2005

An Application of human robot interaction: Development of a ping-pong playing robotic arm

Kalpesh Modi

Ferat Sahin

Eli Saber

An Application of Human Robot Interaction: Development of a Ping-Pong Playing Robotic Arm

Kalpesh P. Modi
Electrical Engineering Department
Rochester Institute of Technology
Rochester, NY, U.S.A.
kpm5163@rit.edu

Ferat Sahin
Electrical Engineering Department
Rochester Institute of Technology
Rochester, NY, U.S.A.
feseee@rit.edu

Eli Saber
Electrical Engineering Department
Rochester Institute of Technology
Rochester, NY, U.S.A.
esseee@rit.edu

Abstract – *This paper describes the design and development of a ping-pong robotic arm as an application of robotic vision. Displaced frame difference (DFD) is utilized to segment the ball motion from background motion. 3-D ball tracking using parametric calibration of single CCD camera is explained. This visual information is temporally updated and further employed to guide a robot arm to hit the ball at a specified location. The results signify the system development based on single camera tracking. System latency is measured as a function of the camera interface, processor architecture, and robot motion. Various hardware and software parameters that influence the real time system performance are also discussed.*

Keywords: Real time machine vision, camera calibration, 3-D imaging, robot vision.

1 Introduction

Human-computer interaction aims at the scientific study of people's communication with computers and applications of associated technologies. Human-robot interaction is an applied area that comprehends behavioral aspects between human and a robot and includes study, design and development of systems for a measure of their joint performance. This performance can be evaluated through techniques which compare human and robot productivity in a team environment [10]. In this work, a ping-pong robot is designed and developed to study the mutual functioning of human and robot.

Vision is our most powerful sense and provides us with a detailed three dimensional knowledge of the environment and perceived changes. Similarly for machines, vision is a potential interface as it facilitates sensing of objects and tracking changes in the scene. This work demonstrates the use of computer vision for performance analysis of the joint human robot system.

1.1 Objective

For extraction of real time knowledge about the dynamically changing environment, visual sensors are included in the feedback loop [7]. Video sequences are acquired through a CCD camera interface to obtain scene information. Motion is detected through the temporal change in the brightness values (gray levels) in the image plane [15, 18]. Motion based image segmentation and object detection techniques are used to locate the ball in the image plane. 3-D image reconstruction is performed based on the geometry of the universal coordinates and frame transformations developed for perspective camera vision. Furthermore, a trajectory path symbolizing the velocity of ball in three dimensions is extrapolated. This motion data is used as feedback to guide the robotic arm to hit the ball at specified location. Figure 1 illustrates a pictorial representation of a camera based robot system developed.

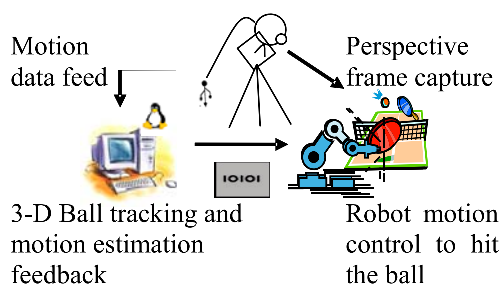


Figure 1. Robot navigation and control set up

Section 2 explains the system theory and the techniques for vision based system development are formulated in Section 3. Section 4 substantiates the use of camera based information as an effective means of feedback for navigating the robot arm.

2 System Theory

This section presents applications of vision in various ping pong playing robots.

2.1 Ping pong playing robot

The principal system components of the ping pong playing robot include a 3-D vision system that locates the ball, a trajectory analyzer that extrapolates the path of ball motion, an expert controller in form of a computational machine that computes the target point, and a robot that hits the ball in a desired approach.

2.1.1 Vision system

The ability to track objects in video sequences allows a robot to rely on vision-based navigation techniques and avoid using active sensors or sophisticated stereo imagers for distance measurement [2]. Information obtained from vision system is used to control the robot's motion in real time. The purpose of vision in the system is to extract information about the ball in space (x, y, z) at particular time (t). This spatial and temporal information is concurrently used to analyze the ball motion. Table 1 reviews vision components for ping pong systems that use stereovision to compute the ball location [4, 8, 13, 14, 17].

Table 1 Vision components for ping pong playing robot

Developer (s)	Vision element	Spatial resoln.	Time resoln.
Andersson, 1986 [14]	Four cameras	756 x 242	60 hertz
Hashimoto, Ozaki, Asano, Osuka, 1987[17]	Binocular camera with four sensors	2048 x 2048	100 hertz
Fassler, Zurich 1990 [13]	Two CCD cameras	422 x 579	50 hertz
Naghdy, Wyatt, Tran, 1993 [8]	Two cameras with frame grabber	1024 x 1024	≈ 60 hertz
Miyazaki, Kusano, 2002[4]	Quick MAG III	640 x 416	60 hertz

Advances in the hardware industry and efficient image and video processing algorithms enable tracking of objects in video sequences and determine motion. This allows camera based vision system to act as sensor information, as compared to the use of active sensor devices [2]. The ping pong playing robot system developed in [3] employs a single CCD video camera along with a standard image-acquisition card based on the chip BT878 (768 x 576 pixels @ 40 hertz) to estimate the ball location using triangulation between ball coordinates and its shadow, through color tracking. In this work, a USB communicative computer camera is interfaced to acquire a frame consisting of (640 x 480) picture elements every 66 milliseconds. The difference in brightness levels of two images is utilized to determine motion in area of interest. Global and local thresholding is applied to segment the ball region from background. Camera calibration based on the geometry of the image plane and analysis of the co-ordinates, is used to determine the location of ball center, relative to its size and position in the image plane. Further, processor architecture is utilized to compute the time for change in ball location and additionally maintain synchronization between the frame capture rate and system processing time. Thus, ball center (x, y, z, t) is computed at the end of each frame.

2.1.2 Trajectory Analysis

Ball position relative to each frame is used to compute its displacement in respective directions. The rate of change of this displacement is measured as the ball velocity and is further utilized to analyze the ball motion in time. The time and position for first and second bounce are estimated in two individual free flight trajectories based on projectile physics. The velocity changes only in the vertical direction changes due to free fall acceleration [5]. An experimental value of coefficient of restitution determines the change in velocity of the ball after the first bounce. This new velocity is used to estimate the second trajectory. Since the robot has to hit the ball before it bounces a second time, a target point relative to the robot's workspace in (x, y, z) is estimated. The robot illustrates motion towards the ball, if the target point within its workspace.

2.1.3 Expert controller

[13, 14, 17] utilize Motorola MC 68020 based architecture to control the robot arm. Transputer architecture is used to control the robot in two degrees of freedom in [8]. In this system, 2.8 GHz. Intel Pentium IV processor performs numerical processing to calculate the distance by which to lead the ball, such that robot can arrive at that position at the same time as the ball. This defines the robot's object retrieval rate. Ping pong requires five degrees of freedom in form of the ball positions along the trajectory, angle at which the ball should be hit and orientation of the paddle [14, 15]. The robot used in this system is a commercial Mitsubishi industrial robotic arm (MIR), RV-2AJ with five degrees of freedom and an operating speed of 0.64 m/s.

In a human ping pong game, a player tracks the ball continuously and hits it at a suitable position and angle after several trials and attempts. The goal is to derive a mathematical paradigm of the human game and teach a robot to plan these strategic skills through effective use of computer vision.

3 System Implementation

Building a system requires a balance between algorithmic requirements and available resources. This section provides a description of techniques and procedures used by the system configuration.

3.1 System configuration

CCD cameras are an important source of geometric information in machine vision and robotics [1]. Video frames are acquired through a CCD based Logitech computer camera. Individual driver based application interface with video for Linux (V4L) is developed to capture 15 frames per second at VGA resolution. Analog to digital conversion is synchronized with the CCD pixel clock. The camera and robot arm are interfaced to an Intel machine. MIR is communicated through a serial interface.

In this work, robot motion is programmed through move master commands of the robot using C programming.

3.2 System Design

In order to avoid a mismatch between user expectations and the ability to realize a computable system, it is important to specify distinctive steps [9]. Subsequent sections explain specific computer vision techniques used to gather the scene information and their application in realizing the system as shown in Figure 2.

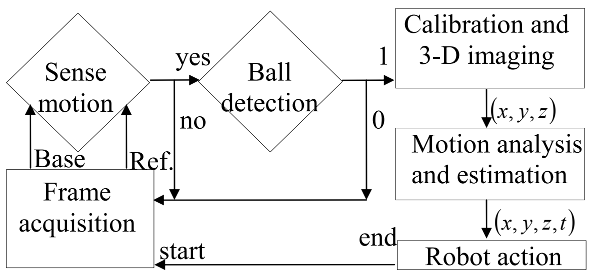


Figure 2. System block diagram

3.2.1 Thresholding and ball detection

Reducing the amount of data to be processed without eliminating the essential information is the key to real time image processing [14]. Object recognition involves recovering information about its reflectance and shape. Displaced frame difference (DFD) of intensity values for specific pixel area of the scene is computed to track changes in the scene [18]. Various point dependent and region dependent global and local thresholding techniques for image segmentation are summarized in [16]. In this work, a preprocessing threshold labeling is performed to segment noise from the area of interest in which motion is sensed. This threshold is gauged as a sum of individual red, green and blue channel values of true color class. The image is further represented at two levels to identify the regions corresponding to changes in brightness levels and hence have the potential of being the ball. These changes are characterized by edges, which are curves corresponding to discontinuities in surface reflectance properties and occluding contours of objects [18]. A fast and high image energy convergent algorithm for first and second order continuity is used to determine various contours [15]. A region dependent threshold based on the retrieved contour points is further applied to fit an ellipse as a single shape approximation for determining the ball silhouette in the image plane. Thus, the topological properties of objects in a scene are used to locate the ball center in image plane.

3.2.2 Camera calibration and 3-D Imaging

This system makes use of camera calibrated sensory information to reconstruct 3-D information of the ball based on the image plane data. Motion accuracy is dependent on camera calibration due to the use of a single camera. Hence, camera calibration is employed as a function of position, orientation and optical characteristics of the camera. A geometric model of the scene based on reference coordinates of universe frame $U(x,y,z)$ based on the robot linear axes, ball frame $B(x,y,z)$ camera frame $C(x,y,z)$, real image frame $F_p(x,y)$ and pixel image frame $I(row, col)$) is developed. A perspective image of the world scene is acquired by the camera as its optical axis is perpendicular to the image plane. Alignment of width centers of this image frame $I(0, col/2)$ and $I(row, col/2)$ with universal axis U_x ensures a zero degree pan in the image plane. Real image frame coordinates $F_p(x,y)$ and ball dimensions are computed from pixel information in the frame as shown in figure below.

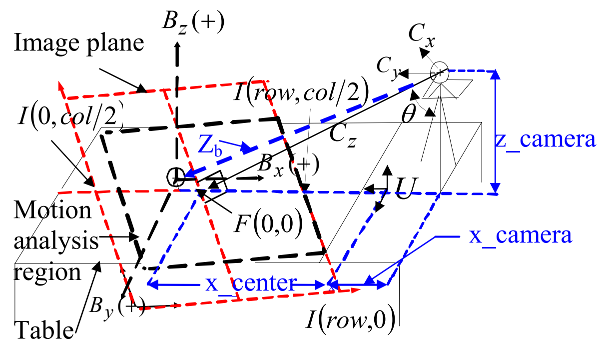


Figure 3. 3-D imaging based on scene and object topology

The fact that size of an object is inversely proportional to its distance from image plane is utilized to compute the ball size as a function of its distance from focal point. An experimental value of this function determines the ball distance Z_b (mm) along the line of focus of the camera. Figure 4 demonstrates the use of this distance to transform from pixel units to real world units (mm) in the image plane.

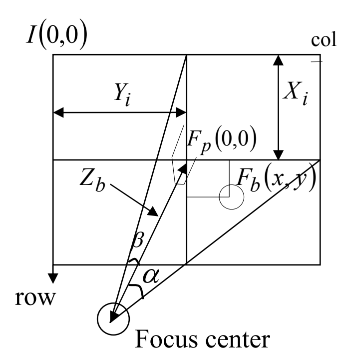


Figure 4. Pixel to millimeter transformation

Experiment based values of α and β are used to map pixel values to physical resolution values using equations (2) and (3).

$$\lfloor X_i = (Z_b * \tan(\alpha)), Y_i = (Z_b * \tan\ (\beta)) \rfloor \quad (2)$$

$$F_b(x) = \left(\frac{F_p(x) * X_i}{col/2}\right), F_b(y) = \left(\frac{F_p(y) * Y_i}{row/2}\right) \quad (3)$$

This locates the ball in real world units on the image plane. Geometric analysis of the scene is used to devise the transformations for 3-D camera vision. Subsequent formulae are derived linear and rotational matrices for single camera vision, determining position and orientation of ball in space, with respect to universal frame U.

$$\theta = \arctan\left(\frac{(x\_center + x\_camera)}{z\_camera}\right) \quad (4)$$

$$T_F^U = \begin{bmatrix} 0 & \cos\theta & -\sin\theta & (Z_b * \sin\theta) - x\_camera \\ -1 & 0 & 0 & 0 \\ 0 & \sin\theta & \cos\theta & -(Z_b * \cos\theta) + z\_camera \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$T_P^F = \begin{bmatrix} F_b(x) \\ F_b(y) \\ 0 \\ 1 \end{bmatrix} \quad (6)$$

Thus, the transformation from universal frame to ball position is calculated as

$$[T_P^U] = [T_F^U] * [T_P^F] \quad (7)$$

$$\begin{bmatrix} B_x \\ B_y \\ B_z \\ 1 \end{bmatrix} = \begin{bmatrix} (F_b(y) * \cos\theta) + (Z_b * \sin\theta) - x\_camera \\ -F_b(x) \\ (F_b(y) * \sin\theta) - (Z_b * \cos\theta) + z\_camera \\ 1 \end{bmatrix} \quad (8)$$

where $B_x, B_{y,} B_z$ are ball centers in respective coordinates with respect to universal frame. Parametric calibration in pan, tilt, focus, and position, proportional to the topological parameters of the scene and ball, enables locating the ball in 3-D from image plane.

3.2.3 Motion estimation

Motion is evaluated as the spatio-temporal displacement of $B_{x,} B_{y,} B_z$ in frame n. Velocity of the ball, which is a vector measurement of the rate and direction of its displacement, is utilized to analyze the locus of projectile ball motion as given below.

$$V_x^n = \frac{B_x^n - B_x^{n-1}}{\Delta_t} = \frac{\Delta_x}{\Delta_t} \quad (9)$$

$$V_y^n = \frac{B_y^n - B_y^{n-1}}{\Delta_t} = \frac{\Delta_y}{\Delta_t} \quad (10)$$

$$V_z^n = \left[\frac{B_z^n - B_z^{n-1}}{\Delta_t} - \left(\frac{1}{2} * g * \Delta_t\right)\right] = \left[\frac{\Delta_z}{\Delta_t} - \left(\frac{1}{2} * g * \Delta_t\right)\right] \quad (11)$$

The vertical motion is influenced by downward acceleration constant $g = -9860\ mm/\sec^2$ as the ball exhibits projectile motion [5]. Δ_t is the time interval measured in milliseconds between two successive ball locations and is maintained at 0.066 seconds through synchronization between the processor ticks and the CCD clocking rate. Figure 5 illustrates an approximation of trajectory path of the ball.

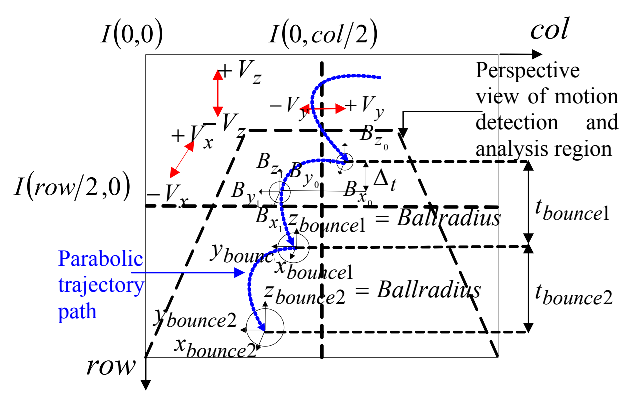


Figure 5. Motion estimation

Subsequent projectile motion equations for trajectory estimation are derived as a function of the velocities with reference to the co-ordinates of ball centers in each frame.

$$B_x^{\ n+1} = B_x^n + V_x^n * \Delta t \quad (12)$$

$$B_{n+1}^y = B_y^n + V_y^n * \Delta t \quad (13)$$

$$B_z^{n+1} = B_z^n + V_z^n * \Delta t \quad (14)$$

3.2.4 Target point computation

Individual ball trajectory before a bounce occurs is calculated based on equations (12)-(14). The time of first bounce is further estimated using the quadratic solution

$$t_{bounce}\Big|_{z=ballradius} = \frac{-V_z^n + \sqrt{\left(V_z^{n^2} - 2 * g * B_z^n\right)}}{g} \quad (15)$$

After the first impact, change in velocity in the vertical direction is restituted by a dimensionless collision coefficient. Since the robot has to hit the ball before the second bounce occurs, time and locations for second bounce are also computed.

Figure 6 illustrates a linear interpolation model developed for the computation of the target point.

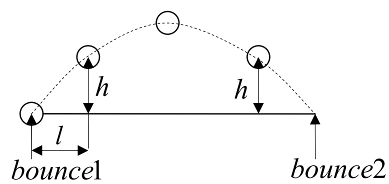


Figure 6 Target point calculation

The hit point for the robot is a $f(x, y, z)$ using

$$T\arg et(x, y, z) = ((l + xbounce2), (ybounce2), (h)) \quad (16)$$

This location is specified to the robot and the robot is programmed to demonstrate movement for hitting the ball. Equation (17) ensures that center of paddle is aligned with the ball center.

$$R_{motion} = R_{\mathrm{hom}\,e} + (T\arg et(x, y, (z - R_{paddle})), a, b) \qquad (17)$$

where, $R_{\mathrm{hom}\,e}$ is the home position of the robot, R_{paddle} is the distance between the robot base and paddle center and a, b are the remaining two degrees of freedom that specify the angle of roll and pitch joints for smooth motion in the robot workspace.

4 System performance and analysis

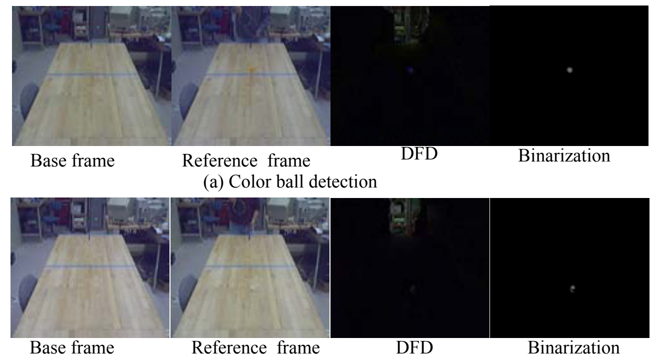
System latency is the time from acquisition of data until it is applied to the control output [15]. A global timestamp measures this latency as a function of processor ticks. Additionally, various local parameters ensure a smooth flow in executing individual blocks of the system as well as their synthesized performance.

4.1.1 Ball detection and 3-D imaging

A global threshold that segments the ball motion and an additional local threshold ascertain the extraction of ball region from the binary image as shown in Figure 7. A comparison of values determining the distance of ball in line of camera axis is as shown in Figure 8 (b) (c). Equations (2) – (8) are further employed to determine 3-D coordinates of ball center as illustrated by Figure 8(d). A performance parameter for this module is scanning for single contour sequence only in the motion-identified region. The system takes 10 ms to perform these computations. In order to maintain synchronization with the frame capture rate, a 45ms delay with 20 % I/O tolerance, based on experimental trials is provided.

4.1.2 Target estimation and robot motion

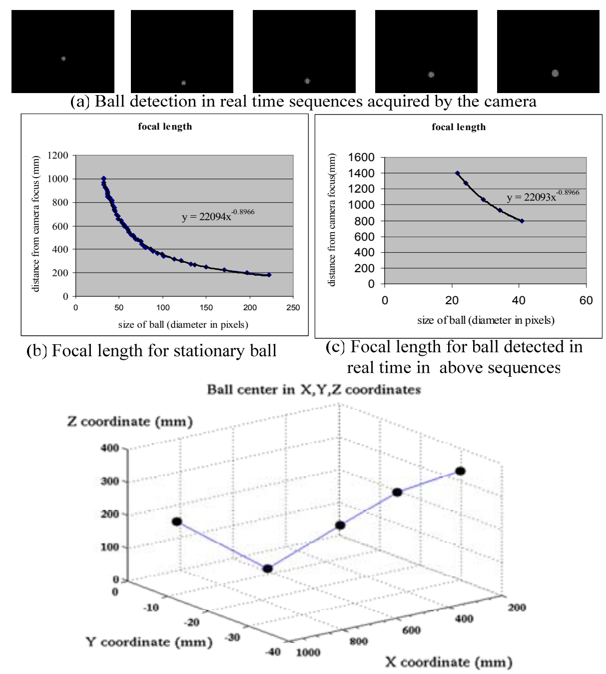
Velocity of the ball in three directions is used to determine the orientation of ball motion with reference to the orientation of robot motion using equations (9) – (11). Since the projectile path of any object is parabolic, the trajectory of the ball is modeled as a parabolic function as shown in Figure (9).



(a) Color ball detection

(b) White ball detection

Figure 7. Ball detection



(a) Ball detection in real time sequences acquired by the camera

(b) Focal length for stationary ball

(c) Focal length for ball detected in real time in above sequences

(d) 3-D coordinates of ball centers in above sequences

Figure 8. 2-D ball location and 3-D

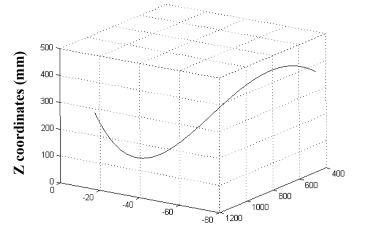


Figure 9. Orientation of ball motion

Δ_t is a decisive performance parameter for target estimation. The system takes 10-20 ms to perform the computations for target estimation, including the previous module. System timing is balanced with an appropriate delay and Δ_t is computed as 30-40 ms.

4.1.3 System setup

Array based algorithmic modules for image math, connected component labeling and estimation are tested using MATLAB® version 7.0.1.24704 (R14) SP1. The system is implemented using a Posix 1003.1c thread based GNU compiler (version 3.3.3) on a Fedora Core 2 kernel (version 2.6.7-1.494.2.2) for pipelined processing. Intel® Image Processing Library functions are used as a benchmark for performing unsigned 8-bit image arithmetic.

5 Conclusion

Processor architecture is efficiently utilized in synchronization with the CCD clocking rate to capture one frame approximately every $1/66^{th}$ of a second and use of frame grabbers and video cameras is successfully avoided. Additionally, ball tracking using single camera based perspective vision was also accomplished through this work. USB version 1.1 is used to interface the camera to the computer. However, VGA resolution in graphics mode is limited to 15 frames per second as there is not enough bandwidth available on the USB bus to squeeze through more, even with compression. Due to low sampling rate, the observed trajectory is not used to predict the future trajectory of the ball. Use of a faster peripheral bus standard IEEE 1394 (a or b) can provide isochronous frame transfer for an increase in sampling rate. Prospective work also involves utilization of all five degrees of freedom of the robot to exhibit the task of hitting the ball at an angle perpendicular to trajectory path of the ball center and train the system for a rally with the human.

References

[1] G. Olson and J.N. Walker, "Advantages of computer cameras over video cameras/frame grabbers for high speed machine vision applications," ELECTRIM Corporation.

[2] M. Mikhalsky and J. Sitte, "Real-time motion tracker for a robotic vision system," Proc. First Canadian Conference on Computer and Robot Vision, pp. 30-34, May 17-19, 2004.

[3] L. Acosta, J.J. Rodrigo, J.A. Mendez, G.N. Marichal, M. Sigut, "Ping–pong player prototype," *Robotics and Automation magazine*, Vol 10, issue: 4, pp. 44-52, Dec 2003.

[4] F. Miyazaki, M. Takeuchi, M. Matsushima, T. Kusano and T. Hashimoto, "Realization of the table tennis task based on virtual targets," Proc. International Conference on Robotics and Automation Vol 4, pp. 3844-3849, May 2002.

[5] D. Halliday, R. Resnick and J. Walker, *Fundamental of Physics*, Wiley, 6th ed., 2002.

[6] L.G.Shapiro and G.C.Stockman, *Computer Vision*, Prentice-Hall Inc., 2001.

[7] K. Hashimoto, T. Ebine, and H. Kimura, "Visual servoing with hand-eye manipulator-optimal control approach," *Robotics and Automation*, Vol 12 issue 5, pp. 766-774, October 1996.

[8] F. Naghdy, J. Wyatt, and S. Tran, "A transputer-based architecture for control of a robot ping pong player," *Parallel computing and Transputers*, pp. 311-317, 1993.

[9] Philip Kahn, "Building blocks for computer vision systems," *Expert*, *Intelligent systems*, Vol 8, issue 6, pp. 40-50, Dec. 1993.

[10] Mansour Rahimi and Waldemar Karwowski, *Human-Robot Interaction*, Taylor and Francis Ltd., 1992.

[11] R.M.Haralick and L.G.Shapiro, *Computer and Robot Vision*, Vol 1, Addison-Wesley, 1992.

[12] D.J. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *CVGIP: Image Understanding*, Vol 55, No. 1, pp. 14-26, 1991.

[13] H. Fassler, H.A. Vasteras, and J.W. Zurich, "A robot ping pong player: optimized mechanics, high performance 3D vision, and intelligent sensor control," *Robotersysteme*, pp 161-170, Berlin: Spring-verlag, 1990.

[14] Russell L. Anderson, *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control*, AT&T Bell Laboratories, MIT Press, 1988.

[15] J.K. Aggarwal, and N. Nandhakumar, "On the computation of motion from sequences of images- a review," Proc. Vol 76, issue 8, pp. 917-935, August 1988.

[16] P.K.Sahoo, S. Soltani, A.K.C. Wong and Y.C. Chen, "A survey of thresholding techniques," *Computer Vision Graphics and Image processing,* Vol 41, pp. 233-260, 1988.

[17] Hashimoto, H., Ozaki, F., Asano, K., and Osuka, K., "Development of a ping pong robot system using 7 degrees of freedom direct drive arm," *Industrial applications of Robotics and machine vision (IECON)*, pp. 608-615,1987.

[18] B.K.P. Horn, *Robot Vision*, MIT Press, Massachusetts, 1986.