Real Time Object Tracking Using Image Based Visual Servo Technique

M. A. El-Bardini, * E. A. Elsheikh, and M. A. Fkirin

Dept. Of Industrial Electronics and Control Eng. Faculty of Electronic Engineering, Menouf, 32952, Menoufya University, Egypt.
dr_moh2000@yahoo.com, mafkirin@yahoo.com
* *Corresponding author. Tel.:+2-048-366-0716 ; fax:+2-048-366-0716;* e-mail: eng_emadelsheikh@yahoo.com

Abstract: In this paper a practical real time object tracking using Image Based Visual Servoing (IBVS) is implemented. The proposed algorithm uses the modified frame difference as an object detection technique integrated with (area, center of gravity and color) as a feature based to segment the detected object. The real position of the interested object is estimated from its features based on camera calibration. A DC-motor position and direction Fuzzy Logic Position Controller (FLC) is implemented to maintain this object in the center of the camera view. Experimental results show that, the developed design is valid to detect and track objects in real time.

Keywords: Real Time Object Tracking, Visual Servoing, Image Processing, Fuzzy Logic Controller, Position Control, DC Motor.

1. Introduction

The Object tracking using camera is an important task within the field of computer vision. The increase of high-powered computers, the availability of high quality inexpensive video cameras and the increasing need for automated video analysis has generate a great deal of interest in object tracking algorithms [1][2]. The real time tracking algorithm consists of two modules; the **first** module is designed to acquire the images, detect the interested moving objects, segment the object or track of such object from frame to frame and estimate the position of this object, then deliver this position as a desired value to the **second** module, which is designed as a *position controller* to maintain the object in the camera view [2][9]. In this case, the information extracted from the vision sensor is used as a feedback signal in servo control, this called Visual Servoing (VS), also known as Vision Based Control [3][4]. The Visual Servoing techniques are broadly classified into, Image Based (IBVS) and Position Based visual servoing (PBVS) [3][4][5][6]. In PBVS, the image features are extracted from the image and a model of the scene and the target is used to determine the pose of the target with respected to the camera [4][7]. While in IBVS, it is also referred to as a feature based technique, because the algorithm uses the features extracted from the image to directly provide a command to motors. And the control law is directly expressed in the sensor space (image space). Many authors consider that the IBVS approach is better of the two, because the image-based approach may reduce computational delay, eliminate the necessity for image interpretation and eliminate errors in sensor modeling and camera calibration [5][6][8].

Various approaches to detect and segment objects has been proposed for visual servoing including, *statistical model based*, *Template-based*, *Background subtraction* based, *feature-based*, *gradient based*, and *Optical flow* object detection methods [10][11][12]. Statistical models are quite computationally demanding in most cases as mentioned in [11]. Template-based method, carries both spatial and appearance information. Templates, however, is suitable only for track objects whose poses does not vary considerably during the course of tracking, where the object appearance is generated from single view [2][13]. Feature-based and Gradient-based or frame difference methods are affected by noise due to intensity changes and some false detection appears in the segmented image [2][11][14].

After segment the object, the camera calibration technique is used to estimate the position of the object [3][15][16]. This position is used as a desired position to the servo controller. The objective of the controller in a vision control system is to maintain a position set point at a given value and able to accept new set point values dynamically. Modern position control environments require controllers are able to cope with parameter variations and system uncertainties [18-21]. One of the intelligent techniques is a Fuzzy Logic [22][23]. The Fuzzy controllers have certain advantages such as; simplicity of control, low cost and the possibility to design without knowing the exact mathematical model of the process [17][24][25].

This work, describes real time vision tracking system which uses a closed-loop control algorithm to track static and moving objects using a single camera mounted on a DC-motor driven platform based on IBVS. The algorithm is designed as two stages, the *first stage* depends on the frame difference with modification in threshold method integrated with (color, area and center of gravity (cog)) as features of the object, to detect and segment moving and static objects. The *second stage* is the proposed servo controller which is designed as a Proportional Derivative Fuzzy Logic Controller (PD-FLC) to overcome the non-linearity in the system. The system was tested in different cases such as; static object static camera, moving object static camera and moving object moving camera for single and multi objects. The practical results show the effectiveness of the proposed technique.

This paper is organized as follows: Section2 describe the problem statement. The Image analysis and feature extraction are explained in Section3, while in Section4 the proposed PD-FLC position controller is illustrated. The practical implementation and results are shown in Section5, and finally conclusions are given in section6.

2. Problem statement

To track moving objects in real time, there are two problems are affects in the system:

1. Problem due to intensity changes

The false detections are appeared in segmented images due to intensity changes.

2. Computational time Problem;

2.1. The time required to detect moving objects is minimized by using frame difference method, because its time is smaller than compared to optical flow methods [26] and Statistical models [11].

2.2. The time required to maintain the object in the center of camera view, is minimized by using PD-FLC as a position controller because it accelerate the transient response of the system .Also the Triangular type membership functions are used to fuzzify their inputs and output due to computation simplicity and ease in real time [17][24]. Analysis of these problems are explained in the following sections.

3. Image analysis and features extraction

The problem due to intensity changes is solved using the proposed algorithm as follows:

1- *Acquiring* the current Images.
2- *Smoothing the acquired images:* using median filter [27] to reduce noise and color variation. In this case, the algorithm able to detect and track objects with specified color.
3- *Frame Difference*: the frame difference serves as an important function to eliminate the background and any stationary objects from the filtered image. It attempt to detect moving regions by subtracting two or three consecutive frames in a video sequence [11] [28], using absolute frame difference as;

$$Diff\,(k) = \left| I_1((i,j),k) - I_2((i,j),k) \right| \quad (1)$$

Where $I_1\,(i,j)$ and $I_2\,(i,j)$ are the current and previous frames respectively.
4-*Threshold*: by applying threshold operation on the difference image to decrease information and eliminate pixels with small intensity values. The threshold value is calculated from the difference image as in [27][29]. But, the noise and false detections are appeared in the output binary image [30]. Therefore, there is a modification in the threshold process by adding two threshold values as follows;

- Applying constant threshold on the difference image to remove noise due to camera and intensity changes. This threshold is determined experimentally. And the output image is a grayscale or RGB depending on the original image, as:

If *diff (i, j) > constant threshold 1*
Then *Out_image (i, j) = diff (i, j),*
Else
Out_image (i, j) =0
End

- The second threshold is estimated for the resulted image after removing the constant noise. The output image is binary image without noise as shown in the results below. Each pixel is calculated as;

$$I_{th}\,(x, y) = 1 \;\; \text{if} \;\; Out\_image\,(i, j) > thresh2$$
$$= 0 \qquad \text{otherwise}$$

5- *Noise removing*: by applying binary area open morphological operation [27], to remove small objects in the segmented image.
6- *Edge detection*: the edge detection approach used, is the Canny Edge detector [30].
7- *Position Estimation:* After segment the interested object based on (color and area) features, the position of the object is calculated with respected to the center of the camera view (**cov**), which used to track the object [28]. It is calculated in pixel as :

$$Position\,(x, y) = cov\,(x, y) - cog\,(x, y) \quad (2)$$

Where *cog* is the center of gravity for the object (pixel). This position is converted into angular position (degree) and becomes a desired position (θd) for the fuzzy logic position controller.

4. Controller Design

The Fuzzy Logic Controller (FLC) is designed as a Proportional, Derivative (PD) controller type to control the position and direction of the DC-Motor according to the desired (estimated) position. The discrete time form of the PD-Controller is:

$$u(k) = k_p\,.e(k) + k_d\,.\Delta e(k). \quad (3)$$

Where $U\,(k)$, $e\,(k)$ and $\Delta e\,(k)$ are the control signal, error signal and error change(Ce) at the sample instant k respectively. And K_P and K_D are the controller parameters. The error is defined as a difference between estimated (reference position $(\theta_d\,(k))$ and previous estimated position ($\theta_d\,(k\text{-}1)$) in degree. $e\,(k\text{-}1)$ is the error value at the previous sample time. For the k^th sampling instant, **(**$e\,(k)$**)** and **(**$Ce\,(k)$**)** can be expressed as:

$$e(k) = \theta_d(k) - \theta_d(k-1)$$

$$\Delta e(k) = e(k) - e(k-1) \quad (4)$$

4.1. Fuzzification

The inputs for the FLC are the error and change of error. The fuzzification strategy converts the crisp input variables into suitable linguistic variables (e.g. Zero "Z", Positive Small "PS"… etc.). The action of the stage is primarily depending on the membership functions (µfs) that describe the input linguistic terms [23]. *Triangular* type membership functions are used to fuzzify the error(e), change of error(Ce) and output control signal(U). Graphical illustration of (µfs) and their ranges for (err, Ce and U) signals are shown in Figure 1 respectively. The µfs are; NVH: Negative Very High, NH: Negative High, NM: Negative Medium, NS: Negative Small, Z: Zero, PS: Positive Small, PM: Positive Medium, PH: Positive High, and PVH: Positive Very High. The ranges for universe of discourses are selected according to the maximum error and change of error values occurred. Therefore, these ranges are in the intervals of [-60 ^{o}, 60 ^{o}] for error, [-20 ^{o}, 20 ^{o}] for change of error and [-1 , 1] for the output control signal (U).

4.2. Rule base design

The relationship between input and output of the system, is called the *rule-base* or *If - then* rules [17]. This relationship must be obtained correctly to improve the performance of the fuzzy logic based control system. This process determine the rules used to track the reference position set point with minimum steady state error. For instance, If the Error (e) is Negative High (NH) and Change of Error (Ce) is Zero (Z) then the output (U) is Negative Very High (NVH). The full set of rules are summarized in the table 1.

4.3. Defuzzification process

At the final step, the determination of both the applicable rules for any fuzzy controller inputs (*e*, *Ce*) and the fuzzy control output (*U*) are performed by inference engine strategy. The fuzzy results obtained from the above rules should be defuzzified to convert them into meaningful crisp values. For this purpose, the center of area (*COA*) method, proposed by [23] is used. It is expressed as;

$$U_{COA} = \frac{\sum_{i=1}^{n} \mu i(u).u}{\sum_{i=1}^{n} \mu i(u)} \quad (5)$$

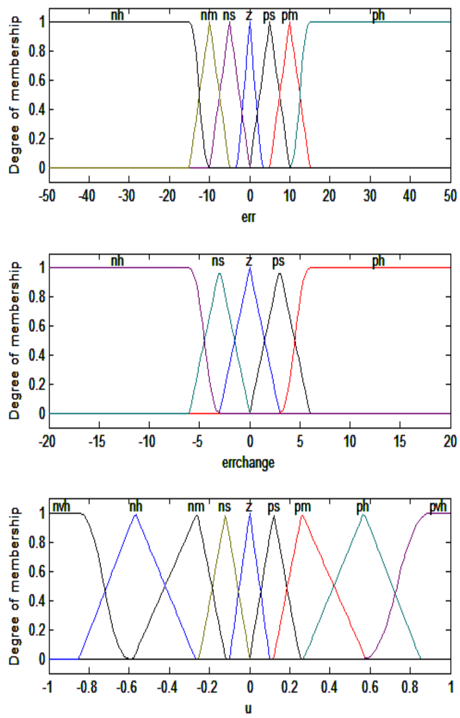


Figure 1. Membership functions for error, change of error and output control signal (u), respectively.

Table 1. Summary of Fuzzy Rules

Ce \ e	Nh	Nm	Ns	Z	Ps	Pm	Ph
Nh	Nvh	Nvh	Nh	Ps	Pm	Pm	Ph
Ns	Nvh	Nh	Nm	Ps	Ps	Pm	Pvh
Z	Nvh	Nh	Nm	Z	Pm	Ph	Pvh
Ps	Nvh	Nm	Ns	Ns	Pm	Ph	Pvh
Ph	Nh	Nm	Ns	Ns	Ph	Pvh	Pvh

5. Experimental results

5.1. System description

The implementation of the visual servoing system is divided into **hardware and software**. The hardware used in the proposed architecture includes the following items:

- PC Pentium 4. CPU 1.8 GHz/512 Mb cache, 512 MB RAM,
- Interface, Motor Drive Circuits and Power Supply Unit, (+5 and +12 VDC).
- Brush type DC-Motor (12 v) and Incremental Encoder with (*105* Slot with *Resolution = 3.429 degree/slot*) to measure the actual position of the camera.
- CCD web camera (¼" color progressive CMOS,640×480pixels) as a vision sensor.

Software implementation: The algorithms are implemented using Matlab software. The block diagram of the system is shown in Figure 2.

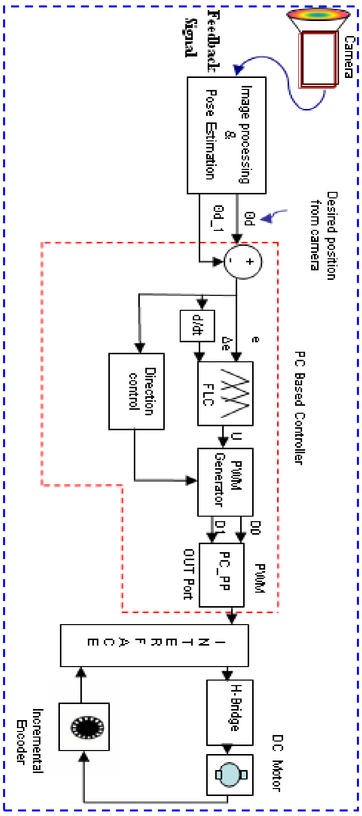


Figure 2. Block diagram of the PC- based DC motor visual servoing system

5.2. Practical results

To test the noise effects on the scene, different experiments are carried to test the proposed algorithm to track objects in the *image frames without controller* to identify any noise in the scene (static scene static camera). Then some testes are carried to track objects in real time (dynamic scene moving camera).

5.2.1. Static scene static camera

a) *Comparison between conventional and proposed frame differences***.**

The conventional frame difference (FD) is depends on one threshold value to produce the inter-frame difference image. When the threshold value is small, some noisy pixels appeared. These noisy pixels will give rise to erroneous object tracking. When the threshold value is large, some moving pixels are lost. The noise is detected in the case of static scene Figure (3-a). But, when applying the proposed algorithm, the noise is removed from the image as shown in Figure (3-b). When the moving object appears into the frame, the noise is decreased, this is because the intensity changes due to moving object is larger than intensity changes due to noise, as shown in Figure (3-c, d).

b) White background, single object detection.

This test is used to detect objects based on object features such as color feature, because the frame difference method cannot detect static objects. Figure 4, shows different locations for the object and their Segmentation. Estimated positions of the object (based on camera calibration technique) with respect to real world positions are summarized in table.2. All values are measured with respect to center of camera view. Table 2 shows that the estimated positions(θe) are close to measured positions(θm). The average processing time in the case of static scene about (0.54 sec).

c) Complex background, multiple object detection.

The modified frame difference is integrated with the object features (color and area) to detect and segment moving and static objects from multiple objects. As shown in Figure (5-a), the algorithm can detect and segment the red color object with specified area from multiple moving objects are appears in the scene. The edge detection for the segmented object at each frame is shown in Figure (5-b). Estimated position (degree) in (x, y) directions are shown in Figure (6-a,b).

5.2.2. Dynamic Scene and Moving Camera (Visual Servoing).

After testing the algorithm to track moving and static objects in image frames (static scene static camera), the algorithm is able to track moving objects in real time. To track objects in real time, the camera is mounted on the shaft of DC-motor. In this case the estimated position is used as a desired angle for the PD-FLC controller. The output of the FLC is a Dynamic Level Control Signal(DLCS).

a) Step Response test to track objects in real time.

In order to test the performance of the proposed real time tracking algorithm, the incremental encoder mounted on the shaft of the motor is used to measure the actual position of the camera. The required object is the red color object, as shown in Figure7. The algorithm detects the red object from the first frame, and then calculates the desired position, as shown in Figure (7-a). The desired position is (-19.2^{o}) into CCW. The FLC generate the DLCS as shown in Figure (7-b), which used to generate Pulse Width Modulated (PWM) signal. The PWM signal is used to control in the speed of the motor to direct the camera into the desired position (*cog*) of the object smoothly without overshoot. As shown in Figure (7-c), the controller able to generate number of loops to apply the PWM signal number of times on the motor, to reach the desired position in very short time. The average processing time observed to maintain the object in the center of view is about (1.3 sec).

(a) (b) (c) (d)

Figure 3. a. Conventional FD static scene, b. Proposed FD, c. Conventional FD for moving object, d. Proposed FD.

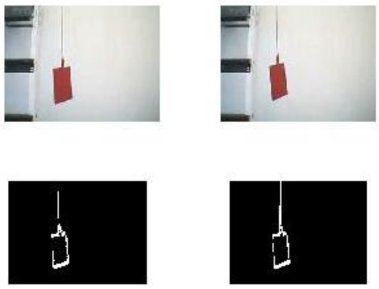
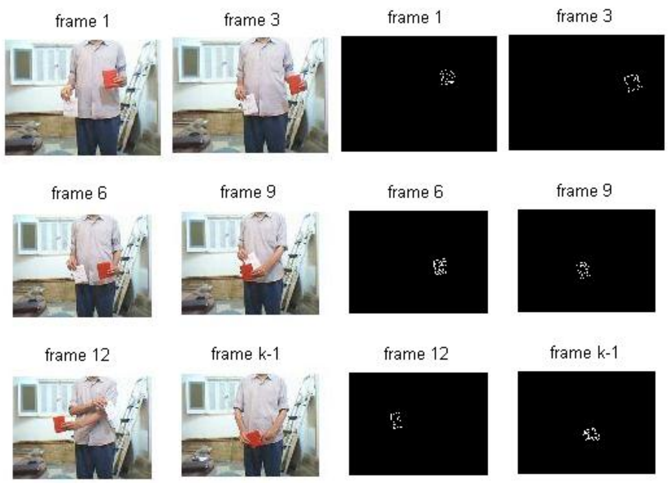


Figure 4. Single object Red color segmentation.

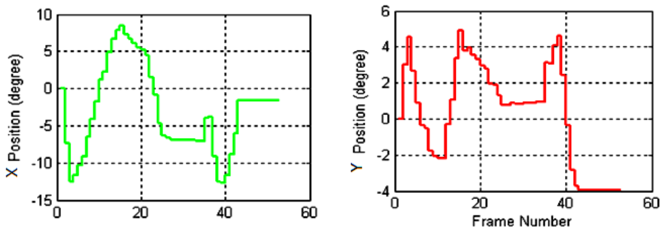
Table 2. Summary for estimated and measured position values.

Position in X-direction (cm)		Position in Y-direction (cm)	
θe	**θm**	**θe**	**θm**
0	0	0	0
9	9.5	9.094	9.5
17.48	18.5	8.65	8.5
37.5	38.5	18.91	18
49	51	27.47	27
67	68	17	16.5



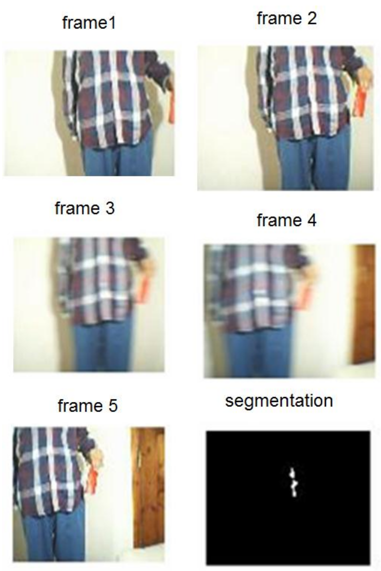
(a) **(b)**

Figure5.a. Motion of multiple objects, b. red color segmentation.

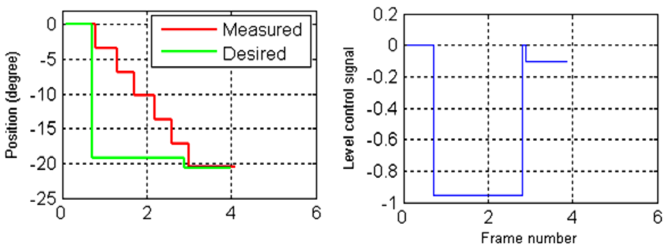


(a) **(b)**

Figure 6. a. Estimated position X-direction (Degree). b. Estimated position in Y direction (Degree).



a. Step response for object tracking



b. Step response and FLC Control signal

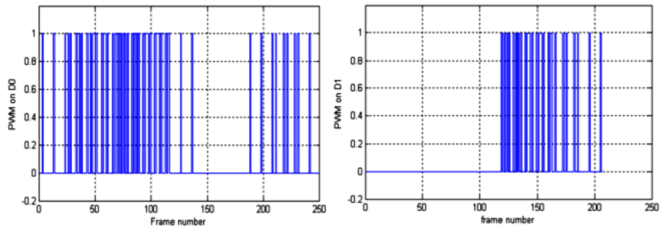


Figure 7 .c- PWM signal

b) Tracking moving objects in real time

This test is used to detect and track red object from complex background, as shown in Figure8. The algorithm tries to maintain the object in the center of camera view from frame 2 until the object stops its movement and becomes in the center of view at frame 95. At frame 118, the red object moved out form the camera view in the other direction instantaneously and then stop. As shown in Figure 8, at frames (141 to 164) the algorithm tries to maintain the object in the camera view and the object becomes in the center at frame 210. Estimated position of the object at each frame with respect to measured position using encoder are shown in Figure 9. The DLCS output control signal of the PD-FLC position controller is used to generate the PWM signal. The average processing time to track objects about (0.98 sec to 3.93 sec).

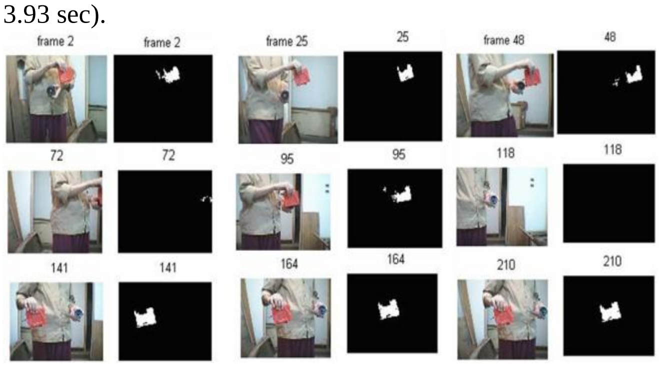


Figure 8. Complex background, Sample of frames to detect and track the red color object.

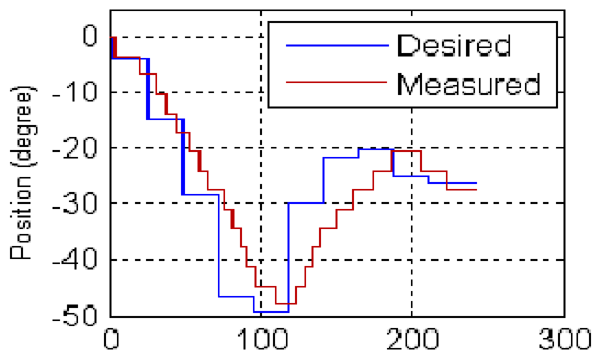


Figure 9. Estimated position w.r.t. measured position.

6. Conclusions

The real time object tracking based on IBVS is implemented. The algorithm merge the feature based and the modified frame difference technique to detect, segment and track Static and Moving objects in real time. The PD-FLC position controller was designed and implemented to generate a PWM signal that used to control the position and direction of the DC-Motor, to maintain the tracked object in the center of camera view, via hardware circuits. A PWM software program has been developed to control the speed of the motor to reach the desired position smoothly without overshoot. Different tests are carried to verify the validity of the proposed algorithm such as, static scene static camera and moving object moving camera. The results show that, the static and moving objects can be detected and segmented without noise and false detections in the scene. The average processing time was observed since the image acquired until the object becomes in the center of camera view is between **0.54 : 3.93** sec. It is depending on the position of the object with respect to the camera view.

References

[1] N. Zuech "Understanding and applying machine vision", Marcel Dekker Inc., 2nd edition Revised and Expanded, 2000.
[2] A. Yilmaz, O. Javed and M. Shah, "Object Tracking: A Survey", ACM Computing Surveys, Vol. 38, No. 4, Article 13, December 2006.
[3] P. I. Corke, "Visual Control of Robots: high performance visual servoing", Research Studies Press Ltd., 1996.
[4] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control", IEEE Trans. on Robotics and Automation, Vol. 12, No. 5, pp. 651-670, October 1996.
[5] D. Kragic and H. Christensen, "Survey on Visual Servoing for Manipulation", Computational Vision and Active Perception Laboratory, Stockholm, Sweden, 2002.
[6] F. Chaumette and S. Hutchinson, "Visual Servo Control Part I: Basic Approaches", IEEE Robotics and Automation magazine, pp. 82-90, December 2006
[7] V. Lippiello, B. Siciliano and L. Villani, "Position-Based Visual Servoing in Industrial Multirobot Cells Using a Hybrid Camera Configuration", IEEE Trans. On Robotics, Vol. 23, No. 1, February 2007.
[8] D. Fioravanti, B. Allotta and A. Rindi, "Image based visual servoing for robot positioning tasks",Springer Science, Meccanica, Vol. 43, pp. 291–305, 2008.
[9] C. Micheloni, G. Foresti "Real-Time Image Processing For Active Monitoring Of Wide Areas", J. Vis. Commun. Image R. Vol. 17, pp. 589–604, 2006.
[10] Y. Yoon, G. DeSouza and A. kak, "Real Time Tracking And Pose Estimation For Industrial Objects Using Geometric Features", IEEE Int. Conf. On Robotics & Automation, Taipci Taiwan, pp.3473-3478, September 14-19, 2003.
[11] P.Y. Yeoh and S.A.R. Abu-Bakar, "Accurate Real-Time Object Tracking With Linear Prediction Method", IEEE Int. Conf. In Image Processing ICIP 2003, Vol. 3, pp. 941-944, September 14-17, 2003.
[12] A. Elgammal, D. Harwood, L. S. Davis, "Non-parametric Model for Background Subtraction" IEEE ICCV99 Frame Rate Workshop. IEEE 7th Int. Conf. on Computer Vision. Kerkyra, Greece, September 1999.
[13] S. Theodoridis and K. Koutroumbas, "Pattern Recognition", Elsevier Academic Press, 2nd edition, 2003.
[14] G. De Cubber, S. Berrabah, H. Sahli, "Color-based visual servoing under varying illumination conditions", Robotics and Autonomous Systems, vol. 47, pp 225–249, 2004.
[15] X. Wan and G. Xu, "Camera Parameters Estimation and Evaluation in Active Vision System", Elsevier Science Ltd, Pattern Recognition, Vol. 29, No. 3, pp. 439-447, 1996.
[16] Z. Zhang, "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations", The Proceedings of the Seventh IEEE International Conference on Computer Vision, Vol. 1 pp. 666-673. 1999.
[17] Paul I- Hai Lin, Santai Wing and John Chou , " Comparison on Fuzzy Logic and PID Control for a DC Motor Position Controller ",IEEE, Industry Applications Society Annual Meeting, volume 3,pp 1930 – 1935, 1994.
[18] G. Cheng and K. Peng "Robust Composite Nonlinear feedback Control with Application to a Servo Positioning System ", IEEE, Trans. On Ind. Elec., vol. 54, No 2, pp 1132-1140 April, 2007.
[19] Tolgay Kara, Ilyas Eker "Nonlinear Modeling and Identification of a DC motor for Bidirectional Operation with Real
Time Experiments", Energy Conversion and Management, volume 45, pp 1087-1106, 2004.
[20] M.A. Fkirin, K. N. Khatab and A. A. Montasser "Identification of Stochastic Processes in Presence of Input Noise", Cybernetics and Systems, an International Journal, Volume 20, pp 489-500, 1989.
[21] M. A. Fkirin "Fixed-interval Smoother in the Identification of Time-Varying Dynamics", INT. J. System SCI., volume 20, NO. 11, pp 2267-2273, 1989.
[22] L. A. Zadeh "A Fuzzy-Algorithmic approach to the Definition of Complex or Imprecise Concepts", Int. J. Man-Machine Studies, pp 249-291, 1976.
[23] D. Driankov, H. Hellendoorn and M. Reinfrank, "An Introduction to Fuzzy Control ", Springer – Verlag Berlin Heidelberg, 1993.
[24] Tipsuwan, Y. Mo-Yuen Chow "Fuzzy Logic Microcontroller Implementation for Motor Speed Control" , The 25th Annual Conference of the IEEE, Industrial Electronics Society, IECON '99 Proceedings, volume 3, pp 1271- 1276, 1999.
[25] C. Pėrez, O. Reinoso and M. Vicente, "Robot hand visual tracking using an adaptive fuzzy logic controller ", WSCG Posters proceedings, February 2-6, 2004.
[26] C. Cėdras, M. Shah " Motion based recognition : a survey ", Image and Vision Computing, Vol. 13, No. 2, March 1995.
[27] Rafael C. Gonzalez and Richard E. Woods "Digital Image Processing", Prentice-Hall, Inc., 2nd edition, 2001.
[28] Y. Shiao and S. Road, "Design and implementation of real time tracking system based on vision servo control", Tamkang Journal of science and Engineering, Vol. 4, No. 1, pp. 45-58, 2001.
[29] R. C. Gonzalez, R. E. Woods and S. L. Eddins, "Digital Image Processing Using MATLAB", Prentice-Hall, Pearson Education, Inc., 2004.
[30] F. Cheng,Y. Chen, " Real time multiple objects tracking and identification based on discrete wavelet transform", Elsevier Science Ltd, Pattern Recognition, 39 , pp. 1126 – 1139, 2006.