A Graphical User Interface for the Initial Path Generation of a Robotic Manipulator for an Arc Welding System

THEODORE PACHIDIS*, JOHN LYGOURAS**,PHILIPPOS TSALIDES***
*Department of Electrical and Computer Engineering
Democritus University of Thrace
67100 Xanthi
GREECE
pated@mail.otenet.gr
**Department of Electrical and Computer Engineering
Democritus University of Thrace
67100 Xanthi
GREECE
ilygour@ee.duth.gr
***Department of Electrical and Computer Engineering
Democritus University of Thrace
67100 Xanthi
GREECE
ptsalidi@eyridiki.ee.duth.gr

Abstract: - In this paper a novel, graphical user interface, is presented. This interface can be used to generate a desired path of the end effector of a robotic manipulator. The path is selected by a static scene of the robot environment, manually either on line using an image capturing system or off line using a stored pair of images or a complex image. On line the system captures a pair of images using a stereovision system with one or two cameras or our pseudo stereovision system. A desired path could be generated by an edge or part of it of the scene image, a line manually designed to the image or a combination of lines of the previous cases. A user can initially process a pair of images selecting from pull down menus a variety of filters, edge detection methods and operations. Then the desired path as a combination of lines is selected from images. Applying our correspondence algorithm, corresponding edges can be found. Finally, a successive number of path points are calculated by means of the stereo system equations, the camera calibration parameters and the hand-eye transformation. In on line operation the capturing system mounted on the end effector can capture images with the desired best view of a scene by moving or rotating, using push buttons, the end effector of the robotic manipulator PUMA 761. Other facilities of the above system are the selection of a variety of colors and shapes, histogram view, magnification, automatic execution of user selected operations and system information. The interface is developed in Visual C++, it runs in a personal computer and communicate with the robot PUMA 761 via ALTER communication port.

Key-Words: - User interface, path generation, robotic manipulator, image processing, stereovision, facilities.

1 Introduction

A robotic manipulator could be used to an arc welding system. In a system like this many problems may be encountered. Some of them are 1) the accuracy of movement of a torch mounted on the end effector of the robotic manipulator, 2) the generation of a proper path which the torch must follow, 3) the flexibility of the system to adapt the specified path to any changes in shape of metal pieces to be welded, 4) the possibility of the system to change the movement rate, or the pattern of movement relatively to metal types or welding surfaces shape.

The above problems could better be manipulated if a stereovision system, mounted on the end effector of the robotic manipulator, parallel with the torch, will be used. A stereovision system could measure with better accuracy the desired path points and adapt an initially specified path to any changes of the metal pieces to be welded. In such system the initial path generation may be quite easy and fast with good accuracy. The previous features will permit the welding of different in shape or in material metal

pieces and to repeat the procedure when it is necessary.
Some researchers in relevant papers propose methods for the initial path generation. In [1-3], Ales Ude and Rudiger Dillman employed the “Teaching by Showing” programming paradigm. The user specified the desired trajectory by moving the object to be manipulated with his hand. The performance was measured with a stereovision system. In [4], B. Brunner et al. proposed the so-called TeleSensorProgramming concept as an easy way to program a robot off-line via learning by showing in a virtual environment. R. M. Voyles and P.K. khosla in [5] proposed a gesture-based programming method of a robot. In [6], Johnson and Marsh presented the basic mathematical and computational framework for a new method of modeling robot manipulator workspaces. In his thesis [7], Sabes studied the planning of visually guided arm movements in two cases: feedback perturbation and obstacle avoidance. In [8], Li Fang Gu studied the visual guidance of a robot by means of a stereovision system and using as features corner points. Paschke and Pauli in [9] implemented the Programming by Demonstration method to reconstruct a smooth 3D trajectory of a gripper, using a stereovision system. In [10], Ruf and Horaud proposed a methodological framework for trajectory generation in projective space. In their paper [11], Zha and Du presented a new approach to the generation and optimization of the position and orientation trajectories in Cartesian task space. Simulation was made in a virtual CAD-based off-line programming environment.
In this paper, a graphical user interface for the initial path generation of a robotic manipulator is proposed. Using this interface an initial path could be calculated quite easy and fast and with good accuracy. The input of the system is a pair of stereo images captured by means of a stereoscopic vision system or a complex image captured by means of our pseudo stereoscopic vision system (PSVS). A desired path is derived by the current scene as a straight or curved edge or combination of edges or user-designed lines connecting i.e. the torch with an object of the scene. For better accuracy in calculations, the selection of the best view of the desired scene is possible by translating and rotating the end effector using push buttons and a user specified step of translation or rotation.
The interface can be used to any other application where the extraction of a desired path is required.
This paper organized as follows. In section 2, the basic concepts used to the software application, as well as theoretical notes of them, are briefly explained. In section 3, the description of the interface is presented. In section 4 experimental results of 3D paths are presented. Finally the conclusion of this paper is presented in section 5.

2 Basic concepts

2.1 Pseudo stereo Vision System (PSVS)

In the presented graphical user interface, as a capturing system an ordinary stereovision system with two cameras, or our pseudo stereovision system (PSVS), can be used.

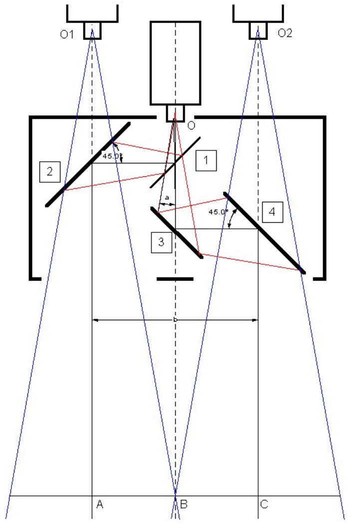


Fig.1 PSV system, mirrors and virtual cameras.

Details for this system are presented in [12]. It is a monocular stereovision system composed of a camera and four mirrors (Fig.1). The first mirror is a thin 50 % beam splitter. The other mirrors are front cover mirrors. Using PSVS a cheaper system is created combining many good features. These are: 1) Creation of two virtual cameras with the same geometric properties, field of view and active focal length, 2) The angular FOV of the apparatus is the same with of the real camera, 3) It has the accuracy of the stereo vision system, 4) It captures a complex image in a single slot. This complex image is composed of the left and right image of an ordinary vision system and consequently this complex image can faster be processed.
The equations giving the coordinates of a point P(x,y,z) in space are:

$$x = \frac{xr}{f} \cdot (z + \frac{b}{2}) + \frac{b}{2} \qquad (1)$$

$$y = \frac{yr}{f} \cdot (z + \frac{b}{2}) \qquad (2)$$

$$z = \frac{b \cdot f}{xl - xr} - \frac{b}{2} + \frac{xl \cdot l}{xl - xr} + \frac{f \cdot m}{xl - xr} \quad (3)$$

The camera coordinate system has as origin the optical center O of the real camera and its z-axis coincides with the optical axis from O. In equations (1)-(3), f is the active focal length of the cameras, b is the baseline length and xr, yr, xl, are coordinates of a point in the image plane. Parameters l and m are related with refraction phenomena due to mirror (1). If mirror (1) is too thin (i.e. Pennicle beam splitter, with thickness 2μm) then m=0, l=0, and the above equations are converted to the ordinary stereovision equations.

2.2 Correspondence Algorithm

The algorithm is described in [12]. The algorithm is based on the concept of seeds. In our application some edges in an image are selected and then the corresponding edges to the second image must be found. By using any correspondence algorithm each corresponding pixel of an edge could be found with some computational cost. Our idea is to find only a small number of pixels (called seeds) of the corresponding edge and then to propagate these seeds to whole edge. The propagation is a passive procedure with very small computational cost. Thus the whole procedure is much faster and using predefined conditions (i.e. processing of a small number of lines in each image) could be used to real time applications. The constraint used for the correspondence either with an ordinary stereo system or with PSVS is only scanning in a scan line each time. So, the algorithm initially finds in each line all the edge pixels and calculates their distances from the initial pixel. In a matrix with counters, the number of reoccurrence of the same distance is stored. When the scanning of the image is finished, as seed pixels are selected those pixels with distance from the initial pixels, equal to the distance of the maximum reoccurrence. The algorithm with the above procedure could found less than the predefined number of pixels but during propagation all edge pixels will be found.

The results, namely, the color of points, their image plane coordinates and the disparity are stored to a file.

2.3 Path Points Calculation Algorithm

Calculation of coordinates of path points involves two stages: 1) the rearrangement of desired pixels so that to be successive path points, and 2) necessary calculations to each pair of corresponding pixels to give the 3D point coordinates. (Fig.2).

2.3.1 Algorithm Description

The objective of the proposed algorithm is to rearrange pixels found through correspondence algorithm, to select the specified number of pixels and to calculate the final path points.

In determining a path, it is desirable, to have a start and an end point. Thus, the first step of this algorithm is to find endpoints of each colored edge as well as cross points. The algorithm could be found points of interest in almost every type of edges (i.e. a line, a closed line, a line with cross points, combination of previous cases). The second step is the rearrangement of pixels between two endpoints of a line each time (end or cross points). The rearrangement is made for all path pixels. The third step is the selection of the specified number of pixels. It must be noted that this number is user selected through an edit box. The selected pixels have equal distances. The forth step is the following. For each pair of pixels, the calculation of the coordinates of the real path point is made. The calculations are presented to the next subsection. The points are designed temporary to the screen as small circles with center the specified point. Then the results are appearing to the list box of the interface and saved automatically to a file as a set of six values per line: x, y, z, yaw, pitch, roll (yaw, pitch, roll are the Euler angles).

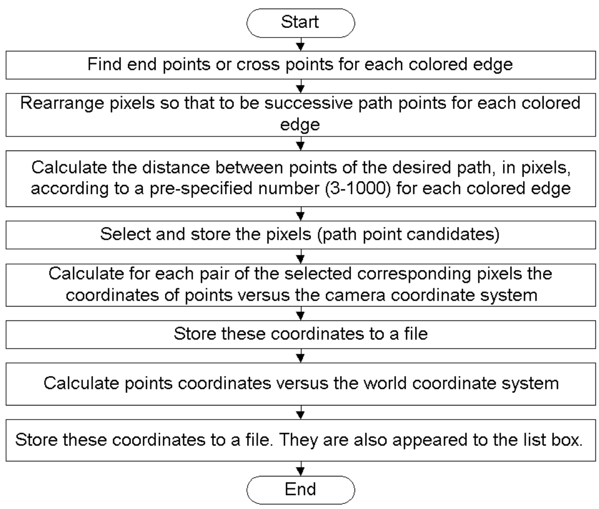


Fig.2 Block diagram of the path points calculation algorithm.

2.3.2 Theoretical Analysis

To calculate coordinates of path points versus the world coordinate system of the robotic manipulator (Fig.3) must be found:

a) The intrinsic parameters as well as the distortion coefficients of the camera. Here, the calibration method of Z. Zhang [13] has been used while for corner detection the algorithm proposed in [14] has been preferred. The transformation matrix with the intrinsic camera parameters is:

$$A = \begin{bmatrix} a_u & c & u_0 \\ 0 & a_v & v_o \\ 0 & 0 & 0 \end{bmatrix} \tag{4}$$

Where u_o, v_o, are the estimated coordinates of the image center in pixels, a_u, a_v are the estimated scale factors in image u and v axes, in pixels again and c the parameter describing the skew of the two image axes.

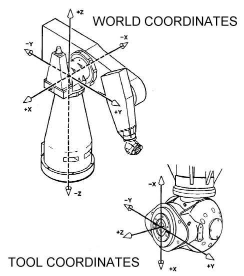


Fig.3 World and tool coordinate system of the PUMA 761 robotic manipulator.

b) The transformation matrix for the hand eye calibration. This matrix is calculated using the linear method in [15]. It is permanently stored to a file and periodically recalculated using the previous method and our interface facilities. The homogeneous matrix is of the form:

$${}^{F}\boldsymbol{T_C} = \begin{bmatrix} {}^{F}\boldsymbol{R_C} & \boldsymbol{X_C} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix} \tag{5}$$

c) Finally the homogeneous matrix which gives the tool coordinates versus the world coordinate system must be derived. This matrix can be found dynamically via the ALTER communication port and after a proper scaling. This matrix is of the following form:

$${}^{W}\boldsymbol{T_F} = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{s} & \boldsymbol{a} & \boldsymbol{X_F} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Using matrix A and pixel coordinates for calculations, equations (1)-(3) can be converted to the following:

$$x_P = \frac{u_r - u_{ro}}{a_u} \cdot (z_P + \frac{b}{2}) + \frac{b}{2} \tag{7}$$

$$y_P = -\frac{v_r - v_{ro}}{a_v} \cdot (z_P + \frac{b}{2}) \tag{8}$$

$$z_P = \frac{b \cdot a_u + (u_l - u_{lo}) \cdot l + a_u \cdot m}{(u_l - u_{lo}) - (u_r - u_{ro})} - \frac{b}{2} \tag{9}$$

The coordinates of a point P(x,y,z), as well as the Euler rotation angles yaw, pitch, roll, versus the world coordinate system can be derived by multiplication of the following matrices. That is:

$${}^{W}\boldsymbol{T_P} = {}^{W}\boldsymbol{T_F} \cdot {}^{F}\boldsymbol{T_C} \cdot {}^{C}\boldsymbol{T_P} \tag{10}$$

Where ${}^{C}\boldsymbol{T_P}$ is the matrix for the transformation of a coordinate system with origin the point P versus to the camera coordinate system. This matrix is of the form:

$${}^{C}\boldsymbol{T_P} = \begin{bmatrix} {}^{C}\boldsymbol{R_P} & \boldsymbol{X_P} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix} \tag{11}$$

The matrix ${}^{W}T_P$ is of the form:

$${}^{W}\boldsymbol{T_P} = \begin{bmatrix} {}^{W}\boldsymbol{R} & \boldsymbol{X} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix} \tag{12}$$

Where the vector $\boldsymbol{X}=[x, y, z]^T$ provides the point coordinates versus the world coordinate system. From the matrix ${}^{W}\boldsymbol{R}$ the Euler angles yaw(ψ), pitch(θ), roll(φ) can be calculated for each point using the following relations:

$$\varphi = \tan^{-1}\left(\frac{r_{21}}{r_{11}}\right) \tag{13}$$

$$\theta = \tan^{-1}\left(-\frac{r_{31}}{\cos\varphi \cdot r_{11} + \sin\varphi \cdot r_{21}}\right) \tag{14}$$

$$\psi = \tan^{-1}\left(\frac{\sin\varphi \cdot r_{13} - \cos\varphi \cdot r_{23}}{\cos\varphi \cdot r_{22} - \sin\varphi \cdot r_{12}}\right) \tag{15}$$

If $r_{11} \rightarrow 0$ and $r_{21} \rightarrow 0$, then $\varphi=0$, $\theta=\tan^{-1}(-r_{31}/r_{11})$ and $\psi=\tan^{-1}(-r_{23}/r_{22})$.

3 Interface Description

The interface was developed in Visual C++. It is part of an application where the communication with PUMA 761 robotic manipulator through ALTER serial port and many other operations like camera calibration, hand–eye calibration, image processing are made.

In Fig.4, the basic form of the interface is illustrated.

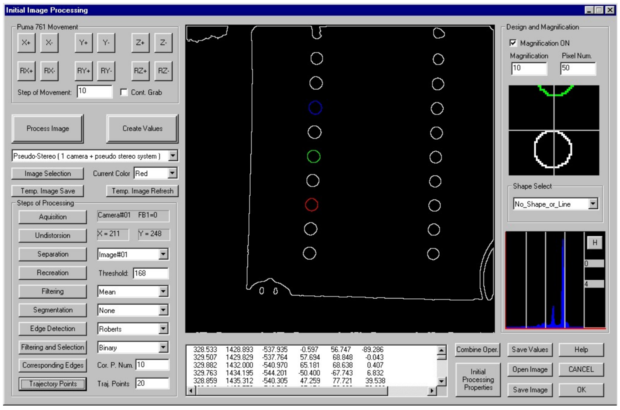


Fig.4 Basic Interface form.

The basic steps of generating initially a desired path are:

1. Using the interface, the end effector is moved to obtain the best view of the scene.
2. The capturing system being used is selected and a proper image or pair of images is captured.
3. The desired filters and operations to process images are selected through pull down menus and edit boxes.
4. The processing procedure is executed step by step or automatically pressing the corresponding push buttons.
5. If the final edge image is the desirable we continue with the selection or the design of the proper edges in the left view of the complex image or in the right image.
6. The maximum number of pixels that could be found to the second image with our correspondence algorithm is selected and the function "Corresponding edges" to find the corresponding edges to the second image is executed. This number is in the range 2 to 100. After the execution the results could be checked selecting the second image or through the list box or having a look to the file where the results are stored.
7. The density of the final path points for each colored edge is determined by changing the value to the edit box "Traj. Points". This number could be from 3 to 1000. Then the function "Trajectory Points" is executed. When the calculations will finish the ordered path points are temporary appeared to the screen and sets with the translation and rotation values for each point are appearing to the list box. The results are also stored to a file.
8. By means of the function "Save Values" previous sets are loaded to a matrix and can be used as the source data to control the robotic manipulator, in real time.

The top and left side of this form is used for the position and rotation control of the end effector. The facility was incorporated to the interface because the selection of the best view of the scene permits the calculation of the desired path points with better accuracy (smaller distance, better slope of an edge).

Continuing to the left side, two modal frame push buttons can be used for the automated calculation of the path in two steps, when proper filters and operations for the pair of images have been selected. The selection of the capturing system is made using the pull down menu which it follows previous push buttons. The selection of the desired color is made using the related pull down menu. In the frame with title "Steps of Processing" two columns with push buttons and boxes are illustrated. The push buttons are related with the execution of the corresponding functions. A user can select the proper processing operations by changing box values (edit boxes, static boxes or pull down menus).

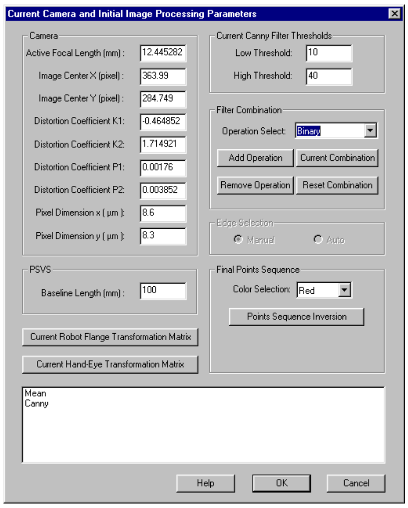


Fig.5 Auxiliary form.

After the completion of the image processing, edges selection is possible by selecting for each edge a different color. The selection is made using the cursor (by planting a seed or selecting an area). It is easy by means of a small window in the right corner of the basic form, where the magnification of the small area around the cursor is possible. Using the shape select pull down menu facilities (right and middle), a user can design extra lines, thus connecting existing paths.

Using this graphical user interface an image can be opened or saved to a media for permanent storage. The histogram of the initial gray scale image is appeared in a window to the right side of the form. In each step of processing the current processed image is appeared in the center of the form as a 512X512 image.

To specify some properties and to check the current settings the auxiliary form of Fig.5 is used.

In this form a user can specify a combination of filters and operations with a desired order and then to process an image just executing the function in the basic form “Combine Oper.”. In the auxiliary form the current camera settings after a previous calibration, the current world-flange transformation matrix, as well as the hand-eye calibration, are presented. The related transformation matrices are stored to files and are loaded during the initialization of the application or are loaded in each execution of the related function.

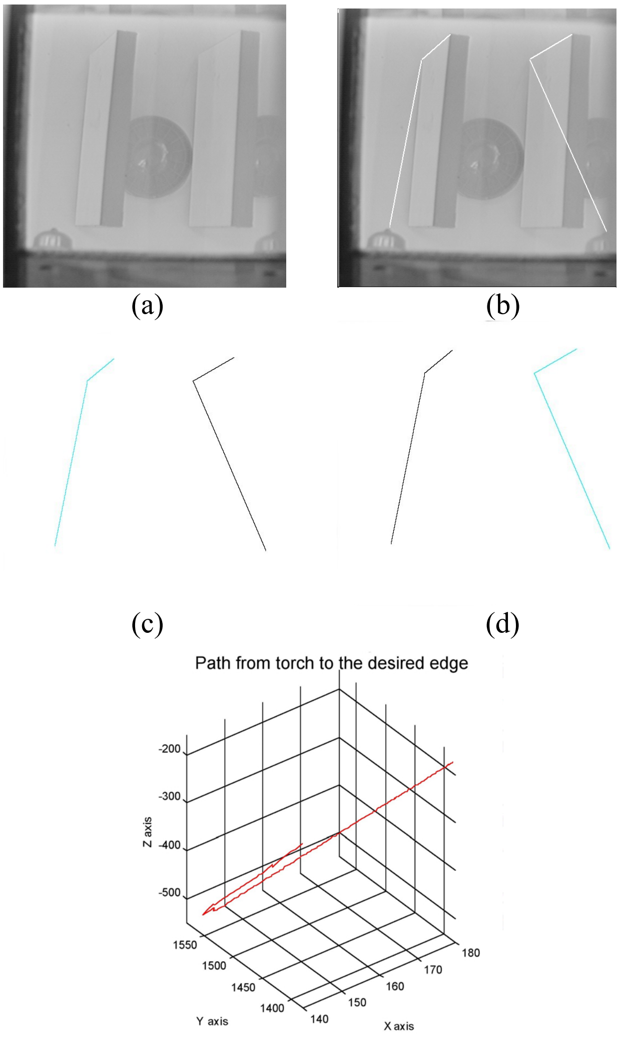
4 Experimental Results

In this section experimental results are presented. A Pulnix TM-520 camera connected with the F64-PCI-DSP frame grabber card of Coreco, was used. The PSVS is mounted to the end effector of PUMA 761 robotic manipulator. A personal computer, Pentium type, was used for the execution of the application.

In Fig.6 the processing stages of a complex image of the scene are presented. In this example, manually designed lines, connect the arc-welding torch (a simulated torch) mounted on the end effector of the robotic manipulator with a desired edge of the 3D object. Before the execution of correspondence, rearrangement and calculation algorithms, the initial image Fig.6 (a) is subtracted from image of Fig.6 (b). Thus an image with only the desired edges is created. The calculated 3D points have a specified order from the torch to the 3D object desired edge. In this way, a path composed of these points is generated.

To indicate the specified order of 3D points, an image of 2D objects was captured (Fig.7 (a)). After the processing (mean filter, binary image with threshold T=175, Roberts edge detector) the path on the selected objects is illustrated as continue line, by means of a 3D map in Fig.7 (b).

To show the accuracy of the measurement system the 30x30 mm square of Fig.7 (a) is used. The depth map of the square points and a 3D map of points as they are calculated versus the world coordinate system, are illustrated in Fig.8 (a) and (b) respectively.

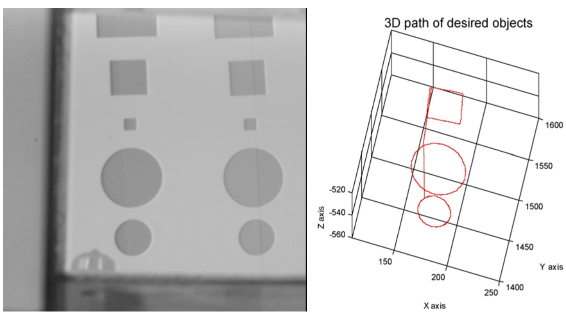


(a) (b)

(c) (d)

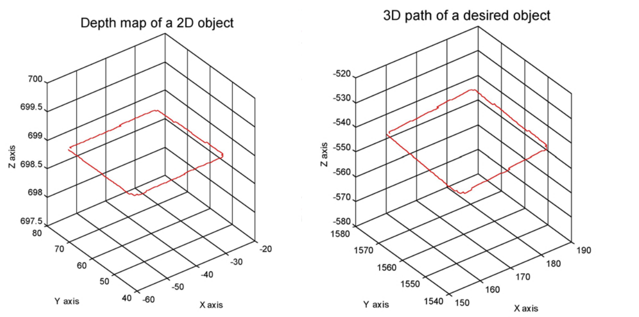
(e)

Fig.6 a) Initial image, b) design of the desired edges, c) left view selection d) right view automatically selected after the application of the correspondence algorithm, e) 3D map of path points.



(a) (b)

Fig.7 a) The initial image, b) the 3D path of three selected objects.



(a) (b)

Fig.8 3D maps of a 30x30 mm square a) versus camera coordinate system, b) versus world coordinate system.

5 Conclusion

In this paper a novel graphical user interface for the initial path generation of a robotic manipulator, is presented. The whole application is based on some new concepts. These concepts permit the generation of a path easier, reliable and faster using simply a personal computer. As a capturing system of scene images can be used an ordinary stereovision system or our PSV system. A path can be pre-calculated, not using the method “Teaching by Showing” but simply designing lines and shapes or connecting edges of objects in the scene. The proposed algorithms in the application for point correspondence and then points rearrangement, permit the generation of successive path points. As the vision system is mounted on the end effector of the robotic manipulator, moving the robot, the best view of the scene can be captured and then the desired path points to be calculated with higher accuracy. This graphical user interface has many facilities making it user friendly.

References:

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

[2] Ales Ude, 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, 1995, pp. 109-131.

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

[5] R. Voyles, P. K. Khosla, Gesture-based Programming, Part I: A Multi-Agent Approach, *Proc ANNIE: Artificial Neural Networks, Fuzzy Logic and Evolutionary Programming for Designing Smart Engineering Systems,* ASME Press, 1996, pp. 299-304.

[6] C. G. Johnson, and D. Marsh, Modeling Robot Manipulators in a CAD Environment Using B-splines, *N.G. Bourbakis, editor, IEEE International Joint Symposia on Intelligence and Systems,* 1996, pp. 194-201.

[7] P. N. Sabes, *The Planning of Visually Guided Arm Movements: Feedback Perturbation and Obstacle Avoidance Studies,* PHD Thesis, MIT, 1996.

[8] L. F. Gu, *Visual Guidance of Robot Motion,* Master Thesis, University of Western Australia, 1996.

[9] M. Paschke, J. Pauli, Vision Based Learning of Gripper Trajectories for a Robot Arm, *International Symposium on Automotive Technology and Automation,* 1997, pp.235-242.

[10] A Ruf, R. Horaud, Visual Trajectories from Uncalibrated Images, *Workshop on New Trends in Image-Based Robot Servoing, IEEE/RSJ Int. Conf. On Intelligent Robots and Systems,* 1997, pp. 83-91.

[11] X. F. Zha, H. Du, Generation and Simulation of Robot Trajectories in a Virtual CAD-Based Off-line Programming Environment, *Int. J. Adv. Manuf. Technology, Springer-Verlag,* 2001, pp.610-624.

[12] T. Pachidis, J. Lygouras, A Pseudo Stereo Vision System as a Sensor for Real Time Path Control of a Robot, *IEEE Instrumentation and Measurement Technology Conference,* 2002. (Accepted)

[13] Z. Zhang, A Flexible New Technique for Camera Calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* VOL.22, NO.11, 2000, pp.1330-1334.

[14] T. Pachidis, J. Lygouras, V. Petridis, A Novel Corner Detection Algorithm for Camera Calibration and Calibration Facilities, *2nd WSEAS Int. Conf. on Signal Processing and Computational Geometry and Vision,* 2002. (Accepted)

[15] H. Zhuang, Z. Roth, R. Sudhakar, Simultaneous Robot/World and Tool/Flange Calibration by Solving Homogeneous Transformation Equations of the Form AX=YB, *IEEE Transactions on Robotics and Automation,* VOL.10, NO.4, 1994, pp. 549-554.