An artificial vision system for a table clearing robot : SERVIROB

J.N. Rodriguez L. Acosta A. Hamilton J.A. Mendez M.S. Saavedra L. Moreno

Department of Physics, Electronics & Systems

University of La Laguna. Spain

Abstract – In this paper a vision system for a highly specialized autonomous robot is presented. The task of the robot is the setting and clearing of tables in a restaurant. The environment in which the robot will operate will be controlled and well known. The first objective of the robot is to navigate around the room to find collectable sites, that is, tables. Specifically, this paper is focused on the task that has to be performed once the robot is located a table. This consists of analyzing the scene and extracting all the necessary information to collect the distinct objects located on it. The application is designed to identify and collect the following objects: dishes, bottles, glasses, forks, spoons, and knives.

Keywords:
Artificial vision, Autonomous robotics, Hough transformation.

I. INTRODUCTION

This work presents a vision system for scene analysis integrated in a service robot application (SERVIROB). This application consists of an autonomous robot for setting and clearing tables in a restaurant. To accomplish this task the robot is autonomously navigating in the room and has to fulfill various objectives. The first one is to identify where the tables are. Once at the table the robot should be able to identify and classify the objects on it. In this study we have considered the following objects:

- White dishes of two different sizes
- Metal forks, spoons and knives
- Clear glasses
- Clear bottles

These objects must be collected and deposited in individual recipients. A specially designed manipulator is mounted on the robot to accomplish this task. The vision module of the complete application analyses the scene environment when SERVIROB is in a position to collect objects. The objective of this module is to provide the necessary information to the main module of SERVIROB so that it can perform pick and place tasks in the table.

The solution proposed is to equip the robot with image analysis software and artificial vision system composed of two cameras.

Fig. 1 shows the set of layers that compound the software structure. The fact of working with a layer structure provides the necessary characteristics in relationship with real time requirements ,module comunications and system scalability. The module comunication is carried out through a set of FIFO (first in first out) files provided by the OS (Operative System). The OS kernel has been modified to change all the interruption requests by a set of macros that allow us to setup a real time layer between the kernel and the robot hardware. In this way the system has a real time response and allow us to setup parameters like priority proccesses and the maximun time for task execution. Over this layer a set of modules has been developed (manipulator Planner, mobile base Planner and Image Capture modules) that provide the interface between real time hardware control layer and high level software like vision module or navigation module. The last layer situated on top of the software structure is the general planner. This layer defines the general robot behavior in order to resolve all posible situations that can be found surrounding the problem that the robot tries to solve.

GVISION: A tool has been developed for the general purpose of capturing and processing images, provides a series of operators which are used when constructing classification strategies. These operators incorporate techniques which allow for improvements in the pre-processing of images, generation of contour maps, the calculation of generalised Hough transforms and moments. This tool has been developed using C++ with the graphics library GTK-- for Linux. In relation with the artificial vision system, the techniques used are based on the description of distinct elements, that can be found on top of tables, using geometric characteristics. The most important tool used in the vision module is the Hough transformation [1]. The geometric characteristics must be general enough to allow the system a margin of error, but at the same time be sufficiently defined so that the object can be identified without errors.

Although the environment is controlled and the objects to be collected are well defined, working in a real scene implies that objects can be overlapped or partially hidden. Also rubbish of the table is a noisy element in the scene that makes difficult the identification. In view of this, the vision system must be capable of extracting the necessary information to identify the objects in the scene even under adverse conditions.

II. DEFINITION OF COLLECTABLE OBJETCS

The objects that can be found in the scene will be described by a set of geometric primitives, more precisely, circumferences and straight line segments. These primitives will constitute the basis of a language that allows the definition of all the objects that can be

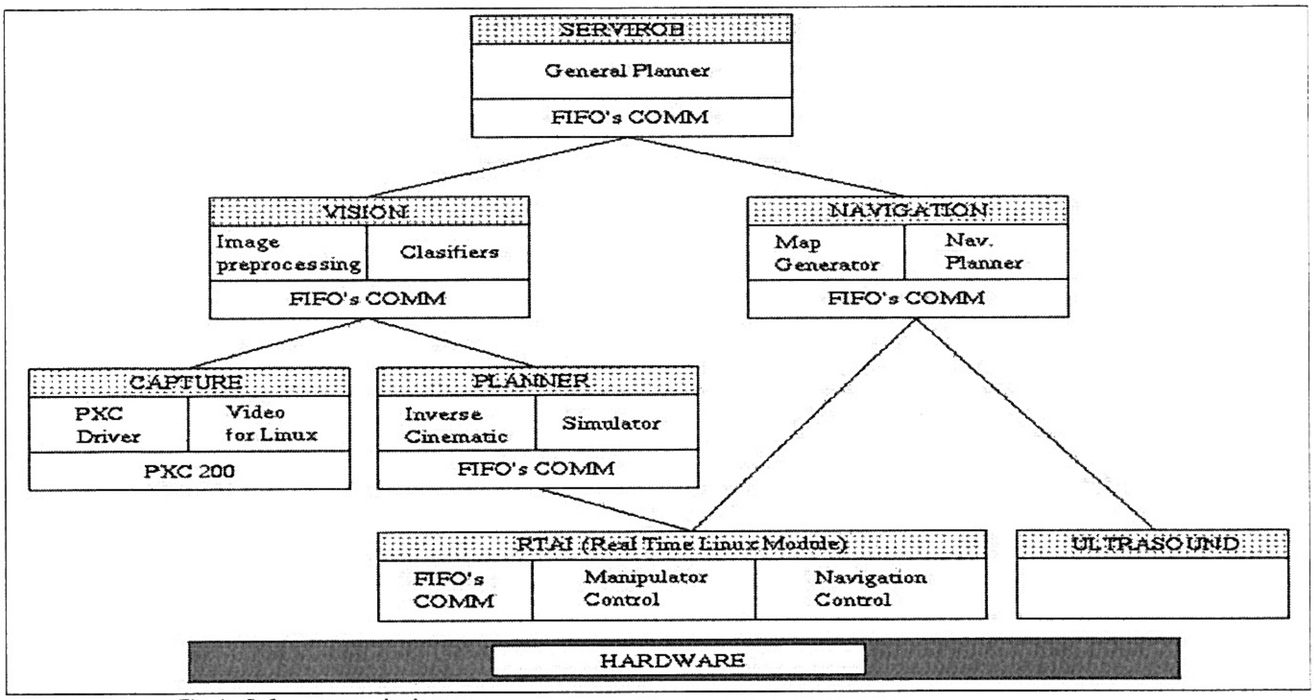


Fig. 1. Software general scheme

identified. Combining the description of all the possible primitives in a object together with the relationships amongst them, a vector of characteristics for each object is obtained. This vector is easily obtained from the images of the working scene, even when working with non-ideal conditions. As an example, in Figure 2 the definition of a spoon is shown.

It can be observed that, in this case, the set of primitives is composed of two circumferences with radius R1 and R2 and two segments with length L. The relationships that these primitives must hold in order to define a spoon are imposed by the relative spatial location between them.

Doing the same with all the objects that the robot must recognise, a knowledge base is built. This knowledge base will be used by the vision system in the processing of the acquired images.

This method has two main advantages. First and most important, because in the procedure isolated characteristics with a given relationship are used to define the objects, the presence of all these characteristics is not necessary in order to obtain a positive detection. In this way, overlapped or partially hidden objects can be identified.

The second advantage is that the use of a visual language to define the objects allows to create a user interface where the number of recognizable objects can be increased easily. This property is important as it allows the application of the system to recognise different types of utensils.

The analysis is carried out by means of an adaptive

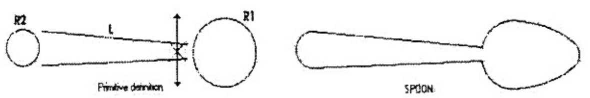


Fig. 2. Definition of a spoon using primtivesi

iterative process. The basic structure of this algorithm is shown in Figure 3. The process iterates until a stop condition is reached. The stop condition is the no detection of an object of type X during a predetermined number of iterations.

The general algorithm repeats the process shown in Figure 3 for each class of predefined object. The object order for detection is also prespecified and is imposed by a set of rules defined by the user depending on the specific problem to be solved. For the application presented in this paper these rules are:

a) Objects whose altitude may affect the manipulator trajectories in the collection task are collected first (bottles and glasses).

b) Once the bottles and glasses (in this order) are collected, forks, spoons and knives are set as the next target. The order is defined by the number of primitives that identify each object. The risk or error is reduced applying first the more restrictive classifiers (the classifiers with more primitives and relationships to hold). These classifiers are more robust to noisy images. Later, the less restrictive classifiers are applied.

It is worth pointing out that, as the number of collected objects increases, the scene noise decreases, thus improving the effectiveness of the less restrictive classifiers.

c) It has to be taken into account that some objects (forks, knives and spoons) can be on the dishes. So these objects are collected before the dishes.

III. ANALYSIS PROCESS

The first step is to capture the work scene. The capture module has been designed to work with two video cameras in order to obtain stereo vision. The cameras provide images with a resolution 384x288 pixels in grey-scale.

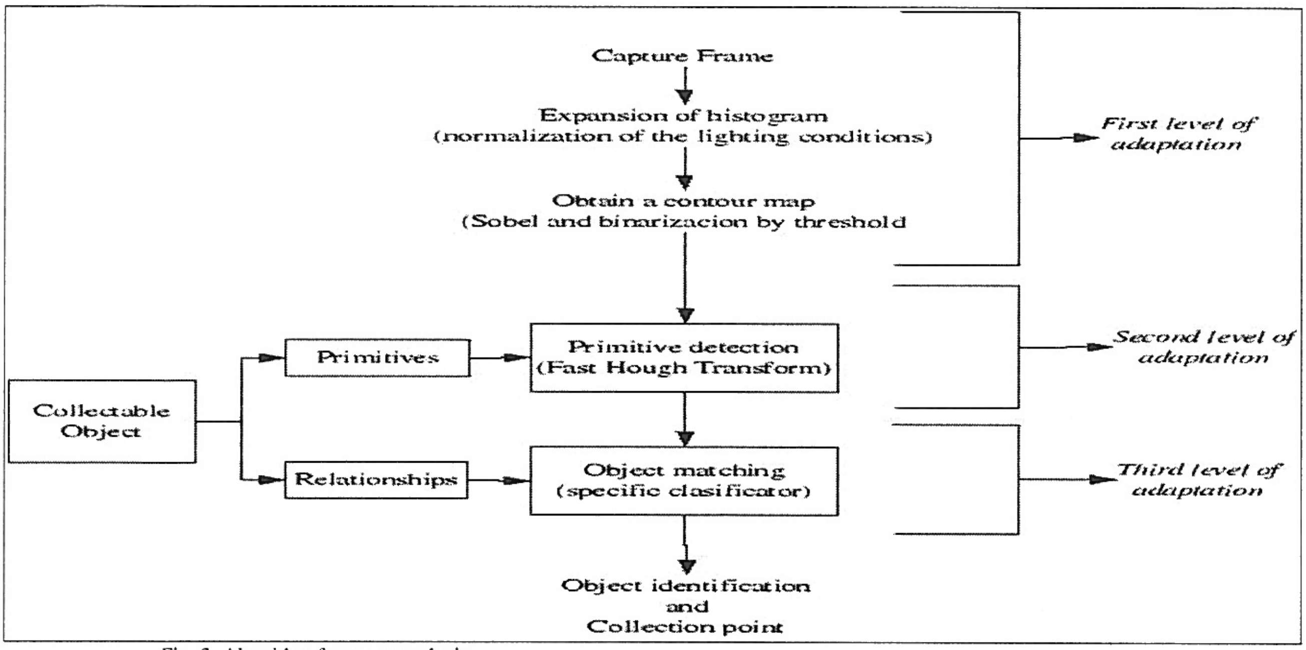


Fig. 3. Algorithm for scene analysis

The fact of working with the geometrical properties of the objects (primitive system), implies that the deformations introduced in the image generated by the camera lenses are a very important error source. The elimination of that error source via software adds a high computational load to the system. So it is important to use high quality lenses to avoid, as much as possible, the deformations in the image.

Once a scene frame is captured, the **preprocessing module** is started. It has to be taken into account that, if a positive detection is obtained in a method iteration, the detected object is collected. This implies a change in the working scene for the next iteration. That is why after the frame capture an adaptation process is carried out. This process is called *first adaptation process* and consists of a normalization of the lighting conditions. This is done because when an object is removed from the scene, the brightness and contrast conditions change significantly. Even the object distribution changes if overlapping existed before the collection.

The preprocessing module allows the application of several operators to achieve normalization of the brightness and contrast conditions in the image. Thus, the primitive detection efficiency is improved. The relief operators implemented are histogram matching, linear piecewise transformations, equalization and non linear parameterized operations.

The system implemented allows captures to try distinct strategies using the operators library in order to minimize the noise in the image. Once the best sequence for the specific conditions is defined, that sequence will be applied automatically on each on line capture.

After the normalization process is finished, the system begins the identification of collectable objects. This task is performed by the **classifier module**. This module first tries to locate primitives in the scene, then verifies the relationships for the specific object and finally generates the votes in the parameter space, that allows to find the object detected with a higher probability of success. Also, at this stage the adaptation parameters are evaluated to check its goodness. The first step in this module is to obtain a contour map of the scene. This map can be obtained in different ways because different operators were implemented to create it. After several experiments it was found that a first order operator, like Sobel operator, together with a binarization operator is the best choice. When Sobel operator is applied, a thick contour map is obtained (Figure 4). Later it will be shown that this map is essential for the construction of the vote map. Next, the specific primitives for the object are identified.

A. Primitive detection

The basic primitives defined in this work are circumferences and straight line segments. The system generates a list of primitives, grouped according to their properties. The circumferences will be grouped by its radio length. The segments will be grouped according to their length. One Hough operator has been implemented for the detection of circumferences and a different Hough operator for the detection of straight lines. Both operators are highly parameterized[2]. These parameters are

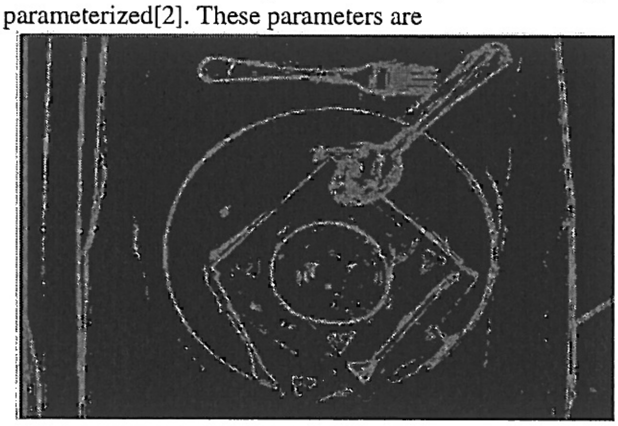


Fig. 4. Contour generation for an acquired image

R min	102 pixels
R max	107 pixels
threshold	0.6
round	2
fit	0.45 percent

Tab. 1. Hough Transform parameters

manipulated by an adaptation process called *second level of adaptation* process. In this way, the tolerance of the operators are tuned. More precisely the Hough operator for circumference detection has five parameters to be adjusted in this level (Rmin, Rmax, threshold, round, fit) as shown in table 1.

R min and R max are the radius range where the search is performed. The threshold parameter is used to select the set of admisible solutions. The round parameter refers to the length of the exploration cell in the image. Finally, the fit parameter indicate the minimun pixel percentage that must appear in the image to be considered as a valid primitive.

In table 1 the stablished values for the detection of the circumference primitives that define a dish are presented.

In this way, and increasing the tolerance gradually, the scene is analyzed to find first the primitives that are less affected by noise. These primitives define the objects that are best candidates to be collected (isolated and well defined). Later, the primitives that are more affected by noise are searched (overlapped and partially hidden). As commented above, the primitives are defined by a set of properties that allows their grouping. For circumferences, the system is asked to look for a given radio. For segments, the system looks for a given length. These are provided to the system by means of a specific range. For instance, for a circumference of radio R, a minimum and a maximum radio around R is provided. The system tries to find primitives into this range. In this way, there is an adaptation to small deviations of the distance between the camera and the table produced by vibrations of the structure or irregularities of the ground. These variations produce changes in the dimensions of the objects and, hence, noise in the scene.

Besides this, a threshold is introduced to indicate the percentage of the primitive that has to be visible to be considered as valid. Accepting partially hidden circumferences or segments as valid, produces a better system behaviour to detect partially hidden or overlapped objects. Figure 5 shows the detection of a valid primitive

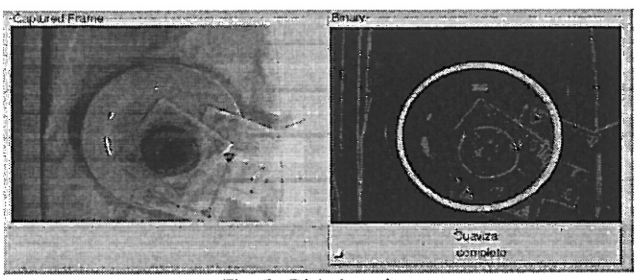


Fig. 5. Dish detection

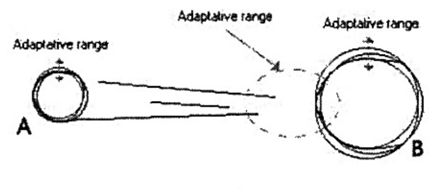
with less than 60 percent of visibility in the captured image. As can be shown the result of the algorithm is a thick circumference. The thikness of this circumference is generated because a great number of primitives is obtained. There are two main reasons that explain this. First, the contour map used is thick. And secondly, the specific set of parameters chosen for this particular case allows multiple primitive to be generated (concentric circumferences).

Once the list of primitives is generated, a matching process is started. In this process, subsets of primitives holding the spatial relationships that define the object are looked for.

B. Object matching

To understand this procedure, the detection of a spoon is considered. In this case, there are three list of primitives: circumferences with radius R1(9,10 pixels), circumferences with radio R2(15,18 pixels) and segments with length L1(70,100 pixels).

For a spoon, the circumference of radio R1 (referred as A) must be at a distance of 120 pixels from the circumference of radio R2 (B). The vector (A,B) provides a slope by means of which the candidate segments to form a spoon are found. Of all the found segments, only those that are located between the two circumferences and minimize the distance (A,B) are considered. Although the system starts from strict spatial relationships amongst primitives, when the verification of those relationships is done, a *third level of adaptation* is defined.



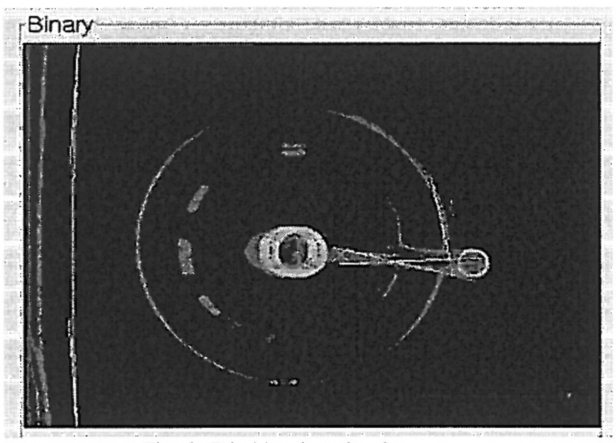


Fig. 6. Primitive detection for a spoon

Algorithm for the detection of spoons (R1, R2, X, Li)

- For all (x, y) so that (x,y) is the center point of R ∈ R1 do
- For all (x', y') so that (x', y') is the center point of R' ∈ R2 do

$$d=\sqrt{(x-x')^2+(y-y')^2}$$

- if $d \geqslant (X - factor_{adaptation})$ and $d \leq (X + factor_{adaptation})$ then
- find segment $(x_1, y_1, x_2, y_2) \in Li$ so that

$$(d_1 \leqslant factor2_{adaptation}) \wedge (d_2 \leqslant factor2_{adaptation})$$

with

$$d_1=\sqrt{(x-x_1)^2+(y-y_1)^2}$$

$$d_2=\sqrt{(x'-x_2)^2+(y'-y_2)^2}$$

or

$$d_1=\sqrt{(x'-x_1)^2+(y'-y_1)^2}$$

$$d_2=\sqrt{(x-x_2)^2+(y-y_2)^2}$$

- if segment exist then spoon matched
- calculate the recollection point $(x_r, y_r)_{spoon\ id}$
- Generate votes

$$votes = f(factor_{adaptation}, factor2_{adaptation})$$

Fig. 7. Algorithm for the detection of spoons

This level allows for small deviations around the original distances of the object definition. This is shown in Figure 6. The tolerance degree increases or decreases depending on the quality of the detection.

It can be observed that the detection process provides a great number of primitives that can be part of the analyzed object. This occurs because of the implementation properties of the Hough operators and because of the edge detection.

As commented above, the operator used to generate the contour map produces a map with thick contours. This fact together with the capability of the primitive detection operator for adaptation to small variations[3] in the size of the objects produces redundant information. This redundancy is used to generate the votes in the parameter space that allows evaluating the quality of the detection. From this information, the object with less error probability is finally chosen. Each primitive holding the constraints imposed by the relationships that define the object is provided with a specific number of votes. More votes are assigned to those primitives that fit better the original model. Figures 8 and 9 show the obtained results when trying to detect a spoon in a noisy situation or of overlapping. In both figures, software represents on the contours map, the primitive ones that have been detected. From these sets of primitive the clasificator algorithms work in search of the recolectables objects. It is possible to be observed in figure 8 like still in the presence of a high level of noise introduced by the napkins, the system is able to detect the circumferences located in the later and previous part of the spoon as well as a small located straight line segment between these circumferences. This set of primitive in this space situation adjusts to the spoon definition that the system has, reason why it provides a positive detection to us and indicate the point of harvesting by means of a mark in form of X. In this project, an effort has been made to endow the system with a certain level of adaptation to cope with real scenes like in a restaurant[4].

The system presents a quite good performace in the detection of objects in presence of noise.

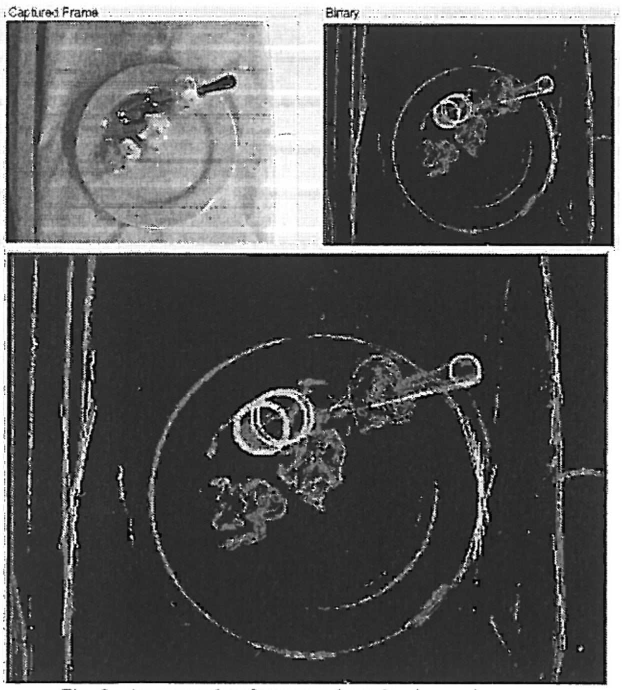


Fig. 8. An example of a spoon detection in a noise scene

IV. EXPERIMENTAL RESULTS

The experimental platform used in tests :

- CPU AMD K6 450Mhz.
- 128 Mb of RAM memory.
- Image capture module based on bt848 chipset.
- Analogic video-camera model CV-S3300.
- GNU-Linux kernel 2.4.21.
- Real-time layer RTAI-24.1.12

Figure 10 shows an example of test scene (384x288 pixels). The objects in figure 10 are numbered with the index in table 2 that represent the recognition order.

Working with the same cutlery for all tests (100 scenes with a mean of 8 objects) the percent of missed objetcs is 7 % and the percent of bad clasyfied objects is 5%.

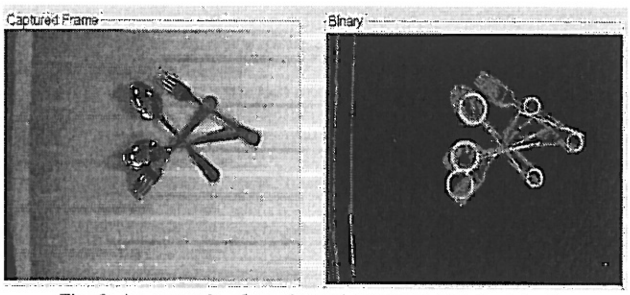


Fig. 9. An example of overlapped spoons and forks detection

Time results for scene analysis:

Object	*Recognition time*
1	1.52 s
2	6.84 s
3	1.29 s
4	1.70 s
5	6.29 s
6	6.20 s
7	5.70 s
8	5.90 s
Total time	35.44 s
Mean time	4.43 s

Tab. 2. experimental results

V. CONCLUSIONS

A versatile vision system has been implemented with a high degree of effectiveness. This system will be integrated in an autonomous robot for setting and clearing tables in a restaurant. With the information provided by the vision system, the robot manipulator will be able to calculate the best trajectories to pick objects from the table. Thus, the vision system provides the target point and the sliding direction to the manipulator.The system has been developed to have certain adaptation level to nonideal conditions. This was implemented using three level of adaptation in the recognition process. The experiments performed revealed a satisfactory performance of the system in the identification of objects, even when noisy images were processed. Successful classification rates have approached 100% when working with a good primitive subset for objects descriptions. This percentage is lower if the primitive subset is too reduced.

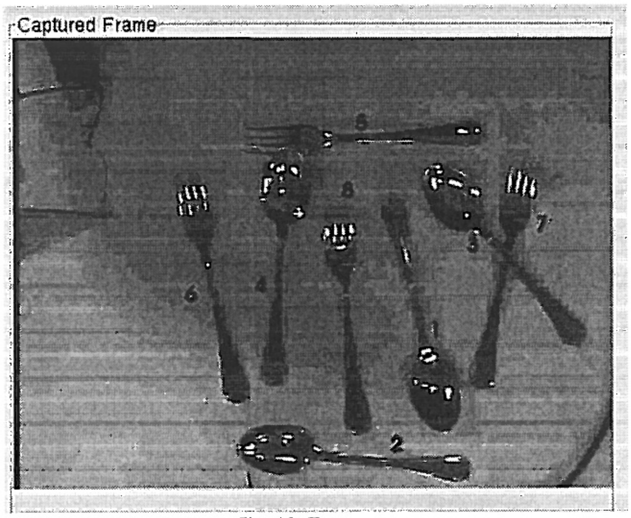


Fig. 10. Test scene

V. ACKNOWLEDGMENT

This work has been financed by the Spanish government, project # DPI2001-3681.
Anonymous reviewers are acknowledged for their interesting and constructive comments.

VI. REFERENCES

[1] - Leavers, V. F. Shape detection in computer vision using the Hough transform / V. F. Leavers London ; Barcelona [etc.] :Springer, cop. 1992 ISBN3-540-19723-0

[2] - A. S. Aguado, E. Montielb and M. S. Nixonc, Invariant characterisation of the Hough transform for pose estimation of arbitrary shapes, Pattern Recognition Volume 35, Issue 5, May 2002, pp 1083-1097.

[3] - A. A. Kassim, T. Tanb and K. H. Tana, A comparative study of efficient generalised Hough transform techniques, Image and Vision Computing Volume 17, Issue 10, August 1999, pp 737-748.

[4] - P.S.Naira, and A. T. Saunders Jr., Hough transform based ellipse detection algorithm, Pattern Recognition Letters Volume 17, Issue 7, 10 June 1996, pp 777-784.