GENETIC ALGORITHM BASED ROBOT MASSAGE

[1] Srinivasan Alavandar, [2] K Adhi vairava Sundaram, [3] M.J.Nigam

[1, 2] Department of Electronics and Computer Engineering, IIT Roorkee

[3]Assoc. Prof., Department of Electronics and Computer Engineering, IIT Roorkee

E-mail: [1]seenu.phd@gmail.com, [2]adhivs@yahoo.com, [3]mkndnfec@iitr.ernet.in

ABSTRACT

In this paper, a new robot massage experimental setup for leg using genetic algorithm based camera calibration is presented. Teach Mover, a five axis articulated robot is used to press the muscle from ankle to knee. The real leg massage problem is approximated by a frustum shaped model, which can be easily extended to real leg massage. Three different sensors that are encoders; mounted at each joint of the robot with six degrees of freedom, a calibrated camera and a grip switch; mounted at the wrist of the manipulator were used. Camera calibration is done with the help of an algorithm proposed by Qiang Ji et. al [1] to estimate internal and external camera parameters using seven control points. The distance between camera and the robot is assumed to be fixed. By estimating the position and orientation of the object, which is the frustum model, the linear trajectory is found which the robot follows. The result shows the feasibility of the use of above-mentioned approach. The algorithm works satisfactorily for wide range of varying parameters i.e. the position and orientation of the model.

Keywords: Massage, Camera Calibration, computer vision and genetic algorithm

1. INTRODUCTION

It has long been realized that manipulator's should be used in osteopathy. Osteopathy is a profession that originated in 1984 when Andrew T. Still devised a drugless technique of curing diseases by massage and other manipulative procedures a technique based on the theory that illness may be caused by the undue pressure of displaced bones on nerves and blood vessels. Central to this is massage between knee and ankle of the leg. The concept of massage may appear to be simple but design and implementation of the system is an extremely complex task.

Estimating pose is done by three different approaches. These may be classified as linear, analytic and non-linear. Linear method's ignore the orthonormality constraint of the rotation matrix and solve the pose of the object as a linear system [2]. The rotation matrix is then made orthonormal after the solution is found. Analytic solutions [3] are closed form solutions that exist for a small number of points and are sensitive to the accuracy of the feature point extraction. In nonlinear methods, the pose of the object is iteratively improved while the system remains nonlinear [4]. Here the nonlinear approach is followed.

2. STATEMENT OF THE PROBLEM

The problem under consideration is massaging automatically the leg of the patient in order to reduce the pain using the TeachMover robot arm. For this purpose the leg is approximated by using a frustum model (object). For massaging automatically, the pose (position and orientation) of the frustum should be estimated. This should be estimated in order to locate the frustum exactly in the world coordinates. There are many approaches in the computer vision domain for estimating the pose of the object. Single camera approach is followed here. Accurate object pose estimation is possible if the vision system observes some features of a known object and uses those features to estimate the object pose.

Let {O}, {W}, {T} and {C} be the object, Robot base, Tool and Camera coordinate frame respectively. All the other coordinates are defined with respect to the robot coordinates. Let the object coordinate be (x_o, y_o, z_o), the robot coordinates be (X, Y, Z), the tool coordinates be (x_t y_t, z_t) and the camera coordinates be (x_c, y_c, z_c). The following figure 2.1 shows the coordinate system used. The problem is to find the pose of the object with respect to robot coordinates $^{W}H_{O}$. Once the pose is found the massage can be executed easily by controlling the robot to follow a straight line. If the distance between the camera and the robot is fixed the pose of the camera with respect to the robot coordinates $^{W}H_{C}$ can be found. Then if the pose of the object with respect to the camera is known $^{C}H_{O}$, the pose of the object with respect to robot coordinates $^{W}H_{O}$ can be found. Once the object pose is known in robot base coordinates, the task is to maintain the pitch of the end effector downwards and the roll along the X-axis of the robot base coordinates. It is assumed that the robot, camera, and the frustum all lie in the same plane. It is also assumed that the dimension of the frustum is known prior. Though the TeachMover arm is an articulated one, it can be made to follow a straight line may be approximated by a series of these curved motions.

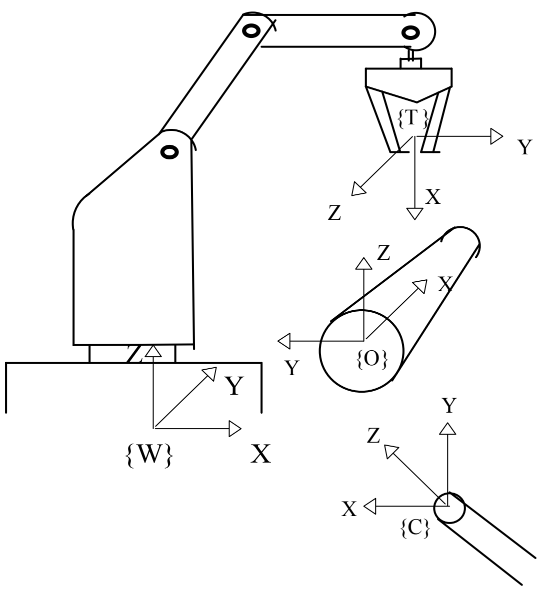

Fig.2.1. Coordinate system used-Right handed coordinated system.

3. GENETIC ALGORITHM BASED CAMERA CALIBRATION

A new genetic algorithm based camera calibration [1] has been proposed recently for calibrating camera based on known 3D coordinates. The algorithm is applied in this thesis for extraction of the pose. Most of the theoretical work supports for encoding the bits in binary and the industry supports for the real valued ones. This discrepancy was solved by Goldberg although not in favor of both of them [9]. Here the algorithm is encoded as the function of pan angle w, tilt angleϕ, and swing angle k of the object as follows:

$r_{11} = \cos\phi \cos\phi$
$r_{21} = \sin\omega \sin\phi \cos k + \cos\omega \sin k$
$r_{31} = -\cos\omega \sin\phi \cos k + \sin\omega \sin k$
$r_{12} = -\cos\phi \sin k$
$r_{22} = -\sin\omega \sin\phi \sin k + \cos\omega \cos k$
$r_{32} = \cos\omega \sin\phi \sin k + \sin\omega \cos k.$
$r_{13} = \sin\phi$
$r_{23} = -\sin\omega \cos\phi$
$r_{33} = \cos\omega \cos\phi$

It should be noted that the Euler angles given in [1] and here is different. This is because; here the pose of the object is estimated with that of the camera. The algorithm uses linear ranking selection and Roulette wheel selection. See [10] and [11] for an in-depth analysis of these selection schemes. The linear ranking selection scheme sorts in the best individual and gives mo re chromosomes for the successful one during selection process.

3.1. Objective Function

Let q be an vector consisting of the unknown interior and exterior camera parameters,

$q = [u_o, v_o, f, s_x, s_y, \omega, \phi, k, tx, ty, tz]^T$

Assuming q as a solution of interior and exterior camera parameters and $q \subseteq Q$, we have

$Q = \{q: q_i \in \lfloor q_i^- ; q_i^+ \rfloor \ ; i = 1, 2, \ldots\ldots, n\}$

where q_i^- and q_i^+ is the upper and lower bounds of q_i. The bounds on parameters can be obtained based on the knowledge of the camera. Any reasonable interval, which may cover possible parameter values, may be chosen as the bound of parameter q_i. An optimal solution of q with M control points can be achieved by minimizing,

$$\sum_{i=1}^{M} [\, g(q, O_i) - X_i)^2 + (k(q, O_i) - Y_i)^2 \,] \qquad q \in Q$$

where $g(q,Oi)$ is given by,

$$X_i = fs_u \frac{r_{11}x_0 + r_{12}y_0 + r_{13}z_0 + t_x}{r_{31}X_0 + r_{32}y_0 + r_{33}z_0 + t_z} + u_0 = g(q, O)$$

and $k(q,Oi)$ is given by

$$Y_i = fs_v \frac{r_{21}x_0 + r_{22}y_0 + r_{23}z_0 + t_y}{r_{31}X_0 + r_{32}y_0 + r_{33}z_0 + t_z} + v_0 = k(q, O)$$

3.2. Mutation

Mutation is a one-dimensional search, which occasionally changes the gene in different directions. The mutation scheme proposed in [1] is an extension of non uniform mutations scheme. Let k be the current optimal point, then the next current point is given by,

$$qk = \begin{bmatrix} q_k^{\,t} + I\Delta(t, q_k^{+}) + (I-1)\Delta(t, q_k^{-}) & \\ q_k^{+} & if\ q_k^{t+1} > q_k^{+} \\ q_k^{-} & if\ q_k^{t+1} < q_k^{-} \end{bmatrix}$$

where $\Delta(t, q_k^{+})$ and $\Delta(t,q_k^{-})$ are the step sizes for the upper and lower bounds of q_k respectively. I is an indicator function assuming the value of 0 and 1 depending on the outcome of a coin toss. The amount of perturbation is determined using the following formulas,

$$\Delta(t, q_k^{+}) = c(q_k^{+} - q_k^{t})(1 - r^{(1-t/(\alpha T))})$$
$$\Delta(t, q_k^{-}) = c(q_k^{t} - q_k^{-})(1 - r^{(1-t/(\alpha T))})$$

where r is the random variable distributed on the unit interval [0, 1]; α lies in [1, 1.5];T is the total number of iterations. The term raised to the power of r, the random variable limits the effect of mutation as the generation proceeds.

3.3. Crossover

Crossover produces new points in the search space. A new individual in generation t+1 can be expressed as linear combination of arbitrary selected individuals from the previous generation t, that is

$$q_i^{t+1} = (1 - \alpha\rho_i)q_i^{t} + \alpha\rho_i q_i^{t+1}$$
$$q_{i+1}^{t+1} = (1 - \alpha\rho_{i+1})q_{i+1}^{t} + \alpha\rho_{i+1} q_i^{t}$$

where α ranges within [0, 1], ρ is a bias factor. Assuming nonnegative fitness function ρ can be determined from the following equations:

$$\rho_\iota = \begin{cases} \varepsilon_\iota & if\ \varepsilon_\iota < 1 \\ 1, & if\ \varepsilon_\iota > 1 \end{cases}$$

$$\varepsilon_\iota = \frac{f(q_i^{t})}{f(q_{i+1}^{t})}\ ,\ \varepsilon_{\iota+1} = \frac{f(q_i^{t+1})}{f(q_i^{t})}$$

where f(*) denotes the GA's fitness function.

4. FRUSTUM MODEL AND OTHER IMPLEMENTATION ASPECTS

4.1. Approximation of the leg

To approximate real leg frustum shape is chosen. The frustum shape essentially is a cone with its tip chopped off. Usually the knee side has a larger diameter and the ankle side has lesser one. The frustum shaped model is constructed in wood. The frustum shape and its dimensions are shown in figure 4.1.

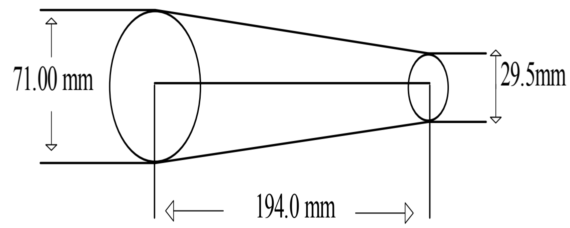

Fig. 4.1 Frustum shape and its dimensions

4.2 3-D Coordinate fixing

To identify the control points we have drawn lines in the frustum with 450 spacing with each other on each face. Then these lines are connected to their corresponding counterparts in the other face. Then using a compass several points have been marked at distances 13, 43, 83 and 160 mm as shown in figure 4.2-1. The 3-D coordinates of these points are found by using trigonometric identities and using the rotation matrices, which are given in mm as follows: P1 = (0, 0, 25); P2 = (12.9262, -20.5891, 20.5891); P3 =(42.7561, -18.333, 18.333); P4 = (82.5292, -15.3249, 15.3249); P5 = (12.9262,-29.1174, 0); P6 = (42.7561, -25.9268, 0); P7 = (82.5292, -21.6727, 0); P8 = (0,-17.6776, -17.6776) ; P9 = (157.1039, -9.6847, 9.6847); P10 = (157.1039, -21.6727, 0). Out of these ten calibrated control points, seven should be chosen by the user. On each control point a checker pattern is pasted with its center aligned to the point. This is done so as to identify the point in the 2D image with less ambiguity.

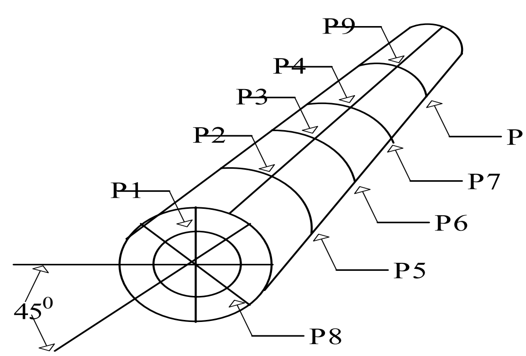

Fig. 4.2-1 Ten control points

4.3. Finding the coordinates in 2D image

For finding the coordinates of the point in 2D image the checker pattern has been pasted and the user has to select the required control points and along with that the 3D coordinates of those selected control points should be specified.

.

4.4. Robot control

After the identification of the control, points the genetic algorithm is executed which gives the value of the homogeneous matrix between the camera and the frustum CHO. Multiplying the CHO with that of the one between the camera and the robot WHC, the homogeneous matrix between the robot and object RHO is found out. This is used to generate the points that the robot has to follow, which are the coordinates of the Z axis of the object frame. Since the robot accepts only the pitch and roll, the roll is calculated by using the rotation matrix. The Y axes of the tool frame {T} is made parallel always to the X axis of the Object frame {O}, along with the X axes of Tool frame {T} to be in the direction opposite to that of the Y axes of the object frame {O}. The Z axes of both object {O} and the Tool {T} frame are in the same direction. The Roll of the end-effector is found using the trigonometric identities.

4.5. Programming algorithm in MATLAB

The algorithm has been written in MATLAB 6.0. The program implements the three results as explained in section 5. To get the images from the camera, Video for Matlab (VFM) toolbox was used. The get cord function gives the 2D co-ordinate of the control points and 3D points. These coordinates are given as input to the genetic algorithm which finds the camera intrinsic and extrinsic parameters. Along with that we need to specify the parameter bound of 8 parameters. The rest three are rotation parameters which are set to the maximum. After the homogeneous matrix between the camera and the object is estimated, it has to be multiplied with that of the robot and the camera. This matrix is found manually after fixing the camera in the workspace. This matrix is fed as a input to the setup. The robot is first initialized to the following coordinates manually (X = 5, Y=0, Z=0, P= -900 and R=00). The trajectory is planned in the Cartesian coordinates and converted to joint angles using the backward arm solution. The robot accepts the joint angle as inputs in terms of motor steps. The base and shoulder joint takes 7072 steps in one revolution, Elbow joint 4158 steps followed by right and left wrist requiring 1536 steps. This information is used to convert from degree to motor steps.

5. RESULTS

The results are twofold. First, the algorithm is tested by keeping the frustum model in various pose. It is followed by robotic massage task. The frustum model was kept at different pose and the algorithm was applied. The parameter bounds were fixed in consonance with the practical situation on the ground and the estimated values for different pose are given in the table 5.3 for the corresponding figures 5.2 (a-d). It can be noted that in figure 5.2-b, the seven control points lie in a single plane, which can be attributed for the poor estimation of the parameters. Although the algorithm iterated for more than 10,000 iterations, the square of the error in this case was high and never decreased. Thus it can be observed that the seven control points should not be in a single plane for the convergence of the algorithm. It was also observed that the scale factors were the most affected one if the control points approximately lie in a single plane. This observation is inline with that of the one reported in [2]. In fact, [2] has removed the scale factors from single plane calibration and advises to calibrate scale factors only in multi plane techniques. Thus, the camera was positioned in such a way that the seven control points can be observed and distributed throughout the image. Also, the frustum was kept in such a way that the bigger circle faces camera all the time. If the smaller side is made to face the camera, it has been observed that some of the control points were not visible and some fall in the same optical flow line. The frustum model can be placed anywhere in the view but the only condition is to align any one of lines drawn on the bigger face and the plane; where camera and the robot is kept, perpendicularly. Our interest is centered on three rotation and translation components rather than all the eleven parameters. Therefore, if these three parameters tend to be constant for more than a specified number of generations, the iteration is stopped.

Finally, the algorithm was applied to the robot massage problem. The distance between the robot and the camera was fixed. The camera was kept facing the frustum. It was kept at Robot base coordinates (9, -4, 2) in inches facing the robot at 450. The frustum was kept at the robot base coordinates (5, -3, 1.3898) in inches.

Table 5.3. Estimated values for different poses

Parameter	Parameter Bound	Estimated Values for fig. 5.2a	Estimated Values for fig. 5.2b	Estimated Values for fig. 5.2c	Estimated Values for fig. 5.2d
f	3,25	5.61	6.04	5.23	3.9
s_x	90,250	173.9.77	116	181.79	157.3
S_y	90,250	179.6	171	161.25	148.7
u_0	280,360	326.9	347.3	324.5	333.4
v_0	210,320	308.4	273.2	288.29	256.2
t_x	-150,150	-25.08	-66.9	-24.8	-22.6
t_y	200,-200	-14.54	11.93	-13.75	-3.14
t_z	0,300	163.8	109.1	149.33	74.7
ω	-π,π	1.662	1.37	1.66	1.71
ϕ	-π,π	0.054	-0.102	0.397	0.086
k	-π,π	0.64	0.6005	1.04	0.705

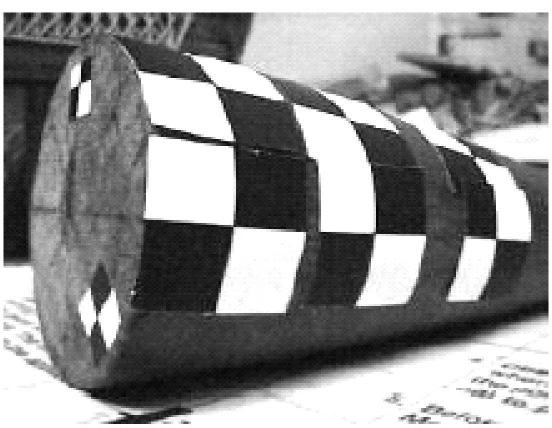

Fig.5.2a

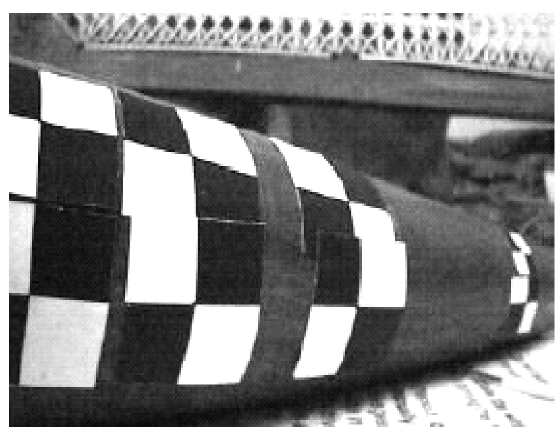

Fig.5.2b

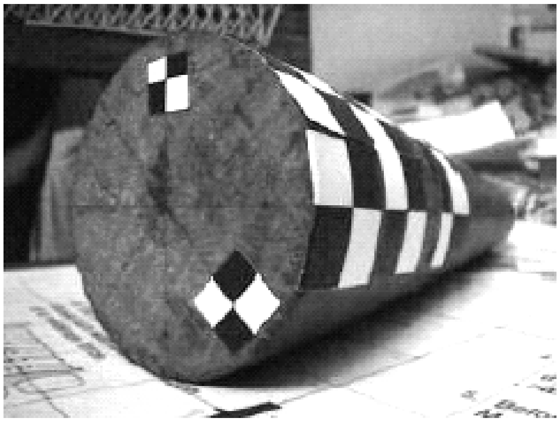

Fig.5.2c

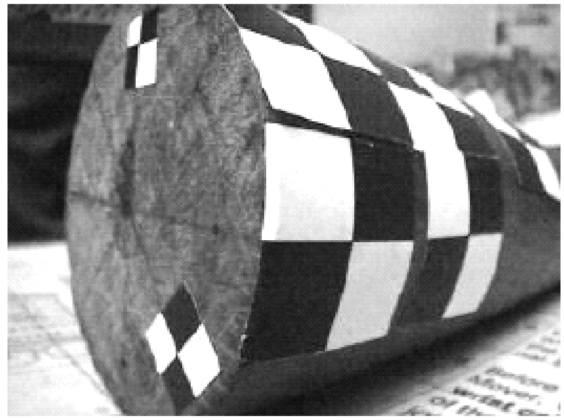

Fig.5.2d

The genetic algorithm based camera calibration program was run and the estimated values were used to determine the exact position of the frustum model relative to the robot base co-ordinate i.e., ${}^{W}H_{O}$ as given below. It was assumed that the frustum model does not move throughout the whole time after the intrinsic and extrinsic parameters were found by using the camera calibration algorithm.

$$
{}^{W}H_{O} = \begin{bmatrix} 0.009501 & -0.9951 & 0.00456 & 4.8933 \\ 0.99664 & 0.00891 & -0.1004 & -2.521 \\ 0.1004 & 0.000001 & 0.99664 & 1.2014 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

This was used to find out points just 10 mm above the end point of the frustum. The robot's centre of the gripper was positioned such that it traverses the centre line of the frustum model and massages the leg model. The actual trajectory shown is parallel to the plane in which the frustum model was kept. The trajectories (figure 5.4-a and 5.4-b) show the line followed by the joint just above the gripper and not of the centre point of the gripper. The condition for positioning the frustum is that any one of the line drawn on the bigger circle of the frustum should be perpendicular to the plane in which the robot and the camera were kept. The photo sequences that the robot followed are shown in figure 5.3. (a -d). The trajectory is actually obtained from the feedback information provided by the position registers mounted at the joint of the robot. The massage is done at four places in the frustum at a distance 7mm, 67 mm, 127 mm, and 187mm from the centre point of the frustum respectively. The centre point of the frustum is the origin of the object frame {O}.

Fig.5.3a. End effector is at point 7 mm from the centre of the frustum

Fig.5.3b. End effector is at point 67 mm from the centre of the frustum

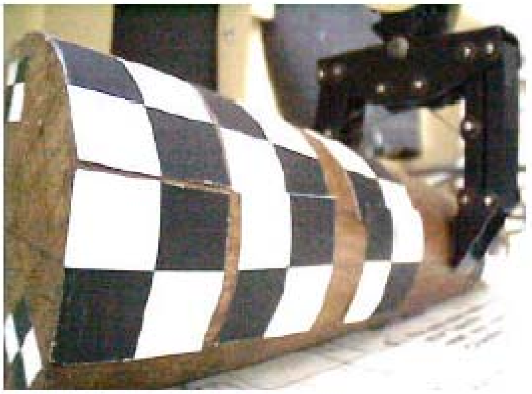

Fig.5.3c. End effector is at point 127 mm from the centre of the frustum

Fig.5.3d. End effector is at point 187 mm from the centre of the frustum

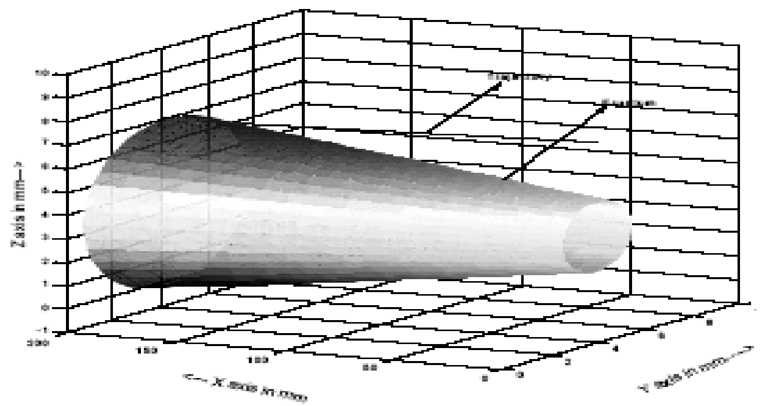

Fig.5.4a. Trajectory followed by robot and effector

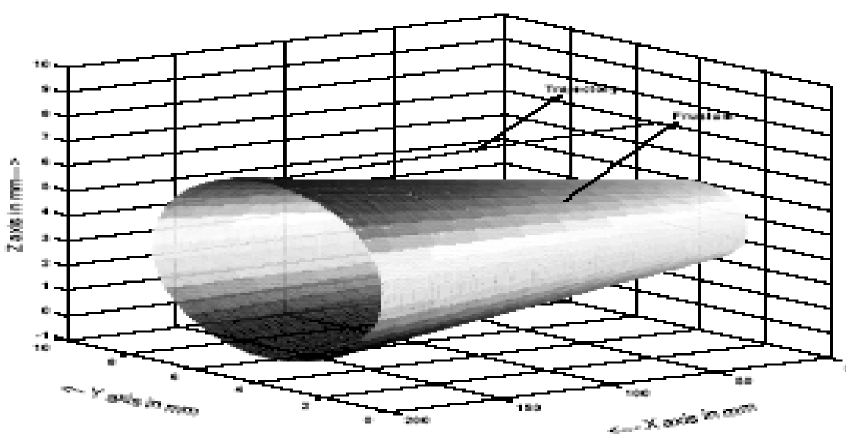

Fig.5.4b. Trajectory followed by robot as seen from camera

6. CONCLUSION

In this paper, a problem of massage robot is examined using robot (arm control), a Personal Computer based camera and a frustum model. The leg is approximated with a frustum shaped model. The distance between the camera and robot is fixed. However, this can be improved either by having another camera at the robot's hand or by estimating the pose of the robot by using a wide-angle camera. The position and orientation of the frustum is estimated by using a camera, which is calibrated, by using a recently proposed algorithm by Qiang Ji et. al. [1]. The results obtained show the feasibility of the approach and the error curves show the convergence of the algorithm. MATLAB program was developed for controlling the robot, getting image coordinates from the camera and implementation of the camera calibration algorithm. The required condition to obtain the results is that the seven control points should be visible and not all of them should be in a single plane. The parameters given to the calibration algorithm can be varied provided it is within a reasonable limit. For real massaging problem the trajectory will not be exactly a linear one. Further, by having a prosthetic hand we could do complex massaging tasks.

REFERENCES

1. Qiang ji and Yongmian Zhang, "Camera Calibration with Genetic Algorithms", IEEE Trans. On Sys., man and Cyb.,vol.31, No.2,pp120-130, March 2001.

2. R.Y.Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off –the-shelf TV cameras and lenses," IEEE J. Robot. Automat., vol. RA-3. pp.323- 344, Aug.1987

3. DeMenthon, D. F., and L.S.Davis, "New exact and approximate solutions of the three point perspective problem.", IEEE Trans. On Pattern Analysis and Machine Intelligence, vol.14, pp.1100-1105, 1992.

4. Kalyanmoy Deb, "Genetic algorithm in search and optimization: The technique and applications", Department of Mechanical Engineering Report, Indian Institute of Technology, Kanpur.(Downloadable from researchindex.org)

5. Microbot TeachMover user manual. S.Hutchinson, G.D.Hager, and P.I. Cork

6. S. A. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. IEEE Trans. Robot. Automat., 12(5):651--670, Oct. 1996.

7. Juyang Weng, Paul Cohen and Marc Herniiou et al. "Camera calibration with distortion models and accuracy evaluation", IEEE Trans. Pattern Anal. Machine Intell. Vol.14, pp 965-980, 1992.

8. Blickle, T., Thiele, L., "A comparison of selection schemes used in genetic algorithms", TIK_Report Nr. 11, December 1995.

9. Goldberg, " Real coded genetic algorithms, virtual alphabets, and blocking", Department of Electrical Engineering, University of Illinois at Urbana-Champaign, U.S.A (downloadable from researchindex.org)

10. Mühlenbein, H. and Schlierkamp, D.: "Analysis of selection, mutation and recombination in genetic algorithms". Technical Report 93-24, GMD, 1993.

11. Zhang B.T., Kim J, "Comparison of selection methods for evolutionary optimization", Evolutionary optimization, Volume 2, Number1 pp.55-70,2000.

12. P.I. Corke, "Visual control of robot manipulators— A review," in Visual Serving, K.Hashimoto, Ed,Singapore: *World Scientific*, pp.1-32,1994.

13. Juyang weng, Paul cohen and Marc Herniiou , "Camera calibration with distortion models and accuracy estimation", *IEEE Transactions on pattern analysis and machine Intelligence*, vol.14. no.10. pp.965-980. October 1992

14. Hanqi Zhuang, Zvi S.Roth, "Camera-aided robot calibration", CRC press, 1996

15. W.M.Spears, "The role of mutation and recombination in evolutionary algorithms", Ph.D.Thesis, George Mason University, 1998.

16. David E. Goldberg, "Genetic algorithms in search, optimization & machine learning", Addision Wesley, 1989.
17. M.Mitchell, "An Introduction to Genetic Algorithms. Cambridge", Prentice-Hall, India,-1996.

18. M. Wilczkowiak, E. Boyer and P. Sturm. "Camera Calibration and 3D Reconstruction from Single Images Using Parallelepipeds" .ICCV - IEEE International Conference on Computer Vision, Vancouver, British Columbia, Canada, pp. 142-148, Vol. I, July 2001.