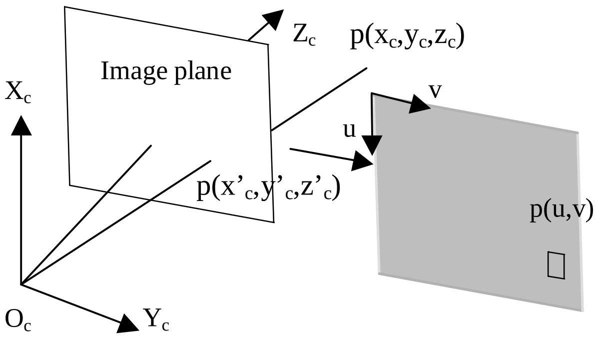

Figure 4.1: *Perspective projection model*

Mathematical formulation of the camera localisation problem

Let us consider a straight 3D line L_i defined by its direction vector $\mathbf{v}_i$ and its position vector $\mathbf{p}_i$ in a co-ordinates system related to the work space frame. $\mathbf{v'}_i$ and $\mathbf{p'}_i$ are the expressions of $\mathbf{v}_i$ and $\mathbf{p}_i$ in the camera frame, l_i is the projection of L_i in the image plane. L_i and l_i belongs to a projection plane passing through the focus point O_c. Let $\mathbf{n}_i$ be the unit vector normal to this plane in the camera co-ordinates system (Figure 4.2). It is possible to deduce $\mathbf{n}_i$ knowing l_i and the intrinsic parameters of the camera. Let $\mathbf{R}$ and $\mathbf{T}$ be respectively the rotation matrix and the translation vector between camera and work space frames. It can be written:

$$\mathbf{v}_i' = \mathbf{R}\mathbf{v}_i$$
$$\mathbf{p}_i' = \mathbf{R}\mathbf{p}_i + \mathbf{T}$$

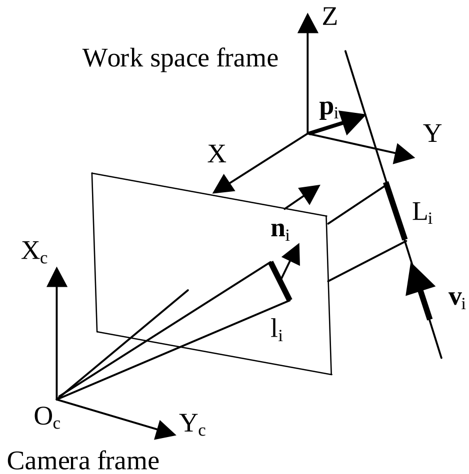

Figure 4.2: *3D line perspective projection*

As L_i and l_i belongs to the same projection plane, the two following scalar product equations are verified :

$$\mathbf{n}_i.(\mathbf{R}\mathbf{v}_i) = 0$$
$$\mathbf{n}_i.(\mathbf{R}\mathbf{p}_i + \mathbf{T}) = 0$$

A set of n 2D-3D straight line matchings (i=1,2,3,…,n) leads to n equation couples.

Computing the rotation

Rotation and translation problems are first separated and then an error function is minimised by a least square method. This error function can be deduced from scalar products seen above. One part of the problem is to find the simplest expression of the function. Using the representation of rotation by a unit quaternion the error function becomes a sum of quadratic terms [21] :

$$f(\mathbf{r}) = \sum_{i=1}^{n} (\mathbf{r}^t A_i \mathbf{r})^2 + \lambda(\mathbf{r}^t\mathbf{r} - 1)^2$$

where $\mathbf{r}$ is the unit quaternion representing the rotation $\mathbf{R}$, λ a positive constant and

$\mathbf{A}_i = \mathbf{Q}(\mathbf{n}_i)^t\,\mathbf{W}(\mathbf{v}_i)$ with

$$\mathbf{Q}(\mathbf{n}_i) = \begin{bmatrix} 0 & -n_{ix} & -n_{iy} & -n_{iz} \\ n_{ix} & 0 & -n_{iz} & -n_{iy} \\ n_{iy} & n_{iz} & 0 & -n_{ix} \\ n_{iz} & -n_{iy} & -n_{ix} & 0 \end{bmatrix}$$

$$\mathbf{W}(\mathbf{v}_i) = \begin{bmatrix} 0 & -v_{ix} & -v_{iy} & -v_{iz} \\ v_{ix} & 0 & v_{iz} & -v_{iy} \\ v_{iy} & -v_{iz} & 0 & v_{ix} \\ v_{iz} & v_{iy} & -v_{ix} & 0 \end{bmatrix}$$

Computing the translation

To compute the translation the previous rotation is first applied on the work space co-ordinate system. The transformation between the new system and the camera co-ordinate system becomes a pure translation. In this configuration, it was shown in [22] that the norm of the projection of the translation vector in the direction of each normal vector $\mathbf{n}_i$ is equal to the distance between a point p_{0i} lying to the line L_i and a plane Π parallel to the projection plane and passing through O'. It is then possible to write :

$$\mathbf{n}_i.\mathbf{T} - \mathbf{n}_i.(\mathbf{R}^{-1}\mathbf{p}_{0i}) = 0$$

This must be right for each line of the 2D-3D matchings set. Theses equations permit to compute **T** translation vector by a least square method with 3 or more 2D-3D line correspondences, with at least three of them not intersecting at the same point.

Discussion

The presented method needs at least three 2D-3D correspondences. It is assumed that in this application context (structured environment) the programme will be able to extract 3 or more straight lines from the segmented image and to match each line with the corresponding edge in the work space. 3 correspondences is the minimum required and for this case the solution may be multiple. Many authors [19, 20] studied the number of solutions for 3 line or point correspondences. One solution is to apply some simple rules to eliminate false solutions [19].

The other problem is convergence and computation time. The more initialisation of rotation and position is close from the final solution the more the convergence will be fast with avoiding local minima. The method presented in [27] uses first an analytical method to find an approximate solution. It is then used to initialise the iterative method.

For off-line localisation time constraint is not very hard. When segmentation does not give enough data it is possible to move the camera and remake the acquisition step. During on-line localisation time constraint is more important but it is possible to use results of odometry to initialise the image based method with a very close position to the solution and make the algorithm converge in a very short time.

Simulation results

The presented method has been tested on synthetic images corresponding to several known positions. The rotation **R** is represented by the three Euler angles phi, theta, psi. The translation **T** is represented by its components (T_x,T_y,T_z). Images contain a set of straight lines which can represent the edges of a corner in the work space formed by two walls and the floor. Image series correspond to a trajectory in which the robot gets closer to the corner and goes around it in the same time. Four lines (figure 4.3) were selected as 2D-3D correspondences. The orientation and the position from where the view was taken is computed for each image. According to the performance of the odometry system, the distance between the initialisation position and the real solution was 10° for the angles and 50 cm for each component of the translation vector. relevant results are presented in figure 4.3.

The first simulation was realized without noise. Results shown that the accuracy is very good and the error is less than 0.01° for the angles and less than 0.001° for the translation.

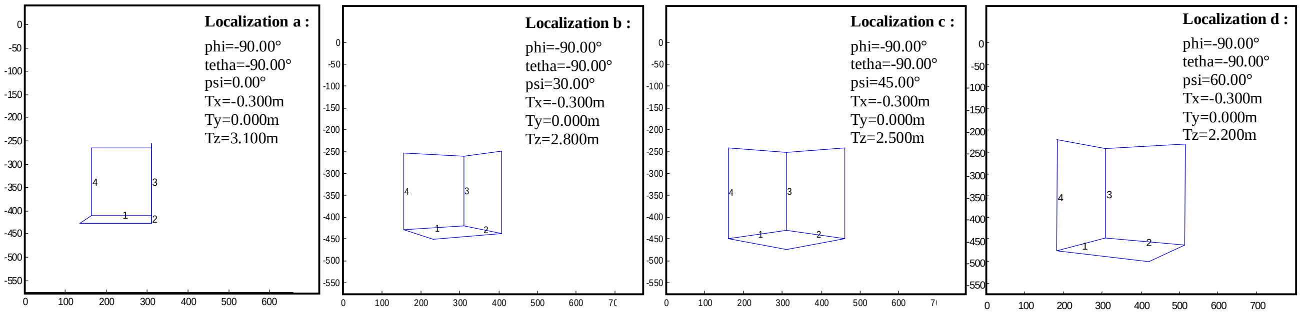

Figure4.3.a: *synthetic images corresponding to different known localisations.*

	Localization a		Localization b		Localization c		Localization d	
	MV	SD	MV	SD	MV	SD	MV	SD
Δphi(°)	0.21	0.14	0.18	0.13	0.19	0.12	0.18	0.13
Δtetha(°)	1.81	1.34	0.31	0.18	0.21	0.13	0.26	0.19
Δpsi(°)	0.02	0.01	1.45	0.93	1.23	0.79	0.97	0.80
ΔTx(m)	0.002	0.001	0.003	0.002	0.004	0.003	0.007	0.005
ΔTy(m)	0.000	0.000	0.001	0.000	0.001	0.000	0.001	0.000
ΔTz(m)	0.020	0.013	0.027	0.022	0.040	0.023	0.050	0.042

Table 4.3.a : *results for nl=0.01.*

	Localization a		Localization b		Localization c		Localization d	
	MV	SD	MV	SD	MV	SD	MV	SD
Δphi(°)	0.32	0.25	0.40	0.26	0.38	0.26	0.41	0.24
Δtetha(°)	4.66	2.96	0.61	0.48	0.40	0.31	0.58	0.35
Δpsi(°)	0.07	0.06	2.99	2.27	2.01	1.54	2.11	1.74
ΔTx(m)	0.005	0.003	0.007	0.005	0.009	0.006	0.014	0.012
ΔTy(m)	0.002	0.001	0.002	0.001	0.003	0.001	0.002	0.001
ΔTz(m)	0.041	0.027	0.063	0.044	0.077	0.051	0.104	0.057

Table 4.3.b : *results for nl=0.05.*

Figure4.3.b: *error mean values(MV) and standard deviations(SD) on computed localisations.*

In the second and the third simulations noise is introduced in the 2D data by adding random values from a uniform distribution in the range [-nl,+nl] to the components of the normal vectors $\mathbf{n}_i$ (nl is the noise level). Tables 4.3.a and 4.3.b represent mean values and standard deviations of obtained errors with 50 tests in each position. For nl=0.01 the accuracy is still satisfying application context since it is assumed that the acceptable errors are about 1° for angles and 10 cm for positions. When nl=0.05 results show that some errors are important (Δtetha=4.66 in positon 'a' or ΔTz=10.4 cm in position 'd'). Note that big errors on translation were obtained when orientation errors were also important. This is because the computed rotation is used to compute the translation. It seems clear that errors in the first step affect results of the second one. The choice of correspondence lines is also important. Results obtained with psi=90° show that the algorithm does not work with this special configuration. Note that for this case we obtained an accurate solution using four other lines.

5. Conclusion

A method of localisation using a single perspective view was presented. Simulation results show that it can be used on assistance mobile robot since it satisfies accuracy exigency of a such application (about 1° for rotation and 10 cm for translation) when noise in 2D data is reasonable. Use of a single view (less image data to treat) and co-operation with odometry (good initialisation of the algorithm) permit also to reduce computing time.

The method can be improved using an error function which optimises simultaneously the translation and the rotation. This will certainly increase the robustness in presence of noise.

In simulations it was assumed that the 2D-3D matching was correctly carried out. This problem is generally solved by testing all the attitudes compatible with the image in a prediction-verification procedure [26,27]. The number of possibilities can be important increasing the computation time. Our objective is to introduce odometry results and logical rules to limit the number of solutions.

References

[1] World Health Organization : "International Classification of Functioning and Disability" - Beta-2 Draft, short version, July 1999.

[2] K. Kawamura, M. Iskarous : "Trends in Service Robots for the Disabled and the Elderly" - Special session on Service Robots for the

Disabled and Elderly People, 1994, pp. 1647-1654.

[3] M. Topping, J. Smith : "The development of Handy 1, a rehabilitation robotic system to assist the severely disabled" - Industrial Robot, Vol. 25, n°5, 1998, pp. 316-320.

[4] H.F. Van der Loos : "VA/Stanford Rehabilitation Robotics Researchand Development Program : Lessons Learned in the Application of Robotics Technology to the field of Rehabilitation" - IEEE Trans. on Rehabilitation Engineering, Vol. 3, n°1, march 1995, pp 46-55.

[5] A. Casals, R. Villa, D. Casals : "A soft assistance arm for tetraplegics" - 1st TIDE congress, April 1993, pp. 103-107.

[6] K. Kawamura, S. Bagchi, M. Iskarous, R. T. Pack, A. Saad : "An intelligent robotic aid system for human services" - AIAA/NASA Conf. On Intelligent Robotics in Fields, Factory, Service and Space, March 1994, vol. 2, pp. 413-420.

[7] R. D. Jackson : "Robotics and its role in helping disabled people" - Engineerinf Science and Educational Journal, Dec. 1993.

[8] H. Neveryd, G Bolmsjö : "WALKY, an ultrasonic navigating mobile robot for the disabled" - 2nd TIDE Congress, Paris, 1995, pp. 366-370.

[9] P. Fiorini, K. Ali, H. Seraji : "Health Care Robotics : a Progress Report" - IEEE Int. Conf. On Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 1271-1276..

[10] P. Dario, E. Guglielmelli, B. Allotta : "Mobile robots aid the disabled" - Service Robot, vol. 1, n°1, pp 14-18, 1995.

[11] E. Guglielmelli, P. Dario, C. Laschi, R. Fontanelli : "Humans and technologies at home : from friendly appliances to robotic interfaces" - IEEE Int Workshop on Rob. and Human Com., 1996.

[12] H. F. Van der Loos, S. J. Michalowski, L. J. Leiffer : "Development of an omnidirectional mobile vocational assistant robot" – proceedings of RESNA'88, eleventh annual conference on rehabilitation technology, 1988, 468-469.

[13] K. Kawamura, M. Cambron, K. Fujiwara, J. Barile : "A cooperative robotic aid system" - Virtual Reality Systems, Teleoperation and Beyond Speech Recognition Conf., 1993.

[14] S. Otmane, E. Colle, M. Mallem, P. Hoppenot : "Disabled people assistance by a semiautonomous robotic system" - SCI'2000, vol. 3 Virtual Engineering and Emergent Computing, pp. 684-689, Orlando, July 2000.

[15] M. Benreguieg, P. Hoppenot, H. Maaref, E. Colle, C. Barret : "Fuzzy navigation strategy : Application to two distinct autonomous mobile robots" - Robotica 1997, vol. 15, pp 609-615.

[16] Hoppenot P. , Benreguieg M., Maaref H., Colle E. and Barret C. : "Control of a medical aid mobile robot based on a fuzzy navigation" - IEEE Symposium on Robotics and Cybernetics, july 1996, pp 388-393.

[17] P. Hoppenot, E. Colle : "Real-time localisation of a low-cost mobile robot with poor ultrasonic data" - IFAC journal, Control Engineering practice 1998, vol. 6, pp.925-934.

[18] P. Hoppenot, E. Colle and C. Barat : "Off line localisation of a mobile robot using ultrasonic measures" - Robotica, 2000, Vol 18, pp. 315-323.

[19] M. Dhome, M. Richetin, J.T. Lapresté, G. Rives : "Determination of the attitude of 3-D objects from single perspective view" – IEEE Trans. On pattern analysis and machine intelligence, vol. 11, N°12, 1989, pp. 1256-1278.

[20] L. Quan, Z. Lan : "Linear N-point camera pose determination" - IEEE Trans. On pattern analysis and machine intelligence, vol. 21, N°8, 1999, pp. 774-780.

[21] R. Horaud, O. Monga : "Vision par ordinateur, outils fondatmentaux" – Hermes, Paris, France, 1993.

[22] Y. Liu, T.S. Huang, O.D. Faugeras : "Determination of camera location from 2D to 3D line and point correspondences" - IEEE Trans. On pattern analysis and machine intelligence, vol. 12, N°1, 1990, pp. 28-37.

[23] O.D. Faugeras : "three dimensional computer vision : a geometric viewpoint" – MIT Press, Boston, 1993.

[24] P. Puget, T. Skordas : "Calibrating a mobile camera" – Image and vision computing, vol. 8,1990, pp. 341-347.

[25] J. Triboulet, F. Chavand : "Calibration of a Multisensor System" - IMACS /IEEE-SMC Int. Conf. CESA'96, Robotics and Cybernetics, pp. 664-669, Lille, 9-12 Juillet, 1996.

[26] M. Mallem, M. Shaheen, X. Dourille, F. Chavand : "A matching method between an image and its 3D-model using a geometric constraints aproach based on contact" – CESA’96 IMACS Multiconference, pp.565-569, Lille, France, 1996.

[27] M. Shaheen : "Reconnaissance d’objets polyhedriques à partir d’un image vidéo pour la téléopération" – Thèse de Doctorat, université d’evry-Val d’Essonne, 1999.