Vol. 00, No. 00, October 2008, 1–20

GUIDE

Camera Calibration Using a Genetic Algorithm

Nirmal Baran Hui[a] and Dilip Kumar Pratihar[b*]

[a]*Department of Mechanical Engg. and Mining Machinery Engg., Indian School of Mines University, Dhanbad, India*; [b]*Department of Mechanical Engineering, Indian Institute of Technology, Kharagpur-721 302, India*

(*v3.3 released January 2008*)

An autonomous robot will have to detect the moving obstacles on-line, before it can plan its collision-free path, while navigating in a dynamic environment. The robot collects information of the environment with the help of a camera and determines the inputs for its motion planner through image analysis. The present paper deals with the issues related to camera calibration and on-line image processing. The problem of camera calibration is treated as an optimization problem and solved using a Genetic Algorithm (GA), so as to achieve minimum Distorted Image Plane Error (DIPE). The calibrated vision system is then utilized for the detection and identification of the objects by analyzing the images collected at regular intervals. For image processing, five different operations, such as median filtering, thresholding, perimeter estimation, labeling and size filtering have been carried out. To show the effectiveness of the developed camera-based vision system, inputs of the motion planner of a navigating robot are calculated for two different cases. It is observed that on-line detection of the shapes and postures of the obstacles is possible by using the developed vision system.

Keywords: On-line obstacle detection, Camera calibration, Optimization, Genetic algorithm, Image processing.

Index to information contained in this guide

*Corresponding author. Phone No. +91-3222-282992, Fax no. +91-3222-282278, Email: dkpra@mech.iitkgp.ernet.in

ISSN: 1478-6451 print/ISSN 1478-646X online

DOI: 10.1080/1478645YYxxxxxxx
http://www.informaworld.com

List of symbols

a	Area of an object image
A^f	Value of size filter
(c_x, c_y)	Row and column numbers of the center of computer frame memory
$\{C\}$	Camera-centered coordinate system
(d_x, d_y)	Center to center distances of two successive pixels
f	Fitness of a GA-string
k_1, k_2, k_3	Lens distortion coefficients
L	Length of the overhang part of the camera tripod
(N_{fx}, N_{fy})	Number of pixels in a line as scanned by the computer
p	Perimeter of an object image
(r_1, r_2, r_3)	Components of the position vector
(s_x, s_y)	Computer uncertainty factors
(t_x, t_y, t_z)	Components of translation vector
$\{W\}$	World coordinate system
(x, y, z)	Coordinates of a point P in the camera-centered coordinate system
(x_d, y_d)	Distorted image plane coordinates
(x_f, y_f)	Row and column numbers of the image pixel in computer frame memory
(x_u, y_u)	Undistorted image plane coordinates
(X, Y, Z)	Coordinates of a point P in the Global coordinate system
Δt	Time interval in seconds
κ	Swing angle
λ	Focal length of the lens
ω	Pan angle
θ	Angle of rotation about X-axis or tilt angle

List of Abbreviations

CCD	Charged Couple Device
CCS	Camera Coordinate System
DIPE	Distorted Image Plane Error
DLT	Direct Linear Transformation
GA	Genetic Algorithm
WCS	World Coordinate System

1. Introduction

Design and development of an autonomous mobile robot is the prime aim of the current field of robotic research. A lot of work is going on throughout the world, to make the robot more and more intelligent and autonomous. An autonomous robot collects information of the environment through sensors and/or cameras and acts upon that environment through actuators, after taking some decision. It involves a number of tasks, such as sensor/camera calibration, obstacles detection, scene modeling, path planning, obstacle avoidance, and others. The performance of a motion planner depends on the collected information of the environment also, i.e., how clearly and distinctly the robot can detect the obstacles. Choice of a sensor plays an important role in this regard. Sonar is found to be the most widely used sensor for obstacle detection, because it is cheap and simple to operate. Borenstein and Koren (1989) used a sonar ring around their robot for obstacle detection. However, a key drawback of sonar lies in the fact that one sensor is required for one distance measurement, that is, in order to obtain an adequate picture of the environment around the vehicle, a number of sensors must be used. Moreover, to achieve the accuracy in detection, the sonars are to be placed perpendicular to the target. Scanning lasers are also available for the said purpose, but have their inherent application limitations and are very costly. The cameras are also useful, as they mimic the human eye. Stereo vision had been extensively used for obstacle detection in a number of applications. Apostopolous *et al.* (1999) used cameras to aid in navigation and obstacle detection for a robot searching for meteorites. Shah and Aggarwal (1997) modeled the scene with the help of a stereo fish-eye lens system. However, the cameras are passive sensors, which require ambient light to illuminate its field of view. Moreover, the accuracy of detection depends on some of the camera parameters. Therefore, a camera will perform well, only when it is properly calibrated. Interested readers may refer to the survey carried out by DeSouza and Kak (2002), on vision-based mobile robot navigation systems developed by the researchers over the last two decades. In the present work, our aim is to develop an efficient and fast obstacle detection technique based on the data collected through a CCD camera, which is placed perpendicular to the field of view. Obstacles considered in the present study are of different height and they are allowed to move in a smooth planar terrain. Thus, the camera requires to detect only the planar objects. However, to improve the accuracy of estimation, calibration of some of the effective camera parameters and development of a noise sensitive image processing technique are required.

Camera calibration is the process of determining the internal camera geometric and optical characteristics, known as intrinsic parameters, and/or the 3-D position and orientation of the Camera Coordinate System (CCS), with respect to a certain World Coordinate System (WCS), which are called extrinsic parameters. The proper choice of these parameters plays an important role in obtaining an accurate relationship between the position and orientation of an object in the WCS and those in the image coordinate system as seen through the computer screen. The existing techniques for camera calibration can be classified into the following categories.

Category I - Direct Nonlinear Optimization: In this category, relations between the 3D coordinates of control points and their image plane coordinates are established. Thereafter, an iterative algorithm is used to obtain the best set of camera parameters, corresponding to which the error will be minimum. Faig's technique (Faig (1975)) is a good example of this category. He used a very elaborate model for imaging and considered at least 17 unknowns for each photo. Although the accuracy

of his technique was excellent, it is very computer-intensive and requires a good initial guess to start the search.

Category II - Closed-Form Solution: One example of this category is the Direct Linear Transformation (DLT) model developed by Abdel-Aziz and Karara (1971). The main advantage of their approach lies in the fact that only linear equations were needed to be solved. However, they did not consider any camera refinement parameters, like lens distortion into the camera model. Therefore, the derived model could not be an exact one and might provide some bad solutions. Hall *et al.* (1982) used a straightforward linear least square method, to solve for the elements of perspective transformation matrix for carrying out 3D curved surface measurement. The computer coordinates were tabulated, but no proof was given, and thus the accuracy remained unknown.

Category III - Two-Plane Method: It involves a direct solution for most of the calibration parameters and some iterative solutions for the other parameters. The existing techniques include those presented by Tsai (1987), Lenz and Tsai (1989). A radial alignment constraint was used to derive a closed-form solution for the external parameters and the effective focal length of the camera. Then, an iterative scheme was used to estimate three parameters: depth component in the translation vector, effective focal length and a radial distortion coefficient. The closed-form solution was obtained for a relatively small number of parameters and the effect of lens distortion was included in the camera model. However, their method could handle radial distortion only and was not extended to other types of distortion. Moreover, they had not fully utilized the information of the calibration parameters, thus the solutions might not be truly optimal. Weng *et al.* (1992) followed the same procedure proposed by Tsai, and they incorporated three types of distortion parameters, such as radial, tangential and prism distortion into the camera model. But, their method failed to provide optimal solutions, which could be due to the fact that they considered a steepest descent method for optimization, which might have local minima problem. Moreover, as the limits of the parameters were not known a priori, it was difficult to make an initial guess. Quite a few researchers studied the method of calibration techniques developed by Tsai (1987) and Weng *et al.* (1992). In this regard, work of Tapper *et al.* (2002) is important to mention. Rather than modeling the resolution parameters of the camera, they used a zoom lens for the said purpose.

Zhang (2000) proposed a new technique to calibrate a camera, in which calibration parameters, such as coordinates of the principal point (c_x, c_y), scale factors (s_x, s_y) and tilt angle (θ) were initially estimated by using the closed-form solution technique and all those parameters including lens distortion parameters (k_1, k_2) were then refined by using a non-linear search technique based on the maximum-likelihood criterion. Although they considered the planar pattern for calibration of the camera, accuracy was found to be comparable with that of the work developed by Tsai (1987). However, they used a steepest descent algorithm for the nonlinear search, which might suffer from the local minima problem. Moreover, initial estimation of all the calibration parameters was time-consuming and numerical procedure might provide with a very crude initial estimation. Recently, understanding the problem of two step camera calibration, quite a few researchers tried to solve this problem in a single step using soft computing-based approaches. Some of them are mentioned here. Lee and Lee (1998) developed a neural network-based camera calibration method. But, they neglected the effect of object height on camera calibration and as a result of which, their method failed to yield the proper relationship among the camera coordinates and world coordinates. Ji and Zhang (2001) tried to solve the problem of camera calibration in a single step by

using a Genetic Algorithm (GA). In their approach, multiple control points of a single object were considered during the calibration of the camera parameters, such as the principal point (c_x, c_y), scale factors due to spatial quantization (s_x, s_y), focal length (λ) and the exterior parameters (components of translation vector – (t_x, t_y, t_z), pan angle ω, tilt angle θ, swing angle κ). Thus, with a small number of control points, calibration accuracy was found to be satisfactory. However, they neglected the lens distortion in the camera model. It is also important to mention that the performance of GA depends on its parameters and no study was carried out by Ji and Zhang (2001) to determine the best set of GA parameters. Thus, there is still a need to find a generalized camera calibration technique, which could be used to calibrate the camera in a dynamic environment. Interested readers may refer to the review paper of Salvi *et al.* (2002), for a more detailed discussion. Klancar *et al.* (2004) proposed a technique to correct distortion effect of an wide-angle camera, which was used for mobile robot tracking. In their approach, if (x,y) and (X,Y) be the coordinates of a pixel in the distorted image and rectified image, respectively, the relations between (x,y) and (X,Y) were obtained as follows.

$$X = Rcos(\phi), Y = Rsin(\phi), \tag{1}$$

where $R = \lambda sinh(\frac{r}{f})$, $r = \sqrt{x^2 + y^2}$, $\phi = arctan2(y/x)$ and λ is the focal length. Thus, the only unknown parameter is the focal length (λ), which was optimized by following the steepest descent method and other all calibration parameters of the camera were neglected.

In the present work, the problem of camera calibration is tackled using a GA. The present experimental set-up is similar to the set-up used by Klancar *et al.* (2004), whereas both extrinsic (component of translation vector along z-axis t_z) as well as intrinsic parameters like focal length (λ), distortion factors (k_1, k_2, k_3) and scale factors (s_x, s_y) of the camera are optimized using a GA to get better estimation of the postures of the robot and the moving objects.

On-line navigation haunts for a fast and noise sensitive image processing technique. Tsukiyama and Huang (1987) reported a scene interpretation approach for navigation of autonomous vehicles. They extracted the edge lines from images based on brightness variation of pixels and these edge lines were classified into three categories: oblique, horizontal and vertical. However, detection of the odd-shaped objects by following their method was found to be the worst. Lee and Shen (1994) proposed a model-based approach to determine location of an automated guided vehicle during navigation. They directly used templates for matching, therefore no special image processing technique was established. However, if the shape of the vehicle changes, their approach will demand for another template. Thus, template formation was itself found to be cumbersome. Cokal and Erden (1997) developed a step-wise image processing method. Initially, they sliced the randomly-captured gray image into a binary image by using a threshold value. Thereafter, Laplacian filters were used for edge detection. However, their method failed to yield a suitable value of threshold, on which the accuracy of the said technique depends and Laplacian filters were found to be less noise sensitive. Moreover, none of the above approaches were suitable to detect the unknown number of obstacles present in the environment. Thus, in the present work, an attempt is made to develop an image processing technique that provides good estimation of the postures of both the robot as well as moving objects in a dynamic environment. The number of objects present in the environment is not pre-assigned and it is identified automatically through labeling of the objects.

In this connection, the authors developed a GA-based camera calibration

method Hui and Pratihar (2005), in which both camera parameters as well as the threshold value were tuned using a GA to minimize the Distorted Image Plane Error (DIPE). However, in that approach, only one directional computer uncertainty (i.e., s_x) factor was considered in the camera model and the GA was run for a particular set of its parameters, which might not be optimal in any sense. In the present work, both the directional uncertainty factors (i.e., s_x, s_y) are included in the camera model to achieve more accuracy and a systematic study has been carried out to obtain an optimal set of GA-parameters. Moreover, Sobel's edge detection technique has been replaced by a perimeter estimator, in the present paper, to obtain the perimeter of the detected objects more accurately.

The remainder of the paper is organized as follows: In Section 2, the camera model used in the experimental set-up has been explained and the image processing technique is discussed in Section 3. The proposed method of tuning the camera parameters, threshold value and value of size filter is explained in Section 4. Results are stated and discussed in Section 5. Some concluding remarks are made in Section 6 and the scope for future work is mentioned in Section 7.

2. The Camera Model

Figure 1 shows the photograph of the experimental set-up, where a color CCD camera, placed on a tripod is used for sensing the environment. The camera sends continuous electrical signals to the computer through a BNC video cable. Thereafter, a frame grabber board, namely vision board (placed in a PCI slot of the computer hardware), is used to convert the continuous signals to 2-D digital images. Figure 2 shows the schematic diagram indicating the camera coordinate system and world coordinate system. Let us suppose that a visible point P is expressed by its coordinate (X,Y,Z) and (x,y,z) in the world coordinate system (WCS) and camera-centered coordinate system (CCS), respectively. The CCS is attached to the center of the image plane and WCS is located at a corner point of the terrain, as shown in the above figure. The camera is mounted on a overhang part (i.e., at point A) of a camera tripod, which allows the camera to tilt through an angle of θ about X-axis. The off-set of the point A from the origin of the WCS is denoted by a vector $\vec{w_0}$ with components $(X_0 - L, Y_0, Z_0)$, and the off-set of the center of the image plane with respect to point A is denoted by a vector $\vec{r}$ having components (r_1,r_2,r_3), when L indicates the length of overhang arm of the camera tripod. It is important to mention that r_2 has been assumed to be equal to 0, in Figure 2.

Our aim is to obtain a camera model based on the geometrical arrangement of Figure 2. It is to be noted that the camera coordinate system can be expressed in terms of world coordinate system and vice-versa, through a set of frame transformations. A perspective transformation will then be applied to obtain the image plane coordinates of any given world point. Thereafter, we consider the radial distortion parameters of the lens to get a more accurate model of the same.

Now, Camera Coordinate System (CCS) can be expressed with respect to the World Coordinate System (WCS) as follows (refer to Figure 2):

(1) Translation to reach origin of the CCS from the origin of the WCS,
(2) Rotation about X-axis by an angle θ.

The origin of CCS may be expressed with respect to the WCS using the following

Figure 1. The photograph of the experimental set-up.

transformation matrix.

$$ {}^{W}Trans_{C} = \begin{bmatrix} 1\,0\,0 & X_0 - L + r_1 \\ 0\,1\,0 & Y_0 + r_2 \\ 0\,0\,1 & Z_0 + r_3 \\ 0\,0\,0 & 1 \end{bmatrix} \tag{2} $$

We then give rotation about X-axis by an angle θ, which is given below.

$$ Rot(x,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sin\theta & 0 \\ 0 & sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

It is important to mention that counter clockwise rotation has been considered as the positive one.

Thus, a point P having coordinate (X,Y,Z) in the WCS, can be expressed in the CCS (i. e., x,y,z) using the following relationship:

$$ \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} X + X_0 - L + r_1 \\ (Y + Y_0 + r_2)cos\theta - (Z + Z_0 + r_3)sin\theta \\ (Y + Y_0 + r_2)sin\theta + (Z + Z_0 + r_3)cos\theta \end{bmatrix} \tag{4} $$

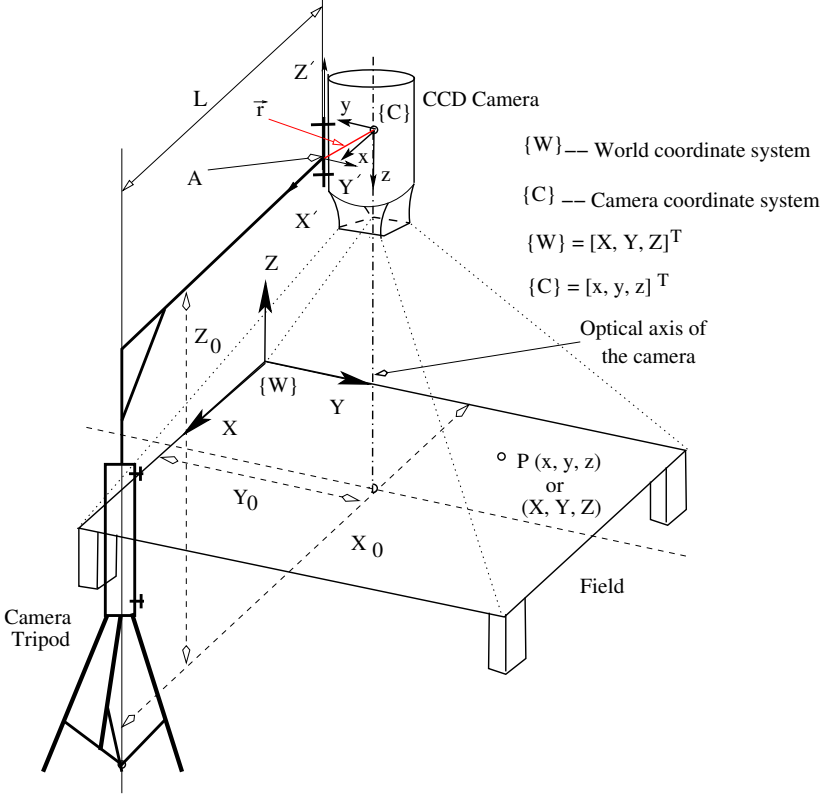

Figure 2. A schematic diagram showing the camera coordinate system and world coordinate system

2.1. *Perspective Transformation*

A perspective transformation projects 3D points onto a plane. A model of the image formation is shown in Figure 3. It is assumed that x-y plane of the camera

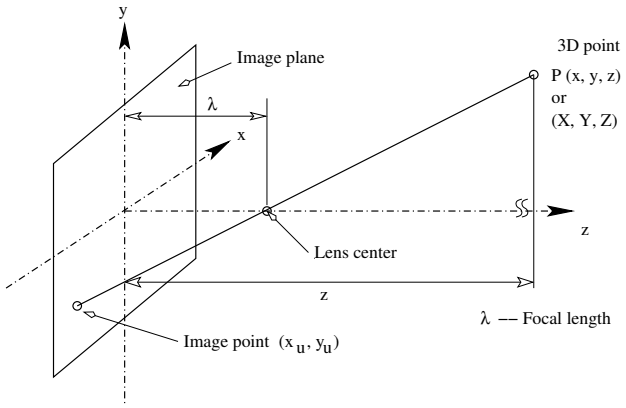

Figure 3. A schematic figure showing the perspective transformation

coordinate system is coincident with the x-y image plane and optical axis of the camera is considered to be along the z-axis of the CCS. Thus, center of the image plane is at origin, and the center of the lens is at coordinate (0,0,λ), where λ indicates the focal length of the lens. Let (x,y,z) be the camera coordinates of any point (P) in a 3-D scene, as shown in Figure 3, and (x_u, y_u) be the projection of the point P(x,y,z) on to the image plane for a perfect pinhole camera model. However, (x_d, y_d) is the actual image coordinate, which may differ from (x_u, y_u) due to lens distortion. Now, transformation from 3D camera coordinate (x,y,z) to

ideal (undistorted) image coordinate (x_u, y_u) using perspective projection (with a pinhole camera geometry) may be obtained using the principles of similar triangles (refer to Figure 3) like the following.

$$\frac{x_u}{\lambda} = -\frac{x}{z-\lambda} = \frac{x}{\lambda - z}, \tag{5}$$

$$\frac{y_u}{\lambda} = -\frac{y}{z-\lambda} = \frac{y}{\lambda - z}, \tag{6}$$

where the negative signs with x and y indicate that image points are actually inverted (refer to Figure 3) and it is assumed that all points of interest lie in front of the lens, i.e., $z \geq \lambda$.

2.2. *Lens Distortion*

It is important to note that the actual cameras may not be perfect and sustain a variety of aberrations. For geometrical measurements, the main concern is lens distortion, which is related to the position of image points in the image plane but not directly to the image quality. It occurs due to several types of imperfections in the design and assembly of lenses composing the camera. There are mainly two types of distortion – radial and tangential (refer to Figure 4). However, most of the researchers Tsai (1987), Zhang (2000) have considered only the radial distortion in their model, as it is found to be the most effective for industrial machine vision applications Weng *et al.* (1992). Radial distortion causes an inward or outward

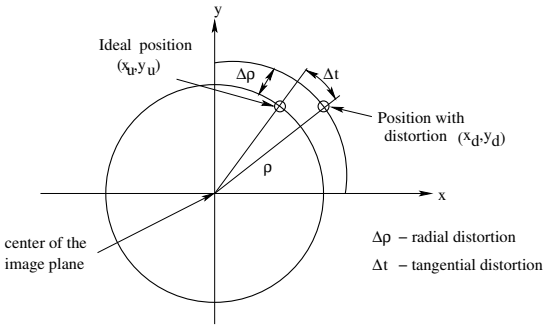

Figure 4. Radial and tangential distortions.

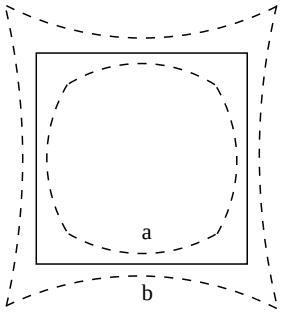

Figure 5. Effect of radial distortion, solid lines: no distortion, dashed lines: with radial distortion (a - negative, b - positive).

displacement of a given image point from its ideal location (refer to Figure 5). This type of distortion is mainly caused by flawed radial curvature of the lens elements. A negative radial displacement of the image points is referred to as barrel distortion. It causes outer points to crowd increasingly together and the scale to decrease. A positive radial displacement is referred to as pincushion distortion. It causes outer points to spread and the scale to increase. This type of distortion is strictly symmetric about the optical axis of the camera. Radial distortion is governed by an expression of the following forms.

$$x_d + \Delta\rho_x = x_u, \tag{7}$$

$$y_d + \Delta\rho_y = y_u, \tag{8}$$

where (x_d, y_d) is the distorted image coordinate on the image plane and

$$\Delta\rho_x = x_d\ (k_1\rho^2 + k_2\rho^4 + k_3\rho^6 + \ldots),$$
$$\Delta\rho_y = y_d\ (k_1\rho^2 + k_2\rho^4 + k_3\rho^6 + \ldots),$$

where $k_1, k_2, k_3 \ldots$ are constants of the infinite series, known as distortion coefficients and $\rho = \sqrt{x_d^2 + y_d^2}$. It is important to note that we have considered up to the third order term of the expansion series, to simplify the model.

2.3. *Computer Coordinate Frame*

The final step of the camera model is the conversion of real image coordinate (x_d, y_d), to computer image coordinate (x_f, y_f). This can be accomplished using the expression given below.

$$x_f = s_x \frac{N_{fx}}{d_x} x_d + c_x, \tag{9}$$

$$y_f = s_y \frac{N_{fy}}{d_y} y_d + c_y, \tag{10}$$

where (x_f, y_f) is the row and column numbers, respectively, of the image pixel in computer frame memory and (c_x, c_y) indicates the row and column numbers of the center of computer frame memory, respectively. Moreover, uncertainty image scale factors, number of pixels in a line as sampled by the computer and the center to center distances between two adjacent pixels in the x and y directions are denoted by (s_x, s_y), (N_{fx}, N_{fy}) and (d_x, d_y), respectively. It is important to mention that uncertainty image scale factors have been introduced to consider the hardware timing mismatch between image acquisition and camera scanning.

Therefore, the final computer image coordinate may be expressed as follows:

$$x_f = \frac{s_x N_{fx} \lambda (X + X_0 - L + r_1)}{d_x (1 + k_1\rho^2 + k_2\rho^4)(\lambda + (Y + Y_0 + r_2) sin\theta + (Z + Z_0 + r_3) cos\theta)} + c_x \tag{11}$$

$$y_f = \frac{s_y N_{fy} \lambda ((Y + Y_0 + r_2) cos\theta - (Z + Z_0 + r_3) sin\theta)}{d_y (1 + k_1\rho^2 + k_2\rho^4)(\lambda + (Y + Y_0 + r_2) sin\theta + (Z + Z_0 + r_3) cos\theta)} + c_y \tag{12}$$

3. Image Processing

The image captured by a CCD camera is stored in the computer memory (in bitmap format) with the help of a vision board/frame grabber placed inside the computer. The following operations are performed on the raw data collected above.

- *Histogramming and Smoothing:* The raw image may contain some noise due to unequal luminance and/or reflection. Thus, a median filter is applied to smoothen the raw image. The main advantage of using the median filter lies in the fact that it can preserve the edge sharpness and produce regions of constant/near constant intensity. For this purpose, a 3×3 window is used. The median of all the pixels lying inside the window is determined. The pixels having intensity below the median value are identified and their intensity values are replaced by the median value.
- *Binarization of the Image:* The smoothed image is binarized with the help of a threshold value. The pixels having intensity below the threshold value are converted into black and the remaining pixels are made white. In the present study, an optimal value of threshold is obtained using a GA.

- *Perimeter Estimation:* Measurement of perimeters, areas and other shape related parameters of the planar digitized objects is an important task in computer vision systems. A large number of techniques are available in the literature. However, counting the number of pixel edges is not the best method, due to the presence of corners and diagonal edges along the perimeter. To eliminate this, a 3×3 perimeter estimator developed by Koplowitz and Bruckstein (1989) has been used, in the present study.
- *Labeling of the Objects:* All spatially close pixels of a binary image having value 1 are grouped into connected components that distinctly represent the different objects. This has been implemented using a component labeling algorithm proposed by Kim *et al.* (2004). It finds all connected components in the image and assigns a unique label, usually an integer to all pixels lying in the same component. Thereafter, area and perimeter of each object have been computed and compactness of each object image is calculated by following the formula given below.

$$Compactness = \frac{4\pi a}{p^2}, \tag{13}$$

where a refers to the object image area and p is obtained using the perimeter estimator as explained earlier. It is obvious that determination of *compactness* and *perimeter* helps in finding and recognizing the known shaped objects very easily. It is also important to note that a *compactness* has been defined in such a manner, so as to have unity value for circular objects. For a more detailed information of the labeling algorithm, interested readers may refer to the book on soccer robotics, written by Kim *et al.* (2004).

- *Size Filtering:* Noise is inherent in computer vision. Some extraneous components could appear in an image, due to noise arising from the unstable resolution of the camera and uneven illumination in environmental lighting. However, these components are usually found to be small in size and ragged contours. Thus, size filtering might remove such noise after the component labeling is over. This involves changing all the pixel values of a component binary image from 1 to 0, if its area is found to be less than an appropriately selected size filter (A^f). In the present study, an optimal value of A^f has been obtained using a GA.

4. Tuning of Camera Calibration and Image Processing Parameters Using a Genetic Algorithm

In Section 2, we have obtained explicit equations for the computer image coordinates (x_f, y_f) of a world point (X,Y,Z). However, it requires knowledge of the following parameters:

- Effective focal length λ,
- Camera off-sets X_0, Y_0, Z_0,
- Angle of rotation θ,
- Radial distortion coefficients k_1, k_2, k_3,
- Uncertainty image scale factor s_x, s_y.

Moreover, binarization of the images depends on the selection of a threshold value and for removing the noises in the form of small, ragged components, a proper choice of size filter (A^f) is necessary. In the present study, X_0, Y_0 and θ are kept constant and all other parameters are varied in a range to minimize the DIPE,

which may be obtained as follows:

$$DIPE = \sqrt{(x'_f - x_f)^2 + (y'_f - y_f)^2}, \tag{14}$$

where (x'_f, y'_f) is the actual position of the target point on image plane and (x_f, y_f) is the estimated position of the target point (obtained through the calibration parameters). Our aim is to obtain the best set of camera parameters for which DIPE will be minimum. Thus, the present problem can be treated as an optimization problem, which can be stated as follows:

$$Minimize \;\; DIPE, \tag{15}$$

subject to

$Threshold_{min} \leq Threshold \leq Threshold_{max}$,
$A^f_{min} \leq A^f \leq A^f_{max}$,
$\lambda_{min} \leq \lambda \leq \lambda_{max}$,
$Z_{0min} \leq Z_0 \leq Z_{0max}$,
$k_{1min} \leq k_1 \leq k_{1max}$,
$k_{2min} \leq k_2 \leq k_{2max}$,
$k_{3min} \leq k_3 \leq k_{3max}$,
$s_{xmin} \leq s_x \leq s_{xmax}$,
$s_{y_{min}} \leq s_y \leq s_{y_{max}}$,
and it should be able to detect exact number of objects.

A binary-coded GA Goldberg (1989) having string length equals to ninety bits (ten for each variable) is used for this purpose. The fitness of a GA-string is made equal to the DIPE, that is,

$$f = DIPE. \tag{16}$$

After the fitness value is assigned to each GA-string, they are modified using three operators, namely reproduction, crossover and mutation. In the present study, tournament selection-based reproduction scheme is adopted and a two-point crossover of probability p_c is used. Finally, a bit-wise mutation of probability p_m is utilized after the crossover. The whole process continues until a termination criterion is reached. In this work, a prespecified number of generation is considered to be the termination criterion.

5. Results and Discussion

The purpose of the present study is to detect the moving obstacles on-line using a camera and image processing algorithm, during navigation of a car-like robot. It is to be noted that the over-head camera is to be calibrated first, before it is used for collecting the image, on-line. The present section is subdivided into two, in the first subsection, results related to the camera calibration are depicted and in the next subsection, results related to positions and sizes of the robot and obstacles are stated, from which two inputs of the robot motion planner, namely *distance* and *angle* are determined. Distance is the Euclidean distance between the robot and the obstacles and angle is nothing but the angle between the line joining the robot and its goal and the line joining the robot and the obstacles.

5.1. *Calibration of the vision system*

The following parameters are used to predict the image plane coordinates:

$d_x = d_y$	= 0.264 mm per pixel
N_{fx}	= 192
N_{fy}	= 256
c_x	= 96
c_y	= 128
$r_1 = r_3$	= 2 cm
r_2	= 0 cm
L	= 113 cm
X_0	= 170 cm
Y_0	= 70 cm
θ	= 180^o

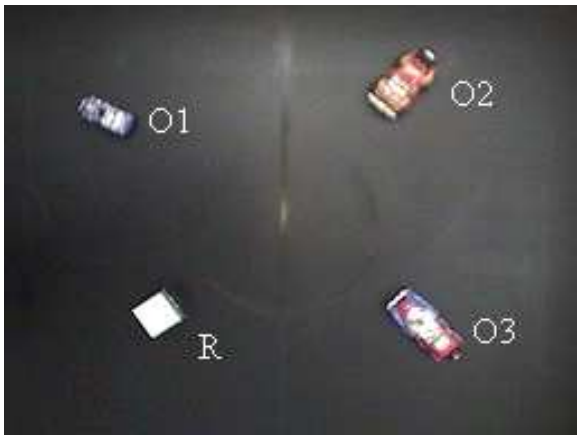

Figure 6. The photograph showing the four objects

A photograph (refer to Figure 6) of the objects is taken with the help of an experimental set-up shown in Figure 1. There are four different objects in the image and the centers of areas of the top surface of the objects represented in the WCS are considered as the four control points during calibration of the vision system and these are measured to be R (31 cm, 21 cm, 5 cm), O_1 (95 cm, 24 cm, 7 cm), O_2 (39 cm, 71 cm, 8 cm) and O_3 (102 cm, 77 cm, 6.5 cm), respectively. During optimization, the variables like *threshold*, size filter (A^f), Z_0, λ, k_1, k_2, k_3, s_x and s_y are varied in the ranges of (100, 230), (20 cm^2, 50 cm^2), (125 cm, 150 cm), (3.5 mm, 8 mm), $(-1\times10^{-6}, 1\times10^{-6})$, $(-1\times10^{-6}, 1\times10^{-6})$, $(-1\times10^{-6}, 1\times10^{-6})$, (0.1, 1.0) and (0.1, 1.0), respectively. As the performance of a GA depends on its parameter setting, experiments are carried out with different sets of parameters to find the most suitable set of parameters. Results of parametric study are shown in Figure 7. The best results are obtained with the following GA-parameters: Crossover probability $p_c = 0.90$, mutation probability $p_m = 0.005$, population size $Y = 100$, maximum number of generation $Maxgen = 90$. The shape of the objects obtained after image processing is shown in Figure 8. The minimum value of DIPE is found to be equal to 0.735214 mm with the following calibration parameters:

Threshold	= 168
A^f	= 38
Z_0	= 149.56 cm
λ	= 3.6 mm
k_1	= -0.000000957

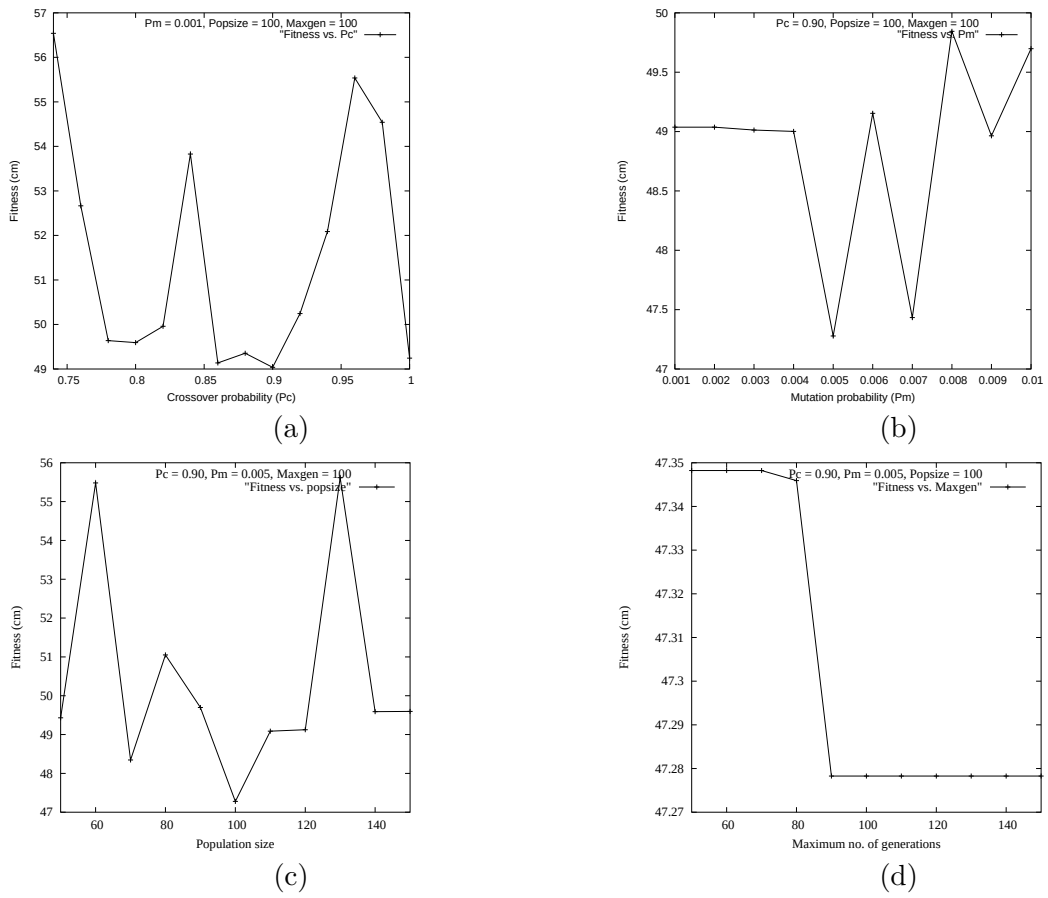

Figure 7. Results of the parametric study to obtain the optimal GA-parameters: (a) Fitness vs. Crossover prob. (P_c), (b) Fitness vs. Mutation prob. (P_m), (c) Fitness vs. Population size, (d) Fitness vs. Maximum no. of generations.

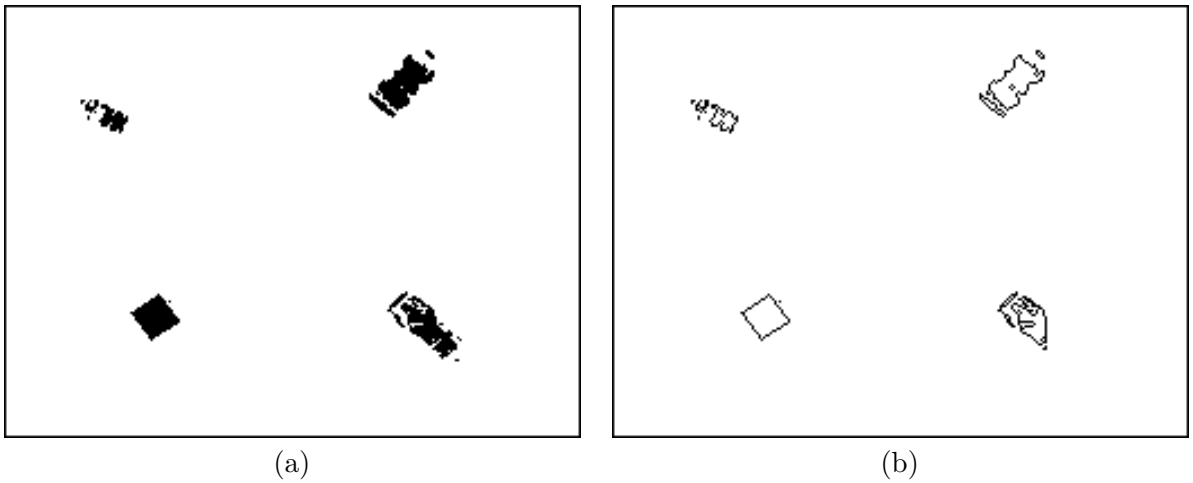

Figure 8. The images showing the shape of the objects obtained after considering: (a) area, (b) perimeter of the objects.

k_2 = -0.000000204
k_3 = -0.000000994
s_x = 0.134
s_y = 0.100

The main advantage of this calibration method lies in the fact that only with the four control points, all the calibration parameters of the vision system have been possible to obtain, efficiently.

5.2. *Determination of the inputs of robot motion planner*

Once the camera and/or the developed vision system is properly calibrated, it is used to compute two inputs, namely **distance** and **angle** of the motion planner, on-line, through a proper processing of the collected images. To test the performance of the developed system, two different cases are considered. In the first case, the robot is allowed to move in the presence of two moving obstacles, whereas four moving obstacles have been considered in the next case.

5.2.1. Case 1: Two moving obstacles

Figure 9 shows the images of the environment at four different instants of time, say $t_1,\ t_1+\Delta t,\ t_1+2\Delta t$ and $t_1+3\Delta t$. The time interval Δt is considered to be

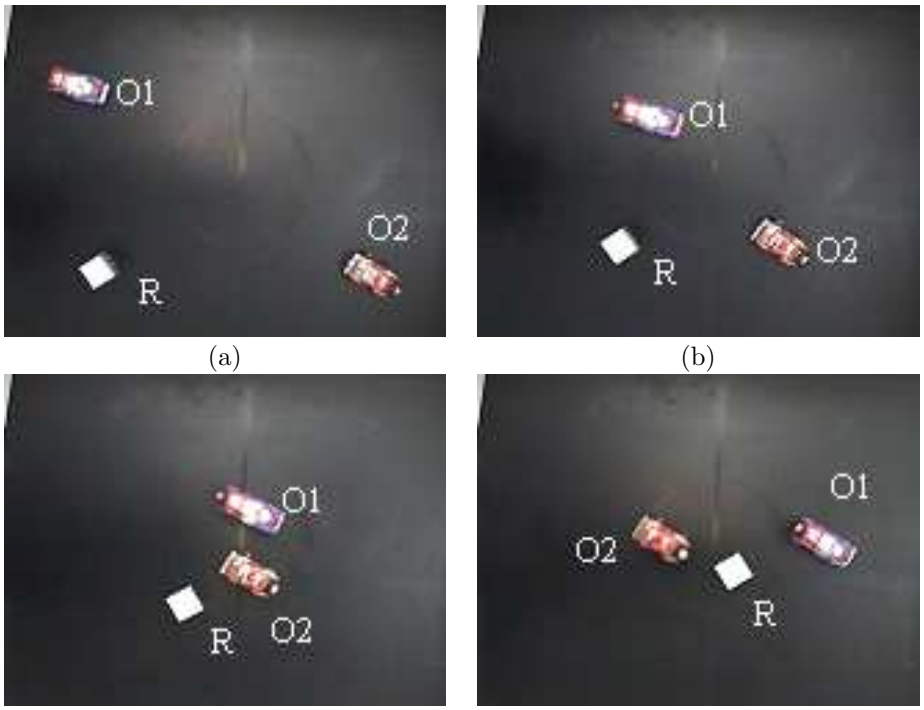

Figure 9. Images showing the movement of both the robot as well as the obstacles at four instants of time: (a) $t=t_1$, (b) $t=t_1+\Delta t$, (c) $t=t_1+2\Delta t$ and (d) $t=t_1+3\Delta t$ – Case 1.

equal to one second. Inputs of the motion planner are calculated based on the fixed goal point (65 cm, 105 cm) and goal reference line making an angle of 116° with positive X-axis of the global coordinate frame shown in Figure 2. Table 1 shows the positions of both the robot as well as the obstacles in the global coordinate frame. Moreover, the distance between the robot and the obstacles, the angle between the line joining the robot and obstacles and the goal reference line as obtained above are shown in Table 1. The radii of the robot (R) and two obstacles ($O_1,\ O_2$) are found to be equal to 5.428 cm, 4.884 cm, 5.272 cm, respectively. An attempt is also made to measure the CPU time of the developed system. The experiment is carried out in a P-IV PC and the CPU time is found to be equal to 0.040 seconds. It indicates that the present vision system might be suitable for on-line implementations.

Table 1. Positions of both the goal as well as two moving obstacles at different instants of time and the calculated inputs of the motion planner based on the goal point (65 cm, 105 cm), goal reference angle = 116° and $\Delta t = 1s$ – Case 1.

Time (t)	Object	X (cm)	Y (cm)	Inputs of motion planner Distance (cm)	Angle (degree)
t_1	Robot	98.241	30.626		
	Obs 1	28.715	26.300	64.776	54.339
	Obs 2	96.862	132.382	96.493	-32.616
$t_1 + \Delta t$	Robot	88.829	50.936		
	Obs 1	40.496	60.470	44.380	68.373
	Obs 2	85.215	108.946	52.580	-53.568
$t_1 + 2\Delta t$	Robot	79.721	70.979		
	Obs 1	50.221	95.894	33.730	-5.947
	Obs 2	74.491	85.831	10.474	-36.331
$t_1 + 3\Delta t$	Robot	70.892	92.205		
	Obs 1	60.948	127.329	31.621	-142.541
	Obs 2	63.028	62.831	25.136	21.780

5.2.2. Case 2: Four moving obstacles

To test the effectiveness of the developed image processing system, experiments are also carried out with four moving obstacles along with the planning robot as shown in Figure 10. The goal point is assumed to be at (5 cm, 180 cm) with

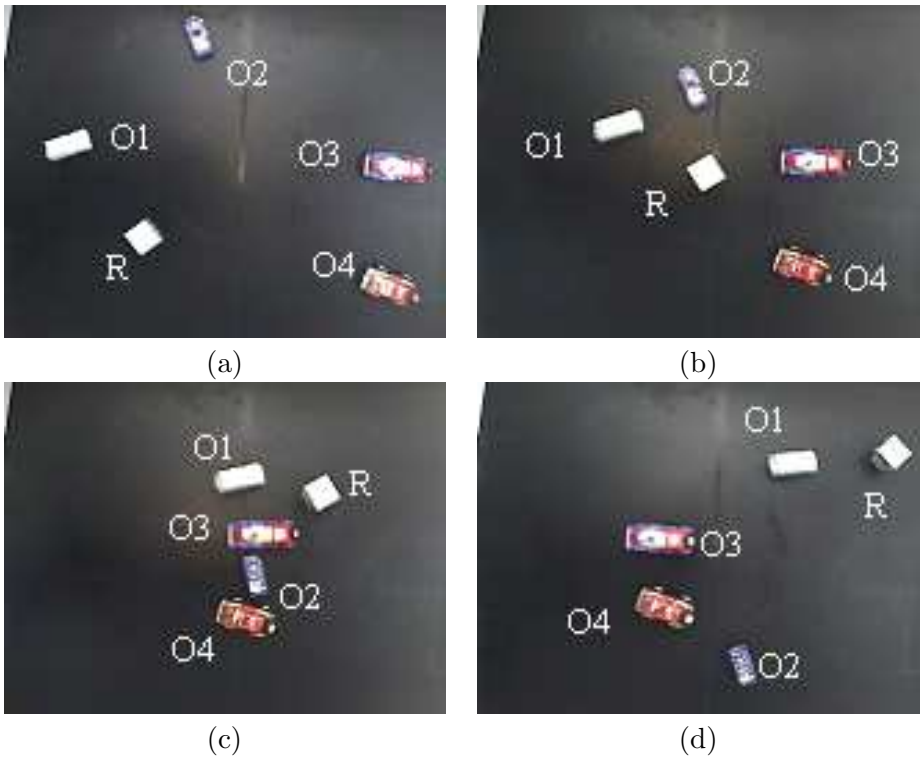

Figure 10. Images showing the movement of both the robot as well as the obstacles at four instants of time: (a) $t = t_1$, (b) $t = t_1 + \Delta t$, (c) $t = t_1 + 2\Delta t$ and (d) $t = t_1 + 3\Delta t$ – Case 2.

respect to the global coordinate frame and the goal reference line makes an angle of 117.772° with the $+ve$ X-axis of the global coordinate frame. Positions of both the robot as well as four moving obstacles are shown for different instants of time in Table 2. The inputs to the motion planner are also calculated and presented in Table 2. The radii of the robot (R) and the obstacles O_1, O_2, O_3, O_4 are seen to be equal to 5.658 cm, 4.765 cm, 2.858 cm, 5.050 cm, 5.200 cm, respectively. The CPU time is coming out to be equal to 0.058 seconds, which makes the developed vision system a perfect choice for computing the inputs of the motion planner, in

a highly complex and changeable environment. It is interesting to note that the developed vision system is very much sensitive to the value of *threshold*, which has been optimized above by using a GA.

Table 2. Positions of both the goal as well as four moving obstacles at different instants of time and the calculated inputs of the motion planner based on the goal point (5 cm ,180 cm), goal reference angle = 117.772° and $\Delta t = 1s$ – Case 2.

Time (t)	Object	X (cm)	Y (cm)	Inputs of motion planner Distance (cm)	Angle (degree)
t_1	Robot	83.431	49.727		
	Obs 1	50.115	22.452	37.587	-19.915
	Obs 2	10.816	69.900	72.507	-74.087
	Obs 3	57.643	144.217	92.896	46.070
	Obs 4	100.903	139.383	86.578	70.097
$t_1 + \Delta t$	Robot	63.111	82.242		
	Obs 1	42.964	50.603	32.309	-88.050
	Obs 2	28.711	77.254	31.902	-141.258
	Obs 3	56.652	122.043	35.272	50.194
	Obs 4	86.324	115.052	35.426	94.838
$t_1 + 2\Delta t$	Robot	43.585	114.600		
	Obs 1	34.308	81.666	29.016	165.305
	Obs 2	69.238	90.381	32.421	-102.984
	Obs 3	56.349	97.437	16.339	102.982
	Obs 4	71.204	88.860	32.989	112.625
$t_1 + 3\Delta t$	Robot	24.959	150.135		
	Obs 1	28.809	113.609	31.528	140.369
	Obs 2	81.011	102.472	70.719	-96.808
	Obs 3	55.615	68.807	81.864	-125.748
	Obs 4	56.619	59.500	91.241	127.145

5.3. *Comparison of the Present Work with Others' Work*

A vision system is developed to detect and recognize the moving objects for the purpose of mobile robot navigation. The vision system consists of a camera placed perpendicular to the field of view, frame grabber inserted on the PC and a image analyzing procedure. However, to get accurate and precise estimation of object positions, parameters of the vision system are to be calibrated. In the past, several attempts were made by various investigators to calibrate the camera parameters. Some of these are mentioned below for the purpose of comparison with the present work.

Zhang (2000) proposed a two-step method of camera calibration. In the first step, some of the camera parameters were estimated through direct transformation of the matrix and in the next step, all those parameters were refined by following Levenberg-Marquardt algorithm. But, the two-step method might be computationally expensive and initial estimation may provide with some crude results. Moreover, the solution provided by the Levenberg-Marquardt algorithm may get trapped into local minima, as it works based on the principle of steepest descent. Later on, after understanding the problem of two-step method, Ji and Zhang (2001) calibrated some of the intrinsic and all the extrinsic parameters of the camera with the help of a GA. However, their method may not provide accurate estimation of the postures of the objects, as they have neglected the distortion effects in the camera model. Klancar *et al.* (2004) et al. made an attempt to correct the effect of distortion of lens in vision system design for mobile robot tracking. They have corrected the radial distortion effects by varying the focal length of the camera only, rather than establishing a relationship between the global and camera coordinate systems. Thus, accuracy in calibration may not be so much good in this case. Although Cokal and Erden (1997) developed a step-wise image processing

technique for detecting the objects, they have not made any attempt to label the objects. Thus, the identities of the objects may get lost with the change in time.

In the present work, both extrinsic as well as intrinsic parameters of the camera are calibrated using a GA, in a single step. Radial distortion factors are included in the camera model and a proper relationship between the global and local coordinate frames is established based on the camera set-up. Labeling of the objects are done to recognize the objects and a small number of control points are considered during calibration. It is important to mention that the developed vision system is able to detect the varying number of objects, which need to be known a-priori.

6. Concluding Remarks

Computer vision and image processing plays an important role in real-time mobile robot navigation. Cameras may replace the sensors, but they require proper calibration to yield the reasonably good results. Moreover, to understand the dynamic environment properly, we need to have a fast and noise sensitive image processing technique in order to compute the postures and geometric properties of the objects present in an image. By keeping these facts in mind, in the present study, a method of calibration and image processing has been developed. The developed vision system is calibrated first and then its effectiveness is tested on two different cases involving two and four moving obstacles, separately. The developed system is able to identify and compute the shapes and positions of the objects through the processing of images taken at different instants of time. Some of the important findings are mentioned below.

(1) The problem of camera calibration could be solved effectively in a single step, even with a less number of control points, if it is posed as an optimization problem and solved using a GA.
(2) The concept of labeling helped us to identify different components present in an image.
(3) Estimation of compactness and perimeter values are helpful in finding and recognizing the different-shaped objects.
(4) The use of size filter can remove the scattered noise components in the labeled binary image.
(5) The CPU time of the developed vision system is found to be reasonably low. Thus, it can be used as a real-time vision-based navigation system.

7. Scope for Future Work

The present work may be extended in the following directions, on which the authors are working at present.

(1) In the present study, the vision system has been used for determining the inputs of motion planner, on-line. However, it will be more interesting to see its performance, after coupling it with a motion planning algorithm.
(2) Since the developed vision system contains only one camera, it is not possible to determine 3D postures of the objects. However, the same can be obtained by using a vision system consists of multiple cameras. But, in such a case, integration of the camera data is to be done efficiently in real-time.

Acknowledgment

This work is supported by the Department of Science and Technology, Govt. of India (Sanction No. SR/S3/RM/28/2003 dt. 12.12.2003). The authors gratefully acknowledge the cooperation of Dr. A. Roy Chowdhury of the Department of Mechanical Engineering, IIT Kharagpur, India.

References

Abdel-Aziz, I.I. and Karara, H.M., 1971. Direct linear transformation into object space coordinates in close-range photogrammetry. University of Illinois at Urbana Champaign Urbana, 1–18.

Apostopolous, D., Wagner, M. and Whittaker, W., 1999. Technology and field demonstration results in the robotic search for antarctic meteorites. 185–190.

Borenstein, J. and Koren, Y., 1989. Real time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems Man and Cybernetics*, 19 (5), 1179–1187.

Cokal, E. and Erden, A., 1997. Development of an image processing system for a special purpose mobile robot navigation. Toowoomba, Australia, 246–252.

DeSouza, G.N. and Kak, A.C., 2002. Vision for mobile robot navigation: a survey. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 24 (2), 237–267.

Faig, W., 1975. Calibration of close-range photogrammetry systems: mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41, 1479–1486.

Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization, Machine Learning*. Addison-Wesley, Reading, Mass, USA.

Hall, E.L., Tio, M.B.K., McPherson, C.A. and Sadjadi, F.A., 1982. Curved surface measurement and recognition for robot vision. 42–54.

Hui, N.B. and Pratihar, D.K., 2005. Some studies on camera calibration and image processing for mobile robot navigation. Pune, India, 384–396.

Ji, Q. and Zhang, Y., 2001. Camera calibration with genetic algorithms. *IEEE Transactions Systems, Man, and Cybernetics–Part A: Systems and Humans*, 31 (2), 120–130.

Kim, J.H., Kim, D.H., Kim, Y.J. and Seow, K.T., 2004. *Soccer robotics*. Springer, Amsterdam.

Klancar, G., Kristan, M. and Karba, R., 2004. Wide-angle camera distortions and non-uniform illumination in mobile robot tracking. *Robotics and Autonomous Systems*, 46, 125–133.

Koplowitz, J. and Bruckstein, A.M., 1989. Design of perimeter estimators for digitized planar shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 611–622.

Lee, M.K. and Lee, J.H., 1998. A 2-D image camera calibration using a mapping approximation of multi-layer perceptrons. *Journal of Control Automation and Systems Engineering*, 4 (4), 487–493.

Lee, P.S. and Shen, Y.E., 1994. Model-based location of automated guided vehicles in the navigation sessions by 3D computer vision. *Journal of Robotics Systems*, 11 (3), 181–195.

Lenz, R.K. and Tsai, R.Y., 1989. Calibrating a Cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 (9), 916–928.

Salvi, J., Armangue, X. and Batle, J., 2002. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35, 1617–1635.

Shah, S. and Aggarwal, J.K., 1997. Mobile robot navigation and scene modeling using stereo fish-eye lens system. *Machine Vision and Applications*, 10, 159–173.

Tapper, M., MacKerrow, P.J. and Abrantes, J., 2002. Problems encountered in the implementation of Tsai's algorithm for camera calibration. Auckland, 66–70.

Tsai, R.Y., 1987. A versatile camera calibration technique for high-accuracy 2D machine vision meteorology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 4, 323–344.

Tsukiyama, T. and Huang, T., 1987. Motion stereo for navigation of autonomous vehicle in man-made environments. *Pattern Recognition*, 20 (1), 105–113.

Weng, J., Cohen, P. and Herniou, M., 1992. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 (10), 965–980.

Zhang, Z., 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (11), 1330–1334.