ISSN 2229-5518

Camera Based Guiding System for a Small Car

Mamon J. Mohammed[1], Loay E.George[2]
Computer of Science, College of Science, University of Baghdad, Iraq

Abstract— This paper presents the development of camera based guiding system for a small car. The tasks handled by guiding system are the processing of collected visual data, deflections from the reference route mark and issue the steering commands. The information coming from a CCD color camera which is mounted on the RC car, are processed in order to determine the position of the car relative to the planned route.

Simple mathematical models and image processing algorithms have been developed and utilized to determine the car position and motion deflection accurately and with lowest possible computational load. In case of outdoor environment the camera based guiding system task become harder, because the developed software should be compensate the changes in both light and the relative appearance of reference mark on the ground which may appear solid or broken area due to the high variation in spectral reflectance beside to the appearance of painted potholes.

The established guiding system includes a module for originating the proper motion control commands. The proportional steering command origination module depends on the position and heading of the vehicle (relative to the planned route), such that the moving vehicle should be stick around the route reference line till reaching its planned destination.

Keyword— Guiding system, Vision based Guidnce system, Line following robot.

1 INTRODUCTION

In order to navigate through its environment a robot vehicle needs to determine its position relative to certain objects or landmarks. If the robot is going to follow a pre-defined path then the information it obtains must provide its position relative to the path. Several techniques have been developed to allow robot determines its position. If the robot is following a pre-defined path, the path itself can be marked. Then, with an electronic or contact sensor, the robot can follow this path. Another method for determining position which has received much attention lately is based upon computer vision [1].

Many of the developed camera-based systems are used to track moving objects. There are many methods used to implement visual servoing or visual tracking by the camera. Some methods use training, known model, or initialization [2, 4, 5]. Other methods involve signature vector or motion prediction [3, 6, 7]. These methods somehow require grid and calibration for position estimation. One of the approaches for visual tracking task is to use Active Appearance Models (AAM). However, it is limited to have all points of the model visible in all frames. Birkbeck [2] have introduced a notion of visibility uncertainty for the points in the AAM in their work, removing the above limitation and therefore allowing the object to contain self-occlusions. Mikhalsky [3] have proposed an algorithm which is based on extracting a signature vector from a target image and subsequently detecting and tracking its location in vicinity of the origin. This process consists of three main phases: signal formation, extraction of signatures, and matching. Leonard [4] have implemented visual servoing by learning to perform tasks such as centering. The system uses function approximation from reinforcement learning to learn the visuomotor function of a task which relates actions to perceptual variations. The research done by Denzler [5] describes a two stage active vision system for tracking a moving object which is detected in an overview image of the scene, and a close up view is then taken by changing the frame grabber's parameters and by a positional change of the camera mounted on a robot's hand. For object tracking, they used active contour models, where the active contour is interactively initialized on the first image sequence. However, errors may occur if there are strong background edges near the object, or if the ROI only partially covers the moving object.

The goal of this paper is developing a camera based guiding system for small car using real time image processing that will follow a specific color of reference route line and originates the proper motion control command.

- *Mamon J. Mohammed has master's degree in computer science in University of Baghdad, Iraq. E-mail: mamoun_87@yahoo.com*
- *Loay E. George has PH.D degree in Computer science in University of Baghdad, Iraq. E-mail: loayedwar57@yahoo.com*

2 GUIDING SYSTEM PLATFORM

This system can be divided into several parts:
- Actuators (Motors and wheels)
- Central Processing Unit
- Optical sensor
- Microcontroller (Digital to Analog Converter)
- Driver

A special "subset" of continuous motors is the servo motor, which in typical cases combines a continuous DC motor with a "feedback loop" to ensure the accurate positioning of the motor. A common form of servo motor is the kind used in model and hobby radio-controlled (R/C) cars and planes. R/C servos are in plentiful supply, and their cost is reasonable [8]. RC car with size 1:6 scale was developed for this project.

The system is equipped with mini laptop as central computing unit, the decision to use x86 compatible processor was driven primarily by the fact that most of available microcontrollers in the market, at the time of developing this project, are not yet completely capable of processing large amounts of vision data in real-time. The micro-controller (Arduino board) have been used to do command translation after being issued from the computer, this microcontroller does not handle any of the primary processing tasks for controlling the car. A mini laptop was placed in the car. Its CPU speed is 1.6 GHz and memory storage 1 GB. The computer was connected to the visual sensor and microcontroller via USB ports. Image data acquisition is done using a USB camera connected to the computer. The used camera is able to deliver an image with resolution of 640x480 pixels. Figure (1) presents the developed system layout.

An application programming interface (API) called "AVICap", which is a window class, was adopted to provide the developed control program with a message-based interface to access the video and acquisition hardware and to control the process of streaming the captured video to a disk. This class allows to check whether the connected webcam is working or not. If not working it, simply, gives a warranty message otherwise it captures the video coming from webcam and display it using a picture box object.

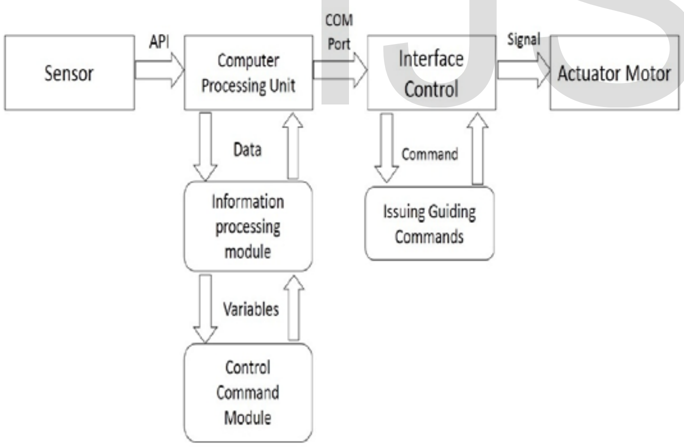


Fig. 1 System structure

3 Proposed System Description

In this paper, the main objective is to present a simple but effective camera-based vision guiding system which acquires color images of the scene in continuous mode (4 frames/second). For each captured frame, the system detects the presence of reference route mark area that has particular color. Then, it determines the coordinates of the center of mass (in pixels) of the detected mark area appeared in the captured image plane. Before jumping to the next frame the system adjusts automatically the scanning window ensure the scanning task is accomplished as fast as possible and to be restricted around reference mark. The guiding system sends an appropriate steering control command to steer the car wheels such that the car's frontal side is directed towards the target route. Figure (2) presents the steps of the established guiding system modules.

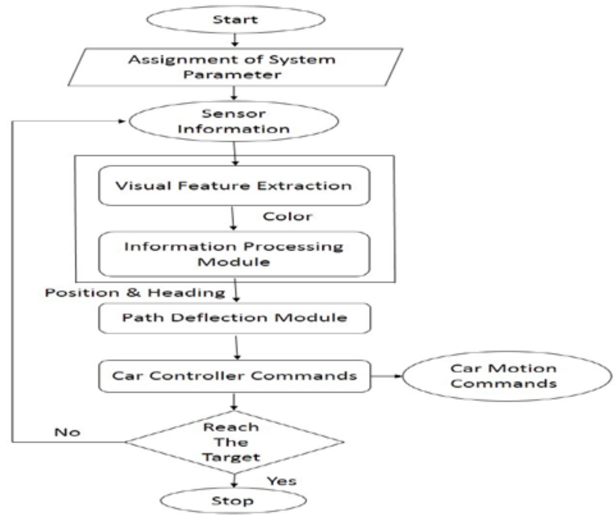


Fig. 2 Guiding system Modules

3.1 Actual Path Parameters Extraction Module

This module is mainly concerned with the extraction the path monitoring parameters (i.e., from the captured image data). The visual data set consist of images snapshots, captured from the video sent by the webcam, they are loaded, repeatedly (i.e., one snapshot per one guiding time slot). Each snapshot is loaded as 24 bit/pixel BMP image file. After loading the image data it is decomposed into red, green and blue color arrays.

3.2 Guiding Based on Web-Camera Data

The established guiding system passes through two phases, in the first phase the user should be interact to make the system capable to determine the dynamic range of "the reference color of the route mark". In the second phase the system switches to the automatic guidance mode, such that it makes its decision. At the first stage of the system the visual data, which sent from the optical sensor as frames, is processed by the processing unit, the outcomes of the processing is mapped to determine the route deflection parameters.

In other words, the whole operation of the established system consists of two phases:

1. The initialization phase: which is developed to predefine some important visual leading parameters; which are required to establish a knowledge about the visual characteristic of the route reference line mark (i.e., its color and variance), and the color variability of the surrounding area.
2. The vehicle guidance phase: in this phase the system should be track the route reference line mark, determine the actual path deflection and heading, and finally issues the proper steering commands.

ISSN 2229-5518

3.2.1 The Initialization Phase

This phase will allows the user to set the required system tracking parameters and conditions. At the beginning step, the user should be manually select a reference area on the captured frames which, as perfect as possible, reflects the color characteristics of the reference line. Figure (3) shows the steps of the initialization phase.

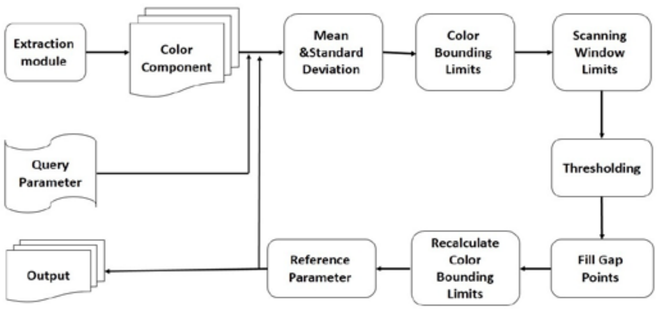


Fig. 3 Initialization phase

The whole conducted tasks in the initial phase are:

A. load a reference frame: the car should be aligned upon the reference route line such that its trace in the captured frames should be clear, and its position in the frame is as close as possible to the central vertical line of the frame.

B. Visual features extraction: As second task, the chosen area by the user for representing the reference route mark is statistically analyzed to determine the color statistical attributes which representing the spectral nature of mark area. The color attributes extraction task implies the following steps:

Step-1 Determine the first order statistical parameters: which includes determination of the mean & standard deviation for the three color components (Red, Green and Blue) of the selected training area.

Step-2 Determine the color bounding limits (i.e., Min & Max of the Reference route Area): After the calculation of mean (μ) and standard deviation (σ) the minimum and maximum values of each color component are assessed according to statistical bases that is determined:

$$\text{Min} = \mu - 2.5\sigma \qquad (1)$$

$$\text{Max} = \mu + 2.5\sigma \qquad (2)$$

Step-3 Calculate the scanning window position; which should has a user predefined width & height values and allocated around the selected mark area.

Step-4 Apply color binarization to assign the nominated reference route mark segment

Step-5 Fill the gap points

Step-6 Re-Calculate the bounding values of the color components: since some of the gap points are flagged as target points, then the bounding color limits should be adjusted

Step-7 Determination of the reference parameters (Xref, Yref, AngRef) using linear regression that is:

Step-7.1 For each line within the search window: scan the pixels of the produced binary image and calculate the number of target points then determine their linear horizontal density and the average value of their x-coordinates.

Step-7.2 Filter out the abnormal lines which have bad linear density. Then determine to overall average of the x-coordinate and the corresponding standard deviation.

Step-7.3 Make linear regresion for the cental points belong to the filtered-in lines.

C. Output: The allocated reference route area, its position, direction, and its color bounding limits are registered as the outcomes of the initial phase.

3.2.2 Vehicle Guidance Phase

This phase started once the guidance system operation mode is switched to guide the vehicle, autonomously, to its destination. Most of the previous steps that are applied on the first frame (i.e., in the initial stage) are reapplied on the next coming frames. The exception is instead of determining the necessary route parameters the outcomes of the previous stage is used to adjust the window search parameters values (i.e., like the bounding values of the color components and the scanning window limits).

The following steps are applied on the next coming frames in a sequential manner (i.e., frame after frame). Figure (4) shows the autonomous guiding steps. The taken tasks in this phase are the following:

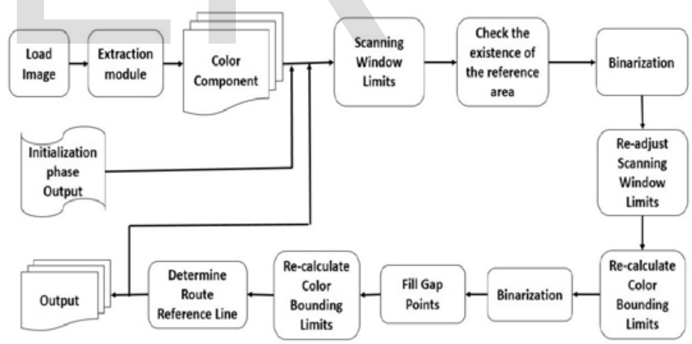


Fig. 4 Vehicle motion phase steps

A. Input: This stage is same as the input stage taken in the initial stage. The horizontal position (Xref) of the reference route line and color bounding limits which was determined in the previous frame are used as input.

B. Processing: some changes have been made in this stage. The involved steps are illustrated in the following:

Step-1: Determine the position of the scanning window (i.e., allocate the left and the right edges of the scanning window). The Xref parameter value is used to determine the horizontal edges position of the scanning area.

Step-2: Check the existence of reference route area. The color bounding values are used to decide whether the tested pixel belong to reference route area (i.e., target area) or not. In case of not enough number of pixels are flagged as target points then the range of the bounding color limits is, gradually, expanded a little bit till collecting enough number of pixels.

Step-3: Make color-based binarization and re-adjust the scanning window.

Step-3.1: Make color binarization for the image region covered by the scanning window, then calculate the horizontal linear density for each line of the produced binary image. Find the total number of target points and determine the x-average value for each scanned line.

Step-3.2: Make the first boundary adjustment for the route reference mark area using the vertical linear density. The average of total number of target points and the corresponding standard deviation is determined to assess the threshold value. Then, check the elements of the 1-D array which holds the number of target points allocated in each column, if the number is more than the threshold value then adjust the width of the scanning area.

Step-3.3: Make the second boundary adjustment using the vertical accumulated density. First, multiply the threshold value by three. Second, use this value upon the accumulated number of target points for each six columns together. Then readjust the width of the scanning area.

Step-3.4: Re-Adjust the color components bounding values.

The rest of the taken steps are similar to the steps taken in the initial phase.

Step-4: Re-apply binarization (to construct a new binary image).

Step-5: Fill the gap points.

Step-6: Re-calculate the bounding values of the color components.

Step-7: Determine the route line reference parameters (Xref, Yref, AngRef) using linear regression.

Step-7.1: Scan horizontally to find the density points and their statistical parameters.

Step-7.2: Filter out the abnormal lines (have bad linear density).

Step-7.3: Make regression using filtered-in good lines.

Step-8: Determine the parameter (i.e., intercept and slope) of the allocated route line.

Step-9: Calculate the vehicle position deflection (i.e., ΔPosition) and heading deflection (i.e., ΔHeading).

When the above steps are accomplished upon a frame, they will repeated again and again upon the new coming frames till the vehicle reach its destination.

C. **Output:** At the end of processing stage, which is applied upon each coming frame, the control commands are determined and send to HW driver. The originated command convey the taken decision for steering the car to ensure its movement along the reference route line. The issued command depends on the relative heading angle deflection of car motion and its position shift relative to the reference line.

3.3 Motion Command Determination

The camera based guiding systems includes a module for originating the proper motion control commands which make the system final response match the planned route requirements. The established closed-loop (feedback) controller uses the information gathered from the vehicle sensors to determine the required commands; and sends them to the vehicle actuator(s). This is the most suitable robust method of control for mobile robots (e.g. vehicles) since it allows the robot adapt its motion with any changes may occurred in its environment.

The established controller, in our project, issues steering commands which are simply depend on both position deflection (ΔP) and heading deflection (ΔH). The position deflection part of the originated guidance command is manipulated linearly while the heading deflection part is manipulated non-linearly as presented in the following equations

1. The originated command (Cmd) is

$$Cmd(\Delta P, \Delta H) = F_1(\Delta P) + F_2(\Delta H) \qquad (3)$$

Where $F_1(\Delta P)$ is the command part which depends on the position deflection; and $F_2(\Delta H)$ is the command part which compensates the heading deflection.

$$F_1(\Delta P) = \beta(P_{actual} - P_{planned} \qquad (4)$$

2. The command position dependent part is linearly proportional with position deflection value:

Where P_{actual} is the current actual position of the vehicle; $P_{planned}$ is the planned position (or the route line); and β is a proportional factor (i.e., the command weight that compensate the position deflection).

3. The command heading dependent part is exponentially dependent on the heading deflection

$$F_2(\Delta H) = \frac{M}{sign(\Delta H) * K(\Delta H)} \qquad (5)$$

$$K(\Delta H) = \begin{cases} 0 & if\ |\Delta H| < T \\ \exp(\alpha|\Delta H|) + 1 & if\ |\Delta H| > T \end{cases} \qquad (5.1)$$

$$sign(\Delta H) = \begin{cases} 1 & if\ \Delta H \geq 0 \\ 0 & if\ \Delta H < 0 \end{cases} \qquad (5.2)$$

Where M is the weight of command heading part; α is a turning decay rate which is used as tuning parameter to control the system sensitivity relative to heading deflection; T is the lowest heading deflection value above it the system makes a response depending on the heading deflection.

The proper values of the control parameters (i.e., β, α) depends on the physical characteristics of the vehicle, there is no fixed set of values that can be adopted to every implementation of the control system. Instead, the (β, α, M) parameters used in commands determination equations (i.e., vehicle) must be *tuned* to the particular platform to which they will be used to control.

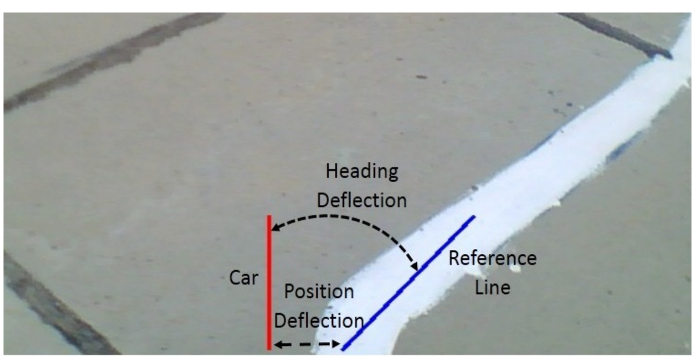
4 Result and Conclusion

In this section an attention is directed toward the results of the tests conducted on the camera based system to define the effect of color values and the allocated reference line parameters. In this set of tests the observations have been focused on analyzing the vehicle motion behavior when it is marched around the reference route line; the motion behavior is qualified in terms of the overall deflection distance and the spend trip time. In this set of tests, the reference route mark was set by drawing a line using white paints on a concrete floor. A set of image frames are captured along the marching path, In Figure (5) a sample frame from the car view presented and the reference route plan and image after the processing to illustrate the behavior of the system and to observe the tracking results. The camera based guiding system uses 4-5 frames per second as the highest video capturing rate.

The test results of camera based guiding system indicated that the system performance are the best when the camera snapshot rate is set 4 frames per second. The average deflection of the car relative to the reference line was 14.79 cm. Figure (6) presents the results of the test.

a.Sample frame from the car view

b.Reference route plan



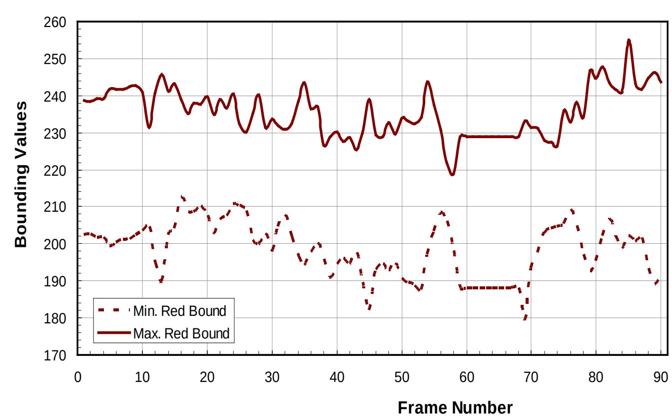
c. Image after processing

Fig. 5 Image taken from the car view, route plan and processed frame

Table (1) presents the results of the conducted test on the guidance system for each frame

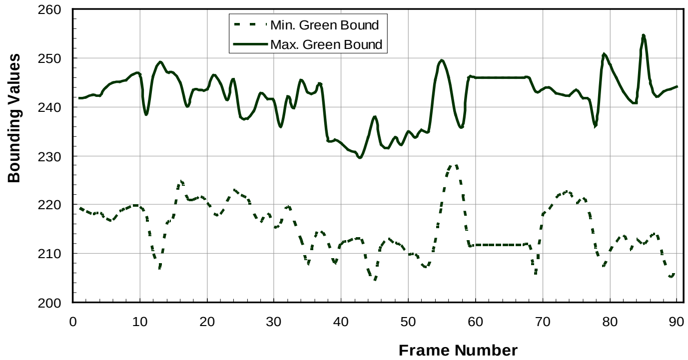
Table 1 Motion command module results

Time/sec.	Angle Deflection	Position Deflection	Position proportional	Turn Proportional	Command
1	0.7	26.9	-13.4	-0.1	-13.6
2	1.0	-0.7	0.4	-0.1	0.2
3	1.8	11.1	-5.5	-0.1	-5.7
4	-13.5	33.6	-16.8	1.8	-15.0
5	-26.3	-109.0	54.5	14.3	68.8
6	-11.2	-65.7	32.8	1.2	34.0
7	-23.2	6.6	-3.3	10.0	6.7
8	-15.3	-39.0	19.5	2.6	22.1
9	-17.9	-7.0	3.5	4.3	7.8
10	-8.8	-234.9	117.5	0.7	118.1
11	-5.3	-100.2	50.1	0.3	50.4
12	4.9	83.2	-41.6	-0.3	-41.9
13	-3.5	-44.5	22.2	0.2	22.4
14	0.2	42.5	-21.2	-0.1	-21.4
15	28.1	-42.0	21.0	-16.6	4.4
16	31.7	139.3	-69.6	-20.3	-89.9
17	-0.1	89.5	-44.8	0.1	-44.7
18	-15.0	-100.2	50.1	2.5	52.6
19	-15.5	14.0	-7.0	2.7	-4.3
20	-16.0	-11.1	5.6	3.1	8.6
21	-31.4	-41.8	20.9	20.1	40.9
22	-39.7	-137.3	68.6	24.1	92.7
23	-26.3	-242.3	121.1	14.3	135.4
24	-19.6	-233.5	116.8	5.8	122.6
25	-32.7	-218.1	109.0	21.1	130.1
26	-25.8	-232.6	116.3	13.6	129.9
27	-22.9	-222.9	111.4	9.7	121.1
28	-13.4	-233.7	116.9	1.8	118.7
29	-24.0	-196.5	98.2	11.1	109.3
30	21.0	-89.0	44.5	-7.3	37.2
31	5.9	76.2	-38.1	-0.4	-38.5
32	-4.7	-84.3	42.2	0.3	42.4
33	5.8	60.0	-30.0	-0.4	-30.4

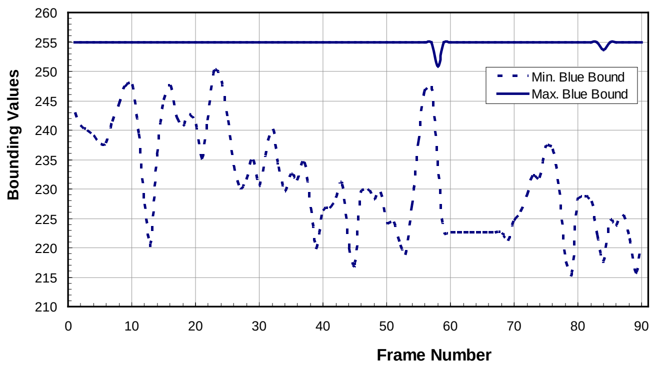


a. Red

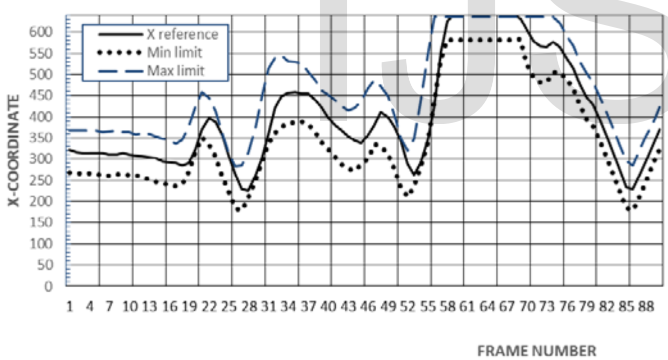
Fig. 6 The color bounding limits values and the scanning window position when the rate of snapshots is set 4 frames/ sec (continue)



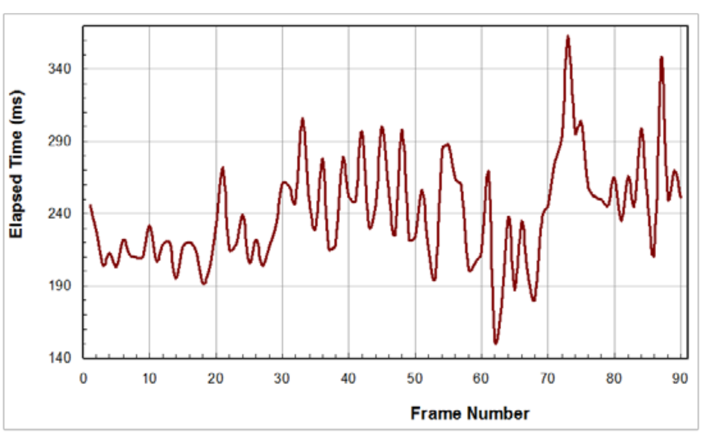
b. Green



c. Blue



d. Scanning Window (in pixels)



e. Elapsed Time (in ms)

Continue of Fig. 6

References

[1] J. W. Courtney, M. J. Magee and J. K. Aggarwalj, "Robot Guidance Using Computer Vision", Pattern Recognition, Vol. 17. No. 6, pp. 585-592, 1984.

[2] N. Birkbeck and M. Jagersand , "Visual tracking using active appearance models", *Proceedings of the First Canadian Conference on Computer and Robot Vision*, pp. 2 – 9, 2004

[3] M., Mikhalsky and J. Sitte, "Real-time Motion Tracker for a Robotic Vision System", *Proceedings of the First Canadian Conference on Computer and Robot Vision*, pp. 30–34, 2004

[4] S. Leonard and M. Jagersand, "Approximating the Visuomotor Function for Visual Servoing", *Proceedings of the First Canadian Conference on Computer and Robot Vision*, pp. 112 – 119, 2004

[5] J. Denzler and D. W. R. Paulus, "Active Motion detection and object tracking", *Proceedings of the IEEE International Conference on Image Processing (ICIP-94)*, Vol. 3, pp. 635 – 639, 1994

[6] J. N. Carter, P. Lappas and R. I. Damper,"Acoustics Evidence-based object tracking via global energy maximization", *Proceedings of the 2003 IEEE International Conference on Speech, and Signal (ICASSP '03)*, Vol.3, pp. 501-504, 2003

[7] Xiao-Jing Shen and Jun-Min Pan, "A simple adaptive control for visual servoing", *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*, Vol. 2, pp. 976 – 979, 2003

[8] Gordon M., "The Robot Builder's Bonanza", *McGraw-Hill*, DOI: 10.1036/0071386718, Second Edition, 2001.