Perception-Based Search and Manipulation in a Semi-Structured Environment

M. Prats, P.J. Sanz and A.P. del Pobil
Robotic Intelligence Laboratory
Universitat Jaume I
Campus Riu Sec, Castellon, Spain
{*mprats,sanzp,pobil*}*@uji.es*

***Abstract*— In this paper we present two new approaches developed in the context of autonomous search for localization and manipulation of books in a semi-structured environment, like a library. Search for books is done by means of visual information which is also integrated with force sensing for book manipulation.**

Interest regions are located using a local segmentation algorithm with automatic threshold selection. From the extracted visual features, we have implemented a fast tracking method by means of model-based matching. We also apply some recognition techniques in order to find the desired book and to guide the gripper over it. Grasping is finally done by integrating visual and force perception into a global hybrid control law.

We have applied the implemented algorithms to the UJI Librarian Robot and results show the robustness of the implemented search algorithm as well as its fast performance. With respect to the grasping system, vision and force coupling are shown to be an adequate combination of sensors for book manipulation.

***Index Terms*— Appearance-based Tracking; Optical Character Recognition; Visually-guided Grasping; Hybrid Force/Vision Control.**

I. INTRODUCTION

There is a big effort in the robotics community to make robots capable of working in unstructured environments, where humans do. However, robots are still highly domain-dependent. We find them in all kind of structured and not too dynamic scenarios. They do what they are intended to, but get very confused when strange things arise, something that they do not know how to deal with. Service robotics works directly with this problem and tries to solve it by providing artificial intelligence to the machines.

In order to have robots closer to humans in unstructured environments, we can first begin with semi-structured scenarios, like libraries. In our laboratory, we are working in a prototype of a librarian robot, designed to work in this partially known environment [3] [4]. We want the system to be able to search and retrieve a book requested by a user. The operation starts when the user requests a book by its name or code, either through Internet or by voice. The robot is then in charge of locating the book in an ordinary library, extract it and take it to the user. The only initial information is the book code, written on a label which is *read* by the vision system. This general application integrates several inter-disciplinary skills like path planning, visual perception or multisensory-based grasping, all linked together by reasoning capabilities.

In this article we pay attention to vision and grasping. With respect to the former, we implement a fast tracking and recognition scheme that allows us to find books at the same time that the robot is moving. In order to add robustness to the method, we exploit some spatio-temporal constraints adding reasoning skills based in probability. In addition, visual perception and force readings are integrated in order to take the book out of the shelf [4].

As far as we know, only two projects in the world follow similar goals that ours. The first one is the Comprehensive Access to Printed Materials (CAPM) project that is being developed at Johns Hopkins University [6]. Researchers in this university are working on a robot able to extract and browse books from a library. However, books are stored in special cases of the same size, placed on a well structured and specially designed library.

On the other hand, japanese researchers at the University of Tsukuba worked on a related system [7]. They designed a teleoperated mobile manipulator for book retrieval and browsing. The main difference with respect to the CAPM project, is that japanese robot had a built-in browsing mechanism so that they had no need for another robot. As we see in [7], they focused their efforts in the book browsing mechanism and the teleoperated user interface, while manipulation and book recognition had to be improved.

For now we are not interested in the book browsing capability, and we have already implemented user interfaces via voice [2], so we concentrate our efforts in autonomous navigation and manipulation modules. In this sense, our work has common points and is complementary to the work that was developed in Tsukuba.

In the following sections, we present a deeper view of the system. We begin in Section II with an overall description of the prototype and the steps involved in the execution of the task. In Section III and IV we focus on the search and tracking algorithms which are part of the vision and recognition module. Manipulation details are given in Section V. Later, some results of the working algorithms are shown in Section VI and finally we end with some conclusions and future work.

II. System description

The problem that we address is the delivery of a book to the user that has demanded it. This is a very general framework that can be divided into different multi-disciplinary tasks, as explained in the following subsections.

A. Human interface

We are interested in a very high level human-robot interface for requesting books. In our library, books are identified by a code or signature placed on a label in a visible place. There is also a database which links titles with signatures. The highest level consists in demanding the book by its title, and two possibilities are considered: the use of Internet or by speech-recognition software, already developed in other projects in our laboratory [2]. The system, then, uses the database and translates the title of the book to an identification label. Moreover, we consider that books in the library are well structured in areas, so that we know where to find it from the signature, approximately in an area of some meters length.

B. Sensory-based navigation

Once we know the approximate location of the book, a map of the library is used to plan a path and take the mobile platform from its current location to the desired one. The navigation is achieved by using sonar and laser sensors that provide collision avoidance capabilities. This is useful in order to interact with dynamic scenarios, like walking people.

C. Book location

The book location phase starts when the mobile manipulator has reached the approximate location found in the database for this book. Because we do not know where the book is placed exactly, we use an intelligent search algorithm, combined with a book recognition module. In this way, signatures of books are read and results are used to guide the manipulator to a next position in an intelligent way, until the book is finally found. The existence of open source Optical Character Recognition (OCR) software like GOCR [5], and the development of new computer vision libraries have lead to a fast recognition module that is presented in this article. This contribution allows the intelligent search algorithm to be executed in a continuous way, with more extracted signatures in less time, and thus reducing the total search time.

D. Book manipulation

The search algorithm ends with the book detection. Then, the grasping module is launched, which is in charge of extracting the book from the shelf. We design a hybrid control law, combining visual information with force readings [4]. The left side of the label is used to guide the gripper in a close force/vision loop. However, we contemplate another possibility for the future: the tracking of signatures, together with known camera motion could be combined following a structure-from-motion approach in order to approximate the 3D position of the book with respect to the camera, and hence with the gripper. In this case, gripper motion would be performed in open loop.

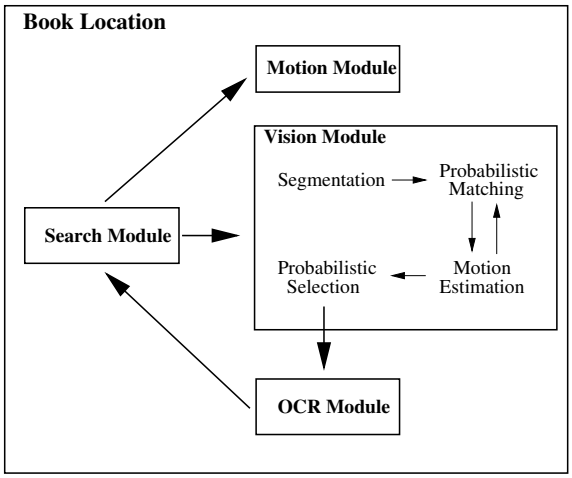

Fig. 1. Modules involved in the location phase.

III. Vision Module

In this article we present an architecture of the vision module that enables the image processing while the camera is moving. We show a probabilistic tracking method in order to match signatures through time, exploiting the spatio-temporal constraints of the problem. The general schema of the location phase is shown in Figure 1, with a detailed view of the vision module.

Using vision, book labels must be segmented and located in the image. Some geometrical information like size or inclination is also required. All this information is sent to the OCR module in order to read the identification codes. From the readings, the search algorithm decides where to move the manipulator so that the book is found as soon as possible.

Note that at some situations, just one image is enough to conclude that the book we are searching is not on that shelf, and to decide to move to another very different position. So, although we are interested in a tracking algorithm, it is a requirement that this algorithm works on single images too. We propose a sequential and modular method in order to achieve these tasks, composed by four steps: label segmentation, probabilistic matching, motion estimation, and probabilistic selection.

We adopt the following constraints and assumptions about the problem:

1) All signatures have white background and black foreground. In addition, they are composed of four text lines (see Figure 2).
2) Signatures are located in a way that there is an horizontal line that crosses the image and intersects all of them. This is an assumption about the vertical location of the labels in the books. Note that this is not a hard restriction, because in our library, signatures are all placed more or less at the same height (see Figure 2).

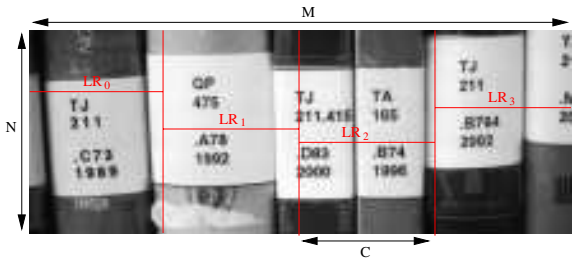

Fig. 2. Intervals and lightest rows.

Fig. 3. Result of the segmentation algorithm.

3) Labels cannot overlap. This is not an assumption but a property of the problem, that can be used to better detect regions.
4) Signatures are rectangular. The vertical size is approximately the same for all. The width depends on the width of the book.
5) We consider a static environment. Changes on the image are only due to camera movements.
6) Signatures can only appear or disappear at the sides of the image, when the velocity is non zero.

A. *Segmentation of signatures*

The first step consists in locating the signatures inside the image. For this, we propose a local segmentation algorithm with automatic threshold detection. For a $N \times M$ greyscale image I, we first divide it in L intervals of $C = \frac{M}{L}$ columns each one. We define the lightness of the row i of interval k as:

$$l_k(i) = \frac{\sum_{j=k\cdot C}^{(k+1)\cdot C} I_{ij}}{C} \tag{1}$$

that is, the mean intensity of the ith row in interval k, being I_{ij} the value of the pixel with column j and row i (in the range $[0, 255]$). Then, we find the lightest row for each interval k:

$$L_k = \mathbf{max}\{l_k(i)\} \quad \forall i \in [0, N-1] \tag{2}$$

$$LR_k = \mathbf{argmax}\{l_k(i)\} \quad \forall i \in [0, N-1] \tag{3}$$

From assumptions 1 and 2, it is easy to see that horizontal lines in rows LR_k will frequently intersect with book signatures. We say frequently, because for an image interval where all the books are white, the lightest row LR_k could not pass through the label. However, this is not a big problem, because since book and signature are of the same color, we can treat it as a big signature and refine the detection in posterior stages.

Merging this property with equations 1 and 2, we can say that L_k corresponds to the mean intensity of labels. In Figure 2 we show the parameters involved in the segmentation process.

The lightest mean row intensity L_k can be used for segmentation purposes. We know that a book signature is the lightest thing we can find into an image, and its color value is around L_k. We can segment locally each interval k using as threshold the value of $\frac{3}{4}L_k$, chosen experimentally. The result will be a binary image where signatures are visible, along with some regions of the same color. Of course, the quality of the segmentation depends on the total number of intervals. For just one interval taking all the image, there is one L_0 that is the mean color of all the signatures in the image. However, there are some signatures that are darker than others, because of the antiquity or due to the shadows. In this case, L_0 would not be a good approximation of labels intensity for all of them, and segmentation would not be precise (note that this would be the equivalent to a global segmentation). This problem is solved by taking more intervals into account and by segmenting the image locally. Of course, the labels of two books are more probable to have a similar intensity than those of six books. Then, higher values for L give a better segmentation, at the cost of execution speed.

Also note that the automatic selection of the threshold makes the segmentation algorithm robust to illumination conditions and illumination changes. Concretely, robustness in global illumination is due to the assumption that signatures are the lightest thing in the image. The algorithm automatically detects the lightest rows and its mean values, which are used to estimate a correct segmentation threshold. Moreover, local illumination changes are detected thanks to the use of intervals which allows to segment separately each part of the image.

We have said that the rows LR_k intersect with all the signatures, and thus they can be used to get the boundaries. The process is as follows:

1) For each interval $k \in [0, L-1]$:
   a) Consider the lightest row LR_k and find in the segmented image the next white pixel in that row that does not belongs to any already detected signature.
   b) That pixel belongs to a new signature, and we can use a standard contour extraction algorithm to get the boundary.
   c) Repeat 1.a and 1.b until the end of the interval

Finally, a curvature analysis on the boundaries is done in order to detect angles and lines. Due to assumption 4, detected boundaries that do not have four lines parallel two by two are rejected. Figure 3 shows the final result of the segmentation algorithm for a real image.

B. Probabilistic matching and motion estimation

If we want to extend our system to work over sequences of images, we can apply the segmentation process for each them. But if we also want to follow the movement of a certain book, then we need a tracking method. In this application, tracking is especially interesting because we want to keep a list of the books that have already been processed by the OCR module. So, if we apply the OCR to all the books of an image, and then the camera moves a little bit to the left so that a new book appears, then we must pass to the OCR the new signature, and not the already processed ones.

We propose a model-based probabilistic approach [1] for matching signatures in two consecutive images. Rather than using the location of the labels in the first image to predict its location in the second image and avoid a whole segmentation, we profit the good performance of the segmentation algorithm and execute a whole segmentation each time. With this, we aim to improve the location of the signatures following a probabilistic framework. For example, segmentation is not always perfect, and it is possible to start with a bad detected signature. Then, we apply segmentation to all images while the camera is moving. If the signature is segmented correctly in one image, our probabilistic algorithm will remember its features for helping future segmentations.

The matching algorithm tries to link books detected in the current image with those detected in the last one. For this, a good representation of a signature is needed. We consider that a label is fully described by its four vertex, represented as points $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ in the two dimensional image space. Points are ordered so that $\mathbf{A}$ corresponds to the top-left corner, $\mathbf{B}$ to the top-right, $\mathbf{C}$ to the bottom-right and $\mathbf{D}$ to the bottom-left. We can describe a signature as a four dimensional vector $\mathbf{S_i} = (\mathbf{A_i}, \mathbf{B_i}, \mathbf{C_i}, \mathbf{D_i})$.

Following with the notation, signatures segmented at a given time t, are represented by the vector $\mathbf{S^t} = (\mathbf{S_1}, \mathbf{S_2}, \ldots, \mathbf{S_n})$, where n is the total number of books in the image. From this, the algorithm tries to match elements of $\mathbf{S^t}$ with elements in $\mathbf{S^{t-1}}$.

The matching is done following a probabilistic approach. Given two signatures, $\mathbf{S_i}$ and $\mathbf{S_j}$, we define a metric for measuring the distance between them as:

$$M(\mathbf{S_i}, \mathbf{S_j}) = \frac{d_{\mathbf{A_i A_j}} + d_{\mathbf{B_i B_j}} + d_{\mathbf{C_i C_j}} + d_{\mathbf{D_i D_j}}}{4} \qquad (4)$$

where d_{xy} is the point-to-point distance between points x and y. We also define the random variable X="distance between two correspondent signatures in two consecutive images", and then:

$$X \hookrightarrow N(v, \sigma^2) \qquad (5)$$

i.e, X is distributed by a Normal with mean v and variance σ^2. The mean v is an estimation of the image velocity and the variance is an estimation of the error in the velocity.

Of course, the distance between the same signature in two consecutive images depends on the image velocity v, that can be estimated either by a calibrated camera with known motion, or by computing the image flow like we do. The variance in the distribution model can be chosen experimentally, but it will not affect the tracking algorithm.

Our system works by computing the distance of all possible matches of signatures between $\mathbf{S^t}$ and $\mathbf{S^{t-1}}$. Then, we apply the probability model in 5 in order to find the most probable pairs. Three cases can happen

1) A signature in $\mathbf{S^t}$ is not matched against any signature in $\mathbf{S^{t-1}}$: This can happen in two situations. The first one is at the start of the algorithm, when $\mathbf{S^{t-1}}$ has no elements. The second one is when a new signature appears in the image, due to camera motion. In both cases, we assign an internal identification number to the book, that will be used for tracking.

2) A signature in $\mathbf{S^{t-1}}$ is not matched against any signature in $\mathbf{S^t}$: This means that the signature has dissapeared. As we have an estimation of the image flow, we can apply some restrictions about the place where signatures can dissappear. For example, for a static image ($v = 0$), no signatures can dissappear. However, if we displace the camera to the right, then signatures can only dissappear through the left side of the image.

Our system takes into account this constraints. If a signature dissapears where it is not possible to, it is created again in $\mathbf{S^t}$ and displaced according to the prediction of the velocity. In this way, we can deal with occlusions or with errors in the segmentation process. The motion of lost signatures is predicted knowing the motion of those which are matched.

3) A signature $\mathbf{S_i} \in \mathbf{S^t}$ is matched against a signature $\mathbf{S_j} \in \mathbf{S^{t-1}}$: That means that $\mathbf{S_i}$ in the current image is the same that $\mathbf{S_j}$ in the last image, so we have successfully matched the signature. But once more, segmentation problems can arise and it is possible to have a bad extraction in $\mathbf{S_i}$, while $\mathbf{S_j}$ is good. To cope with this problem, we define two additional random variables: H="height of a signature" and W="width of a signature". These variables are also distributed according to a Normal distribution, with a mean and a variance that can be chosen experimentally.

When two signatures are matched, the probability models are computed on $\mathbf{S_i}$ and $\mathbf{S_j}$. If $\mathbf{S_i}$ has a very low probability of being a good signature, and $\mathbf{S_j}$ has high probability, then we overwrite $\mathbf{S_i}$ with the good one and displace it in the image by using the image flow estimation. With this we ensure that if a label is well detected at a given time, future problems with it will be solved using the good prediction.

But in the majority of the cases, $\mathbf{S_i}$ and $\mathbf{S_j}$ will be good, and then the match will be used to update the image flow to be used in future predictions. The image flow is a 2D vector $\mathbf{f}$ computed as follows:

$$\mathbf{f} = \frac{\mathbf{t_A} + \mathbf{t_B} + \mathbf{t_C} + \mathbf{t_D}}{4} \qquad (6)$$

with:

$$\mathbf{t_A} = \mathbf{A^t} - \mathbf{A^{t-1}} \quad (7)$$

$$\mathbf{t_B} = \mathbf{B^t} - \mathbf{B^{t-1}} \quad (8)$$

$$\mathbf{t_C} = \mathbf{C^t} - \mathbf{C^{t-1}} \quad (9)$$

$$\mathbf{t_D} = \mathbf{D^t} - \mathbf{D^{t-1}} \quad (10)$$

The velocity used in the probability distribution is defined as the module of the image flow, $v = |\mathbf{f}|$.

C. Probabilistic selection

The last step that we have implemented in the vision module is the probability-based selection. The same random variables H and W defined in the last section are used here to eliminate those signatures with a low probability of being good. With this, we can discriminate books that are too wide for being grasped by the robot, for example.

Moreover, this stage is in charge of selecting those books that must be passed to the OCR module. In this way, just new signatures are selected, i.e. those in the current image which have not been found in the last one.

IV. OCR MODULE

The recognition module takes as input a set of signatures and it first tries to locate text lines inside them. For each signature, we take perpendicular lines between points $\mathbf{A}$ and $\mathbf{D}$ and compute the lightness for each line in a way similar to that of section III-A. The maxima of this function represent the background while the minima represent text lines. In this way, the four lines of text are located.

After that, we compute the angle of the signature by the following equation:

$$\alpha = \arctan \frac{\mathbf{A_y} - \mathbf{B_y}}{\mathbf{B_x} - \mathbf{A_x}} \quad (11)$$

A rotation of $-\alpha$ is applied to the signature before it is passed to the OCR, in order to convert inclined signatures in straight ones. The utility of locating the separation between text lines is that each line can be passed separately to the OCR, and then we can apply some restrictions about the characters that can appear in each line. For example, we know that the first line of text is composed only by letters, while the last line is a year composed just by numbers. If we process the lines separately, we can indicate this constraints to the OCR in order to reduce errors.

As OCR core, we use the GOCR software [5], that is being developed under the GNU Public License. This software is able to extract the text included in images. It works with all kind of fonts, and can be trained for better performance in particular applications.

V. BOOK MANIPULATION

Special fingers have been designed to accomplish the task of extracting a book from a bookshelf (see Figure 4). Moreover, some initial hypothesis must be assumed:

1) Books must be graspable by our robot arm. So, its physical properties (i.e. length, wideness, weight, etc.) will be always suitable to the available manipulation capabilities (i.e. geometry of the gripper, etc.).
2) The side of the books from the robot point of view, oriented towards the outside the shelf, are all of them, approximately, in the same spatial plane.
3) Moreover, it is assumed that the books are not pressed together on the shelf in such a way as to impede the insertion of the gripper fingers.

Fig. 4. Experimental environment.

In our system, when the OCR phase finds the requested book, his position on the image is returned to the grasping system that has to advance the gripper towards the book. We propose a closed loop based on visual and force information that gives us a robust, reliable and fast method for accomplishing this task [4].

VI. EXPERIMENTAL RESULTS

The UJI Librarian Robot consists of a *Mitsubishi PA-10* manipulator mounted over a *PowerBot* mobile platform, as we see in Figure 4. All the system is controlled autonomously from the computer inside the mobile robot. It is an AMD Athlon XP, running at 1.6 GHz, with 512 Mb of RAM. The operating system installed is Linux. It is in charge of communicating with the mobile robot microcontroller through the serial port, with the manipulator and force sensor through two different PCI cards, to the gripper via an expansion input/output board, and to the camera using the IEEE1394 interface. The camera is actually a stereo system, altough we are no exploiting this property. It is a MEGA-D Stereo Head, constructed by Videre Design. It can capture video at different resolutions, at a speed of 24 frames per second.

For our tests, we have used images with a size of 640x240. The results show that the total execution time for segmentation and tracking is 42 milliseconds. It means that the algorithm can be applied at video rate.

Times are longer when a signature must be passed to the OCR module. The total time for recognizing the text inside a signature is about 130 milliseconds, which means that in one second, the system is able to fully process 7

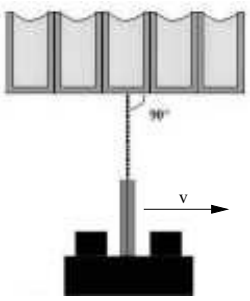

Fig. 5. Motion of the camera.

books. Note that during the tracking, the OCR module is called just when a new signature appears, so that this extra time is only consumed at a few frames.

For testing the performance of the tracking algorithm, we placed the manipulator in front of a bookshelf, and moved it in a direction parallel to the books while running the program (see Figure 5). The arm was moved at a speed of about 1 cm/s, and the total time of the experiment was 80 seconds. In this time, 22 out of 24 books were successfully located and tracked, which gives a 91% of success.

With respect to the text recognition module, 32 characters failed to be recognized in the total of 24 labels. This gives a mean of 1.3 wrong characters per signature. In addition, 50% of the labels were successfully fully recognized, 30% had one or two misses, and in the other 20% more that 2 characters failed. In all the cases, the main reason for missed characters was a bad binarization. We aim to improve this results by implementing another local binarization stage, after the label has been located.

Finally, during the execution of the task, the room was submitted to illumination changes, and it had no effects on the algorithm.

Regarding the grasping process, a ratio of 1.5 seconds per 1 centimeter of book is needed on average before the book is grasped (e.g. a book with a width of 4 cm requires 6 seconds).

VII. Conclusion & future work

We have presented a system that is able to locate labels of books from an image and to extract text lines included inside them. We have extended it in order to be used over video sequences by means of an intelligent model-based tracking method. Moreover, we have used this tracking algorithm in order to guide the grasping task, while paying attention to force readings.

The algorithm first applies some well-known segmentation techniques in order to locate interest regions in the image, that correspond to book signatures. Some geometrical features are computed from the located boundaries. Width and height of the label, as well as the orientation angle with respect to the horizontal are of special interest.

The current set of labels is compared with the last one by means of a metric defined in function of the geometrical features. A probability distribution based in the image flow is then applied and matches of signatures are located. Two more probability distributions are defined to describe the most probable width and height of a label. These distributions are used to detect errors in the segmentation and to correct them. In addition, good predicted matches are used in a learning step in order to estimate camera motion and to improve tracking performance.

With these properties, the algorithm is able to detect which signatures are seen for the first time. These labels are passed to the OCR module that applies inclination correction and finally extracts the text. If the recognized book is that we are searching for, the grasping module is called and the book is taken out of the library by means of vision and force integration.

This algorithm has been tested in the UJI Librarian Robot, and experimental results show the good performance of the system that is able to process images at video rate. The program was tested over a video sequence with 24 different books, and 91% of them were successfully located. However only 50% succeeded to be fully recognized by the OCR module. The system was also proved to be robust to light conditions and occlusions.

As future research lines, we aim to improve the recognition module by designing an additional layer for automatic segmentation of text that would increase OCR performance. We also plan to integrate the vision module along with the intelligent search algorithm for camera motion planning. In addition, we are also interested in the design of a more general control law enabling a better control in all the steps of the book handling, by the definition of new visual features that allow us to control more degrees of freedom, like the gripper aperture.

Acknowledgements

This paper describes research carried out at the Robotic Intelligence Laboratory of Universitat Jaume-I. This work was supported in part by the MEC under projects DPI2001-3801 and DPI2004-01920, and by the Generalitat Valenciana under the FPI grant CTBPRB/2004/052.

References

[1] D. Lowe. Robust model-based motion tracking through the integration of search and estimation. In *International Journal on Computer Vision*, pages 113–122, 1992.

[2] R. Marín, P. Vila, P. Sanz, and A. Marzal. Automatic speech recognition to teleoperate a robot via web. In *IEEE International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, October 2002.

[3] M. Prats, R. Ramos-Garijo, P. Sanz, and A. del Pobil. Autonomous localization and extraction of books in a library. In *Intelligent Autonomous Systems 8*, pages 1138–1145, 2004.

[4] R. Ramos-Garijo, M. Prats, P. Sanz, and A. del Pobil. Recent progress in the uji librarian robot. In *IEEE International Conference on Systems, Man & Cybernetics*, pages 3912–3917, The Hague, The Netherlands, October 2004.

[5] J. Schulenburg. Gocr. Available on: *http://www-e.uni-magdeburg.de/jschulen/ocr/*.

[6] J. Suthakorn, S. Lee, Y. Zhou, R. Thomas, and S. Choudhury. A robotic library system for an off-site shelving facility. In *IEEE International Conference on Robotics and Automation*, pages 3589–3594, 2002.

[7] T. Tomizawa, A. Ohya, and S. Yuta. Remote book browsing system using a mobile manipulator. In *IEEE International Conference on Robotics and Automation*, pages 256–261, Taipei, Taiwan, September 2003.