View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by HAL-CEA

The modelling system PYRAMIDE as an interactive help for the guidance of the inspection vehicle CENTAURE

Philippe Even, Lionel Marcé, Joël Morillon, Robert Fournier

To cite this version:

Philippe Even, Lionel Marcé, Joël Morillon, Robert Fournier. The modelling system PYRAMIDE as an interactive help for the guidance of the inspection vehicle CENTAURE. [Research Report] RR-1061, INRIA. 1989. <inria-00075498>

HAL Id: inria-00075498

https://hal.inria.fr/inria-00075498

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNITE DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 1061

Programme 6
Robotique, Image et Vision

THE MODELLING SYSTEM PYRAMIDE AS AN INTERACTIVE HELP FOR THE GUIDANCE OF THE INSPECTION VEHICLE CENTAURE

Philippe EVEN
Lionel MARCE
Joël MORILLON
Raymond FOURNIER

Août 1989

RR_1061

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Publication Interne n° 480 - Juin 1989 - 22 Pages

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone : 99 36 20 00
Télex: UNIRISA 950 473 F
Télécopie: 99 38 38 32

The modelling system PYRAMIDE as an interactive help for the guidance of the inspection vehicle CENTAURE

—

PYRAMIDE; outil interactif de modélisation géométrique pour l'aide au pilotage du véhicule d'inspection CENTAURE

Philippe EVEN*et **Lionel MARCE**
IRISA, INRIA, Rennes, France,

—

J. MORILLON et R. FOURNIER,
CEA, UGRA/SETA, Fontenay-aux-roses, France

Abstract

Teleoperation of vehicles and manipulators can become much easier if a geometric model of the environment and of the robot is provided. However in case of an accident in hostile environment, we do not have always such models. In order to answer this drawback, we conceived a Generalized Information Management System called PYRAMIDE, capable of modelling a 3-d world from camera views, of locating the robot in this environment and of producing synthetic displays for the assistance of the operator while guiding a robot. This article relates an experimental phase where PYRAMIDE was coupled with the intervention vehicle CENTAURE. Some results on the performance are given, and the potentialities offered in case of intervention are discussed.

Résumé

L'accès à un modèle géométrique de l'environnement et du robot peut faciliter grandement la téléopération des véhicules et des manipulateurs. Cependant, lors d'un accident dans un environnement hostile, de tels modèles ne sont pas toujours disponibles. Pour pallier à cet inconvénient, nous avons conçu un outil de gestion des informations généralisées, appelé PYRAMIDE, capable de modéliser un monde 3-D à partir d'images issues d'une caméra embarquée, de localiser le robot dans son environnement, et de produire des vues synthétiques pour aider l'opérateur lors du guidage du robot. Ce papier décrit une phase expérimentale où PYRAMIDE était couplé au véhicule d'intervention CENTAURE. La présentation des résultats obtenus est appuyée d'une discussion des potentialités du logiciel en cas d'intervention.

*sous contrat CFR avec le CEA

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L. A. 227)
UNIVERSITÉ DE RENNES 1 I. N. S. A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

1 Introduction

Our study occurs after conclusions of advanced teleoperation research carried out into the frame of the French ARA project [Espiau86]. The issue is remote control of manipulators and vehicles for intervention in case of an accident in hostile environment. The goal is to provide the operators with synthetic informations from a geometrical data base in order to fully benefit of the potentialities of a Computer Aided Teleoperation system. These synthetic feedbacks are made adequate by using the operator as a decision maker to indicate what kind of information he needs and what precision the tasks requires.

The mobile robot CENTAURE-I was used during an experimental phase to evaluate the consistency of the principles we propound. This robot is aimed at *following the natural path of a man in an installation* for achieving an intervention mission [Clement et al.87]. It cumulates the advantages of wheels, tracks and legs, combining an articulated track principle with a movable payload platform which provides active control of stability. In industrial environments, lot of passages are conceived anthropomorphically, and become obstacles for a robot (stairs, narrow places, ...). The nice abilities of CENTAURE to cross most of these obstacles are not always easy to use, due to the loss of geometrical informations between the remote site and the control station. This reason puts CENTAURE as the best candidate to evaluate our system. We indicate first how a geometrical data base can be used to improve a mobile robot teleguidance.

To increase system capacities, we can improve data feedback and exhibit them in such a way that operators feel like working in the remote worksite (telepresence concept [Tachi et al.85]). In some circumstances, for instance in bad visibility, the operators are provided with degraded data, which can be partly restored if an environment model of the world in which the robot is running is available. Our philosophy is based upon the concept of Generalized Information Feedback [Andre et al.86].

A geometrical data base in which the robot is located, can be used to build up a realistic synthetic image of the environment, as it would be seen through robot sensors. In this way some sources of image blurring (smoke, obstacles, ...) can be fictitiously eliminated. The synthetic image can also be symbolic. The geometric model allows transformation of data to be displayed to operators in an easily understandable way. For instance range to an obstacle can be displayed with a visual indicator of bar graph type. This geometric model allows operators to record data, for instance a sign posted direction, or guiding marks, in order to avoid their short term memory overloading.

But the geometrical model affords more than to improve operators information; it also permits the programming of tasks. Trajectories and targets can

be specified or automatically computed [Faverjon84], potential areas can be defined, for obstacles avoidance or objects handling [Khatib85, Andre83]. In the same way as in robotics, the geometrical model can be used for characteristics evaluation during system conception. Operators can for instance estimate if cameras position is accurate (occlusion test [Sakane et al.85]). This simulation also allows the display of command effects. This aspect is very important during task debugging. Operators can for instance choose a good way to reach a goal (reach test) and also to get one's hand when no error is allowed. The goal can be to limit error risks in case of irreversible commands as to provide operators with anticipation possibilities when simulation is running on-line. As for aircraft simulators in aeronautics, a geometrical model allows operators training and the reduction of the learning stage.

2 The PYRAMIDE geometric modelling system

The goal of PYRAMIDE is to create, to fill up, or to modify a geometrical 3d model of a robot environment. This one is usually unknown. Some objects are possibly normalized. Sometimes the operator disposes of a priori geometrical data bases; however some objects may be shifted. The operator himself estimates the degree of precision necessary for his work; typically he can fully model work areas and only partially model the obstacles. These features allow to reduce the modelling time and the display time (the less numerous the objects are to visualize, the more possible is animation). They also allow the suppression of useless data which focus and seize attention.

The modelling process is a cooperation between a priori CAD data bases, an embedded multisensor system including on board cameras, operator's knowledge and the computation power of an IRIS graphics workstation. Computers can provide geometrical models of simple volumes, brought together and scaled to form more complex objects. These virtual objects are set up by operators in the real world of the vehicle and their projection is mixed to the projection of real objects. Operators match very quickly the two projections in tuning up parameters of the synthetic model.

The PYRAMIDE geometrical modelling unit architecture is based on a controller that shares messages between four modules. The data base allows the world model updating, the robot model actualization, the selection of a geometrical primitives library and the dynamic management of data. To insure a flexible man-machine interface conception an Object-Oriented approach was chosen. The different kinds of informations are shared between six windows and managed with interactive interface tools (mouse, potentiometers, button

box, ...). The geometrical reasoning module is a library of programs that solve some 3d problems. Amongst these are methods to compute the camera attitude parameters, to retrieve the depth of an object in the scene from its image projection, and to compute the model uncertainties. These points are discussed more thoroughly in [Even et al.88].

3 Integration of PYRAMIDE in the control station

An experimental phase held in december 1988 at the teleoperation site of the UGRA laboratory. The PYRAMIDE system was coupled with the control station of CENTAURE. The scope was to estimate the feasability of the modelling method in realistic conditions and the relevance of the built data base to help an operator when piloting the robot. Amongst the tested fonctions were the modelling process, the resetting of data bases on the environment, the robot location, the training possibilities, and the assistance tools for the pilotage.

A serial bus RS-232 was used to drive data between PYRAMIDE and CENTAURE control station. Images from an embedded CCD camera equipped with a $6.4 \times 4.8mm$ pickup area and a $8.5mm$ format lens were displayed on a TV monitor. An IBM PC AT equipped with a MATROX PIP 1024 acquisition card was used to digitalize the video images. The eight grey levels scaled 512×512 images were then transmitted to PYRAMIDE on ETHERNET with FTP protocol.

The lens was calibrated at IRISA [Chaumette et al.89] with a high accuracy method. Images of a 48 points rectangular grid were acquired at five distinct heights of the camera. These image points were identified to the known 3-d points. Then a convergent algorithm was used to estimate the internal and external camera parameters with high accuracy (table 1). The optical center is located from the lower left corner of the image. The focal length is defined in pixels by taking into account the pixel size.

Parameters	X axis	Y axis
Optical centre	267 pels	283 pels
Focal length	722 pels	1019 pels
Y / X ratio	1.41	
Distorsion rate	-8.26×10^{-7}	

Tab. 1: *The lens internal parameters.*

The knowledge of the radial distorsion rate makes possible the off-line image correction. So we can estimate the influence of this factor on our method.

4 Geometrical resetting of the data base on the environment

In this section we discuss some techniques to link the frame of the camera with a frame chosen in the 3d scene. Many difficulty levels are faced according to the prior knowledge on camera external parameters and to the required accuracy of the resetting.

4.1 Full resetting

When operators have not any previous knowledge on the position and attitude parameters of the camera, a full resetting must be proceeded. Six parameters (three translations and three rotations) are to be computed. This work can be performed in two steps; first we determine the camera attitude parameters, then the computed frame is translated to its correct location in the 3-d world.

The computation of the attitude parameters can not be easily done manually. In our implemented method, operators must indicate three image segments, parallel to the 3d world frame axes. These segments do not necessary concur. Typically one of them corresponds to a vertical line and the remaining ones lie in horizontal planes and are orthogonal. Most images of structured environments include such features. From the equations of these three segments, two analytical solutions are computed; one provides operators with a convex frame, the other with a concave one. The implemented algorithm generalizes a method that has been developped at LIFIA / IMAG in Grenoble [Horaud et al.89]. More details can be found in [Even et al.89]. The computed frame is displayed at an arbitrary location on the image. Then the operator selects it, translates it at a convenient place, often an obstacle vertex, and specifies the height of the camera in order to make sure the frame lies on the floor.

To evaluate the accuracy of this method, many series were executed by an operator with a good knowledge of the method. The image we chose comprises many segments with distinct features. These segments are labelled on figure 1. In each series, 3 segments were chosen and a full resetting operation was executed around 26 times. The influence of the segment length was tested. We also evaluate the influence of the distorsion. To do it, we worked on the distorted video image and on a corrected version, and we chose segments at different locations (central or peripheral) and orientations (radial or tangent). The table 2 gives the experimental conditions for each series.

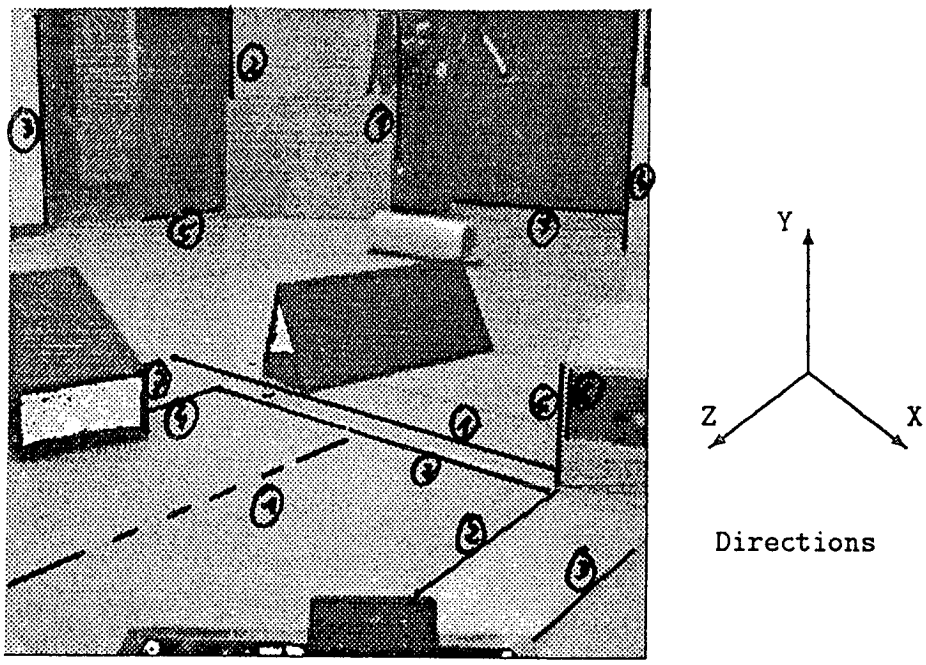

Fig. 1: *Labelling the segments.*

Series	Segments X axis	Y axis	Z axis	Chosen image	Execution mean time
1	2	14	1	video	20 s.
2	2	14	1	corrected	20 s.
3	2	3	1	video	20 s.
4	2	3	1	corrected	20 s.
5	2	1	1	video	20 s.
6	2	1	1	corrected	20 s.

Tab. 2: *Experimental conditions of each series.*

The figure 2 gives the 5 computed parameters and the standard deviation for each series. The real values are displayed in bold. The accuracy obtained with good segments ($\approx 200pels$ and with radial orientations), is about $10cm$ for the positional parameters (P_x, P_z), and 0.2^o, 2°, and 1° respectively for the R_x, R_y, and R_z rotational parameters (series 1). Very minutious pointings do not affect considerably the accuracy, The distorsion correction is not very effective with such segments (series 2). On the other hand, important offsets appear when one of these segments lies in the peripheral part with tangent direction (series 3). As shown on series 4, this offset is avoided on the corrected image. If one of the segments is short (under $100pels$), the mean value is still correct but important standard deviations occur (series 5 and 6).

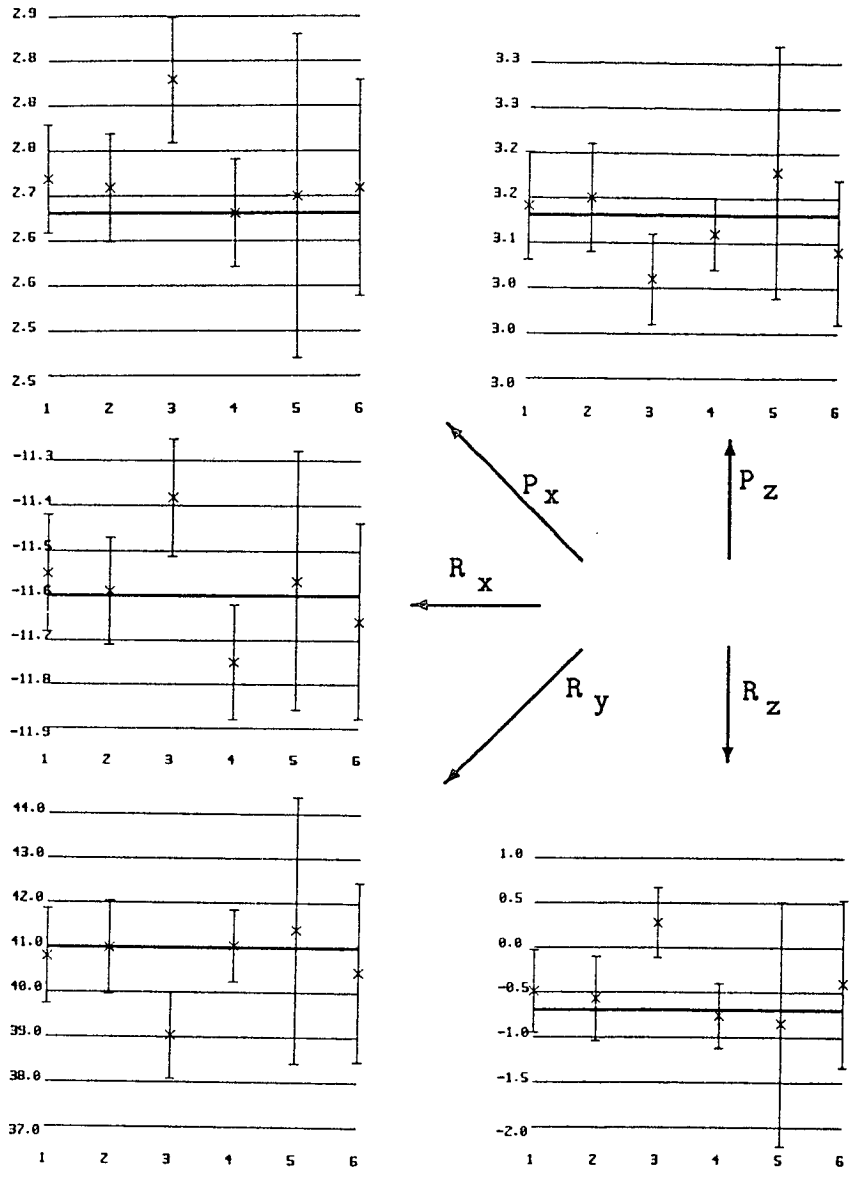

Fig. 2: *The camera position (cm) and attitude (°) for each series.*

To estimate the easiness of use, many unexperienced operators were observed and interviewed. As first result, a poorness of the method is the bad knowledge on the accuracy that can be reached in each experimental situation. The operators try to get the best matching possible. So the execution times increase

drastically. Some feedback on the reached accuracy should be produced from the geometrical features of the retained segments. We notice also that a precise pointing on a screen is a visualy very tiresome work. In order to avoid this focus, some automatic segments extraction should be implemented so that the operator must just indicate what segment to be kept.

4.2 Partial resetting

If we have more knowledge on the camera external parameters, a partial resetting can be performed. When modelling an environment, the robot moves on the floor, so the height and the trim of the camera do not change. Only three parameters must be reset, that are two translation parameters from the previous location to the present one, and a rotation parameter corresponding to the azimuthal variation. This can be done manually. First the operator translates the model frame at a location that is visible in the new image. This location is the center of azimuthal rotation of the whole model. Then he selects the whole built model. The wire-frame display is superimposed on the video image. With the mouse and a potentiometer, the operator slides and orientates it until it matches correctly the 2d view.

The robot can be roughly located in a very short time. If a good accuracy is necessary (for example when obstacles are near), more attention is required from the operator to obtain the best matching. In case of a good matching, the robot location is given with an error less than $5cm$. However perfect matchings are seldom. The reason is that during the robot displacement, the trim of the camera changes slightly (a few tenth of degrees). This drawback comes mainly from the floor irregularities, the displacement of the inertial centre and the elasticity of the tracks. In order to reduce this bad effect, buttons have been programmed to adjust the roll and pitch angles. Nevertheless the adjustment of these parameters is difficult. This operation requires from operators some ability to manipulate objects in 3 dimensions. So a few minutes are often necessary to get an already exact matching.

5 Modelling an environment

When modelling a teleoperation environment, the goal is to acquire enough geometrical informations to provide the operator with efficient synthetic feedbacks for an assistance during the pilotage phases. In the frame of intervention scenarios a critical requirement is to build this model as quickly as possible. So the operator manages a compromise between the modelling time and the model accuracy.

5.1 The model acquisition

At the beginning of the scenario, the mobile robot enters a roughly known room, that must be crossed along to reach a valve. Some obstacles are scattered over the floor. The only knowledge the operator has is the height of the camera. So after the first image acquisition he must achieve a full resetting. When this is done, he knows the position and the attitude of the camera related to the the frame that was chosen. So he can now establish geometrical constraints to make the modelling process easier. For example the mouse movements on its plate are linked with objects slidings on the floor, or with vertical displacements. In order to perform the modelling task in best conditions, the resetting operation must be executed as carefully as possible.

To model an obstacle, the operator selects a primitive, translates it from the scene frame to the obstacle, rotates and scales it so that the wired shape fits the obstacle at best. Some geometrical constraints may be used to get a better precision and speed up the task; they are for example identical scalings along both horizontal axes to keep bases square or circular, or contact detection to join objects side by side. If an obstacle shape is not included in the primitive box, the operator can gather many primitives, or take the obstacle in a surrounding volume.

The main visible obstacles are built to obtain a first model. Then the operator guides the robot in this modelled part of the environment and stops it at a location where new features and a known one are visible. From this known feature he achieves a refined partial resetting. To ensure a good modelling some care is required. To make the resetting easier and more precise some surfacing details are often useful. So lines, grids or blobs on the floor may be preferably modelled. When the resetting is over, the operator updates the model.

5.2 Accuracy and execution time

The figure 3 shows the environment modelled during the displacement from the door to the valve and the positions where the robot stopped. Five images (exp1 to exp5) were used to acquire the model. This one was compared with the real world caracteristics. Good accuracy is noticed in the surroundings of the robot path. The dimensions of obstacles near the robot (less than 5 meters) are given with an error less than 5 cm, and their position with an error less than 15 cm. Significant errors are noticeable on more distant obstacles. A very simple model but accurate enough to provide an effective help can be obtained in about fifteen minutes.

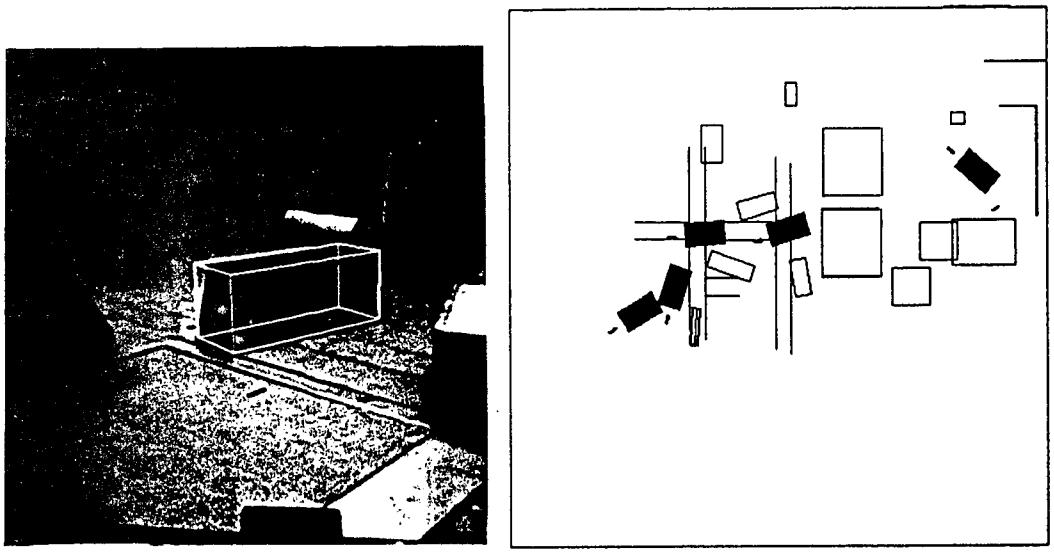

Fig. 3: *An image of the scene and a top view of the model.*

To give a better idea of the performance, an experiment was registered. A model was built by an experimented user from five views (exp1 to exp5). In order to obtain an accurate global model, careful partial resettings were performed. The table 3 gives the model acquisition time as the sum of a resetting time, a modelling time and a displacement time, the number of objects and primitives that have been handled on each view, and the number of objects and primitives added to the model. Here for example, 22 objects were built with 32 primitive in forty one minutes. On an image, the number of handled objects may differ from the number of modelled objects; some objects are modified due to the loss of informations on the previous views. The global time seems to be rather long, but it will be largely saved during the future displacements.

Images	Time				Handled		Acquired	
	Resetting	Modelling	Pilotage	Sum	Obj.	Prim.	Obj.	Prim.
exp1	40"	8'20"	2'	11'	13	14	13	14
exp2	40"	2'20"	3'	6'	5	5	3	3
exp3	2'30"	5'	4'	11'30"	7	15	5	13
exp4	1'	0	5'	6'	0	0	0	0
exp5	40"	2'50"	3'	6'30"	2	2	1	2
Global:	5'30"	18'30"	17'	41'	-	-	22	32

Tab. 3: *Acquisition of a model and its acquisition time.*

6 The simulation tools

Some simulation tools have been implemented from the built data base and proposed to many operators with different knowledge levels. The graphics are updated using the mouse or the joystick that usually controls CENTAURE.

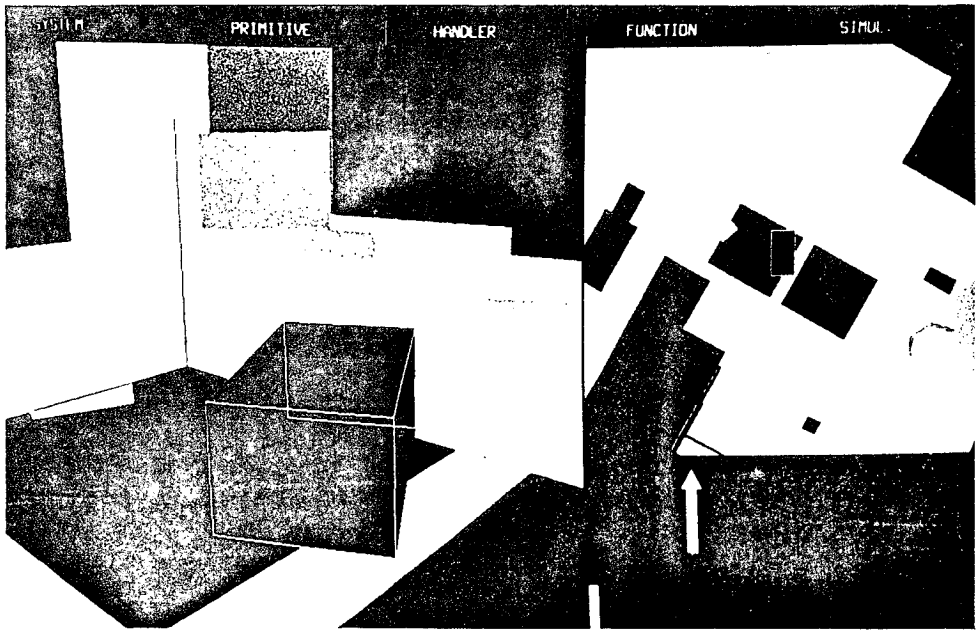

Fig. 3: *A simulation phase.*

Two modes have been tested to update the perspective view. In static mode a cuboid with the dimension of CENTAURE is driven over a video image or a coloured synthetic picture. To obtain a fast display time, only that cuboid is refreshed. This kind of pilotage is rather difficult. Indeed the robot may often go out of the field of view. Moreover it is hard to locate the robot with regard to the obstacles. This is due to the loss of the cuboid hidden parts removal when this one goes behind an obstacle. The only trustable indication is then the position of the cuboid base.

In dynamic mode the graphics is updated in a way to simulate what should be seen from an embedded camera on the fictitious vehicle. This synthetic feedback could be provided when the video information is lost. We meet here the same problem as when we pilot a real robot with the TV feedback, i.e. we do not have any distance or dimension information. This problem is made more drastic

on the synthetic display because of the regularity of the displayed surfaces. For instance a correct estimation of the moment when to turn in front of an obstacle requires a good training.

Two animation modes are also proposed to update the orthographic top view. In static mode the environment lies steady in the background and a rectangle is displayed on it for CENTAURE. This tool brings a very noticeable comfort for the robot pilotage. The static mode requires a good knowledge of the elements that are included in the data base. Indeed no height information is visible. So a table may appear as an obstacle, even though the robot can move under it. Moreover when the robot turns back, the command effects are reversed; so many operators turn on the right instead of the left.

In dynamic mode the rectangle lies steady at the center of the view, whereas the environment is updated around it. On this local view, the command effects reversal problem vanishes. So many users like it better. Meanwhile the goal to be reached does not necessarily appear on the view. So some tools should be added to provide the operator with orientation. Moreover this animation mode requires a larger display time.

An on-line collision detection module has been added. The collision test is just graphical and makes use of the IRIS workstation graphics abilities. So its implementation does not increase by far the display time. This tool is useful to know if the robot can pass between two obstacles or under one.

		Top view	
		static	*dynamic*
Perspective view	*static*	< 100 ms	200 ms
	dynamic	200 ms	400 ms

Tab. 4: *The refresh time for each simulation mode.*

The table 4 gives the refresh time of these various modes for a scene with an average complexity. For each dynamic mode selected, about 200 facets are to be updated. With a dynamic mode on both view, the image flickering is visible, but as the robot does not go fast, this flickering is still not troublesome for the pilotage. However if a complex object with a lot of facets enters the field of view, some delay arises between the command and the display.

7 The pilotage experiments

These many tools have been integrated into complete simulation experiments. A scenario was conceived in which some enrolled operators tried to pilot CEN-

TAURE from a start point, and to lead it in front of a valve through a path cluttered with many obstacles. Two methods were suggested and observed.

7.1 Pilotage with the synthetic feedbacks

In the first method an execution operator drives the robot using the TV feedback and also the synthetic ones (pilotage phase). The synthetic feedbacks are updated in accordance to the relative positional measurements that are given by an odometry sensor. This one includes two odometry wheels located under the robot centre. The measurements are sent every $x \times 100ms$ to PYRAMIDE, where x is a entire value fixed in relation with the simulation modes the operator chooses (table 4). They are encoded as two translational coordinates X, Y and a rotational coordinate ϕ (figure 5), and transmitted on the RS 232 serial bus.

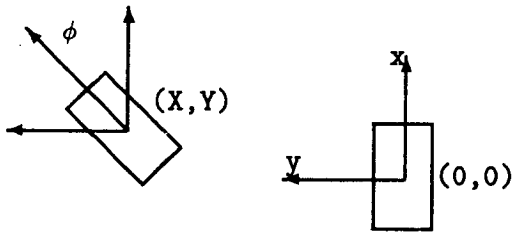

Fig. 5: *The odometry measurements.*

The odometry module is subject to some uncertainties that depend on many parameters (floor irregularities, sliding frictions, ...). As the robot goes on, these uncertainties cumulate. In order to give the operator an idea on the relative location of the vehicule, this one is displayed on the top view with a blue rectangle that increases in accordance to the path covered. The magnification ad hoc formula is given below. T stands for the current size and T_i for the initial one. k_d is the magnification factor in distance. It is set to the initial value $k_i = 4 \times 10^{-5}$. $k_o = 4 \times 10^{-7}$ is the magnification factor in angle. As indicated in this formula, the magnification factor in distance increases after each rotation that has occured.

$$T = T_i + k_d * \sqrt{X^2 * Y^2}, \qquad \text{with} \qquad k_d = k_i + k_o * |\phi|$$

The initial size of the robot is displayed as a red line superimposed on the magnifying rectangle. As the uncertainty on the vehicle location increases, the rectangle overlays the surrounding obstacles. That moment comes at best to proceed to a resetting of the robot position on its environment. So the execution operator stops the vehicle and the supervision operator acquires an image and transmits it to PYRAMIDE. Once the resetting phase is over, the pilotage

phase can go again.

This pilotage methodology was submitted to several operators. They found the synthetic display to be a very friendly help. Two ways to use it have been noticed. Many users look at the synthetic top view from time to time, just to check the robot position before steering near a corner. For instance when an obstacle is run along, it disappears from the camera field of view at a certain time. From that time, not any information is accessible to know when to turn without any risk to send the back-sided camera against the obstacle.

Other users better pilot the robot on the top view, and check sometimes the TV image agrees to the fictitious robot environment. The resetting phase is long and arduous. So the operators have a tendency to delay it. This fact is rather troublesome because when they decide to reset, there is not always enough geometrical details in the video image to act easily. Advertised operators do not wait and reset the robot position as plenty of details are still present.

The overall time spent to lead the robot from its start point to the valve varied from about ten minutes up to near an hour. The overall pilotage phase mean time was about four minutes and was negligible compared with the resetting phases durations. Two to six resettings were executed. As these operations were exaggeratedly careful, they were also time consuming.

7.2 Pilotage using the simulation tools

The second method begins with a simulation phase were the execution operator drives the simulated robot on the PYRAMIDE top view. When the simulation is over, the final position is encoded as two entire values, an angle θ and a distance r, and sent to CENTAURE on the serial bus as an order for an automatic displacement phase. This displacement is servoed by the odometry measurements, so resetting phases must be held sometimes.

As main result, the execution operator task is made easier. In counterpart the overall pilotage phase (simulation and automatic displacement) takes more time than in the first method. An other particular interest of that method is a decreasing number of resetting phases. Indeed when a pilot leads his robot with video or synthetic feedbacks, disturbing rotations are introduced while steering. These rotations are also visible during a simulation, but they are not taken into account in the order. So as the odometry uncertainties are especially sensitive to the rotations, the odometry measurements are longer reliable.

8 The fine modelling of a complex object

The last experimented functionality of PYRAMIDE was the fine modelling of a complex object, here a valve. While modelling this valve, many difficulties arised. First plenty of complex shapes are included (pipes, collars, bolts, an elbow, a wheel). Then the object leaned against stairs. So many rotations occured. To benefit of all the potentialities offered by PYRAMIDE, this work was achieved by an experimented user.

First a model was built from a single view. 12 cylinders were assembled in 9 minutes to form the valve. Some results on geometrical features are given in table 5. Other points of view showed a split object with large errors on lot of parameters.

To reach a better accuracy, we acquired an other view of the valve in such a way we could use the interactive stereometry module. The principle of this module is to match a graphical primitive on the chosen detail of the valve on the first view, then to set a modelling constraint to slide the primitive in that first view viewing pyramid, until the matching is reached on the second view. In order to get a good precision, the image acquisitions must be followed by precise full resettings.

Features	Measured value	Single view	Int. stereometry
Wheel diameter	20 cm	21 cm	21 cm
Wheel to pipe axis distance	36 cm	9 cm	36 cm
Pipe diameter	8 cm	8 cm	9 cm
Wheel to pipe base height	60 cm	56.5 cm	61.5 cm
Valve tilt (R_x)	$\approx 10^o$	0°	7°
Valve steering (R_y)	-45°	-23.1^o	-51°

Tab. 5: *Some modelled features of the valve.*

We obtained a model of the valve in about fourty minutes. Fifteen minutes were necessary to execute the resettings. The built valve (figure 6) includes 22 cylindrical primitives and 2 cuboidal ones. As shown on table 5, the values of the geometrical features are closer to the real ones. Some a priori knowledge was used to add the bolts. First the operator modeled one bolt with the interactive stereometry principle. When this bolt was built, the perspective and the top views were zoomed, and this first bolt was duplicated, positionned at the same height (on the collar). The eight duplicated bolts were then distributed around the collar on the top view.

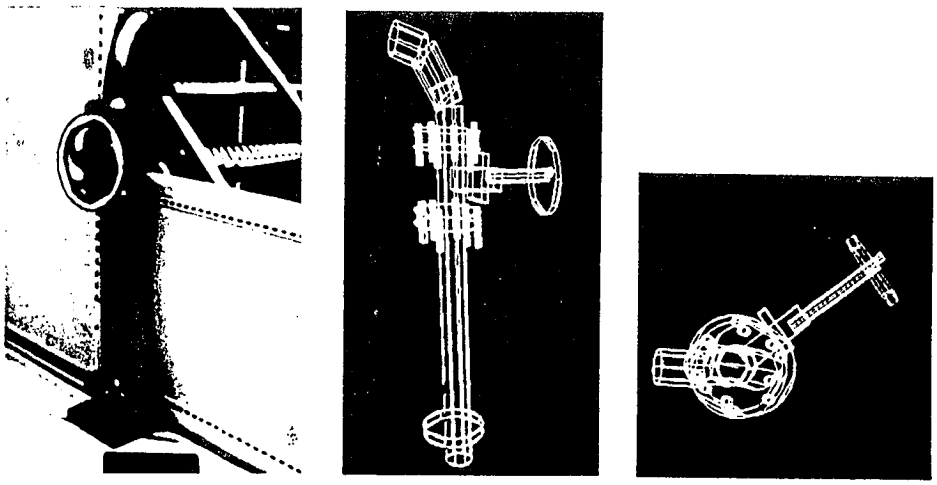

Fig. 6: *The valve and its model.*

In the present version of PYRAMIDE, the fine modelling of fine objects is not enough simple to get quickly a precise model. Some geometrical tools (principally the interactive stereometry module) can be used, but more automaticity must be provided and a data base of standard objects (bolts, collars, ...) should be used.

9 Conclusion

The Generalized Information Management System PYRAMIDE has proved to procure an effective help for the guidance of the mobile robot CENTAURE in case of an accident in a hostile environment. When there is not any a priori model of the environment, a simple model can be built in a very short time. This potentiality can be used to send CENTAURE in the damaged world as a scout to get a model other heavier robots will use. The metric informations provided by the built data base can be used by CENTAURE itself; it is quite possible to know if the robot will pass between two obstacles or to acquire an obstacle height and give that information to an automatic obstacle crossing module [Fournier et al.88]. After all the robot can be located accurately enough to enable synthetic feedbacks to help the operator during the guidance phases. In order to produce safer movements, the displacements can be simulated on the

synthetic displays and sent as an order for an automatic displacement. Accurate models of complex objects can also be obtained, and used for automatic tasks; however this operation is time consuming.

To remedy the remaining defaults of PYRAMIDE, further works are going on. Topological tools may be estimated to speed up the modelling process while improving the model consistency. Such tools are based on higher level graphics primitives management. Use of telemetric measurements should also provide stronger potentialities.

Acknowledgements

The experiments took place in the UGRA department of CEA (French Nuclear Agency) of which we wish to thank the members, particularly MM. Rouyer, Fraize and Clement for their invaluable help and their friendly collaboration, and MM. Cabirol, Dupont, Mrs. Destolles and the whole team of the UGRA for giving us their sympathetic user suggestions for improvement.
We thank also Mr. Chaumette for providing us with the camera calibration method.

Bibliographie

[Andre et al.86] G. André and A. Fournier.

The generalized information feedback concept in computer aided teleoperation.

RoManSy 86, Cracow, September 1986.

[Andre83] G. André.

Conception et modélisation de systèmes de perception proximétrique. Application à la commande en téléopération.

Thèse D.I., IRISA-Rennes I, Octobre 1983.

[Chaumette et al.89] F. Chaumette and P. Rives.

Réalisation et calibration d'un système expérimental de vision composé d'une caméra mobile embarquée sur un robot-manipulateur.

Publication Interne 454, IRISA - RENNES, Février 1989.

[Clement et al.87] G. Clement and E. Villedieu.

Mobile robot for hostile environments.

Proc. of the Int. Topical Meeting on Remote Systems and Robotics in Hostile Environments, 270–277, Pasco, Washington, March 29 - April 2, 1987.

[Espiau86] B. Espiau.

Advanced teleoperation.

RoManSy 86, Cracow, September 1986.

[Even et al.88] P. Even and L. Marcé.

Pyramide: an interactive tool for modelling of teleoperation environments.

IEEE Int. Workshop on Intelligent Robots and Systems, 725 – 730, Tokyo, Oct. 31 - Nov. 2, 1988.

[Even et al.89] Ph. Even and L. Marcé.

3d modelling of a teleoperation environment with pyramide.

Third Topical Meeting on Robotics and Remote Systems, 11,1,1 – 11,1,7, Charleston, South-Carolina, March 13-16, 1989.

[Faverjon84] B. Faverjon.

Obstacle avoidance using an octree in the configuration space of a manipulator.

IEEE Int. Conf. on Robotics, 504–512, Atlanta, March 13-15, 1984.

[Fournier et al.88] R. Fournier, P. Gravez, M. Dupont, and J.P. Gaillard.

Computer aided teleoperation of the centaure remote controlled mobile robot.

Int. Symp. on Teleoperation and Control, 97–105, Bristol, July 12-15, 1988.

[Horaud et al.89] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle.

An analytic solution for the perspective 4-point problem.

Computer Vision, Graphics, and Image Processing, 1989.

[Khatib85] O. Khatib.

Real-time obstacle avoidance for manipulators and mobile robots.

IEEE Int. Conf. on Robotics and Automation, 500–505, St. Louis, March 25-28, 1985.

[Sakane et al.85] S. Sakane, M. Ishii, and M. Kakikura.

Hand-eye simulator: a basic tool for off-line programming of visual sensors.

'85 ICAR, 103–110, Tokyo, September 9-10, 1985.

[Tachi et al.85] S. Tachi and H. Arai.

Study on tele-existence (ii): three-dimensional color display with sensation of presence.

'85 ICAR, 345–352, Tokyo, September 9-10, 1985.

LISTE DES DERNIERES PUBLICATIONS INTERNES IRISA

PI 474 **ALGEBRAICALLY CLOSED THEORIES**
Eric BADOUEL
22 Pages, Mai 1989.

PI 475 **QUELQUES OUTILS GRAPHIQUES POUR LA MODELISATION DU CONTROLE D'EXECUTION EN ROBOTIQUE DE COOPERATION**
Jean-Christophe PAOLETTI, Lionel MARCE
52 Pages, Juin 1989.

PI 476 **SIMULATION REPARTIE DE SYSTEMES A EVENEMENTS DISCRETS: PARTIE 1 : MODELISATION ET SCHEMAS D'EXECUTION**
Philippe INGELS, Michel RAYNAL
26 Pages, Juin 1989.

477 **PROGRAMMING WITH MALI - UNIFICATION OR ORDERED TYPES**
Olivier RIDOUX
18 Pages, Juin 1989.

PI 478 **CLASSIFICATION OF CONCEPTS DESCRIBED BY TAXONOMIC PREORDONNANCE VARIABLES WITH MULTIPLE CHOICE**
Israël-César LERMAN, Philippe PETER
16 Pages, Juin 1989.

PI 479 **A SIMPLE GRAPH CONSTRUCTION OF SEMILINEAR REACHABILITY SETS OF VECTOR ADDITION SYSTEMS**
Gilles LESVENTES
16 Pages, Juin 1989.

PI 480 **THE MODELLING SYSTEM PYRAMIDE AS AN INTERACTIVE HELP FOR THE GUIDANCE OF THE INSPECTION VEHICLE CENTAURE**
Philippe EVEN, Lionel MARCE
22 Pages, Juin 1989.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique