Robot Path Generation by Viewing a Static Scene from a Single Camera

Dimitrios Aristos*, Theodore Pachidis**, John Lygouras***
Department of Electrical and Computer Engineering
Democritus University of Thrace
67100 Xanthi-Greece
*daristos@hotmail.com **pated@mail.otenet.gr *** ilygour@ee.duth.gr

Abstract - In this paper a new method for robot path generation on a plane surface is proposed. As a sensor a single camera mounted on the end-effector of a robotic manipulator is used. Processing an image of a static scene of the environment of the manipulator, where the desired track is contained, the 3D path points are accurately calculated. The accuracy of the calculation is achieved by combining two well-known methods for camera calibration and hand/eye – robot/world calibration. The method was implemented in a PUMA 761 robotic manipulator using a proper program developed in C. Finally the accuracy of the calculation was checked in the same robot using a software application in Visual C++.

I. INTRODUCTION

A major problem in robotics is the accurate calculation of a desired robot path. Several researchers have proposed methods where the initial path of a robotic manipulator could be generated. In some cases, a vision system was used as a sensor for the robot path generation. "Teaching by showing" is one method where a stereovision system is used to generate a robot path [1-4]. According to this method an object is moving while a stereovision system captures successive stereo images. Then, by processing these images the desired path is generated. A recent proposed method in [5] uses again a stereo system or the pseudo stereovision system (PSVS) [6]. According to the last method, edge points of an image captured by means of an ordinary stereo system or the PSVS of a static scene can be converted to 3-D path points.

In this paper a new approach to this problem is proposed. The basic concept is to find again robot path points but using a single camera and exploiting the accuracy in calculations during camera calibration and hand/eye-robot/world calibration. Several camera calibration methods have been proposed [7-12]. In this approach Z. Zhang method for camera calibration is used. The intrinsic and the extrinsic camera parameters are obtained. Transformation matrices, which relate robot base coordinate system with the world coordinate system and camera coordinate system with the end effector coordinate system, are necessary to be estimated. The estimation is possible if some of the proposed from several researchers methods are used. The most commonly methods used for the evaluation of the above transformations are the linear method, the closed-form method and the non-linear minimization method [13-17]. The method used in this approach is the linear method proposed in [14]. A combination of the previous selected methods and using a simple image processing procedure permit the accurate calculation of successive robot path points.

This paper is organized as follows. In section II, the camera calibration problem is briefly described. In section III, the linear method for the solution of the hand/eye - robot/world calibration problem is presented. In section IV, the processing of an image containing the desired track, captured by the camera, is briefly explained. In section V, a method for the calculation of real 3D coordinates of the robot path is proposed, based on pixel coordinates found during the image processing part of the procedure. In section VI, the experimental results for the camera calibration and hand/eye-robot/world calibration problems are provided. Finally in section VII, the conclusions of this work are presented.

II. CAMERA CALIBRATION

An analogue camera, mounted on the end-effector of the robot, is used to observe a plane surface (i.e. the surface of a table). The world coordinate frame is located on the surface of the table. Every point in 3-D space can be described with respect to the world coordinate frame, so it can be denoted by $\mathbf{M}=[X,Y,Z]^T$. The augmented vector is found by adding 1 as the last element: $\mathbf{M}'=[X,Y,Z,1]^T$. If this certain point is observed by the camera, it is transformed into a pixel into the digital image, which is finally stored into the memory of the host PC. This pixel, that corresponds to the real point $\mathbf{M}$, can be denoted by $\mathbf{m}=[u,v]^T$, where u,v are pixel coordinates of the captured image. Respectively, the augmented vector of $\mathbf{m}$ is $\mathbf{m}'=[u,v,1]^T$. A camera is modeled by the usual pinhole, so the relationship between a 3D point $\mathbf{M}$ and its image projection 2D point m is given by:

$$s \cdot \boldsymbol{m}' = \boldsymbol{A}_{in} \cdot [\boldsymbol{R}\ \boldsymbol{t}] \cdot \boldsymbol{M}' \quad \text{with } \boldsymbol{A}_{in} = \begin{bmatrix} a & c & u_0 \\ 0 & b & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

where s is an arbitrary scaling factor. The 3×3 rotation matrix **R** and the 3×1 translation vector **t** are called extrinsic parameters of the camera, and relate the world coordinate frame to the camera coordinate frame. It is obvious that the extrinsic parameters change if the camera is moved. The intrinsic parameters are given by the 3×3 matrix $\boldsymbol{A}_{in}$, and describe the internal characteristics of the camera, such as the pixel coordinates of the principal point (u_0,v_0), the scale factors a, b in image's u and v axes, and the parameter c describing the skewness of the two image axes.

The camera calibration problem is to find the intrinsic parameter matrix $\boldsymbol{A}_{in}$, which is stable for a certain camera, and the extrinsic parameters **R**, **t** for various positions of the camera. A model pattern with known geometry is usually used for the calibration process, such as a pattern with fixed geometric features, printed on a laser printer. This is necessary because certain points **M** of the pattern (i.e. its corners or edges) must have known coordinates [X,Y,Z] with respect to a fixed coordinate frame. The model plane is put on the surface of the table (so we can assume that Z=0) and the camera is moved to acquire different aspects of the non-moving pattern. Using an edge or corner detection algorithm, the images, captured by the camera, can be digitally processed. In this way, the pixels **m**=(u,v) that correspond to certain points of the pattern can be found. If at least three images of the same pattern are acquired, it is possible to calculate the intrinsic parameters of the camera (matrix $\boldsymbol{A}_{in}$) and the extrinsic parameters for the different (at least three) positions of the camera. The algorithm for the solution of the camera calibration problem can be found in [11-12].

III. ROBOT/WORLD AND HAND/EYE CALIBRATION

In many cases it is necessary to relate the robot base coordinate frame to the world frame, or the end-effector frame to the camera frame. The calculation of the unknown transformations **X**, **Z** between these coordinate frames is usually named as "Robot/World and Hand/Eye Calibration". Fig.1 depicts the robot geometry, the corresponding coordinate frames and the transformation matrixes **A**, **B**, **X**, **Z** among them. Transformation **B** is known by the solution of the manipulator's forward kinematics problem, which is usually done automatically by the software of the robot. Transformation **A** has been computed by the camera calibration procedure and includes the extrinsic parameters of the camera (rotation and translation **R**, **t**). Although matrixes **X** and **Z** are constant for certain robot geometry, the elements of the matrixes **A**, **B**, change every time the end effector of the robotic manipulator and consequently the camera mounted on it is moving.

The linear method for the computation of the unknown **X**, **Z** uses quaternion algebra for the representation of the rotation matrixes. A method for the conversion of a rotation matrix into the corresponding unit quaternions, and vice-versa, can be found in [18]. The Hand/Eye and Robot/World calibration problem refers to the solution of a homogenous matrix equation of the form **AX**=**ZB** for at least two different positions of the robot. This matrix equation can be decomposed into a rotation equation and a position equation:

$$\mathbf{R}_A\mathbf{R}_X = \mathbf{R}_Z\mathbf{R}_B \tag{2}$$

$$\mathbf{R}_A\mathbf{t}_X + \mathbf{t}_A = \mathbf{R}_Z\mathbf{t}_B + \mathbf{t}_Z \tag{3}$$

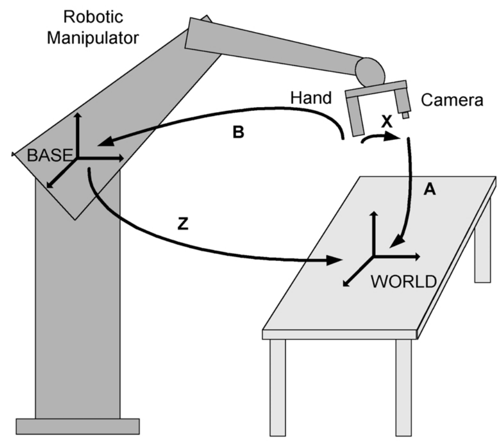

Fig.1 The robot geometry including the robot base, the end-effector, the camera and the world coordinate frames. The unknowns are the Robot/World (**Z**) and the Hand/Eye (**X**) transformations.

A unit quaternion, $\mathbf{q}_X$, which corresponds to a rotation matrix $\mathbf{R}_X$ can be denoted as:

$$\mathbf{q}_X = (q_0, q_1, q_2, q_3) = (q_0, \mathbf{q})$$

Let $\mathbf{q}_A = (a_0, \mathbf{a})$, $\mathbf{q}_B = (b_0, \mathbf{b})$, $\mathbf{q}_X = (x_0, \mathbf{x})$ and $\mathbf{q}_Z = (z_0, \mathbf{z})$ be the unit quaternions that correspond to the rotation matrixes $\mathbf{R}_A, \mathbf{R}_B, \mathbf{R}_X$ and $\mathbf{R}_Z$ respectively. From equation (2) we can obtain a system of three linear equations with six unknowns of the form:

$$\underbrace{\mathbf{J}}_{3\times6} \cdot \underbrace{\mathbf{u}}_{6\times1} = \mathbf{b} - (b_0/a_0)\mathbf{a} \tag{4}$$

where $\mathbf{u}^T = [u_1, u_2, u_3, u_4, u_5, u_6] = [\mathbf{x}^T/z_0, \mathbf{z}^T/z_0]$ is the unknown vector. The matrix **J** is known and can be computed from $\mathbf{q}_A$ and $\mathbf{q}_B$. More detailed information about the computation of **J** can be found in [14]. Equation (4) refers to the rotations of only one robot position between the coordinate frames of Fig. 1. Every additional robot position yields three additional equations of the form of (4) with the same unknowns, so at least two robot positions are required to obtain a 6×6 system of linear equations. Solving for $\mathbf{u}$, the components of $\mathbf{q}_X$ and $\mathbf{q}_Z$ can be determined using the equations below:

$$\begin{aligned} z_0 &= \pm(1+u_4+u_5+u_6)^{-1/2} \\ \mathbf{z} &= z_0[u_4, u_5, u_6] \\ \mathbf{x} &= z_0[u_1, u_2, u_3] \\ x_0 &= \pm(1-\|\mathbf{x}\|^2)^{1/2} \end{aligned} \tag{5}$$

The computation of the rotation matrixes $\mathbf{R}_X$ and $\mathbf{R}_Z$ with respect to the quaternions $\mathbf{q}_X$ and $\mathbf{q}_Z$ becomes trivial:

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2+q_2^2-q_1^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2+q_3^2-q_1^2-q_2^2 \end{bmatrix}$$

The remaining unknowns are the translation vectors $\mathbf{t}_x$ and $\mathbf{t}_z$. Equation (3) can be rewritten as a 3×6 system of linear equations of the form:

$$\underbrace{\mathbf{F}}_{3\times 6} \cdot \underbrace{\begin{bmatrix} \mathbf{t}_X \\ \mathbf{t}_Z \end{bmatrix}}_{6\times 1} = \mathbf{d} \tag{6}$$

with:

$$\begin{aligned} \mathbf{F} &= [\mathbf{R}_A \quad -\mathbf{I}_{3\times 3}] \\ \mathbf{d} &= \mathbf{R}_Z \cdot \mathbf{t}_B - \mathbf{t}_A \end{aligned}$$

At least two different positions of the robot are required to form a 6×6 system of linear equations and obtain a unique solution vector. In practice, however, many more robot positions are required to obtain accurate and reliable results. For the solution of a linear system with more equations than the unknowns, the Singular Value Decomposition algorithm can be used [19].

IV. IMAGE PROCESSING

The camera mounted on the end-effector of the robot manipulator observes the scene on a plane surface (i.e. the surface of a table). On the plane surface there is a pattern of the track that the end-effector of the robot must follow. In this case we have drawn a curved line on a piece of paper, as shown in Fig.2. The purpose of the image processing part of the procedure is to find the central pixels of the track (the black line) from one end (bottom of image) to the other (top-right of image). Many image processing methods can be used for this purpose. One way is to convert the 8-bit image into binary (1-bit) in order to distinguish the foreground (black line) from the background. Another way is to perform edge detection through the whole image, as the edges correspond to pixels of the track. This method involves scanning of the whole image for the detection of areas where neighbor pixels have large variations of color values. For example the pixels that correspond to the background of the image have similar color values close to white. An edge is detected if the variation between neighbor pixel values exceeds a predetermined threshold value. Concerning the image in Fig.2, this is possible only for the image area including the track, because this is the only part of the image where abrupt variations of pixel color values occur.

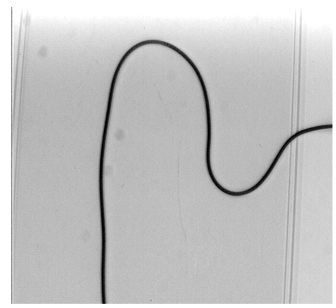

Fig.2 Image of a track on a plane surface captured by a camera mounted on the end-effector of the robotic manipulator PUMA 761.

It is also known that a small amount of distortion is introduced in an image captured by a camera. However the most important factor that must be taken into consideration is the lens distortion of the camera, especially radial distortion. The reader is referred to [11-12] for a complete description of the model used to describe radial distortion. During the camera calibration process it is possible to estimate the coefficients k_1 and k_2 of the radial distortion, which can be used before the image processing procedure, to remove this distortion from the image. This task is necessary if better accuracy is needed for the calculation of the real world coordinates that follows.

V. CALCULATION OF WORLD COORDINATES

The final step of the procedure is the calculation of the real world coordinates of the path that the end-effector must follow. Based on the central pixels of the track, which have been found during the image processing, we must find a way to transform pixel coordinates into real world coordinates. Provided that the camera calibration problem has been solved, the intrinsic parameters (array $\boldsymbol{A}_{in}$) and the extrinsic parameters (rotation $\mathbf{R}$ and translation $\mathbf{t}$), for a certain position of the camera with respect to the observed planar surface, are known. To obtain accurate results, the camera must not be moved from this specific position. Otherwise, the values of the rotation array **R** and translation vector **t** between the camera and the real world coordinate frames will change and will not be known any more.

The basic equation (1) of the camera calibration problem transforms a scene point **M**, with real world coordinates [X,Y,Z], into the corresponding pixel in the digital image array, with pixel coordinates **m**=[u,v].

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}_{in} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

In this case we are interested in the inverse procedure, which is the calculation of the real world coordinates [X,Y,Z] that correspond to a certain pixel [u,v] of the image captured by the camera. Provided that Z=0 for all the scene points of the observed planar surface, the above array equation can be rewritten as:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} ar_{11}+u_0r_{31} & ar_{12}+u_0r_{32} & ar_{13}+u_0r_{33} \\ br_{21}+v_0r_{31} & br_{22}+v_0r_{32} & br_{23}+v_0r_{33} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} at_x+u_0t_z \\ bt_y+v_0t_z \\ t_z \end{bmatrix} \overset{Z=0}{\Leftrightarrow} \quad (8)$$

$$\begin{bmatrix} ar_{11}+u_0r_{31} & ar_{12}+u_0r_{32} & -u \\ br_{21}+v_0r_{31} & br_{22}+v_0r_{32} & -v \\ r_{31} & r_{32} & -1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ s \end{bmatrix} = \begin{bmatrix} -at_x-u_0t_z \\ -bt_y-v_0t_z \\ -t_z \end{bmatrix}$$

From the solution of the above 3×3 system of linear equations we obtain the desired real world coordinates that correspond to a pixel **m** of the image. The scaling factor s is of no interest and there is no need to be found.

So the track of the image shown in Fig.2 is transformed into a real world 2D path, with world coordinates found using equation (8).

The final step is to transform the real world coordinates into robot base coordinates, as the position of the end-effector is controlled by the robot's software with respect to the robot base coordinate frame. This is very simple to be done, as far as the transformation **Z** from the world coordinate frame to the robot base coordinate frame is known from the solution of the hand/eye and robot/world calibration problem. The desired transformation takes place with the multiplication of **Z** with the real world coordinates X,Y,Z, using the equation (9):

$$\begin{bmatrix} RobotBase\_X \\ RobotBase\_Y \\ RobotBase\_Z \\ 1 \end{bmatrix} = \mathbf{Z} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{z(11)} & r_{z(12)} & r_{z(13)} & t_{z(x)} \\ r_{z(21)} & r_{z(22)} & r_{z(23)} & t_{z(y)} \\ r_{z(31)} & r_{z(32)} & r_{z(33)} & t_{z(z)} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (9)$$

VI. EXPERIMENTAL RESULTS

In this section some experimental results are given for the camera calibration problem and the hand/eye-robot/world calibration procedure. A PUMA 761 robot manipulator is used, with an analogue camera PULNIX TM-520 CCIR mounted on its end-effector. The images of the pattern shown in Fig.3 have 512×512 pixel resolution and 8-bit color depth. The chessboard model pattern shown in Fig.3 is printed on a laser printer and has been designed so that the distance between two neighbor inner corners is 2cm in both horizontal and vertical directions.

The source code for the solution of the camera calibration problem is developed in C. The program processes a number of images of the pattern taken from various orientations of the camera. Then, the sub-pixel coordinates for all corners of each image are found, and the intrinsic and extrinsic parameters of the camera are calculated combining these coordinates with the real world coordinates of the corners. The accuracy of the final results increases as the number of the pattern images increases. Table 1 shows the intrinsic parameters of the camera found for various numbers of input images.

The intrinsic parameter c is found to be very close to zero, which means that the angle between the two image axes u and v is very close to 90°. The extrinsic parameters are also calculated for each position of the camera and will be used for the computation of the unknown transformations **X** and **Z** (Hand/Eye and Robot/World Calibration).

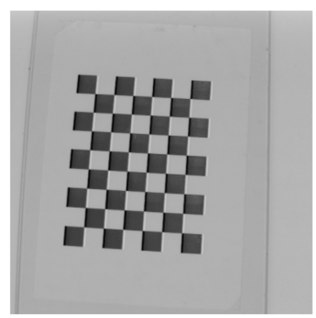

Fig.3 Image of the chessboard pattern as observed from an unknown orientation.

The source code for the computation of the transformations **X** and **Z** is also developed in C, using, as inputs for n different positions of the robot manipulator, matrixes $\mathbf{A}_n$ (extrinsic parameters) and $\mathbf{B}_i$ (transformations from the end-effector coordinate frame to the robot base coordinate frame for i robot positions). Table 2 shows the translation vectors of the transformations **X** and **Z** found, for various numbers of robot positions.

TABLE 1

Number of images	a	b	u0	v0
4	1375.907	1386.020	356.523	322.127
5	1366.065	1374.967	353.078	321.958
6	1353.795	1363.750	352.583	332.687
7	1513.394	1505.001	352.215	207.627
8	1429.292	1430.947	344.972	277.136
9	1440.030	1439.174	343.899	266.191
10	1440.947	1439.924	344.080	265.762
11	1448.901	1446.719	344.741	259.759
12	1450.796	1448.004	345.413	258.451
13	1448.997	1446.579	344.978	260.033
14	1445.087	1443.219	344.882	262.513
15	1442.564	1441.088	345.219	264.667
16	1442.430	1440.977	346.082	264.648
17	1445.876	1444.094	346.161	262.189
18	1443.631	1442.149	346.186	263.741
19	1443.453	1442.005	346.294	264.130
20	1446.424	1445.119	345.760	263.520

TABLE 2

Number of robot pos.	Translation vector of transformation **X**			Translation vector of transformation **Z**		
	X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)
4	-15.56	22.01	-7.17	-37.29	-150.49	52.82
5	-11.80	14.96	-7.52	-31.05	-143.23	51.28
6	-12.05	12.09	-10.86	-30.56	-142.48	56.34
7	-8.41	13.39	-12.45	-26.33	-143.69	55.65
8	-8.81	13.41	-13.27	-24.51	-144.70	56.41
9	-5.23	11.36	-12.72	-20.32	-142.73	56.74
10	-6.43	12.30	-13.48	-20.71	-143.94	56.81
11	-4.06	12.58	-13.16	-17.84	-144.12	56.31
12	-3.76	13.56	-13.21	-17.86	-144.87	55.95
13	0.68	15.75	-12.92	-12.22	-147.50	56.09
14	1.42	16.08	-12.86	-11.53	-147.86	56.14
15	1.23	15.82	-12.77	-12.31	-147.51	56.20
16	1.04	15.61	-12.62	-12.93	-147.20	56.21
17	0.87	15.32	-12.59	-13.32	-146.97	56.53
18	0.95	15.10	-12.56	-13.22	-146.86	56.80
19	0.74	14.92	-12.52	-13.81	-146.58	56.83
20	0.84	15.69	-12.50	-13.82	-147.05	55.98

The rotation matrixes of the transformations **X** and **Z** converge more quickly, and their values are more stable with respect to the number of the different robot positions. For 20 robot positions their values are:

$$\mathbf{R}_X = \begin{bmatrix} 0.998038 & 0.062158 & -0.007486 \\ 0.06128 & -0.945377 & 0.320168 \\ 0.012824 & -0.319999 & -0.947331 \end{bmatrix}$$

and

$$\mathbf{R}_Z = \begin{bmatrix} 0.999939 & -0.007211 & 0.008356 \\ 0.007253 & 0.999961 & -0.005114 \\ -0.008319 & 0.005174 & 0.999952 \end{bmatrix}$$

The rotation matrix $\mathbf{R}_Z$ is very close to the identity matrix $\mathbf{I}_{3\times3}$. This means that the axes of both real world and robot base coordinate frames are almost parallel.

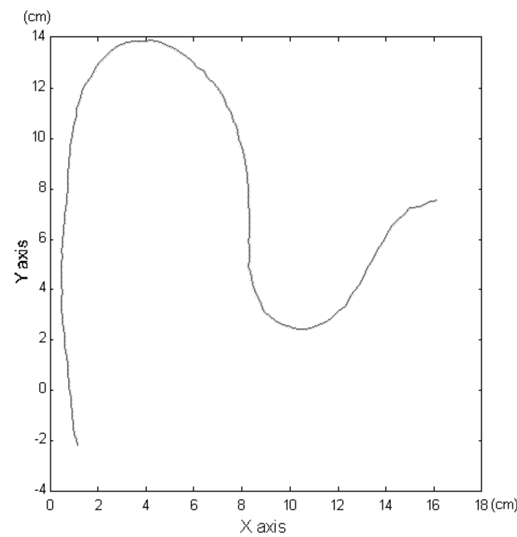

Fig.4 The track of the image in Fig.2 transformed into real world coordinates (in cm).

In order to archive higher accuracy using less robot positions, it is possible to use the non-linear minimization method. The most commonly non-linear method used is the Levenberg-Marquardt algorithm, which makes a maximum likelihood estimation to obtain more accurate results. The initial estimates required can be provided by the results of the standard linear method. The failure of the linear method to converge at a small number of robot positions is mostly attributed to the presence of noise in the values of the transformations **A** and **B**, which are used for the computation of the unknown transformations **X** and **Z**. Furthermore, non-linear methods provide parameters related to the quality of the solution and the confidence associated with the solution.

Path points coordinates to the world and the robot base coordinate systems, can now be found using of equations (8) and (9) respectively. Fig.4 and Fig.5 illustrate path point coordinates of the track shown in Fig. 2, based on the calibration parameters previously computed. If the Z axes of the world coordinate frame and the robot base coordinate frame were completely parallel ($r_{z(31)}=r_{z(32)}=0$

and $r_{z(33)}=1$) then the 3D track of Fig.5 would be parallel to XY plane and all points of the line would have a constant coordinate RobotBase_Z. In practice it is most common the Z-axes of the two coordinate frames to be almost parallel. For this reason, the plane where the track line is found in Fig.5 is not completely parallel with the XY plane of the world coordinate system and there is a small variation of the coordinate RobotBase_Z.

The robot base coordinates were tested on the PUMA 761 robot manipulator, where the end-effector tracked the path generated by the proposed method with high accuracy.

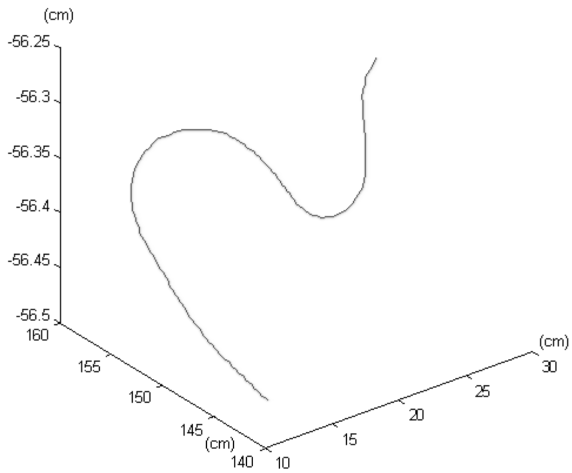

Fig 5. The track of Fig. 3 transformed into robot base coordinates (in cm).

VII. CONCLUSION

A new method of finding a robot path on a plane surface is presented. A single camera, mounted on the end-effector of the robot manipulator, was used as a sensor. The robot path points are calculated using an image of a static scene of the environment of the manipulator, where a desired track was appeared. The calculation of the final path points is the result of the combination of two well-known methods for camera calibration and for hand/eye – robot/world calibration. A software application written in C, using the previous two methods and simple image processing procedures is used to calculate the desired robot path points based on the image captured by the camera. With the proposed method, the desired robot path points can be found with high accuracy. The method was used to simulate a gantry system with a robotic manipulator.

REFERENCES

[1] Ales Ude and Rudiger Dillmann, "Trajectory Reconstruction from Stereo Image Sequences for Teaching Robot Paths," in *Proc. 24th Inter. Symp. Industrial Robots*, 1993, pp. 407-414.

[2] Ales Ude and Rudiger Dillmann, "Vision-Based Robot Path Planning," *Advances in Robot Kinematics and Computational Geometry, Kluwer,* 1994, pp.505-512.

[3] Ales Ude, Rudiger Dillmann, "Robot Motion Specification: a Vision-Based Approach," *Surveys on Mathematics for Industry,* vol. 5, pp. 109-131, 1995.

[4] B. Brunner, K. Arbter, G. Hirzinger and R. Koeppe, "Programming Robots via Learning by Showing in a Virtual Environment," in *Proc. Virtual Reality World'95, IDG Conferences and Seminars,* 1995, pp.63-72.

[5] T. Pachidis, J. Lygouras and P. Tsalidis, "A Graphical User Interface for the Initial Path Generation of a Robotic Manipulator for an Arc Welding System," in *Proc. WSEAS ICRODIC*, 2002. (Accepted)

[6] T. Pachidis, J. Lygouras, "A Pseudo Stereo Vision System as a Sensor for Real Time Path Control of a Robot," in *Proc. IEEE Instrumentation and Measurement Technology Conference*, 2002, pp.1589-1594.

[7] O. Faugeras and G. Toscani "The calibration problem in stereo," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, June 1986, pp. 15-20.

[8] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3 no. 4, pp. 323-344, Aug. 1987.

[9] S. J. Mayback and O. D. Faugeras, "A theory of self-calibration of a moving camera," *The International Journal of Computer Vision*, vol. 8 no. 2, pp. 123-152, Aug. 1992.

[10] O. Faugeras, T. Luong and S. Mayback, "Camera self-calibration: theory and experiments," in *Proc. 2nd ECCV*, May 1992, pp. 321-334.

[11] Z. Zhang. "A Flexible New Technique for Camera Calibration," *Technical report MSR-TR-98-71*, *Microsoft Research*, December 1998.

[12] Z. Zhang, "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations," IEEE Transactions on Robotics and Automation, vol. 22, no. 11, pp. 1330-1334, November 2000.

[13] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration", *IEEE Journal of Robotics and Automation* , vol. 5, no. 3, pp. 345-358, June 1989.

[14] H. Zhuang, Z. Roth and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation of the form AX=YB," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 549-554, August 1994.

[15] F. Park and B. Martin, "Robot sensor calibration: solving AX=XB on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717-721, October 1994.

[16] H. Zhuang, K. Wang and Z. S. Roth, "Simultaneous calibration of a robot and a hand-mounted camera," *IEEE Transactions on Robotics and Automation,* vol. 11, no. 5, pp. 649-660, October 1995.

[17] Fadi Dornaika and Radu Horaud, "Simultaneous robot/world and hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617-622, 1998.

[18] Jain, R. Kasturi, B. Schunck, *Machine Vision*. McGraw-Hill, 1995.

[19] W. Press, B. Flannery, S. Teukolsky and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.