

Pseudo Stereovision-based Path Generation Method for a Robotic Arc Welding System

THEODORE PACHIDIS*, JOHN LYGOURAS**
Department of Electrical and Computer Engineering
Democritus University of Thrace
Vas. Sofias 12 – 67100 Xanthi
GREECE
*pated@mail.otenet.gr, **ilygour@ee.duth.gr

Abstract: - In this paper a vision-based path generation method for a robotic arc welding system, is presented. The vision system used is the Pseudo Stereovision System (PSVS) that captures a complex image in a single shot. A part of the recently developed *HumanPT* robotic software application permits to implement the algorithms proposed. A desired welding path can be generated, not using a “Teaching by Showing” like method, but using a complex image captured by means of PSVS from a scene of the robotic environment. From this complex image, one or more edges or parts of them, lines manually designed and or a combination of lines of the previous cases, as well as lines in complex images of successive scenes can be used to generate the path that a torch follows. By means of the graphical user interface (GUI) provided, a user can initially process 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. The correspondence algorithm proposed can be implemented in complex images and it is based on seeds. Finally, a successive number of equidistant path point locations are calculated by means of the path points calculation algorithm also proposed. In on line operation, PSVS mounted on the end-effector can capture complex images with the desired best view (welding view) of a scene permitting this way to calculate the desired path point locations with better accuracy (smaller distance, better slope of an edge). The application is developed in Visual C++, it runs in personal computers under Windows as operating system and communicates with PUMA 761 robot through ALTER communication port at 38400 bps.

Key-Words: - Pseudo Stereovision System, *HumanPT* robotic application, complex image processing, ALTER communication, correspondence, arc welding.

1 Introduction

A robotic manipulator can be used in an arc welding system. In a system like this many problems may be encountered. Some of them are 1) the movement accuracy of a torch mounted on the end-effector of the robotic manipulator, 2) the generation of a proper path the torch must follows, 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 like PSVS, mounted on the end-effector of the robotic manipulator together with the torch, might be used. A stereovision system can measure with better accuracy than other systems the desired path points and adapt an initially specified path to any changes of the metal pieces to be welded. In such a system the initial path may be

generated quite easy and fast. 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 specifies the desired trajectory by moving the object to be manipulated with his hand. The performance is 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 is made in a virtual CAD-based off-line programming environment.

In this paper, an integrated system for the initial path generation of a robotic welding system is proposed. The input of the system is a complex image captured by means of our pseudo stereovision system (PSVS) [12-13]. It differs than other vision-based methods ("Teaching by Showing" like methods) because only a complex image of a static scene can be used to generate a robotic path. A path can be calculated quite easy, fast and with good accuracy. A desired path is generated by using a straight or curved edge, or combination of edges belonging to the complex image, or user-designed lines connecting edges and feature points. For better accuracy in calculations, the selection of the best view of the desired scene is possible by translating and rotating the end effector (and the attached on it PSVS) by means of the GUI (part of the recently developed *HumanPT* robotic application). By combining the results of more complex images the desired path for a procedure can be generated. The whole system can be used to any other application where the extraction of a desired path is required. *HumanPT* application is executed in personal computers under Windows as operating system and exploits the only feature that Unimation provides for real time (RT) communication with PUMA 761 robot, the ALTER communication port. The communication is stable by increasing the baud rate at 38400 bps (The standard baud rate proposed by Unimation is 19200 bps).

This paper is organized as follows. In section 2, the basic concepts used to the software application, as well as theoretical notes, are briefly explained. In section 3, the basic steps of the algorithm proposed are presented. In section 4 experimental results are given. Finally the conclusions are presented in section 5.

2 Basic concepts

2.1 Pseudo stereo Vision System (PSVS)

In the GUI of the presented system, as a capturing system, an ordinary stereovision system with two cameras, or PSVS, can be used.

Details for PSVS are presented in [12-13]. 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 first surface 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 stereovision system, 4) It captures a complex image as the result of superposition of the left and the right view, in a single shot. This complex image can faster be processed.

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 (Fig. 1).

To calculate coordinates of path points with respect to camera coordinate system, the intrinsic parameters as well as the distortion coefficients of the camera must be found. Here, the calibration method of Z. Zhang [14] has been used while for corner detection the algorithm proposed in [15] has been preferred.

We define the following symbols:

$u_0, v_0, u_{0R}, v_{0R}, u_{0L}, v_{0L} \rightarrow$ Estimated coordinates of the image center in pixels of the real, the right (virtual camera 1) and the left camera (virtual camera 2) respectively.

$a_u, a_v, a_{uR}, a_{vR}, a_{uL}, a_{vL} \rightarrow$ Estimated scale factors in image u and v axes, in pixels again of the real, the right (virtual camera 1) and the left camera (virtual camera 2) respectively.

$c \rightarrow$ Parameter describing the skew of the two image axes.

$u_R, v_R, u_L, v_L \rightarrow$ Estimated coordinates of a 3D point in the image plane of the right (virtual camera 1) and the left camera (virtual camera 2) respectively.

$b \rightarrow$ The baseline length.

$m \rightarrow$ Parameter describing the parallel displacement of the virtual optical axis from O_2 along X axis due to refraction phenomena in mirror (1).

$l \rightarrow$ Parameter describing the shifting of the optical center O_2 along the optical axis (Fig. 1) due to refraction phenomena in mirror (1).

$\omega_1 = 90^\circ - \theta_i \rightarrow$ Angle forms the optical axis with a mirror plane (θ_i is the incidence angle).

$\theta_r \rightarrow$ Refraction angle (Fig. 1).

$d \rightarrow$ Thickness of mirror (1).

The transformation matrix with the intrinsic camera parameters is:

$$A = \begin{bmatrix} a_u & c & u_0 \\ 0 & a_v & v_0 \\ 0 & 0 & 0 \end{bmatrix} \quad (1)$$

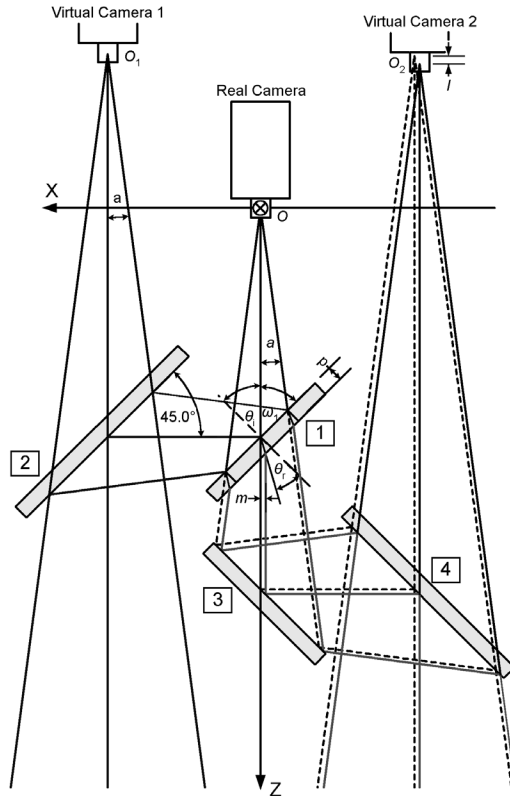


Fig. 1 PSVS mirrors arrangement and virtual cameras are presented. The displacement of the second virtual camera due to refraction phenomena of mirror (1) is also illustrated.

Equations giving point coordinates $P(x_p, y_p, z_p)$ in 3D space with respect to camera frame are the following:

$$x_p = \frac{1}{a_u} \cdot \left(z_p + \frac{b}{2} \right) \cdot (u_R - u_{0R}) + \frac{b}{2} \quad \text{or} \quad (2)$$

$$x_p = \frac{1}{a_u} \cdot \left(z_p + \frac{b}{2} - l \right) \cdot (u_L - u_{0L}) - \frac{b}{2} - m$$

$$y_p = \frac{1}{a_v} \cdot \left(z_p + \frac{b}{2} \right) \cdot (v_R - v_{0R}) \quad \text{or} \quad (3)$$

$$y_p = \frac{1}{a_v} \cdot \left(z_p + \frac{b}{2} - l \right) \cdot (v_L - v_{0L})$$

$$z_p = \frac{a_u \cdot (b + m) + l \cdot (u_L - u_{0L}) - b}{(u_L - u_{0L}) - (u_R - u_{0R})} - \frac{b}{2} \quad (4)$$

The constant m is equal to:

$$m = \frac{d \cdot \cos(\omega_1 + \theta_r)}{\cos \theta_r} \quad (5)$$

The constant l is equal to:

$$l = \frac{d}{\cos \theta_r} \cdot (1 - \sin(\omega_1 + \theta_r)) + \frac{d}{\cos \theta_r} \cdot \cos(\omega_1 + \theta_r) \cdot \tan \omega_1 \quad (6)$$

For mirror (1) thickness equal to 1mm (as in PSVS), $m=0.329$ mm and $l=0.377$ mm (refraction indices, $n_{\text{glass}}=1,5$ and $n_{\text{air}}=1$).

2.2 Correspondence Algorithm

The algorithm used is described in [12]. It belongs to high-level feature-based algorithms and particularly to algorithms that can find correspondences in curves [18-20]. It is based on the concept of seeds and it is implemented either in complex or in a stereo pair of images. In the first stage, the algorithm finds only a small number of pixels (called seeds) of the corresponding edge (sub-sampling) and then propagating these seeds as 4-neighbor pixels creates the edge. The propagation is a passive procedure with smaller computational cost. Thus the algorithm is faster than other algorithms and using predefined conditions (i.e. processing of a small number of lines in each image) it can be used in 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. In the novel form of the algorithm three criteria are used to select the corresponding edge:

1. A candidate corresponding edge must have the same horizontal upper and lower limits of the initial edge with probably a small predefined permissible deviation measured in pixels.
2. A candidate corresponding edge must have the same number of pixels of the initial edge extended by a predefined percentage of the initial number of pixels.
3. The most probable candidate edge corresponds to the maximum population of pixels at the same distance from the initial image. At this distance, at least one pixel of the candidate edge is detected. (Criterion of the most probable edge).

A number of different criteria have been studied. It was found that the above three criteria permit a more reliable detection of a corresponding edge. After the corresponding edges have been created

(detection and propagation), the corresponding points in each pair of edges can be found by mapping the points one by one. In this second stage, the density of points, that is, the number of corresponding points in a pair of edges is determined by means of a pre-specified number (N_C). The corresponding pairs of points are equally distributed and their locations and disparities are stored in a matrix. Using this algorithm to the proposed method of path generation the selection and processing of 11 independent edges with different colors, in the same scene, is possible. The results, namely, the color of points, their image plane coordinates and the disparities are stored to a file. Thus from this processing step the different behavior of the robot end-effector and the torch mounted on it can be determined for each part of a desired robotic path (i.e. part in which the robot is moving with constant velocity or part for welding in which the torch is activated).

2.3 Path Points Calculation Algorithm

Coordinates calculation 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. The algorithm is described by the flowchart of 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, first the algorithm finds endpoints of each colored edge as well as cross points. The algorithm can find 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). Then the rearrangement of pixels between two endpoints (end or cross points) of a line each time follows. The rearrangement is made for all path pixels. The next step is the selection of the number N_T of the desired points in each edge. This number is user selected through an edit box. The selected pixels have equal distances. Then for each pair of pixels, 3D points coordinates of the desired path are calculated. The calculations are presented in the next subsection. The points are designed temporary to the screen as small circles with center the specified point. The results appear to the list box of the GUI and are saved automatically to a file as a set

of six values per line: $x, y, z, \psi, \theta, \phi$ (ψ, θ, ϕ are Euler angles yaw, pitch, roll respectively).

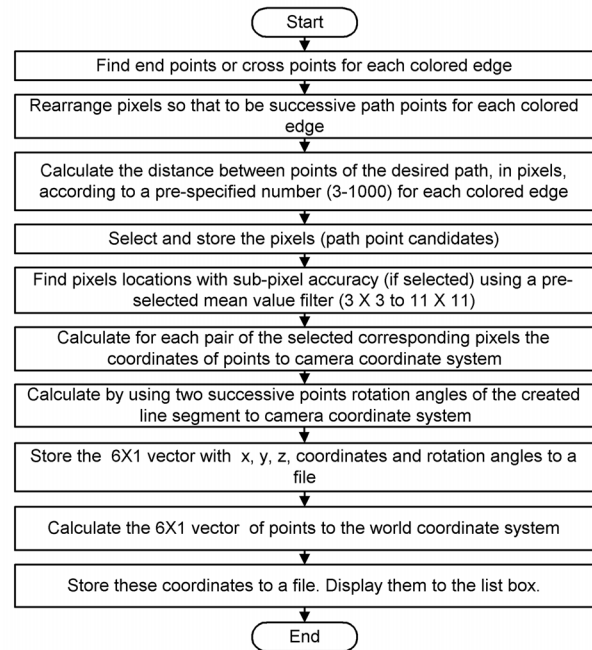


Fig. 2 Block diagram of the path point's calculation algorithm.

2.3.2 Theoretical Analysis

In order to calculate the vector $X(x, y, z, \psi, \theta, \phi)$ with respect to the world coordinate system of the robotic manipulator (robot base frame), except point coordinates to camera coordinate system, the rotation matrix for a coordinate system attached to each point might be found. We introduce a procedure, better adapted to the requirements of the proposed method for a robotic arc welding system. The method is supported by PSVS but can also be used with an ordinary stereovision system with parallel cameras. Assuming that the desired edge is a line segment defined by two points, $P_1(x_1, y_1, z_1)$ as the start point and $P_2(x_2, y_2, z_2)$ as the end point, the three angles roll (ϕ_c), pitch (θ_c), yaw (ψ_c) with respect to camera coordinate system might be calculated. The objective is to find rotation angles around X_C, Y_C, Z_C axes of the camera coordinate system so that Y_C axis to be parallel to the line segment. By definition, in sequence, angle yaw represents rotation around X_C -axis, angle pitch rotation around Y_C -axis of the once-rotated system and finally, angle roll rotation around Z_C axis of the twice-rotated system. The 3x3 rotation matrix will be $R(\psi_c, \theta_c, \phi_c) = R(\phi_c) \cdot R(\theta_c) \cdot R(\psi_c)$.

To calculate angle yaw(ψ_C), the line segment is projected to $Y_C Z_C$ plane. This projection forms an angle with Y_C -axis, which represents the angle, the camera coordinate system rotates around X_C -axis, so that the projection to be co-linear with Y_C -axis, and as a consequence the line segment to be found on $X_C Y_C$ plane. This angle is equal to:

$$\tan \psi_C = \frac{z_2 - z_1}{y_2 - y_1} \Rightarrow \psi_C = \tan^{-1} \frac{z_2 - z_1}{y_2 - y_1} \quad (7)$$

Case studies: If $|z_2 - z_1| < \varepsilon \Rightarrow \psi_C = 0^\circ$, where $\varepsilon > 0$ is a small real number. In this case, the projection of a line segment to image plane through the PSVS (reception of a complex image) is always two parallel lines. In any other case these lines are not parallel.

Angle pitch can be found by the projection of the line segment to the $X_C Z_C$ plane as it arises of the once-rotated system. This projection is always co-linear with X_C -axis and angle pitch is taken by definition equal to zero, $\theta_C = 0$.

Finally to calculate angle roll, the line segment is projected to $X_C Y_C$ plane. This projection forms an angle with Y_C -axis, which represents the angle, the camera coordinate system have to rotate around Z_C -axis, so that the line segment to be co-linear with Y_C axis. Thus the angle roll is calculated from the relation:

$$\tan \varphi_C = \frac{x_2 - x_1}{y_2 - y_1} \Rightarrow \varphi_C = \tan^{-1} \frac{x_2 - x_1}{y_2 - y_1} \quad (8)$$

Case studies: If $|x_2 - x_1| < \varepsilon \Rightarrow \varphi_C = 0^\circ$, where $\varepsilon > 0$ is a small real number. In this case, the projection of a line segment to image plane through the PSVS (reception of a complex image) is always two parallel lines parallel to Y_C axis. The rotation matrix arising by using these angles is the following:

$${}^C \mathbf{R}_P = \begin{bmatrix} c\varphi_C & -s\varphi_C c\psi_C & s\varphi_C s\psi_C \\ s\varphi_C & c\varphi_C c\psi_C & -c\varphi_C s\psi_C \\ 0 & s\psi_C & c\psi_C \end{bmatrix} \quad (9)$$

Where, $c\psi_C \rightarrow \cos \psi_C$, $c\varphi_C \rightarrow \cos \varphi_C$, $s\psi_C \rightarrow \sin \psi_C$, $s\varphi_C \rightarrow \sin \varphi_C$.

Knowing these angles, PSVS and the welding torch mounted on the flange of the robotic manipulator can be rotated around the three axes of the world coordinate system (by means of the GUI developed for this purpose) obtaining in this way the best view, which is the desired welding view, of the desired path. Thus, during the welding procedure, the welding torch will track the desired path with the

desired orientation. To find the final homogeneous matrix to the world coordinate system the next two steps are necessary.

1. To calculate the homogeneous transformation matrix ${}^F \mathbf{T}_C$ (solution of the hand eye calibration problem). This matrix is calculated by using the linear method in [16]. It is permanently stored to a file and periodically recalculated using the previous method and our *HumanPT* application facilities.

2. To create the homogeneous matrix ${}^W \mathbf{T}_F$ providing the flange coordinates to the world coordinate system. This matrix can be found dynamically through ALTER communication port working in our system at 38400 bps and after a proper scaling. It is available by the system each 28 ms (operation cycle of the PUMA robot).

The homogeneous transformation matrix ${}^W \mathbf{T}_P$ of the frame attached on a point $P(x,y,z)$ with respect to the world coordinate system can be derived by multiplication of matrices as follows:

$${}^W \mathbf{T}_P = {}^W \mathbf{T}_F \cdot {}^F \mathbf{T}_C \cdot {}^C \mathbf{T}_P \quad (10)$$

Where ${}^C \mathbf{T}_P$ is the matrix transforming the coordinate system with origin the point P to the camera coordinate system. This matrix is of the form:

$${}^C \mathbf{T}_P = \begin{bmatrix} {}^C \mathbf{R}_P & \mathbf{X}_P \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (11)$$

Where $\mathbf{X}_P = (x_P, y_P, z_P)$. ${}^W \mathbf{T}_P$ matrix is of the form:

$${}^W \mathbf{T}_P = \begin{bmatrix} {}^W \mathbf{R}_P & \mathbf{X}_{3xl} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (12)$$

Vector $\mathbf{X}_{3xl} = [x, y, z]^T$ provides point coordinates with respect to the world coordinate system. ${}^W \mathbf{R}_P$ matrix has the form of matrix in (17) and Euler angles yaw(ψ), pitch(θ), roll(φ) can be calculated for each point using relations in (18).

$${}^W \mathbf{R}_P = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (13)$$

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

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

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

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 Path Generation Algorithm

The GUI used is part of the recently developed *HumanPT* robotic application (Fig. 3). A description of this form and more details concerning the GUI can be found in [17]. By means of *HumanPT* the stable communication with PUMA 761 robotic manipulator through ALTER serial port is achieved. Many other operations like camera calibration, hand-eye calibration, image processing, real time control, visual servo control are also performed. The basic steps of generating initially a desired path are:

1. Using the GUI, the end-effector is moved to obtain the best view of the scene.
2. The capturing system being used is selected and a complex image is captured (here only the case of PSVS is examined).
3. The complex image is duplicated to keep the compatibility with other stereovision systems.
4. The desired filters and operations to process images are selected through pull down menus and edit boxes.
5. The processing procedure is executed step by step or automatically pressing the corresponding push buttons.
6. If the final edge image is the desired we continue with the selection or the design of the proper edges in the left view of the complex image.
7. The maximum number of pixels that could be found to the second image with the correspondence algorithm proposed is selected and the function "Corresponding edges" (to find the corresponding edges) is executed. This number (N_C) 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.
8. The density of the final path points (measure of velocities) for each colored edge is determined by changing the value to the edit box "Traj. Points". This number (N_T) could be from 3 to 1000. Then the function "Trajectory Points" is executed. When the calculations will finish the ordered equidistant path points are temporarily appearing to the screen and sets with the translation and rotation values for each point are listed to the list box. The results are also stored to a file.

9. By means of the function "Save Values" previous sets are loaded in a matrix, are stored in a file and can be used as the source data to control the robotic manipulator, in real time (28 ms cycle).

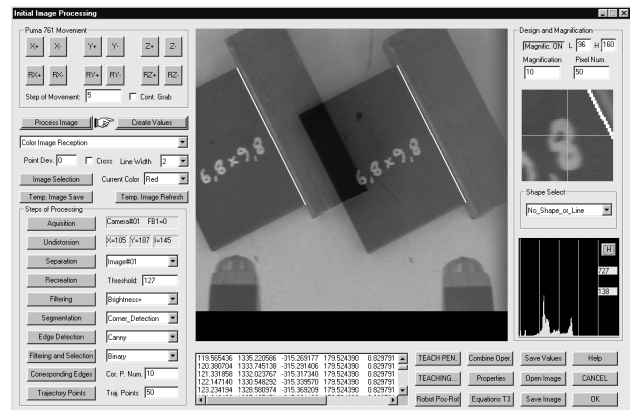


Fig. 3 Basic GUI form.

4 Experimental Results

In this section experimental results are presented. A Pulnix TM-6705 camera connected with the F64-PCI-DSP frame grabber card of Coreco, is used. The PSVS is mounted on the end effector of PUMA 761 robotic manipulator. The application is executed in a personal computer, Pentium II/350 MHz. A second computer communicates with the controller of the robotic manipulator PUMA 761 through ALTER communication port at 38400 bps. These two computers are connected together through an Ethernet-based local network.

In Fig. 4 the processing stages of a complex image of the scene are presented. The PSVS and the arc-welding torch are driven from the initial view of Fig. 3 to the final view of Fig. 4 (a) by means of the push buttons incorporated for this reason to the GUI. In this paradigm, 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. The desired path is the sum of two sub-paths. The first one will drive the end-effector from its current location to the start point of the "welding path" (for the approach 20 points were selected) and the second one will be the "welding path" (100 points were selected). The processing is made in two separate stages. Before the execution of correspondence and path points calculation algorithms, the initial image, Fig. 4 (a), is subtracted from image of Fig. 4 (b). Thus an image with only the desired edges is created (Fig. 4(c),(d)). The calculated 3D

points have a specified order, from the torch to the 3D object's desired edge. For this paradigm, pixel accuracy was selected. The result is illustrated to the 3D map of Fig. 5.

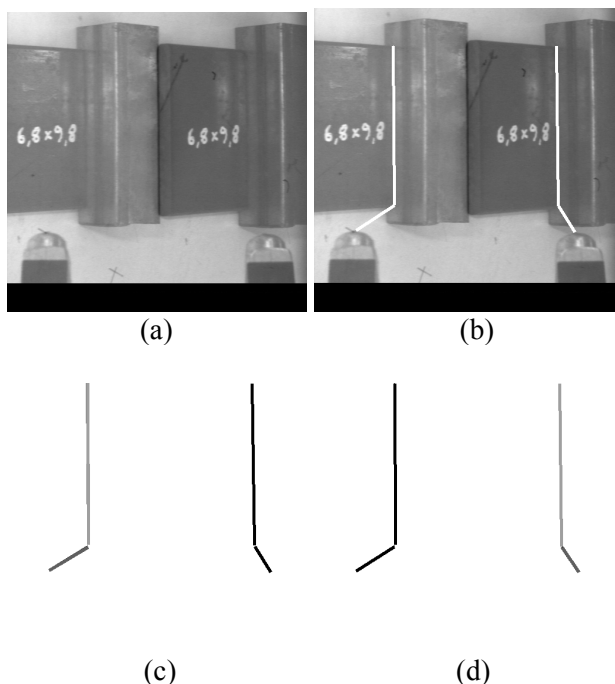


Fig. 4 a) Initial image, b) design of the desired edges, c) left view manually selected (two different colors) d) right view automatically selected after the application of the correspondence algorithm.

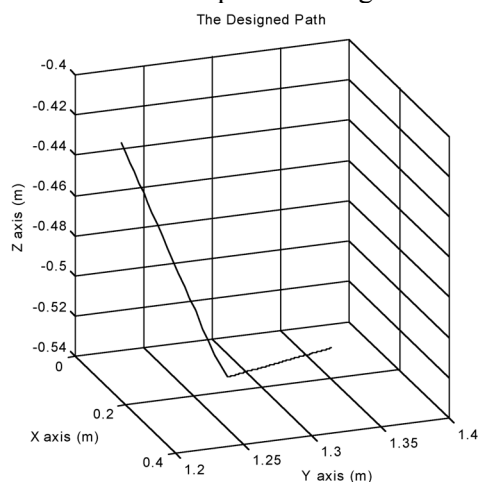


Fig. 5 3D map of path points for the paradigm of Fig. 4.

To indicate the specified sequence of 3D points, a complex image of 2D objects was captured (Fig. 6 (a)). After the processing (mean filter, conversion to binary image with threshold $T=128$, Roberts edge detector, closing operation) the path on the selected objects is illustrated as a continue line, by means of a 3D map in Fig. 6 (b). The start point (indicated with a small triangle) and the sequence of path

points are selected by means of the GUI. The path points are selected to be $N_T = 200$.

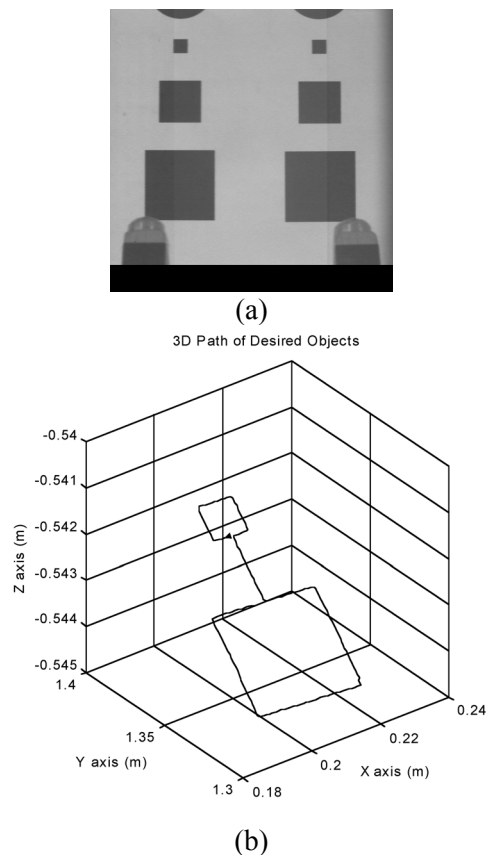


Fig. 6 a) The initial image, b) the 3D path of two selected objects.

Table 1

Errors with respect to the camera and to the world frame (mm)

	x_p	y_p	z_p
ME	0.0988	-0.045	-0.008
MAE	0.1963	0.1525	0.0381
MaxE	0.781	0.3965	0.059
	x	y	z
ME	0.1513	-0.082	0.0228
MAE	0.2083	0.1421	0.3091
MaxE	0.834	0.701	0.641

To show the accuracy of the measurement system, first, the 30x30 mm square of Fig. 6 (a) (in the middle) is used. The path points are selected to be $N_T = 100$. By using the estimated and the real point position vectors with respect to camera frame and to world frame, the mean error (ME), the mean absolute error (MAE) and the maximum error (MaxE) are calculated. The results are presented in Table 1. The depth map of the square points (points to camera coordinate system) is illustrated in Fig. 7

(a). The 3D map of points as they are calculated to the world coordinate system with crosses “+” and the path as it tracked by the robot using the low level point to point step controller (PPSC), with continue line, are illustrated in Fig. 7 (b).

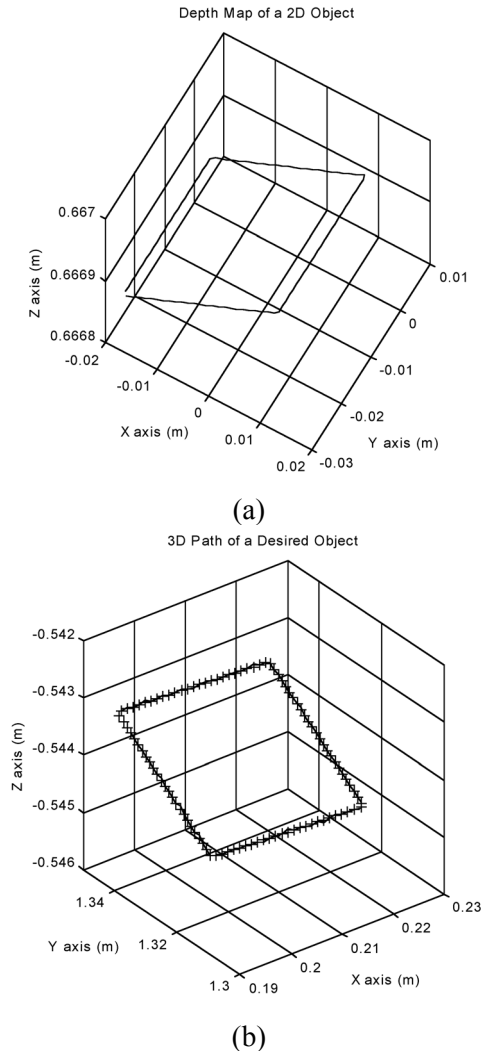


Fig. 7 3D maps of a 30x30 mm square a) with respect to the camera coordinate system, b) with respect to the world coordinate system.

As a second paradigm an arc of about 90 degrees of the complex image of Fig. 8 (a) is selected. The selection is illustrated to Fig. 8 (b). The initial image of Fig. 8 (a) is subtracted from image of Fig. 8 (b). Thus an image containing only two arcs remains. After the implementation of the processing algorithms for $N_T = 20$ points, the absolute errors for position and orientation are illustrated to Fig. 9. In this figure, absolute errors concern the desired path as it was estimated with the previous procedure and the path that the end-effector follows as it tracks the desired path, in real time.

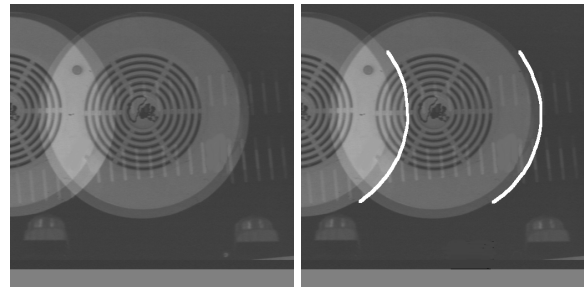


Fig. 8 a) The initial complex image. b) The initial complex image with the selected arc.

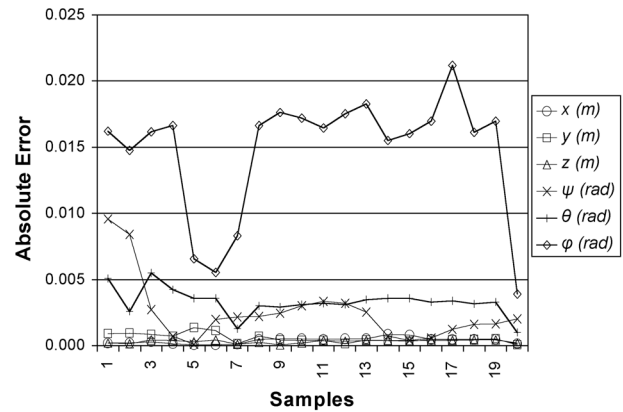


Fig. 9 Absolute errors for position and orientation of the path generated by the arc.

5 Conclusion

In this paper a vision-based path generation method for a robotic arc welding system, 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 personal computers. Scene complex images are captured by means of PSVS. A path can be generated, not using a “Teaching by Showing” like method but simply designing lines and shapes or connecting edges of objects in the scene. The algorithms proposed in the application for point correspondence and then path point calculation, permit to generate successive path points and to determine the behavior of the robot end-effector (and of the welding torch mounted on it) for each discrete part of the desired path. The method proposed for angles estimation to camera coordinate system provides the possibility of tracking any curved line by the system. 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 achieved and then the desired path point coordinates can be calculated with higher accuracy.

The described software is executed in personal computers. The system proposed can be used in

applications in space or submarine vehicles where it is difficult to determine a welding path with other methods. It can be also implemented to shipyards for damage repairs, where robotic welding systems are used. Moreover can be used to generate a robotic path for any task or procedure.

References:

- [1] A. Ude and R. Dillmann, Trajectory Reconstruction from Stereo Image Sequences for Teaching Robot Paths, *24th Inter. Symp. Industrial Robots*, 1993, pp. 407-414.
- [2] A. Ude and R. Dillmann, Vision-based Robot Path Planning, *Advances in Robot Kinematics and Computational Geometry*, Kluwer, 1994, pp.505-512.
- [3] A. Ude and R. 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, and R. Koeppel, 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 and 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 and 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 and R. Horaud, Visual Trajectories from Uncalibrated Images, *Workshop on New Trends in Image-Based Robot Servicing, IEEE/RSJ Int. Conf. On Intelligent Robots and Systems*, 1997, pp. 83-91.
- [11] X. F. Zha and 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 and 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, pp. 1589-1594.
- [13] T. Pachidis and J. Lygouras, Pseudo Stereo Vision System: Modifications for Accurate Measurements in 3-D Space Using Camera Calibration, *Proc. IEEE/ISA Sensors for Industry Conference (Sicon/02)*, 2002, pp. 66-70.
- [14] 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.
- [15] T. Pachidis, J. Lygouras and V. Petridis, A Novel Corner Detection Algorithm for Camera Calibration and Calibration Facilities, *Proc. 2nd WSEAS Int. Conf. on Signal Processing and Computational Geometry and Vision*, Rethymno, 2002, pp. 6911-6916.
- [16] H. Zhuang, Z. Roth and 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.
- [17] 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, *Proc. 2nd WSEAS Int. Conf on Robotics, Distance Learning and Intelligent Communication Systems (ICRODIC 2002)*, 2002, pp.1601-1607.
- [18] U. Dhond and J. Aggarwal, Structure from Stereo – A Review, *IEEE Transactions on Systems, Man and Cybernetics*, Vol.19, No.6, 1989, pp. 1489-1510.
- [19] G. Jones, Constraint, Optimization and Hierarchy: Reviewing Stereoscopic Correspondence of Complex Features, *Computer Vision Image Understanding*, Vol.65, No.1, 1997, pp. 57-58.
- [20] J. Goulermas and P. Liatsis, Hybrid Symbiotic Genetic Optimization for Robust Edge-based Stereo Correspondence, *Pattern Recognition*, Vol. 34, 2001, pp.2477-2496.