Proceedings of the 2004 IEEE
Conference on Robotics, Automation and Mechatronics
Singapore, 1-3 December, 2004

ROBOLEO – An Interactive Seeing Robot

B. Lammen , A. Ruckelshausen
Faculty of Engineering and Computer Science
Fachhochschule Osnabrück
University of Applied Sciences
Osnabrück, Germany
b.lammen@fh-osnabrueck.de

Abstract—**A robotic system that serves as a platform for research in human-robot interaction, robot vision, model based planning, and automation of flexible production processes is introduced. A robot equipped with a vision system and distance sensors interacts with a human in a non-planned variable environment. As an application, a mechanical balance game between a robot and a human player is presented.**

I. Introduction

Robot vision and sensor integration are gaining increasing importance for future robot applications in flexible production processes. Robots are intended to carry out tasks in non-planned variable situations. Robots have to operate in flexible environments where e.g. the position of processed parts or the order of processing can not be determined in advance. This can also include human-robot interaction.

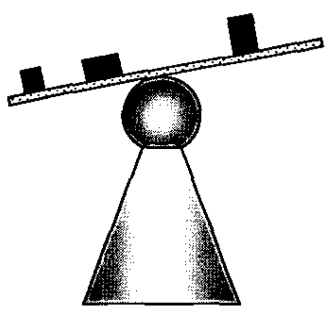

Figure 1. Bamboleo balance game

The robotic system ROBOLEO is presented in this paper. The initial idea of the project is based on the simple balance game called Bamboleo [1] shown in fig. 1. Objects of different weight and shape are placed on a board lying on a ball. The players alternately remove objects from the board. The equilibrium conditions change and the board moves into a new position. If a player causes the board or objects to fall down the last player loses.

In ROBOLEO, the game is carried out between an interactive, seeing robot and a human player. The robot is equipped with a CCD-camera and distance sensors to recognize the game situation.

The system offers an intuitive approach to different fields of research and industrial applications such as human-robot interaction, sensor integration for flexible production processes, and model based planning algorithms.

II. System Structure

The system structure of ROBOLEO is shown in fig. 2. An industrial robot is mounted above the board. The system is equipped with a CCD-camera for the recognition of the objects on the board.

At present, cylindrical and cubic objects with a black surface are used. The image processing provides the position, orientation, and shape of the objects on the board. For a complete description of all objects in 3-D-world coordinates, the inclination of the board has to be known. Considering the radius of the ball, the inclination angles can be derived from the signals of two distance sensors mounted.

The intelligent combination of image processing and distance measurement enables the robot to identify and pick up any object in its work space. The decision about which object shall be removed is based on a model of the balance game and on strategic considerations. The image processing and all algorithms are executed by a PC. The PC transfers the coordinates of the object that is to be removed to the robot control.

III. Robot Vision and Sensor System

The position of the objects is of three dimensional nature, thus distance sensors are used to measure the orientation of the platform. Based on this information, the image processing of the camera data determines the exact position and orientation of each object on the platform.

The coordinates of three points are necessary to define the position of the platform. Since the radius of the cork ball underneath the board is known, the additional two points are measured with distance sensors (see figure 2). Based on experiences from the authors [2] optoelectronic sensors are used. The triangulation sensor ODS96 from Leuze [3] with a measurement range from 100 to 600 mm and a resolution of even better than 0,5 mm satisfies the specifications needed for this application. With an analog-to-digital conversion the sensor data become available for the PC software. Using the given information of the thickness of the platform as well as the radius and the position of the cork ball, the three dimensional position of the surface of the platform (and thus the normal vector) is known.

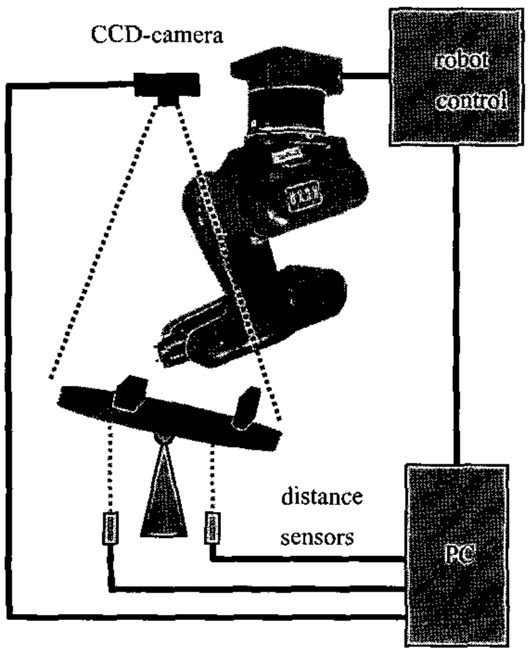

Figure 2. System structure

The task of the image processing is the two-dimensional determination of the objects, which can be transferred into three dimensions by taking into account the above described triangulation sensor measurements. In the first development stage of ROBOLEO the objects have the same height and differ with respect to their cylindrical or cubic shape, which has to be detected.

The black-and-white camera VCAM-020 from Phytec [4] has been used, including a 1/3" CCD image sensor with 500 x 582 pixels. The camera is connected to a PCI frame grabber card. Due to the black surface of the objects as compared to the white board, standard image processing algorithms can be applied to determine the position of the objects [5,6]. Using a threshold, the 8 bit grey level image is converted into binary data. After pre-processing and segmentation algorithms the compactness K of the resulting objects is determined (U: perimeter, A: area):

$$K = \frac{U^2}{4\pi A} \qquad (1)$$

Due to the limitation with respect to the shape of the objects, the selectivity of K is sufficient to distinguish between the two type of structures. Since the orientation of the board results in variations of the distances from an object to the camera, this optical distortion has to be corrected in order to determine the centre of each object. Finally the position and the orientation of each object has been calculated and is available for the robot control.

IV. Model Based Game Algorithm

The sensor system of ROBOLEO provides all information necessary to calculate the inclination of the board and the position and orientation of all objects in 3-D world coordinates. Afterwards an algorithm is required to choose the object that is to be removed by the robot.

A first heuristic strategy would be to simply choose the nearest object on the lower side of the board. A more offensive strategy is to remove an object on the higher side of the board and thus create a critical situation for the following player. Such algorithms can lead to a reasonable behaviour of the robot. However, model based algorithms achieve better results.

Taking into account the mass properties and the geometrical distribution of the objects given by the sensor system, the dynamic equations of motion and the conditions for stability of the mechanical system can be evaluated.

Experiments have shown that in most cases it is sufficient to consider just the static equilibrium of the system. For simplicity it is assumed that the centres of gravity of all objects are lying on the surface of the board, i.e. the height of the objects is neglected. Furthermore, the thickness of the board is assumed to be zero.

A. Fundamental Geometrical Considerations:

Fig. 3a shows the geometry in a static equilibrium situation. After being placed on the ball, the board moves into a static position with a normal vector

$$\vec{n} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \quad \text{where } |\vec{n}| = 1 \qquad (2)$$

and an inclination angle α. The board contacts the surface of the ball in point A. The position $\vec{r}_A$ of A with reference to the inertial frame in the centre C of the ball can directly be calculated from $\vec{n}$ as

$$\vec{r}_A = R \cdot \vec{n} \qquad (3)$$

where R is the radius of the ball. The inclination angle α is given by

$$\cos\alpha = n_z . \qquad (4)$$

In case of a horizontal position of the board, B would be the contact point (see fig. 3b). With respect to B, the contact point A rolls along the circumference of the ball over a distance of

$$\ell = R\alpha . \qquad (5)$$

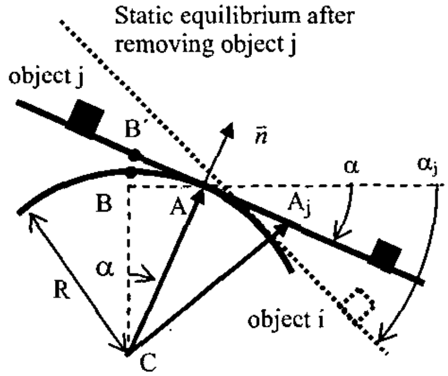

Figure 3a. Static equilibrium

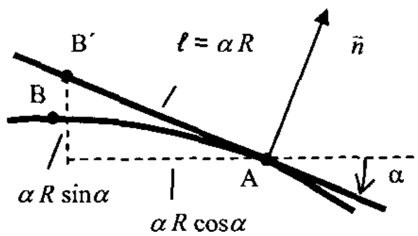

Figure 3b. Detail for calculation of

B, as a fixed point of the board, is transferred to B' because of the inclination. B' is lying in the plane between the vertical axis of the inertial frame and $\vec{n}$ (see fig. 4b) in a position

$$\vec{r}_B{}' = \vec{r}_A - \alpha R \begin{pmatrix} \cos\alpha\cos\beta \\ \cos\alpha\sin\beta \\ \sin\alpha \end{pmatrix} \tag{6}$$

where β is the angle of the projection of $\vec{n}$ into the horizontal plane of the inertial frame in C.

$$\tan\beta = \frac{n_y}{n_x}. \tag{7}$$

B. Algorithm

Presuming stability, the board takes another static position after an object is removed. Before the robot removes one of the objects the model based algorithm calculates the new static inclination angle α_j. This is done for each particular object j.

The radius R of the ball, the mass m_b of the board, and the masses m_i of all objects i (i=1,2, ..., k) are parameters of the algorithm. The normal vector $\vec{n}$, the position $\vec{r}_A$ and the positions $\vec{r}_i$ of all objects i are known from the sensor system.

The position $\vec{r}_b$ of the centre of gravity of the board itself is not provided directly, however it can be calculated from $\vec{r}_A$ as follows. The sum of all torques with respect to the contact point A has to be balanced. This leads to

$$\vec{r}_b = \frac{1}{m_b}(\vec{r}_A m_t - \sum_{i=1}^{k} \vec{r}_i \cdot m_i) \tag{8}$$

where m_t is the total mass:

$$m_t = m_b + \sum_{i=1}^{k} m_i \,. \tag{9}$$

In the next step of the algorithm, the static situation after the removal of objects can be considered. Taking away an arbitrary object j causes a new centre of gravity of the remaining system in a point A_j fixed to the board. The position vector $\vec{r}_{A,j}$ can be calculated as

$$\vec{r}_{A,j} = \frac{1}{m_{t,j}}(\vec{r}_b m_b - \sum_{i=1,i\neq j}^{k} \vec{r}_i \cdot m_i) \tag{10}$$

where $m_{t,j}$ is the total mass without object j:

$$m_{t,j} = m_b + \sum_{i=1,i\neq j}^{k} m_i \,. \tag{11}$$

After a dynamic transition, stability presumed, the point A_j contacts the ball. According to the distance

$$\ell_j = \left|\vec{r}_{A,j} - \vec{r}_B{}'\right| \tag{12}$$

the board takes a new inclination angle

$$\alpha_j = \frac{l_j}{R}. \tag{13}$$

The calculations above are executed for every object j on the board. Experiments have shown, that stability is lost, if the static inclination angle α_j exceeds a critical value $\alpha_{crit} = 10°$.

Different strategies of choosing an object can now be applied. Any object with an inclination angle $\alpha_j < \alpha_{crit}$ is possible.

The static algorithm above can be improved by taking into account effects due to the dynamic transition between equilibrium positions and possible violations of friction conditions. The simulation of the dynamic equations of motion of the system will be an issue of future work in ROBOLEO.

V. Human-Robot Interaction

The static algorithm described above has been implemented on a PC. Figure 4 shows the graphical user interface (GUI) of ROBOLEO. The human player can select either an "offensive" or a "defensive" mode. The robot, respectively, removes the object that results in the smallest inclination angle or the nearest angle smaller then α_{crit}. The GUI displays the image of the CCD-camera.

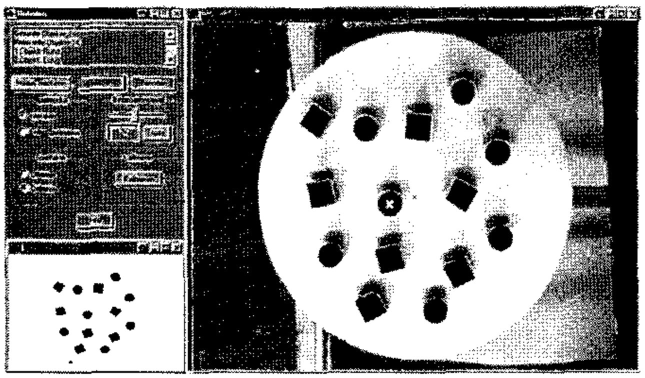

Figure 4. Graphical user interface

The object that shall be removed in the present situation is marked by a green cross. In the lower left corner of the GUI the image after binarization and segmentation is shown.

The positon and orientation of the chosen object is transfered to the robot control. For the realization of ROBOLEO, an industrial robot Kuka KR3 [7] is used. Fig. 5 shows the ROBOLEO test setup. The robot is mounted in a hanging position above the board. Fig. 5 also shows the CCD-camera for object recognition and the distance sensors beneath the board

During interactive operation, the robot and a human player alternately remove objects from the board. For that purpose the player has to enter the workspace of the robot. In conventional industrial applications such situations are avoided for safety reasons. During the presence of a person in the work space the robot drives are deactivated

In ROBOLEO, the robot stays in a rest position while the human player makes his move. The robot control software has been modified in such a way that the robot can only be activated after a confirmation button has been pressed.

Safe interactive operation of the robot is a major issue of further research with ROBOLEO. In a first step an optoelectronic light grid with parallel beams for contact less monitoring and detection of human access to the workspace has been integrated with the robot control.

VI. Summary and Conclusions

A balance game between a human and a robot demonstrates the intelligent combination of sensor information from image processing and distance measurement for robotic control. The human player and the robot alternately remove an object from the game board being placed on a ball. Moreover, the applied methods and technologies show potential for innovative industrial applications such as flexible production processes. Furthermore ROBOLEO offers a test setup for future research in human robot interaction and further development of sensor based robotics.

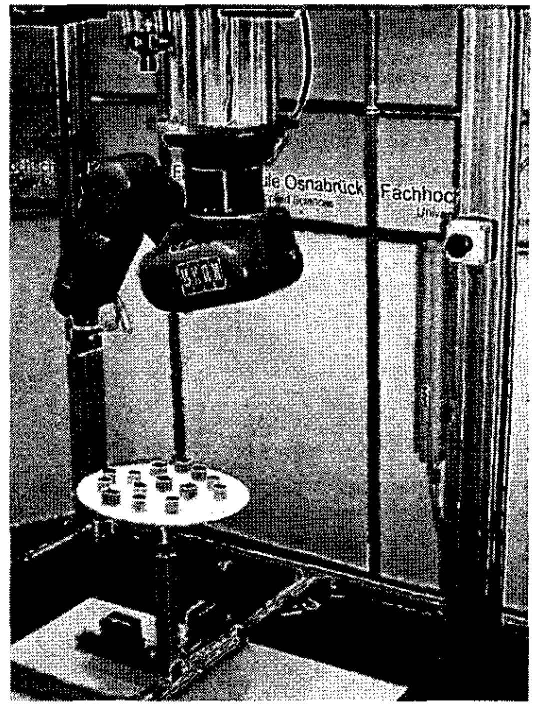

Figure 5. ROBOLEO test setup

Acknowledgment

This work has been supported by the German Department of Research and Technology (BMBF) as part of the aFuE-program and by the Wilhelm Müller foundation.

References

[1] Zoch GmbH, Brienner Str. 54a, 80333 München, Germany, www.zoch-verlag.com (see "Bamboleo")

[2] G. Thösink, J. Preckwinkel, A. Linz, A. Ruckelshausen, J. Marquering, "Optoelectronic Sensor System for Crop Density Measurement", Landtechnik 59 (2004), no. 2, pp. 78 - 79

[3] Leuze electronic GmbH + Co KG, In der Braike 1, 73277 Owen / Teck, Germany, www.leuze.de

[4] PHYTEC Technologie Holding AG, Robert-Koch-Strasse 39, 55129 Mainz, Germany ; www.phytec.de

[5] P. Soille, „ Morphologische Bildverarbeitung", Springer Verlag, 1998

[6] S. Battmer, N. Ellermann, A. Linz, A. Ruckelshausen, U. Ruhnau, K. Tovar, G. Weidner, "Fast image processing for online detection, tracing and observing of microorganisms", Bornimer Agrartechnische Berichte, Heft 31 (2002), ISSN 0947-7314

[7] KUKA Roboter GmbH, Hery-Park 3000, 86368 Gersthofen, Germany, www.kuka-roboter.de